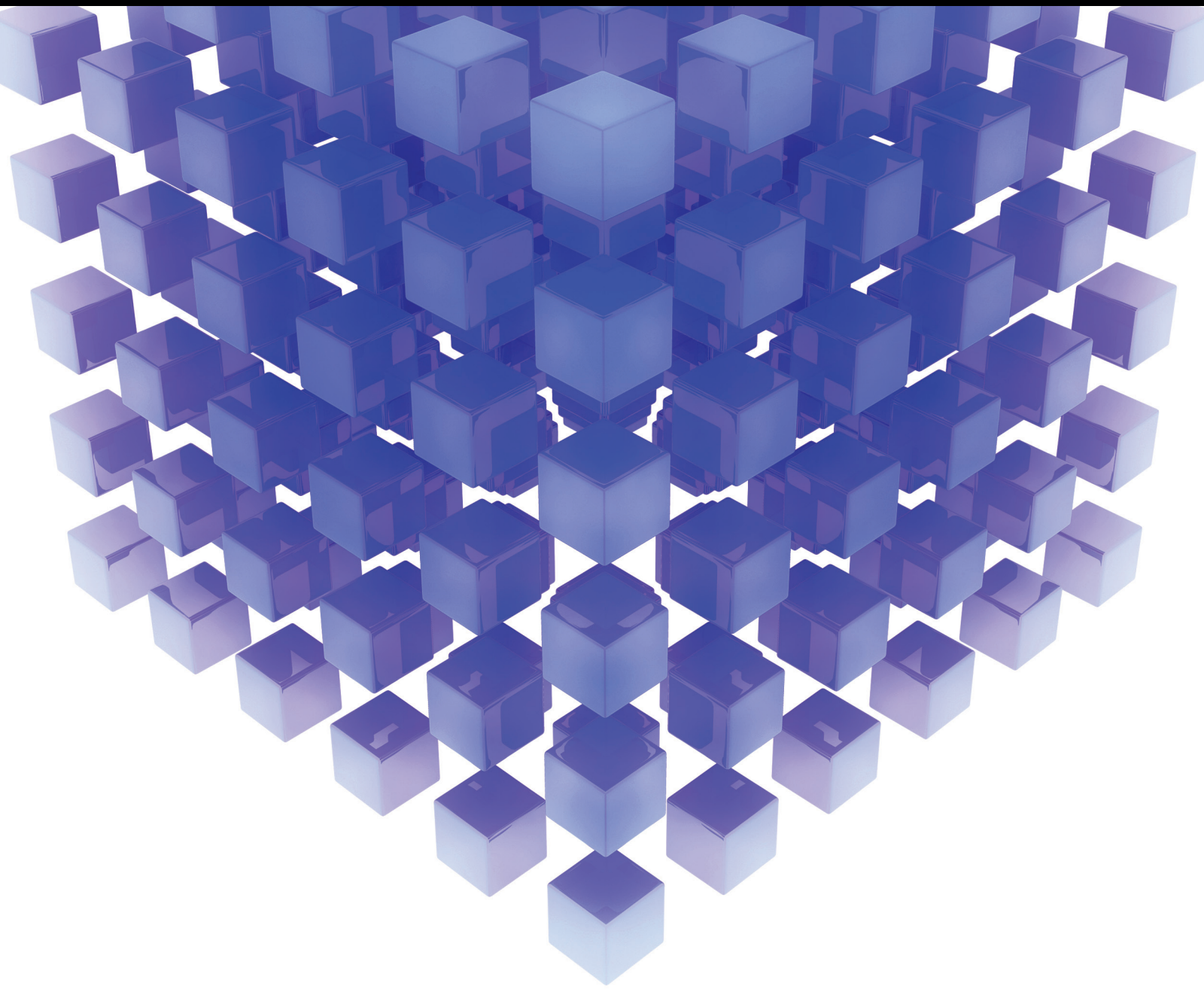
Mathematical Problems in Engineering

Computational Intelligence in Image Processing 2018

Special Issue Editor in Chief: Erik Cuevas
Guest Editors: Daniel Zaldivar, Gonzalo Pajares, Marco Perez-Cisneros, and Raul Rojas



Computational Intelligence in Image Processing 2018

Mathematical Problems in Engineering

Computational Intelligence in Image Processing 2018

Special Issue Editor in Chief: Erik Cuevas
Guest Editors: Daniel Zaldivar, Gonzalo Pajares, Marco Perez-Cisneros, and Raul Rojas

Editorial Board

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 6952803, 3 pages
https://doi.org/10.1155/2018/6952803

Editorial

Computational Intelligence in Image Processing 2018

Erik Cuevas,[1] **Daniel Zaldívar**,[1] **Gonzalo Pajares**,[2] **Marco Perez-Cisneros**,[1] **and Raúl Rojas**[3]

[1]*Departamento de Electrónica, CUCEI, Universidad de Guadalajara, Avenida Revolución 1500, Guadalajara, JAL, Mexico*
[2]*Departamento de Ingeniería de Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain*
[3]*Institut für Informatik, Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany*

Correspondence should be addressed to Erik Cuevas; erik.cuevas@cucei.udg.mx

Received 27 February 2018; Accepted 27 February 2018; Published 5 August 2018

Computational intelligence (CI) represents a set of robust information processing approaches for knowledge management and decision-making. CI methods are considered as useful tools for the development of advanced systems which maintain intelligent capabilities such as learning, adaptation, and evolution for solving complex problems. Examples of popular CI schemes include artificial neural networks, fuzzy systems, evolutionary algorithms, decision trees, multiagent systems, knowledge-based systems, rough set theory, and hybridization of these models.

Image processing is a progressive and fast-moving research discipline. Recent advances in image processing have produced an explosion in the use of images in a diversity of engineering and scientific applications. Therefore, each new approach that is developed by engineers, mathematicians, and computer scientists is quickly identified, understood, and assimilated in order to be applied to image processing problems.

Standard image processing techniques frequently face great difficulties when they operate over images containing information that is incomplete, noisy, imprecise, fragmentary, not fully reliable, vague, contradictory, deficient, and overloading. Under such conditions, the use of computational intelligence approaches has been recently extended to address challenging real-world image processing problems.

The use of computational intelligence approaches in image processing has increased in all engineering areas. Such a fact is evident from a quick look at special issues, congresses, and specialized journals that focus on such a topic. The central purpose of this special issue is to bridge the gap between computational intelligence techniques and challenging image processing applications. The final goal is to expose the cutting-edge research and applications that are going on across the domain of image processing, particularly those whose contemporary computational intelligence techniques can be or have been successfully employed.

The special issue received several high-quality submissions from different countries all over the world. All submitted papers have followed the same standard of peer-reviewing by at least three independent reviewers, just as it is applied to regular submissions to the Mathematical Problems in Engineering journal. The primary guideline has been to demonstrate the wide scope of computational intelligence algorithms and their applications to image processing problems.

The paper authored by S. Lee and I. Kim presents a deep neural network model for effective video captioning. Apart from visual features, the proposed model learns additionally semantic features that describe the video content effectively. In this model, visual features of the input video are extracted using convolutional neural networks such as C3D and ResNet, while semantic features are obtained using recurrent neural networks such as LSTM. The approach also includes an attention-based caption generation network to generate the correct natural language captions based on the multimodal video feature sequences. Various experiments, conducted with the two large benchmark datasets, Microsoft Video Description (MSVD) and Microsoft Research Video-To-Text (MSR-VTT), demonstrate the performance of the proposed model.

X. Liu and X. Gao propose a method based on the subspace optimization to improve the image restoration process. This method corrects the search directions of primal alternating direction method by using the energy function and a linear combination of the previous search directions. In addition, the convergence of the primal alternating direction method is proven under some weaker conditions. Thus, the convergence of the corrected method could be easily obtained by the equivalence between the direction method and the previous direction. Numerical examples are given to show the performance of the proposed method finally.

C. Jungan et al. propose a novel clustering algorithm called k-deviation density based DBSCAN (kDDBSCAN). The method extends the DBSCAN method by exploiting a new density definition. Various datasets containing clusters with arbitrary shapes and different densities are used to demonstrate its performance and investigate its feasibility. The results show that kDDBSCAN performs better than DBSCAN.

O. O. Olugbara et al. present a new algorithm based on perceptual color difference saliency along with binary morphological analysis for segmentation of melanoma skin lesion in dermoscopic images. The new algorithm is compared with existing image segmentation algorithms on benchmark dermoscopic images acquired from public corpora. Results of both qualitative and quantitative evaluations of the new algorithm are encouraging as the algorithm performs excellently in comparison with the existing image segmentation algorithms.

The paper by A. Stubendek and K. Karacs proposes a combined shape descriptor for object recognition with offline and online learning methods. The descriptor is composed of a local edge based part and global statistical features. The approach also presents a two-level, nearest neighborhood type multiclass classification method, in which classes are bounded defining an inherent reject region. In the first stage, built on the global features, class candidates get prefiltered, in contrast to the second stage, where the projected features are compared. The experimental results show that the combination of independent features leads to increased recognition robustness and speed. The core algorithms map easily to cellular architectures or dedicated VLSI hardware.

H. Xu et al. present a study on crack detection of industrial explosives. The proposed algorithm consists of the following steps: (1) Image preprocessing was performed according to the defect features of industrial explosives cartridge, and we developed an improved visual attention-based algorithm. This proposed algorithm features a parametric analysis that can be implemented on the image according to the conspicuous maps with the introduction of the concept of defect discrimination. (2) As compared with other algorithms, our method can realize real-time multitarget detection function. (3) A new analysis method, the IPV-WEN algorithm, was proposed to analyze the cartridge defects based on performance indices. Through comparison and experimentation, it was revealed that this method can achieve a detection accuracy of 97.9%, with detection time of 34.51 ms, which satisfied the requirement in the industrial explosives production.

The paper authored by W. Zhao et al. presents a denoising algorithm combined with SGK dictionary learning and the principal component analysis (PCA) for noise estimation. At first, the noise standard deviation of the image is estimated by using the PCA noise estimation algorithm. And then it is used for SGK dictionary learning algorithm. Experiments' results show that (1) the SGK algorithm has the best denoising performance compared with the other three dictionary learning algorithms, (2) the SGK algorithm combined with PCA is superior to the SGK combined with other noise estimation algorithms, and (3) compared with original SGK algorithm, the proposed algorithm has higher PSNR and better denoising performance.

M. Yousaf et al. propose an effective novel technique to improve the performance of content-based image retrieval (CBIR) on the basis of visual words fusion of scale-invariant feature transform (SIFT) and local intensity order pattern (LIOP) descriptors. SIFT performs better on scale changes and on invariant rotations. However, SIFT does not perform better in the case of low contrast and illumination changes within an image, while LIOP performs better in such circumstances. SIFT performs better even at large rotation and scale changes, while LIOP does not perform well in such circumstances. Moreover, SIFT features are invariant to slight distortion as compared to LIOP. The proposed technique is based on the visual words fusion of SIFT and LIOP descriptors, which overcomes the aforementioned issues and significantly improves the performance of CBIR. The experimental results of the proposed technique are compared with another proposed novel features fusion technique based on SIFT-LIOP descriptors as well as with state-of-the-art CBIR techniques. The qualitative and quantitative analyses carried out on three image collections, namely, Corel-A, Corel-B, and Caltech-256, demonstrate the robustness of the proposed technique based on visual words fusion as compared to features fusion and state-of-the-art CBIR techniques.

S. Cheng et al. present a method for water-quality monitoring. The approach is based on the Tracking-Learning-Detection (TLD) framework and eXtreme Gradient Boosting (XGBoost). Firstly, TLD captures 3D coordinate information of fish by using video. Then, it calculates the parameters of the fish movement which can reflect the change of water quality through the processing of the fish body information. The data coordinate information will be more prominent via the data processing. The model was used to analyze and evaluate fish behavior parameters under unknown quality to achieve the purpose of water-quality monitoring.

The paper authored by P. V. V. Kishore et al. presents an approach to detect Indian classical dance forms. With this propose, a new segmentation model is developed using discrete wavelet transform and local binary pattern (LBP) features. Then, a 2D point cloud is created to detect local human shape changes in subsequent video frames. The classifier is fed with 5 types of features calculated from Zernike moments, Hu moments, shape signature, LBP features, and Haar features. The method also explores multiple feature fusion models with early fusion during segmentation stage and late fusion after segmentation for improving the classification process. The extracted features input the AdaBoost multiclass classifier

with labels from the corresponding song (tala). The classifier has been tested with online dance videos and on an Indian classical dance dataset prepared in our lab. The algorithms were tested for accuracy and correctness in identifying the dance postures.

E. Chavolla et al. propose an algorithm based on a modification of the HSV color model in order to improve the accuracy of the results obtained from the color segmentation by using the K-means++ algorithm. The proposal gives a better segmentation and fewer erroneous color detections due to illumination conditions. This is attained by shifting the hue and rearranging the H equation in order to avoid undefined conditions and increase robustness in the color model.

Acknowledgments

Finally, we would like to express our gratitude to all of the authors for their contributions and the reviewers for their efforts to provide valuable comments and feedback. We hope this special issue offers a comprehensive and timely view of the area of applications of computational intelligence in image processing and we hope that it will grant stimulation for further research.

Erik Cuevas
Daniel Zaldívar
Gonzalo Pajares
Marco Perez-Cisneros
Raúl Rojas

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4763050, 18 pages
https://doi.org/10.1155/2018/4763050

Research Article

Shape Recognition Based on Projected Edges and Global Statistical Features

Attila Stubendek and Kristóf Karacs

Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Prater 50/A, Budapest 1083, Hungary

Correspondence should be addressed to Attila Stubendek; stubendek.attila@gmail.com

Received 16 September 2017; Revised 12 January 2018; Accepted 13 February 2018; Published 19 April 2018

Academic Editor: Daniel Zaldivar

A combined shape descriptor for object recognition is presented, along with an offline and online learning method. The descriptor is composed of a local edge-based part and global statistical features. We also propose a two-level, nearest neighborhood type multiclass classification method, in which classes are bounded, defining an inherent rejection region. In the first stage, global features are used to filter model instances, in contrast to the second stage, in which the projected edge-based features are compared. Our experimental results show that the combination of independent features leads to increased recognition robustness and speed. The core algorithms map easily to cellular architectures or dedicated VLSI hardware.

1. Introduction

Recognizing shapes is an essential task in computer vision, especially in understanding digital images and image flows. A wide spectrum of application areas relies on shape recognition, including robotics, healthcare, security systems, assistance for the impaired.

The goal of computer vision is to generate answers to visual queries which are based on the input image. Depending on the query, several levels can be identified in a vision problem. A typical categorization distinguishes between detection, localization, and recognition. In the detection part, the presence of an object is examined; localization determines the position; in comparison, recognition identifies the detected objects, possibly considering their context in the visual scene. However, the definition of the object depends on the task [1, 2]. In typical computer vision systems, the result is computed from the image through its features, as a verified hypothesis [3, 4]. Similar to queries, features may incorporate local details as well as global image properties. If patches or complete contours are extracted from the image, the shape of the resulting region is one of the most important local features beside color, texture, and other details [5].

The key to efficient shape recognition is to use an appropriate representation that comprises all important characteristics of a shape in a compact descriptor. A shape description is considered to be efficient from a recognition point of view, if

(i) the representation is compact,

(ii) a metric for the comparison of the feature vectors can be efficiently computed,

(iii) the representation is insensitive to minor changes and noise,

(iv) the description is invariant to several distortions.

The most basic classification of shape descriptions distinguishes between contour-based and region-based techniques. Each method extracts specific features that encompass some meaningful aspects of the information in the shape. Using only one feature type thus limits the description power of the descriptor in terms of discriminative power and classification performance [6].

Contour-based shape features describe the shape based on its contour lines in various representations, such as contour moments [7, 8], centroid distances and shape signatures [9–11], scale space methods [12], spectral transforms [13, 14], and structural representation [15, 16]. Common drawbacks of contour methods are the complexity of feature matching,

representation of holes and detached parts of the shape, and noise sensitivity [6].

Region-based techniques describe the shape based on every point of the shape and represent mainly global features of the shape. Moment invariants are derived as statistical features of the shape points [17]. Orthogonal moment descriptors such as Zernike and Legendre descriptors employ polynomials instead of the moment transform kernels [18–20]. Complex shape moments are robust and matching is straightforward; however, lower order of moments poorly represents the shape, but higher orders are more sensitive to noise and difficult to derive [21]. Generic Fourier descriptor represents the shape as the 2D Fourier transformation of the polar-transformed shape.

The requirement of compactness stands for the maximal level of independence of the feature data without sacrificing comparison and recognition performance. In other words, redundancy in the feature vector is accepted if it significantly simplifies the subsequent processing of the vector, thus accelerating the classification, and may increase the accuracy of the recognition. Combining different features allows catching different essences of the shape, and although it may introduce redundant data, at the same time it also increases robustness [22–24]. However, employing compound feature vectors requires a decision method that suits the different parts of the description. In machine learning, several ensemble classifiers are known, which handle compound features, like boosting, bagging, or stacking [25–27].

Representations of the same real-world object may differ due to several effects such as lighting conditions, camera settings, position, and noise. The major challenges of object detection are to ignore the differences in the representation resulting by sensing and preprocessing and to recognize if the difference is caused by different input objects. Several invariance requirements are often standard expectations to shape recognition methods, but the exact group of requirements has to be defined to each individual task, considering other parameters as well, such as hardware ones.

The principal motivation of our work was to create methods for portable vision application, where safety and reliability are the primal goals, such as aid for the visually impaired, and also other vision-based recognition systems. The requirements towards the application outline the specifications of the used algorithms. We aim to recognize mainly rigid, not flexible objects in video images, but due to various image acquisition conditions and poor image quality, significant amount of noise has to be handled and several invariance requirements have to be fulfilled. The application is valuable only if it is reliable and it is not critical to classify all frames but false answers can easily cause dangerous situations. Thus minimizing false-positive errors has priority over maximizing cover ratio. Finally, we preferred that kind of algorithms that are appropriate for dedicated VLSI architecture but provide real-time processing even on standard cell phone CPU and GPU.

Visual environments containing real-word objects normally encountered by humans contain a practically infinite number of object classes. Depending on the task, out of these classes, the number of relevant ones may be orders of magnitude smaller than the number of irrelevant classes; thus representing each irrelevant class with a representative instance is not efficient, if at all feasible. Hence, our primary goal is to develop a framework that can handle multiclass recognition problems, with only a few classes considered relevant, which requires performance evaluation metrics adapted to this flavor of multiclass classification.

The paper is organized as follows. In Section 2, we review the issue of invariance requirement of a recognition tool. In Section 3, we describe the performance evaluation methods used in the paper. In Section 4, we present our proposed compound description method, the Global Statistical and Projected Principal Edge Description. In Section 5, a gradual classification method is presented including a limited nearest neighborhood decision. The related online and offline learning method is presented in Sections 6 and 7. Finally, in Section 8, we show our results and, in Section 9, we conclude with future directions.

2. The Role of Description and Classification

We investigate classic machine learning decomposition and the role of edges and their appropriate and efficient representation. The estimation of the ground truth is based on limited sensing, resulting in different representation of essentially same objects. The key point of the recognition is a model that draws boundaries of output classes. However, classes may differ based on various traits; thus the selection of discriminative features is also essential. From this point of view, we will divide recognition to feature extraction and classification.

In this paper, we investigate shape recognition that models the decision based on supervised learning, where the model is built up based on previously labeled inputs denoted as templates; the set of already known inputs is denoted as training set. Independently from the exact type and behavior of the classifier, the classification is a comparison of the input to labeled elements from the training set (or a model built up from the set), where the decision is a function of the representation. The difference between the representations of the same object is a result of various distortions that occur during the image acquisition and preprocessing. Note that distortions affect also the elements of the training set.

The input shape S_i^* is a result of a T transformation of the original shape S_i, where γ denotes the parameter(s) of the transformation and P is the set of all possible parameters of the transformation:

$$T_{\gamma_i}(S_i) = S_i^*, \quad \gamma_i \in P. \tag{1}$$

The input shape S_t^* is a result of a T transformation of the original template shape S_t:

$$T_{\gamma_t}(S_t) = S_t^*, \quad \gamma_t \in P. \tag{2}$$

The output class of S_i^* is a decision function $\widehat{D}$, depending on one or more labeled shapes $S_{t1}^*, \ldots, S_{tn}^*$, comprising the representative set R:

$$\widehat{D}\left(S_i^*\right) = D_R\left(S_i^*\right) \qquad \bigcup_{i=1}^{n} S_{ti}^* = R. \tag{3}$$

The task of the recognition is not the reconstruction of the original shape by mathematical operations but to classify independently from transformations that distort the original and the template shapes and thus to estimate the ground truth C.

$$\widehat{D}\left(S_i^*\right) \approx C\left(S_i\right). \tag{4}$$

From this aspect, the transformation can be also considered as noise and noise is considered as a transformation.

In the next paragraphs, we give an overview of possible distortions of a shape in an object recognition problem and formalize deviations mathematically. Then we try to define the ability to represent similarity by formalizing tolerance and invariance in general and especially for the target shapes. Finally, we give an overview about possible solutions of ensuring invariance and tolerance in a description-based recognition system.

2.1. Distortions in a Shape Description Problem. To find all the possible deviations of a shape, we go along the process where the binary shape is generated from a real-word object. However, shape generally can be defined as a multidimensional set of points; in this paper, we only focus on 2D shapes that are projections of 2D, flat objects in a 3D space and characteristic silhouettes of 3D images (2D representation of 3D objects from different viewpoints, where the object has to be modeled or multiple shapes are needed to reconstruct the object, is not the subject of this paper).

Applying the constraints above, during image acquisition by a camera, where the 3D-2D transformation and the sampling take place, the following geometric and pixel-level deviations may occur:

(a) Rotation of the object on its plane compared to the camera axes

(b) Position difference of the object relatively to the camera, which can be split to

  (ba) distance difference of the camera and the object

  (bb) position difference of the projected camera origin and the object

(c) Angular deviation of the object plane normal-vector and the camera projection direction

(d) Appearance of noise due to sensing limitations and sampling errors

(e) Some part of the shape being missing or the shape being joint with another pattern

Note that, from practical considerations, geometric variances can be represented in other spaces too. If we consider the characteristic motives of the shape to be larger than the sampling rate, the deviance in (d) is limited only to the sensing noise. However, inappropriate focusing may also cause loss of details of the shape which in most of the cases exceeds the sampling error.

The shape is generated from the input image by various image processing algorithms, such as segmentation, patterns extraction, and morphological operations. Here, we will not investigate these preprocessing phases, but generally it can be stated that the shape generation is a binarization of some characteristic pattern of the image; thus the deviation (e) may befall due to the various lighting condition and unambiguous shape edges.

Summarizing the deviations, we can name those variations, of which shape recognition may be independent or the similarity index should be proportional to the deviation. From the aspect of the shape, the distance variation appears in different scales of the shape. Positioning variance results in a different location of the shape on the image canvas; rotation of the image in its plane also results in a rotated shape. Angular deviation of the image plane together with positioning difference results in perspective variance. Not only do binarization ambiguity and noise result in misplaced edge pixels on the desired shape but also both of them may lead to detached shape parts or to holes in the original shape.

2.2. Decomposition Model for Shape Similarity. Variance in the appearance of an object can be modeled in a mathematical sense as noise. We call shapes to be similar if the difference is due to different observation properties and processing noise. If the shape is rigid, observation property is reduced only to geometrical transformations. To achieve classification consistency across various distortions, we identify two different aspects, invariance and tolerance, with respect to these distortions.

Invariance of a recognition engine with respect to a particular type of deviation is defined as the ability to return the same result for all inputs that only differ in the given deviation.

$$\widehat{D}\left(T_\gamma\left(S\right)\right) = \widehat{D}\left(S\right) \quad \text{for } \forall \gamma \in P. \tag{5}$$

We speak about tolerance to an effect if difference in the input causes no difference in the output to a certain limit L_T:

$$\widehat{D}\left(T_\gamma\left(S\right)\right) = \widehat{D}\left(S\right) \quad \text{for } \forall \gamma \in P,\ \|\gamma\| < L_T. \tag{6}$$

Note that norm for transformation parameter is substantially an abstract function, which cannot be measured directly but only can be estimated based on the transformed shape. Similarly, the limit L_T also represents an abstract value. Both the norm and the limit are determined by the actual interpretation of the similarity.

Tolerance can be defined as a limited, local invariance, and, vice versa, invariance is a global tolerance. Due to this, invariance with respect to an effect implies tolerance

to the whole domain, while invariance can be achieved by overlapping regions of tolerance.

The human similarity metric highly depends on the actual task; thus no general statement can be defined on which deviations should be eliminated and which should be tolerated during shape recognition. The environment in some cases does provide some references regarding the projection details. Some of the parameters described above might be fixed, previously adjusted (e.g., relative orientation or position of the camera and the object), or can be derived from the image metadata (e.g., distance of the focused subject of an image and angular difference from the horizontal plane). In these cases, deviations in the given parameters result in a different shape; thus invariance is needed only if the human notion of the shape is not dependent on the distortion, and only tolerance is required if the given parameters are not exact or the human perception does tolerate deviations with a certain limit.

The transformations above can be characterized by the possible outputs applying the transformations. The range Q of transformation T is defined as the set of all possible results of transformation T on a shape S:

$$Q_T(S) = \left\{U,\ U = T_\gamma(S),\ \gamma \in P\right\}. \tag{7}$$

In the case of reversible transformation $T_\gamma(\cdot)$, the inverse transformation is denoted here as $T_{\gamma^{-1}}(\cdot)$. To represent noise as transformation, we chose the parameter γ as a shape, and the noise transformation $T_\gamma(S) = S \oplus \gamma$, where $\oplus$ stands for the logical X-OR operation. By using this formalization, the random property of the noise transformation is ensured in random selection of parameter γ. This annotation allows us to represent the noise as a reversible operation, where $\gamma^{-1} = \gamma$.

We denote shapes S and U to be separated by transformation T if there are no parameters γ_1 and γ_2 of T which transform S and U to the same shape:

$$\begin{aligned} &\nexists \gamma_1, \gamma_2 \in P: \\ &T_{\gamma_1}(S) = T_{\gamma_2}(U) \\ &Q_T(S) \cap Q_T(U) = \emptyset \end{aligned} \tag{8}$$

If the transformation is reversible, then S and U are separated by transformation T:

$$\begin{aligned} &\nexists \gamma:\ T_\gamma(S) = U \\ &S \notin Q_T(U). \end{aligned} \tag{9}$$

If we assume that output classes are separated by transformation T and no reference system is given, the recognition should be invariant to transformation T. If the classes are not separated, the recognition should only tolerate the difference caused by transformation T.

Without any assumptions about the noise, adding sampling and preprocessing noise to a shape (noise transformation) may result in an arbitrary distortion; no shapes are separated by noise transformation, and thus the recognition should only be tolerant to the noise transformation. If the noise is bounded, the result space is limited.

Adding sampling and preprocessing noise (noise transformation) theoretically may result in arbitrary shape. The geometric transformations, except for the 90-degree perspective distortion, are closed transformations; thus invariance with respect to rotation, scale, and transition and tolerance to perspective distortion are standard requirements in case of shape recognition. However, distortions affecting a shape cannot be handled separately. Sampling noise when doing a low resolution scale or a flat perspective view can be significant. Hence, scale invariance and perspective tolerance are limited to scales where essential details of the shape are still present.

Invariance and tolerance regarding different distortions can be ensured in various ways. Feature extraction generalizes the shape from the specific aspect independently from those effects that are irrelevant for the classification, and classification performs a decision based on a complex distance. Hence, feature extraction is generally responsible for ensuring invariance and classification for tolerating difference to a specific limit. However, as we described in Section 2.2, invariance can be achieved by continuous tolerance and tolerance is a partial invariance; thus encoding similarities may occur in different parts of the recognition unit. In addition, there are many classifiers that also include generalization power (i.e., kernel functions).

3. Performance Evaluation in Multiclass Classification

In open-world multiclass recognition problems, only a relatively small subset of the classes is considered relevant for the given task. This is similar to a binary classification scheme with only positive and negative labels, with the difference that inside the positive class we need to be able to differentiate between several "positive" labels, which are considered relevant by themselves, as opposed to the irrelevant ones, among which no differentiation is necessary. More precisely, the relevancy attribute partitions the set of classes into the relevant and the irrelevant subsets.

For an appropriate evaluation performance, metrics need to be adapted to this nature. Due to the prevalence of the positive-negative property for this multiclass case, it makes sense to rely on classic binary performance metrics, including recall and precision. To be able to use them, we need to extend the binary confusion matrix scheme of positive and negative decisions. Since we do not differentiate between irrelevant classes, all decisions from and into irrelevant classes are counted as true-negative (TN). True-positive (TP) counts all correct positive, that is, relevant, classifications; false-negative (FN) refers to the number of decisions where a relevant input was classified as irrelevant. False-positive decisions are split into two categories: $\mathrm{FP}_{\mathrm{Rel}}$ indicates the number of false classifications between relevant classes, while $\mathrm{FP}_{N\mathrm{Rel}}$ counts decisions where an irrelevant input is classified as a relevant one.

Using this extended taxonomy, precision and recall can be defined as follows:

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}_{\mathrm{Rel}} + \mathrm{FP}_{N\mathrm{Rel}}},$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}_{\text{Rel}}}. \tag{10}$$

For F_β, being a weighted average of precision and recall, the definition does not need to be changed, with recall being more important for $\beta > 1$, and precision weighted more important for $\beta < 1$:

$$F_\beta = \left(1 + \beta^2\right) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}. \tag{11}$$

As we primarily target real-time recognition tasks on video sequences, type II errors have a much lower cost than type I errors. Hence, we have used the value $\beta = 0.05$, which reflects this preference.

4. The Global Statistical and Projected Principal Edge Description

Since shapes have different properties depending on several aspects and the distinctive characteristics may be encoded in different aspects, a descriptor compound of independent shape features may provide more accurate representation. As mentioned earlier, the most important aspects are scope (global-local) and basis (region and edge). We suggest a shape description denoted as Global Statistical and Projected Principal Edge Description (GSPPED) that combines these shape features in order to represent different aspects.

The descriptor consists of global statistical features and principal edge descriptors representing local characteristics. Structurally the descriptor is divided into three parts:

(a) A highly expressive header including eccentricity and area fill ratio

(b) A region-based feature set with histogram moments representing global shape properties

(c) A contour-based edge description employing modified Projected Principal Edge Distribution description for shapes

4.1. General Region-Based Global Features. Moments and general statistical features derived from moments are frequently used descriptors in shape and pattern recognition [6, 28]. A series of moments express the properties of a shape from basic features to details [17]; however, moments of higher orders are more vulnerable to noise and variances in shape. Thus, in vision applications, where patterns belonging to the same class may vary due to camera position or segmentation, using higher-order moments is less effective [21].

The header part of the proposed description aims to depict the shape in the most compressed and expressive way. We are searching for that kind of combinations which are perceptually linear but may be calculated by nonlinear operations from easily measurable operands. Eccentricity and area ratio describe the basic outline of the shape; however, they are only suitable to use as primary features in filtering obviously false matches [6]. Besides, they are simple scalars encompassing understandable and most characterizing information for a human. The smaller the eccentricity is, the closer the shape is to a circle, while shape with eccentricity value of one is a line. The area ratio is the ratio of the area occupied by the shape and the area of the minimal rectangle covering the shape.

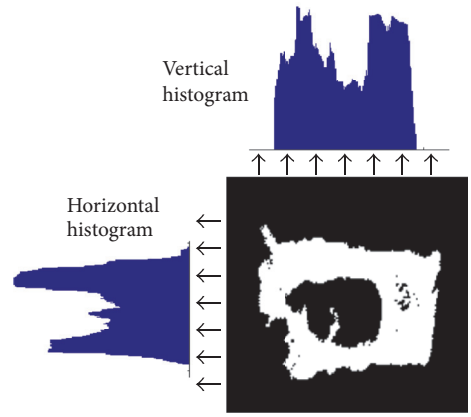


Figure 1: Vertical and horizontal histograms of a shape.

The region-based feature set consists of the first four moments of horizontal and vertical histograms of the shape (Figure 1). Using more moments would enable us to describe the shape in more detail, but we would lose the general recognition ability. Thus, we used the first four moments: mean, variance, skewness, and kurtosis. For the sake of simplicity but not losing dimensional information, the moments are computed from the histograms of the shape. This solution reduces computational complexity compared to 2-dimensional moment calculation and provides advantages when the descriptor is computed on VLSI architecture. The distribution of the region-based features is shown in Figures 2 and 3.

4.2. Contour-Based Features

4.2.1. The Projected Principal Edge Distribution. Projected Principal Edge Distribution (PPED) is a grayscale image descriptor that characterizes principal edges of 64×64 pixels' moving image window developed for recognizing anatomical regions in X-ray images. To highlight important edges, for every pixel, a local threshold is defined as the median of differences of neighboring pixel values in a 5×5 pixels' window around the pixel. Edges are detected in four directions (0, $\pi/4$, $\pi/2$, and $3\pi/4$) with a convolution, where values below the actual pixel threshold (defined above) are set to zero. To select the principal edges only, for every pixel location of the four edge maps, only the largest edge value is kept and the values of the location on the other three maps are set to zero. Edge maps are then projected in the same direction as the convolution and normalized to a length of 16 values. Finally, smoothing is applied to reduce noise.

For every window position on the input image, a separate feature vector is computed and compared to the labeled

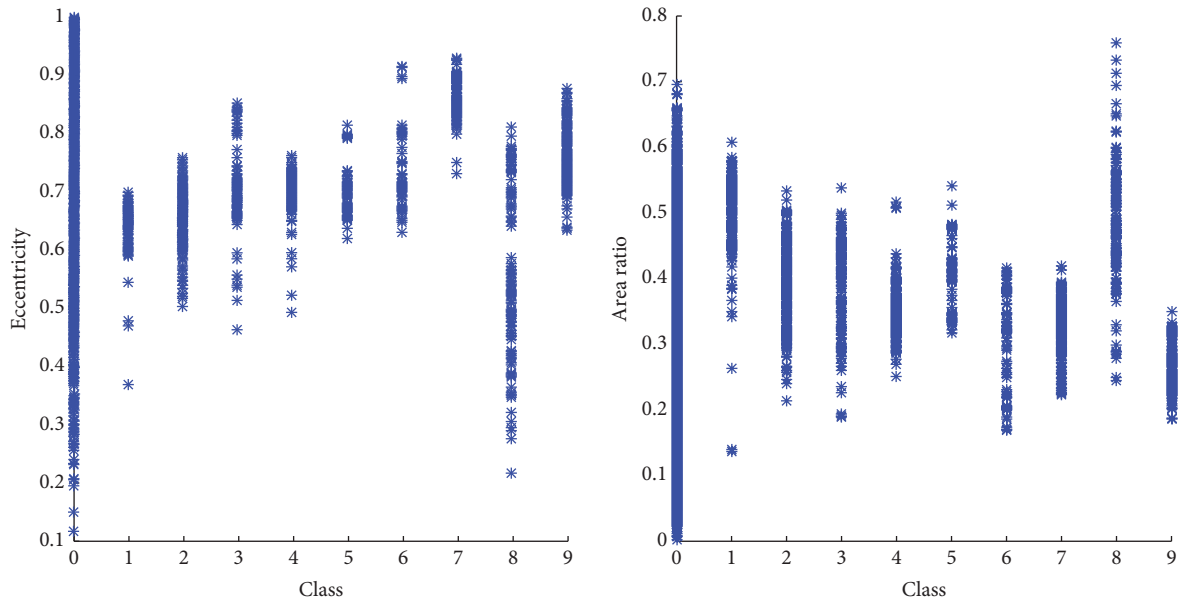


Figure 2: The distributions of the eccentricity and the area ratio on the Hungarian Forint Banknote pattern dataset (for details, see Section 8.). Classes 1–9 represent different patterns from banknotes; other irrelevant shapes are denoted as class 0.

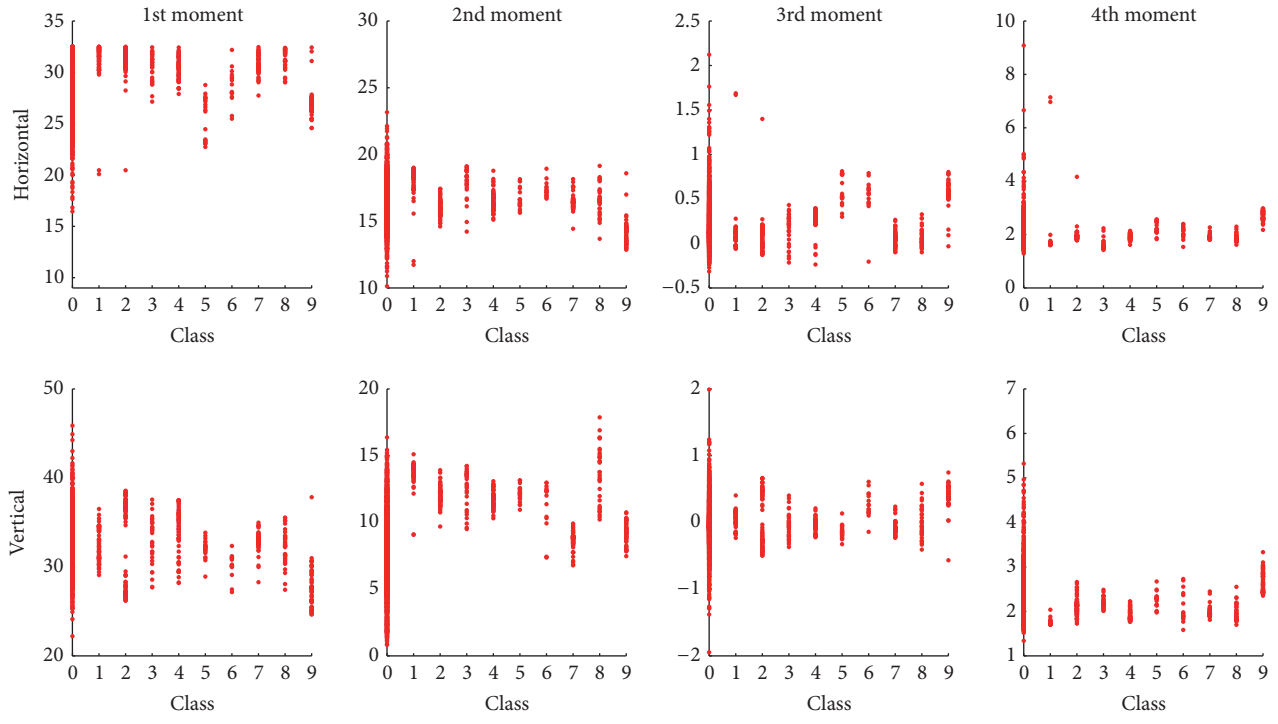


Figure 3: The distributions of the vertical and horizontal moments on the Hungarian Forint Banknote pattern dataset (for details, see Section 8.). Classes 1–9 represent different patterns from banknotes; other irrelevant shapes are denoted as class 0. The figure shows that these values do not contain enough discriminative power to classify the patches but provide a good guide to filter and reject obviously different shapes.

templates, choosing the closest instance to classify the input [29].

Note that PPED and relative descriptors are not rotation-invariant, and scale invariance is ensured by using fixed window sizes and scaling.

4.2.2. The Extended Projected Principal Shape Edge Distribution (EPPSED). The core of the contour-based edge description is based on the principle used by the PPED. The edge values are detected in four directions; principal edges are selected and then projected and concatenated; the result

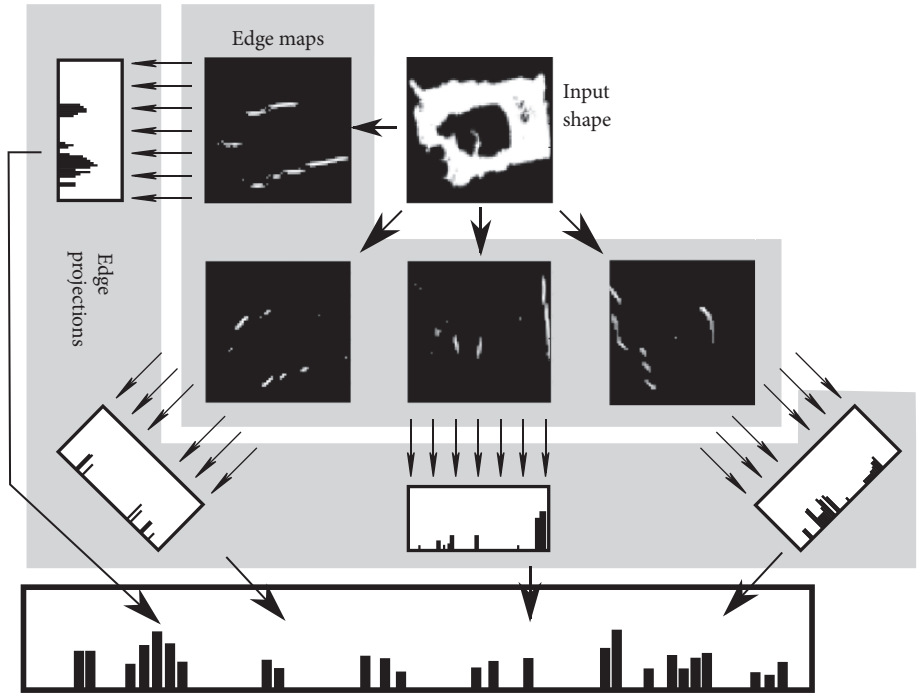


Figure 4: Construction of the EPPSED feature vector. Edges are detected in four directions; then thresholding and maxima selection are applied; finally projections are concatenated and normalized.

for one shape is a 64-element feature vector. The essential difference between the methods is in selecting the object, calculating the thresholds and the maxima of the four edge maps, and ensuring scale and rotation invariance. The method is shown in Figure 4.

The input of a shape recognition task is the shape: a binary or a grayscale image with a binary mask, where the borders, the edges of the shapes, are detected by the pattern extractor or the segmentation algorithm. Thus, finding the border of the shape is the task of the preprocessor, not the shape descriptor. From another aspect, the differences between neighboring pixel gray-values in a binary image are 0 or 1 (pixel value 1 for in-shape pixels and 0 for others); consequently, the median value is also binary. Hence, using the median of differences as a threshold is unnecessary. We experimented with different threshold values, and we concluded that the best results can be achieved by using a threshold value of $\theta_{\text{global}} = 2$.

An essential difference between the EPPSED and the PPED lies in the thresholding method and in choosing the maximal edge value. Our aim is to design a cross-architecture algorithm, where architecture-dependent computing does not influence the output significantly. Using hard-thresholding (t_{hard}) may result in ambiguous behaviors near the threshold value for almost identical edge values; thus we use a soft-thresholding (t_{soft}) method with no discontinuity:

$$t_{\text{soft}}(x) = \begin{cases} \max\left[0, \dfrac{\theta}{b}x + \left(1 - \dfrac{\theta}{b}\right)\right] & \text{if } x < \theta \\ x & \text{if } x \geq \theta, \end{cases} \tag{12}$$

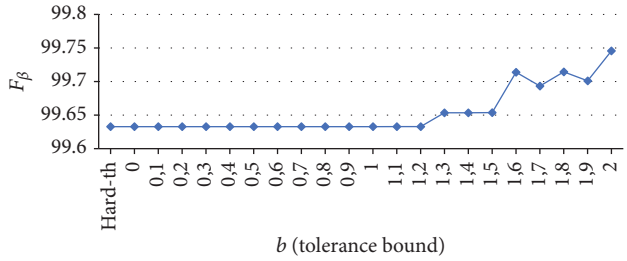


Figure 5: The F-measure depending on the tolerance bound b in the soft-threshold method. Hard-th corresponds to the F-measure value achieved by hard-thresholding.

where θ is the threshold value or the maximal edge value and b is the tolerance bound.

We chose the optimal tolerance bound b based on F_β measured on the databases mentioned in Section 8 (Figure 5). The results showed that small tolerance values do not have a significant effect, but using higher values leads to improved classification performance, with the best result achieved using a tolerance bound of $b = 2$, which was used for all experiments described in the paper. Noting that $\theta_{\text{global}} = b$, we concluded that using a global threshold for the edge values is unnecessary.

Another major deficiency of the PPED is that it is not invariant with respect to rotation. Rotation invariance can be ensured at various phases of feature-based object recognition. One approach generates a rotation-invariant feature vector applying an adequate mathematical transformation while generating the description. Another possibility is to solve

the rotation-invariance in classification technique whether by using rotationally redundant training set or by applying preprocessing on the feature vector [18–20, 30]. Since the PPED algorithmically is not rotation-invariant, only angular normalization or employing a rotationally redundant template bank may provide invariance. The latter solution can easily result in a huge and complicated database. To achieve rotation invariance, we chose to detect a characteristic angle and normalize the shape angularly. The orientation of the shape (defined as the declination of the major axis of the ellipse having the same second moment) serves well as a characteristic angle, since it is consistent in the sense that orientation values of similar shapes are close to each other (mathematical orientation may significantly deviate from the orientation value estimated by a human observer).

Orientation is a value within $(-\pi, \pi)$; thus rotation by the orientation provides invariance to rotation by $k * \pi$, resulting in two distinct possibilities. To make the rotation unambiguous, the shape is rotated by π if the center of mass of the shape is located on the right side.

To achieve scale-invariant shape analysis, the shape is normalized to fit in a window sized 64×64 pixels, preserving the original aspect ratio. It has been shown earlier that using larger sizes is unnecessary. Due to angular normalization, the shape fully fills the horizontal space; thus positioning is limited only to the vertical alignment, where the shape is moved to have the same distance between the borders and the square box on the two sides.

We summarized the construction of the EPPSED feature vector by Pseudocode 1.

5. Multilevel Classification

Adapted to the structure of the GSPPED descriptor, we propose a two-step classification method that adapts to the compound characteristics of the GSPPED descriptor, but it can be used in general as well.

Nearest neighborhood classifiers are typical when using PPED-type descriptors. The drawback of the nearest neighborhood method is that it might be slow (due to many comparisons) [31]; representation set scales poorly, and there is no option to reject inputs not belonging to any relevant class. The GSPPED, as a compound descriptor, enables us to use a special comparison method, since the parts of the vector represent different features. Compound classifiers are frequently used techniques to handle separate parts, but generally they do not exploit the meaning of each part of the vector.

We suggest a two-step classification scheme that allows using the different parts of the descriptor individually. Shape classification is performed by comparison of the descriptor to labeled points in the feature space denoted as templates. In the first step, global and statistical features are compared; then, if a satisfactory match is achieved, the final decision is computed from the differences between the contour features.

We call the set of templates used for comparison the representative set, which is a subset of the training set. Every template in the training set is labeled by its semantic class. Depending on the task, several classes are chosen as relevant ones, whereas every input vector outside of these is considered to be nonrelevant (nonrelevant classes). Although the nonrelevant subset typically comprises many classes, it can be handled as a single class due to the lack of need to differentiate between them.

5.1. Filtering. The first phase of the decision selects candidates from the representative set for the second phase by rejecting obviously dissimilar template vectors. An input descriptor matches the labeled template vector if the number of elements with a difference higher than the threshold is under a certain limit.

$$\begin{aligned} \text{comparison}\,(f,t) &= \begin{cases} \text{match}, & \text{if } E(f,t) \le \text{th}_G \\ \text{reject}, & \text{if } E(f,t) > \text{th}_G \end{cases} \\ E(f,t) &= \sum_{i \in \text{filters}} e_i(f,t) \\ e_i(f,t) &= \begin{cases} 1 & \text{if } |f_i - t_i| \le \text{th}_i \\ 0 & \text{if } |f_i - t_i| > \text{th}_i, \end{cases} \end{aligned} \tag{13}$$

where f is the input shape feature, t is the template vector, th_G is the global filtering threshold, and th_i is the threshold for the ith feature used for filtering.

Other definitions of $e_i(f,t)$ can also be considered with continuous error values; for the sake of simplicity and simple computation, we chose a discrete function.

The threshold values th_G and **th** were determined based on preliminary measurements and genetic algorithm results. The fitness value of a filter vector $\mathbf{z}$ was chosen as follows:

$$f(\mathbf{z}) = \sum_{x \in R} -\,\text{penalty}\,(x, \mathbf{z})$$

$$\begin{aligned} &\text{Penalty}\,(x, \mathbf{z}) \\ &= \begin{cases} 0 & \text{if } C(x) = \widetilde{D}(x, \mathbf{z}) \\ 1 & \text{if } C(x) \ne \widetilde{D}(x, \mathbf{z}),\ \widetilde{D}(x, \mathbf{z}) \text{ is not relevant} \\ P & \text{if } C(x) \ne \widetilde{D}(x, \mathbf{z}),\ \widetilde{D}(x, \mathbf{z}) \text{ is relevant} \end{cases} \end{aligned} \tag{14}$$

$$\widetilde{D}(x, \mathbf{z}) = \widehat{D}_R(x) \quad \text{using filters } \mathbf{z},$$

where $C(x)$ is the class of x and $\widetilde{D}(x, \mathbf{z})$ is the predicted class of element x from parameter set R using the filter vector $\mathbf{z}$. The false-positive penalty value P represents the priority between the precision and recall. If $P > 1$, the precision is prioritized, and if $P < 1$, the recall is maximized. The resulting filter values are denoted by $\mathbf{z}^*$ and were computed using fitness function defined above with $P = 50$ in 200 epochs and population size of 100 individuals.

The goal of the filtering is to reduce classification time by allowing only a highly reduced sample set into the second phase and to increase precision by excluding elements that fall close to the input in the feature space of the second phase but are trivially dissimilar based on this lower dimensional subspace. However, not only does filtering result in a slight

$$\mathrm{EV}(\rightarrow) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathrm{EV}(\searrow) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\mathrm{EV}(\uparrow) = \mathrm{rot}\,90(\mathrm{EV}(\rightarrow)) \quad \mathrm{EV}(\nearrow) = \mathrm{rot}\,90(\mathrm{EV}(\searrow))$$

```
function EPPSED(S)
    N=64
    % preprocessing
    rotate(S, -orientation(S))
    resize(S, [N, N], fit)
    if horizontal_mass_center(S) > N/2 then
       rotate(S, 180)

    directions ≔ [↑, ↗, →, ↘]
    % generate edge maps (EM) for every direction
    for dir in directions
       EM(dir) ≔ convolution2d(S, EV(dir))
    % for every location threshold the edge maps
    for i in [1..N], j in [1..N]
         for dir in directions
           θ ≔ max_{dir2 in directions}(EP(dir2)[i, j])
           EMT(dir) ≔ t_soft(EM(dir), θ)
    % project thresholded edge maps and scale them
    for dir in directions
       PR(dir) ≔ histogram(EMT(dir))
       scale(PR(dir), N/4)
    EPPSED ≔ [PR(↑), PR(→), PR(↗), PR(↘)]
end
```

Pseudocode 1

increase in precision but also we could achieve significantly higher recall rate (Figure 6).

The explanation of the anomaly is the consequence of the second phase of the classification explained in Section 5.2. The Adaptive Limited Nearest Neighborhood model learns the limits of acceptance also on filtered results and maximizes the classification precision. Assuming that filtering is based on data orthogonal to the data used in the second phase, it might filter out templates that in the second phase would determine lower acceptance radius for some instance. To verify the hypothesis, the frequency of acceptance radius lengths was measured in the function of the usage of filtering on the same representative set.

As is seen in Figure 7, filtering allows bigger acceptance radii, resulting in higher recall in the final classification. The mean of the acceptance radii is 113.4 if filtering is applied. Without filtering, the mean is reduced to 75.35, and only few representative instances have higher radius than 110 (radii shown in this paragraph are distances in the EPPSED feature space. Typically, the values of each dimension are in the interval from 0 to 200).

We also measured the speed of classification which depends on the number of comparisons to the template vectors. Filtering reduces the average lookup time by 85–95%.

Filter values were computed to fit one actual shape set; thus these values may not be suitable for other sets. Since generated values are in the same order of magnitude with the standard deviation of the measured moments on the training set, we tested the standard deviation (as well as the half and the double of the standard deviation) on the same test sets. Results are summarized in Table 1. Precision does not depend on the filter values and recall is significantly lower using the standard deviation compared to z^*; however, they are clearly higher than without filtering.

5.2. Adaptive Limited Nearest Neighborhood Classification. The second phase of the classification defines the final class of the input employing Limited Nearest Neighborhood Classification method. The reason to choose nearest neighborhood (NN) classifier is due to its suitability to be implemented on dedicated VLSI architecture; it can be easily learned and extended with new knowledge by inserting new representative instances.

An important property of the nearest neighborhood method is that there is always a nearest element to every input vector if no additional constraints are specified. This can be a disadvantage in some cases if the distance between the closest element and the input is high.

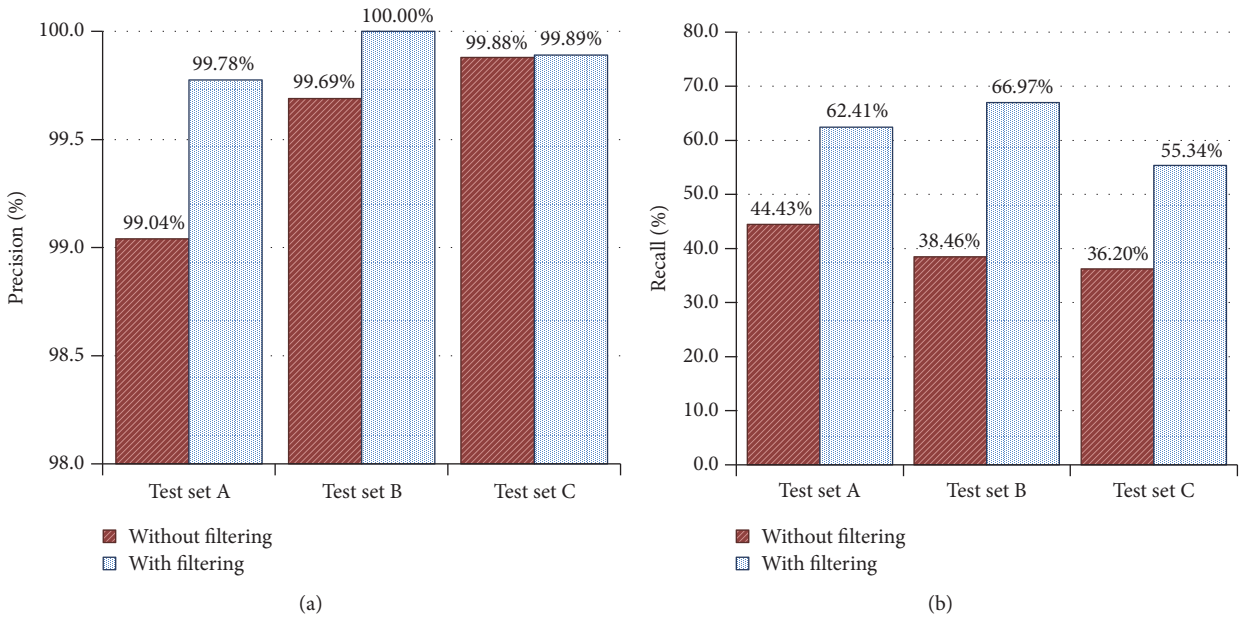


Figure 6: Classification precision (a) and recall (b), tested on three different test sets A, B, and C, depending on the usage of filtering phase.

Table 1: Recall depending on the filtering. $\mathbf{z}^*$ denotes the filter vector obtained from genetic algorithm; std denotes the standard deviation of the relevant classes.

	$\mathbf{z}^*$	std	std/2	std ∗ 2	No filtering
Test set A	**62,41%**	50,82%	54,23%	47,61%	44,43%
Test set B	**66,97%**	60,29%	62,72%	59,02%	38,46%
Test set C	**55,34%**	48,91%	47,66%	47,47%	36,20%

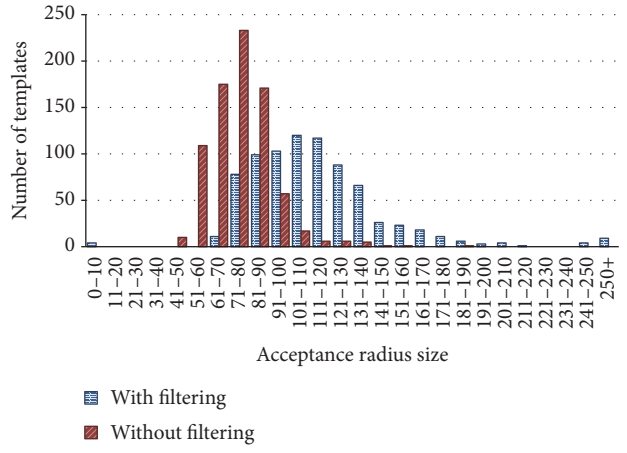


Figure 7: Frequency of the acceptance radii in case of filtering (blue dotted) and without filtering (red lines). Filtering allows the acceptance radii to take higher values in average but even zero (if no other comparable elements remain after filtering) and overly high values as well (if only few and distant templates remain to compare).

The inability to reject a hypothesis results in a type I error when irrelevant parts of the input space are not covered with training instances. To make the classifier able to maximize the precision, we propose an Adaptive Limited Nearest Neighborhood (AL-NN) method that allows the rejection of irrelevant inputs Y by defining an acceptance radius individually for all training instances ($C(Y) \in \mathcal{C}_{NRel}$, where $\mathcal{C}_{NRel}$ is the set of irrelevant classes and $\mathcal{C}_{Rel}$ is the set of relevant classes). By setting an upper bound for the distance of an input and a representative instance, we can limit the set of inputs that may get classified to the corresponding class to a hypersphere in the feature space, which we call the acceptance region of the instance.

Acceptance regions of different shapes can also be considered. If the dimensions can be typically regarded as independent and the noise is not significant, the usage of a hypercube as an acceptance region is justifiable. Employing a hyperellipsoid allows different limits in different dimensions, but it is effective only in case of low-dimensional spaces. Since we work with a 64-dimensional feature vector, the degree of freedom would be impractically high. Additionally, in the proposed edge-based shape description, dimensions are typically equivalent as the amount and distribution of noise are the same in all dimensions; dimensions are weakly dependent, and we expect similar tolerance in all dimensions; thus we opted to use hyperspheres as acceptance regions.

Using the same radius for all representative instances would be computationally easier, but it would result in a disproportionately large representative set to represent in-class regions and also boundary regions with the same

radius. Furthermore, irregular boundaries might increase the inefficiency of the cover if the radius is determined based on the radius of the highest curvature.

The acceptance radius indicates the extension of the class, the region in the feature space where the characteristics of the instance are valid. The clues in determining the acceptance radius as a boundary measure for a representative instance are the closest known instances that belong to another class and the instance with the maximal distance that belongs to the same class. In case of a relevant sample, it is worthwhile to distinguish relevant instances from other classes and irrelevant instances to make the representation more flexible.

We define the set of all irrelevant examples (N), and for every example we define the set of other instances of the same class (SP(x)) and the set of instances of all other relevant classes (OP(x)):

$$\begin{aligned} \mathrm{OP}(x) &= \{y \mid y \in R,\ C(y) \in \mathscr{C}_{\mathrm{Rel}},\ C(y) \neq C(x)\} \\ N &= \{z \mid z \in R,\ C(z) \in \mathscr{C}_{N\mathrm{Rel}}\} \\ \mathrm{SP}(x) &= \{w \mid w \in R,\ C(w) = C(x)\}. \end{aligned} \tag{15}$$

To be able to handle the three cases in a unified manner, a partial acceptance region function is introduced ($r_A^\lambda(x)$), which expresses the threshold for a given set A and a given threshold function λ:

$$r_A^\lambda(x) = \begin{cases} \lambda(\{d(x,v) \mid v \in A(x)\}) & \text{if } A(x) \neq \emptyset \\ \infty & \text{if } A(x) = \emptyset. \end{cases} \tag{16}$$

The final acceptance radius will be the smallest of the partial acceptance radii. An example x from the training set R from the class $C(x) \in \mathscr{C}_{\mathrm{Rel}}$ will get an acceptance radius $r(x)$ (Figure 8):

$$\begin{aligned} r'_{\mathrm{OP}}(x) &= \eta_{\mathrm{OP}} \cdot r_{\mathrm{OP}}^{\min}(x) \\ r'_{N}(x) &= \eta_{N} \cdot r_{N}^{\min}(x) \\ r'_{\mathrm{SP}}(x) &= \eta_{\mathrm{SP}} \cdot r_{\mathrm{SP}}^{\max}(x) \\ r(x) &= \nu \cdot \min\left(r'_{OP}(x), r'_{N}(x), r'_{SP}(x)\right), \end{aligned} \tag{17}$$

where ν serves as a shared vigilance parameter, which affects how cautious do we want to be, and can be used to move on the precision-recall trade-off curve.

In our experiments, we used $\eta_{\mathrm{OP}} = \eta_{\mathrm{SP}} = 1$, because this safely excludes other relevant samples from the acceptance region but still tries to include as many samples from its own class as possible. For the irrelevant classes, we have set the threshold parameter η_N more conservatively to 0.5 so as to enable the omission of the irrelevant elements from the representative set, as this choice does not allow acceptance regions to intersect with acceptance regions of irrelevant samples. Thus, if an input is not within the acceptance region of any relevant elements, then it is refused as being a nonrelevant input. With these parameter values, setting $\nu = 1$ results in strong preference for a high precision over a high recall rate. In this paper, where it is not specified, we used $\nu = 1$.

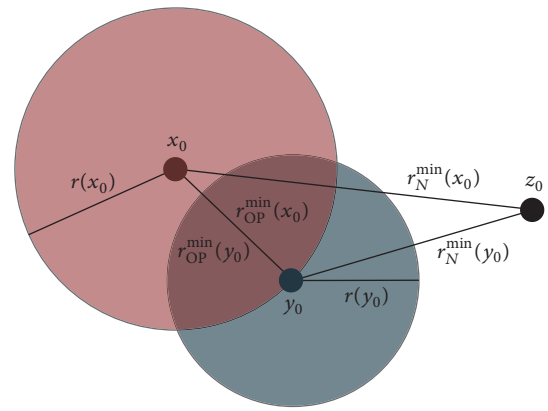


Figure 8: Definition of the acceptance range. x_0 and y_0 represent relevant elements from different classes, z_0 is the closest irrelevant element. Acceptance threshold for $y_0(r(y_0))$ is set to the half of the distance to the closest irrelevant element ($r_N^{\min}(y_0)$). Acceptance threshold for x_0 ($r(x_0)$) is chosen as the distance to the closest element of another class ($r_{\mathrm{OP}}^{\min}(x_0)$). Since z_0 is an irrelevant template, it is not included in the representative set; thus no acceptance region is defined for it.

Finding the optimal representation set in general is hard; hence, it is important that slight overrepresentations do not degrade the recall rate and thus the generalization capability of the model does not change considerably. This is satisfied by the formula proposed above, as it does not let the radius of the acceptance region to decrease if a new instance of the same class is added to the representative set.

6. Optimizing the Representative Set

Another major disadvantage of the nearest neighborhood classification is that the manually built training/representative set might be disproportionately large, making the classification very slow.

The representative set can be optimized by eliminating unnecessary points so that the resubstitution results do not change significantly on the training set. Omitting points may lead to a small decrease of the cover, but most of the omissions can be regarded as noise filtering, thus making the model eventually more robust.

Selection of unnecessary points can be carried out based on the analysis of the representative set by minimizing the set size while preserving approximately the same cover. A point Y is unnecessary (U) from the aspect of classification if the classification result remains the same for all the points of the space (i.e., for an arbitrary input) if Y is removed from the representative set:

$$U = \{\mathbf{x} \in F\colon D_R(\mathbf{x}) = D_{R\setminus\{Y\}}(\mathbf{x})\}, \tag{18}$$

where F is the feature space and $D_R(\mathbf{x})$ is the decision for feature vector x using the model learned by representative set R.

In a nearest neighborhood model, the class is determined by the nearest labeled point. A representative instance Y is unnecessary if, for every point in the feature space that is

classified to Y as the closest template point, the second closest template point belongs to the same class as Y.

$$\begin{aligned} &\forall \mathbf{x} \in \mathbf{F}\ \exists \mathbf{Z} \in \mathbf{R} \setminus \{\mathbf{Y}\}\ \forall \mathbf{w} \in \mathbf{R} \setminus \{\mathbf{Y}\}: \\ &d(\mathbf{x}, \mathbf{Y}) \leq d(\mathbf{x}, \mathbf{w}) \longrightarrow \\ &d(\mathbf{x}, \mathbf{Z}) \leq d(\mathbf{x}, \mathbf{w}), \end{aligned} \tag{19}$$

where $d(x, y)$ is the distance between points x and y.

The boundary surface B between classes is the set of points that are equally distant from support vectors of different classes. A template point Y is unnecessary if it does not influence the boundary surfaces. In an n-dimensional feature space, such a boundary surface is $n-1$-dimensional, and apart from the singular cases when points lie in one hyperplane, complete shadowing of Y can be achieved with at least n necessary points of the same class. Therefore, if the number of representative points of a class and the dimension of the feature space have the same order of magnitude, only a negligible portion of the representative set can be unnecessary.

In the Adaptive Limited Nearest Neighborhood method proposed above, acceptance regions of each representative instance provide a good estimate of their contribution to the global cover. We propose an iterative optimization algorithm for the Adaptive Limited Nearest Neighborhood classification which reduces the mutual cover of the representative set elements. As initialization, the points of the representative set are ordered in a queue P. The set S is initialized as empty:

$$\begin{aligned} &S \coloneqq \emptyset \\ &P \coloneqq (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}), \\ &\forall x_i, x_j\ i < j \longrightarrow \\ &H_m(\mathbf{x_i}) < H_m(\mathbf{x_j}) \\ &H_m(\mathbf{x_i}) = \sum_{\substack{j=1 \\ j \neq i}}^{n} h_m(\mathbf{x_i}, \mathbf{x_j}) \end{aligned} \tag{20}$$

$$h_m(x_i, x_j) = \begin{cases} 1, & \text{if } d(\mathbf{x_i}, \mathbf{x_j}) \leq m \cdot r(\mathbf{x_i}) \\ 0, & \text{if } d(\mathbf{x_i}, \mathbf{x_j}) > m \cdot r(\mathbf{x_i}), \end{cases}$$

where $r(\mathbf{x_i})$ is the acceptance radius of $\mathbf{x_i}$ and $H_m(\mathbf{x_i})$ is the number of instances in the representative set which are closer to $\mathbf{x_i}$ by $m \cdot r(\mathbf{x_i})$.

The first element of P is taken out from P and moved to S, and all other instances are removed from P which are covered by it.

$$\begin{aligned} &p \coloneqq P[1] \\ &S \coloneqq S \cup \{p\} \\ &\text{remove } \{x \in P \mid C(p, x) = 1\} \text{ from } P. \end{aligned} \tag{21}$$

The iteration ends when P is empty, and S will be the reduced representative set.

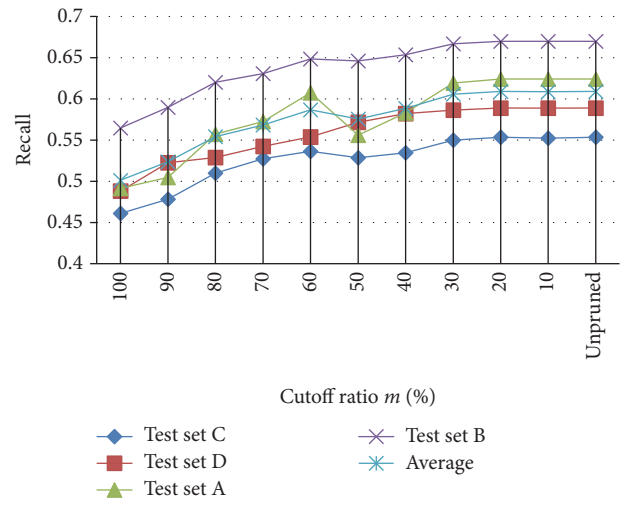


Figure 9: Recall as a function of the cutoff ratio.

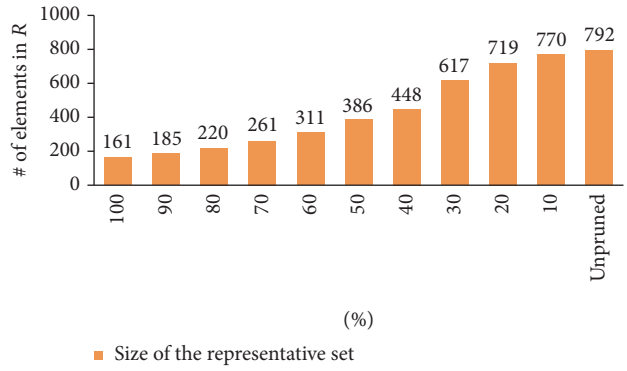


Figure 10: Size of the representative set as a function of the cutoff ratio.

We tested the optimization algorithm from cutoff ratio $m = 1$ to $m = 0$, where $m = 1$ stands for deleting any covered representative element and $m = 0$ stands for unpruned model, where even identical elements may remain in the set. Results show that from $m = 1$ to $m = 0.5$ the recall increases from 0.5 to 0.6 nearly linearly, and from $m = 0.5$ to 0 only a small increase can be noticed. Almost parallel to that, the size of the reduced representative set increases slightly to $m = 0.4$ and after a significant increase saturates after $m = 0.2$. The size of the resulting representative set is shown in Figure 10, and the recall depending on the cutoff ratio is shown in Figure 9.

We also tested different orderings of the set P. Ordering based on acceptance radius showed lower performance with the same representative set size. Ordering based on the ratio of the volume of intersections and the acceptance hypersphere produced almost the same results as the method above but with significantly more complex computation. Another way to optimize the representative set is to perform several tests and remove instances that did not play a role in a certain number of classifications. However, this empirical method would require additional data that cover the largest portion of the feature space.

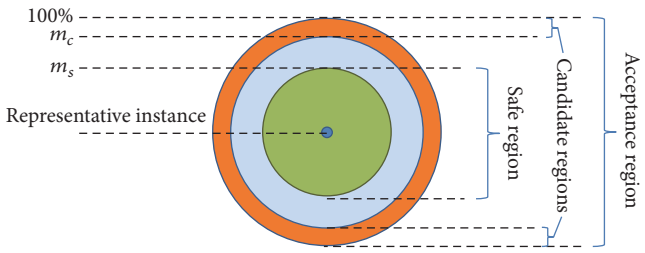


Figure 11: Acceptance region, safe region, and candidate region of a representative instance in the feature set.

7. Extending the Representative Set

Adaptive extension of a learned AL-NN model can be carried out easily by inserting new point to the representative set and setting the acceptance radius of the inserted element based on the original training set. The extension can bring higher recall rates by covering previously uncovered regions of the feature space.

The main challenge in extending the representative set is to select new instances to be inserted adequately. On one hand, to cover new areas, a candidate has to be far from the existing representative elements. On the other hand, an automatic update should only insert elements that are classified correctly with a reasonably high confidence, that is, ones close to a labeled point. If both conditions hold, we declare the insertion of the new instance to be safe.

As we showed in Section 5.2, the acceptance regions clearly bound the coverage in the representative set; thus both conditions can be formalized based on acceptance thresholds. The real challenge in selecting new instances is that the two conditions are contradictory. To resolve the contradiction that the distance of the candidate sample should be low (for high confidence) and high (to gain significant coverage) at the same time, we rely on temporal information.

We developed an automatic extension algorithm for the AL-NN model, which uses temporal information in the update method, if available (Figure 11). We define a decision to be safe if a test set element is closer to the representative example than the half of its acceptance threshold.

$$d(t,c) \leq m_s r(t) \qquad m_s = 0.5. \tag{22}$$

The choice of the value $m_s = 0.5$ was based on the quick decision (presented in Section 5.2) threshold.

An element is chosen as a candidate for insertion if it is at the edge of the acceptance region.

$$d(t,c) \geq m_c r(t). \tag{23}$$

A candidate is only inserted into the representative set if neighboring frames contain patches that were classified in the same class with a safe decision. The radius of neighboring frames is chosen based on the processing frame rate and the median translation of the image. In the shape set we used, the total processing time is between 0.1 and 0.3 seconds, while the images were taken by a cell phone camera moving slightly upon a table; thus the frame radius was set to involve only directly neighboring frames.

We tested the extension with $m_c = 0.5$ to $m_c = 0.95$. We added new elements from three different test sets (test sets A, B, and C) and measured the improvement on an independent test set (test set D). Results are summarized in Table 2. Details of the test sets are described in Section 8.

Table 2: Results of online learning algorithm depending on the candidate ratio m_c.

m_s	# of added instances	# of new recognized instances
0.5	91	35
0.75	24	15
0.9	8	15
0.95	4	6

Table 3: Experimental results of the proposed GSPPED shape descriptor and the two-level classification algorithm including the AL-NN classification. Test sets A-D contain shape images from live tests performed with participation of visually impaired subjects; test set E was generated in laboratory.

Test set	F_β	Precision	Recall	Images
Test set A	99,63%	99,78%	62,41%	7008
Test set B	99,88%	100%	66,97%	6482
Test set C	99,69%	99,89%	55,34%	6171
Test set D	99,54%	99,71%	58,89%	13895
Test set E	99,93%	99,95%	92,94%	13113

8. Experimental Results

The description and the classification method presented in this paper have been tested in the framework of the Bionic Eyeglass. The Bionic Eyeglass [32, 33] is a portable device to help blind and visually impaired people in everyday navigation, orientation, and recognition tasks that require visual input. The development of the device is ongoing at present; the finalized algorithms are now implemented on different platforms (Android, iOS, and FPGA). The Bionic Eyeglass integrates several functions requested by visually impaired people, namely, banknote recognition [34], crosswalk detection, and public transport number reader.

The five shape datasets contain several thousands of shape images, including irrelevant inputs that do not belong to any class. The GSPPED was extracted in average of 29.5 milliseconds on a standard computer (Core2 Quad CPU @ 2.66 GHz, 4 GB memory). The test results and the exact size and source of the test sets are indicated in Table 3.

The representative set was produced from a training set containing 1073 shapes; the initial representative set contained 792 shapes. Shapes in the test sets represent characteristic graphical patches (portraits and drawings) extracted from banknotes and also shadows, joined patterns, and other patches from the background. We used 9 relevant classes containing highly varying shapes due to morphologic

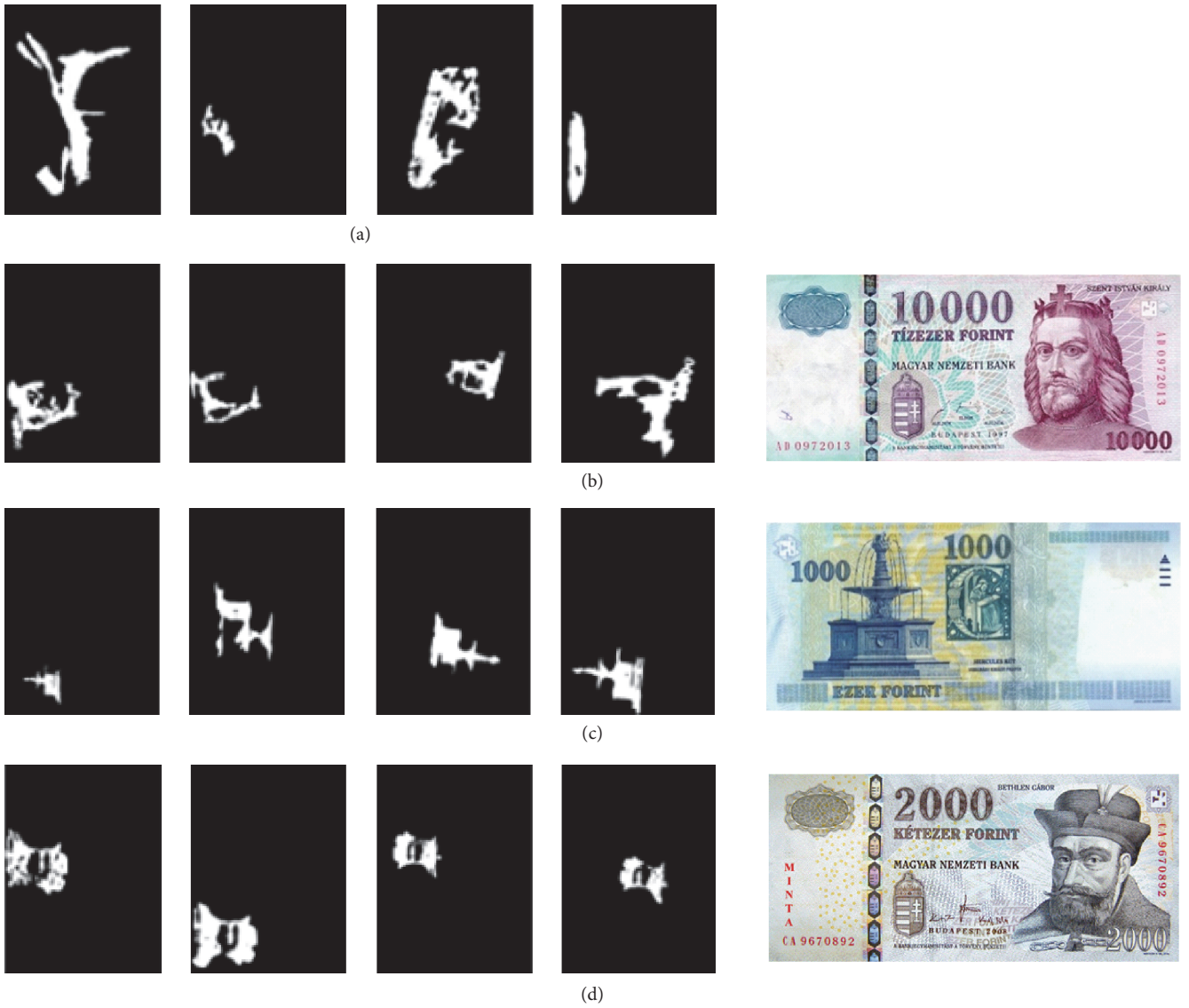


Figure 12: Fragment of the test sets. Row (a) shows shapes from irrelevant classes and rows (b–d) show relevant shapes of banknotes; the source banknotes are shown next to the shape images.

extraction. The average lookup time was 1.8 ms. Examples of input images are shown in Figure 12.

We compared our results achieved on test sets A–D (excluding test set E) with other shape descriptions (Complex Zernike Moments and Generic Fourier Descriptor) and classification methods. To allow for different weights for prioritization of precision over recall, AutoMLP, FF-NN, and SVM models were trained using a cost matrix with false-positive to false-negative penalty rates ranging from 5 to 100; in case of the AL-NN, we changed the vigilance parameter ν from 1.0 to 1.25, with an appropriate adjustment of the filter limit vector $\mathbf{z} = \nu^2 \mathbf{z}^*$.

First we compared the AL-NN classifier to a feed-forward neural network (FF-NN), an AutoMLP, a k-NN model, and a SVM on the shape feature vectors obtained from the GSPPED.

The best results were reached by the neural networks, FF-NN, and AutoMLP. We tested the FF-NN containing 2 to 5 hidden layers and trained from 100 to 1000 epochs. AutoMLP was trained for 20 cycles of 10 generations and 5 MLPs per ensemble. The best performances achieved by the models are shown in Figure 13. Since the SVM (with radial basis function and polynomial kernels) and the k-NN (for k from 1 to 10) models could not achieve precision rate above 90% with any parametrization, they are not included in Figure 13.

In order to investigate the efficiency of the GSPPED shape descriptor, we compared it with the Generic Fourier Descriptor (GFD) [35] and with the Complex Zernike Moments Descriptor (CZMD) [36, 37], trained on the same train set, using the AutoMLP classifier. The feature vector of the GFD contained 85 elements with angular frequency of 16 and radial frequency of 4. The CZMD contained 121 feature

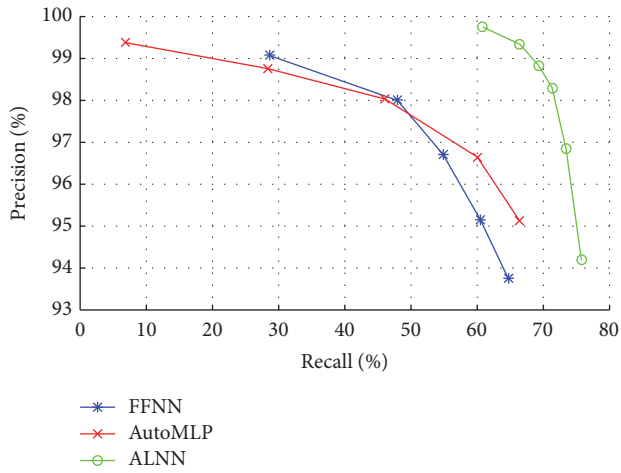


Figure 13: Precision and recall of the shape classification by FF-NN, AutoMLP, and the presented Adaptive Limited Nearest Neighborhood (AL-NN) classification. The source data is constructed by the GSPPED shape descriptor.

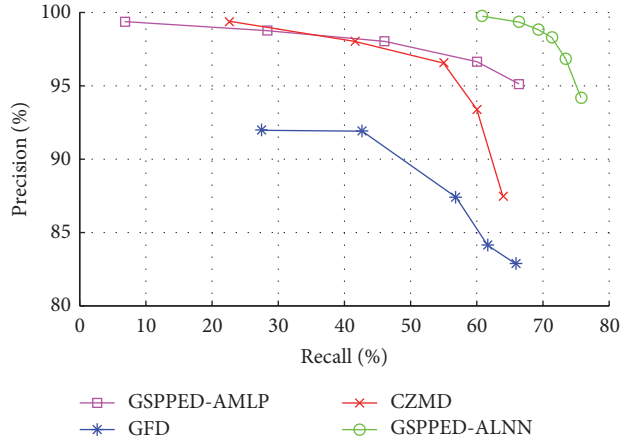


Figure 14: Precision and recall of the shape classification, comparing the performance of the Complex Zernike Moments Descriptor, the Generic Fourier Descriptor, and the GSPPED descriptor.

elements with the highest order of 20. Results are shown in Figure 14.

The GSPPED and the Complex Zernike Moments Descriptor evidently outperform the Generic Fourier Descriptor. When classified by the AutoMLP, the GSPPED slightly outperforms the CZMD; however, with high penalty coefficient, the CZMD provides better recall.

We also compared the descriptors based on McNemar's test. CZMD and GSPPED with AL-NN differ significantly ($p = 1.27e - 4$), and GSPPED with AL-NN also exceeds the performance compared to GFD significantly ($p < 1e - 15$).

8.1. Effect of Noise. To measure the sensitivity of the developed shape description and classifier, we repeated the test on noisy images and compared the results with the results obtained with the Complex Zernike Moments Descriptor and the Generic Fourier Descriptor. Based on our datasets, we observed that deviations in the extracted shape images do not occur in pixel-level additions or removals but in joining with other blobs or in removal of some parts of the shape (also see Figure 15). To model this kind of noise, we added and removed several randomly generated blobs to and from the original shape. The total area of the blobs is given as a ratio (w) to the shape area.

In the case of the CZMD and GFD, results show consistent decrease both in recall and in precision. GSPPED provides lower recall on high noise ratio than the other two descriptors; however, the precision is significantly higher compared to the CZMD and the GFD (Figure 16). These results might highlight the nature of GSPPED and AL-NN: the generalization capability of the AL-NN classification method using GSPPED is somewhat limited, but it is still comparable to other methods; at the same time, this combination provides outstanding discriminative power.

9. Conclusion

We presented a new shape description and classification method. Key characteristics of our approach are the compound descriptor and classifier that join the region and contour-based features. We suggested an online learning method to extend the representative set and increase performance. We proposed a representative set optimizing algorithm as well.

The core idea behind our method is the two-level description and classification: for an input shape, low-level, global statistical information is extracted to roughly select the set of similar objects and to reject obviously different templates. In the second stage, local edge information is investigated to find the closest known shape but with the ability to reject the match. The refusal is based on the acceptance radius that is specified individually for every item in the representative set according to the properties of the local proximity in the feature set.

Results demonstrate a high precision rate (99.83%) and an acceptable recall rate (60.53%), which fulfil the requirements for a safety-oriented visual application processing an image flow. The reason to have lower cover is that input frames contain highly deformed shapes, which, for sake of reliability, are classified as nonrelevant inputs. The recall is acceptable, as long as a continuous input is available. Compared to other classifiers, none of the tested ones could outperform the AL-NN in precision, and the same recall could only be reproduced with significantly lower precision. If a final decision is made based on multiple input frames and multiple clues, the false-positive error can be minimized to be practically negligible.

The computation time of the descriptor (~30 ms) and the classification time (~2 ms) allow real-time recognition even on standard CPUs in computers and phones, and the architecture core of the algorithm is easily adaptable to locally connected cellular array processors.

The proposed algorithms were implemented on cell phones and FPGAs with the purpose of providing a reliable vision aid for blind and visually impaired people. One of the drawbacks of the GSPPED we have found is the high

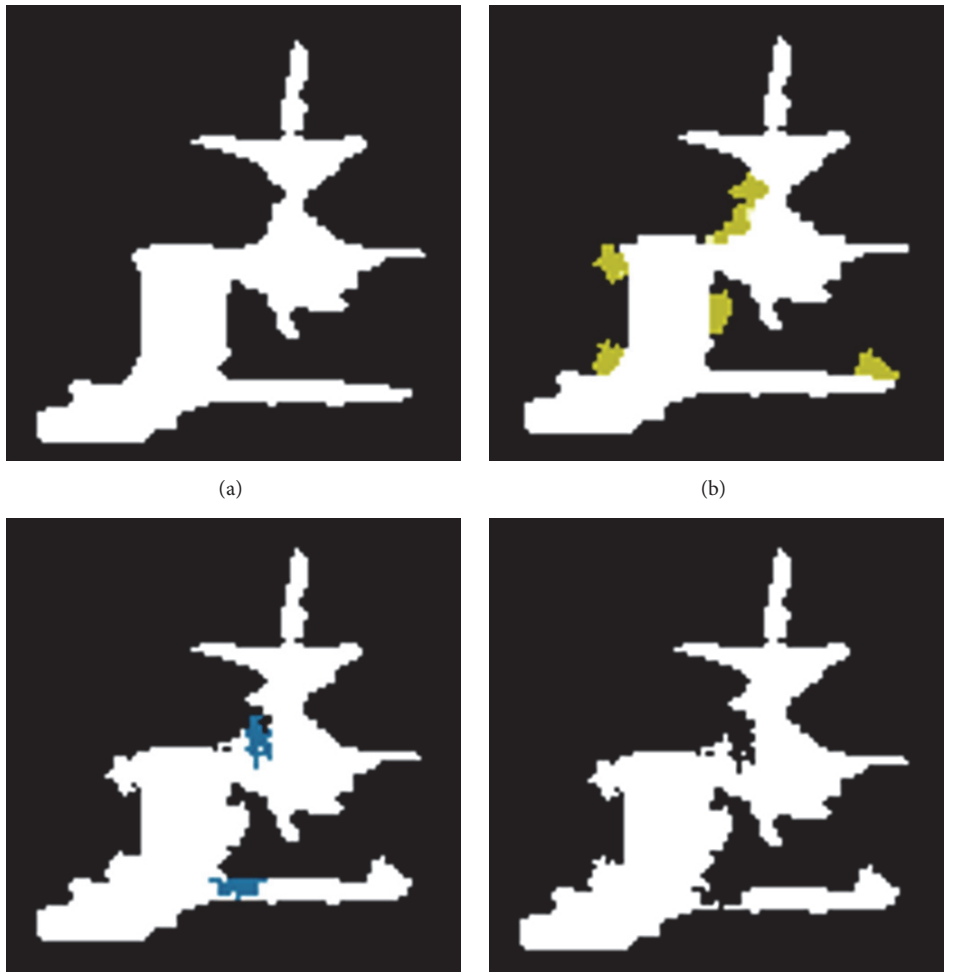


Figure 15: Example of blob-level shape noise with manipulation ratio $w = 0.2$. In (a), the original shape is shown, in (b), the additions are shown, in (c), removals are highlighted, and the final noisy shape is shown in (d).

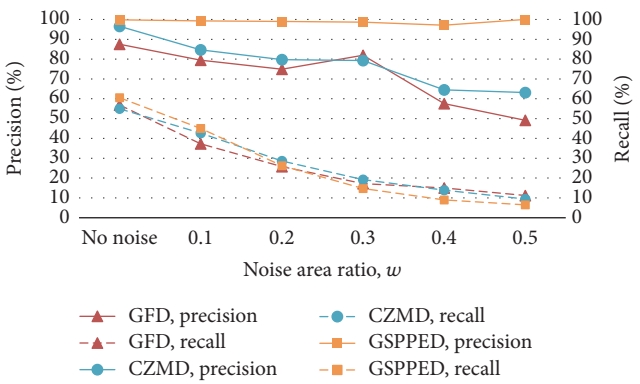


Figure 16: Classification recall and precision of the GSPPED, the Complex Zernike Moments Descriptor, and the Generic Fourier Descriptor, depending on the noise ratio w that represents the ratio of the number of manipulated pixels to the total area of the shape. The CZMD and GFD features were classified by the AutoMLP algorithm, while GSPPED features were classified with the AL-NN method.

sensitivity to positioning and scaling, depending on minor variations. We will focus on designing and employing a more robust translation and scale normalization method. We also plan to investigate the possibility of taking more training elements into account when defining the acceptance threshold, similar to the k-NN method.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to highlight the contributions of the late Tamás Roska to this work. They are very grateful for his ideas and guidance that formed an essential basis of the whole research work. The support of the Swiss Contribution, the Bolyai János Research Scholarship, and the Pázmány Peter Catholic University is gratefully acknowledged.

References

[1] A. Andreopoulos and J. K. Tsotsos, "50 Years of object recognition: directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[2] A. Andreopoulos, S. Hasler, H. Wersing, H. Janssen, J. K. Tsotsos, and E. Körner, "Active 3D object localization using a humanoid robot," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 47–64, 2011.

[3] J. Tsotsos, "The Encyclopedia of Artificial Intelligence," in *Image Understanding*, pp. 641–663, John Wiley and Sons, Canada, 1992.

[4] S. Dickinson, "What is Cognitive Science?" in *Object Representation and Recognition*, pp. 172–207, Basil Blackwell publishers, Object Representation and Recognition, 1999.

[5] K. Prasad, "Dilip, Survey of the problem of object detection in real images," *International Journal of Image Processing*, vol. 6, no. 6, p. 441, 2012.

[6] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[7] S. A. Dudani, K. J. Breeding, and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE Transactions on Computers*, vol. 26, no. 1, pp. 39–46, 1977.

[8] L. Gupta and M. D. Srinath, "Contour sequence moments for the classification of closed planar shapes," *Pattern Recognition*, vol. 20, no. 3, pp. 267–272, 1987.

[9] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, vol. 54, Academic Press, New York, NY, USA, 1991.

[10] P. J. van Otterloo, *A Contour-Oriented Approach to Shape Analysis*, Prentice-Hall International (UK) Ltd, New Jersey, NJ, USA, 1991.

[11] A. C. Evans, N. A. Thacker, and J. E. W. Mayhew, "Pairwise representation of shape," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, vol. 1, pp. 133–136, IEEE, The Hague, Netherlands, 1992.

[12] H. Asada and M. Brady, "The Curvature Primal Sketch," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 2–14, 1986.

[13] G. Eichmann et al., "Shape representation by Gabor expansion," in *Proceedings of the Hybrid Image and Signal Processing II*, vol. 1297 of *SPIE*, pp. 86–94, 1990.

[14] Q. M. Tieng and W. W. Boles, "Recognition of 2D object contours using the wavelet transform zero-crossing representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 910–916, 1997.

[15] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, 1961.

[16] W. I. Grosky and R. Mehrotra, "Index-based object recognition in pictorial data management," *Computer Vision Graphics and Image Processing*, vol. 52, no. 3, pp. 416–436, 1990.

[17] M. K. Hu, "Visual pattern recognition by moment invariant," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[18] H. S. Kim and H.-K. Lee, "Invariant image watermark using zernike moments," *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[19] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[20] R. B. Yadav, N. K. Nishchal, A. K. Gupta, and V. K. Rastogi, "Retrieval and classification of objects using generic Fourier, Legendre moment, and wavelet Zernike moment descriptors and recognition using joint transform correlator," *Optics & Laser Technology*, vol. 40, no. 3, pp. 517–527, 2008.

[21] C.-H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496–513, 1988.

[22] J. Iivarinen, M. Peura, J. Srel, and A. Visa, "Comparison of combined shape descriptors for irregular objects," in *Proceedings of the 8th British Machine Vision Conference*, 1997.

[23] M. Hasegawa and S. Tabbone, "A shape descriptor combining logarithmic-scale histogram of radon transform and phase-only correlation function," in *Proceedings of the 11th International Conference on Document Analysis and Recognition, ICDAR '11*, pp. 182–186, September 2011.

[24] S. Khanam, S. Jang, and W. Paik, "Shape retrieval combining interior and contour descriptors," in *Proceedings of the International Conference FGCN*, pp. 120–128, 2011.

[25] T. G. Dietterich, "Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.

[26] T. G. Dietterich, *Multiple Classifier Systems, Chapter Ensemble Methods in Machine Learning*, vol. 1857 of *Lecture Notes in Computer Science*, 2000.

[27] D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[28] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman and Hall Computing, Boca Raton, Fla, USA, 1993.

[29] M. Yagi and T. Shibata, "An image representation algorithm compatible with neural-associative-processor-based hardware recognition systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 5, pp. 1144–1161, 2003.

[30] M. Yang, K. Kpalma, and J. Ronsin, "A survey of shape feature extraction techniques," Peng-Yeng Yin, Pattern Recognition, IN-TECH, pp. 43–90, 2008.

[31] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Supervised machine learning: a review of classification techniques," *Informatica*, vol. 31, no. 3, pp. 3–24, 2007.

[32] K. Karacs, A. Lázár, R. Wagner, D. Bálya, T. Roska, and M. Szuhaj, "Bionic eyeglass: an audio guide for visually impaired," in *Proceedings of the IEEE Biomedical Circuits and Systems Conference Healthcare Technology, BioCAS '06*, pp. 190–193, IEEE, London, UK, December 2006.

[33] K. Karacs, M. Radvanyi, A. Stubendek, and B. Bezanyi, "Learning hierarchical spatial semantics for visual orientation devices," in *Proceedings of the 10th IEEE Biomedical Circuits and Systems Conference, BioCAS 2014*, pp. 141–144, Switzerland, October 2014.

[34] A. Stubendek, K. Karacs, and T. Roska, "Shape description based on projected edges and global statistical features," in *Proceedings of the International Symposium on Nonlinear Theory and Its Applications (NOLTA '14)*, 2014.

[35] D. Zhang and G. Lu, "Shape-based image retrieval using generic Fourier descriptor," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 825–848, 2002.

[36] A. Tahmasbi, F. Saki, and S. B. Shokouhi, "Classification of benign and malignant masses based on Zernike moments,"

Computers in Biology and Medicine, vol. 41, no. 8, pp. 726–735, 2011.

[37] F. Saki, A. Tahmasbi, H. Soltanian-Zadeh, and S. B. Shokouhi, "Fast opposite weight learning rules with application in breast cancer diagnosis," *Computers in Biology and Medicine*, vol. 43, no. 1, pp. 32–41, 2013.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 8738316, 11 pages
https://doi.org/10.1155/2018/8738316

Research Article

A Multitarget Visual Attention Based Algorithm on Crack Detection of Industrial Explosives

Haibo Xu, Buhai Shi, and Qingming Zhang

School of Automation Science and Engineering, South China University of Technology, Wushan Rd., Tianhe District, Guangzhou 510640, China

Correspondence should be addressed to Haibo Xu; 201510101991@mail.scut.edu.cn

Received 4 September 2017; Revised 26 January 2018; Accepted 11 February 2018; Published 26 March 2018

Academic Editor: Marco Perez-Cisneros

This paper is a novel study on crack detection of industrial explosives. The proposed algorithm consists of the following steps: (1) image preprocessing was performed according to the defect features of industrial explosives cartridge, and we developed an improved visual attention based algorithm. This proposed algorithm features a parametric analysis that can be implemented on the image according to the conspicuous maps with the introduction of the concept of defect discrimination ξ; (2) as compared with other algorithms, our method can realize real-time multitarget detection function; (3) a new analysis method, the IPV-WEN algorithm, was proposed to analyze the cartridge defects based on performance indices. Through comparison and experimentation, it was revealed that this method can achieve a detection accuracy of 97.9%, with detection time of 34.51 ms, which satisfied the requirement in the industrial explosives production.

1. Introduction

With the rapid development of the Chinese economy and the technical innovation in the industry of industrial explosive materials, the production scale of industrial explosives has been continuously expanding. Under such circumstance, the vigorous improvement in the continuity and automation level of the production line has become an inevitable trend. However, due to various factors, such as malfunctioning of automatic industrial explosives packing equipment, quality of the raw materials, and interference to the production environment, numerous defects acquired during the packaging process might be detected on explosive cartridges, which can affect the efficiency and quality of explosives production [1–3]. Therefore, real-time and efficient defect detection and classification of industrial explosives have become key factors for the improvement of production quality and personnel security.

Crack defect is one of the most common detectable cartridge defects. This is a defect on the surface of the cartridge, which looks normal in shape but has breaks on the surface. It mainly refers to the difference in the crack information between the cartridge and the standard cartridge. It indicates the brightness inconformity between the pixel subset and pixel block within the standard cartridge. It can also be attributed to the uncertainty in the distribution of crack defects and randomly distributed crack scale, depth, and position, thus making the forecasting difficult. Furthermore, the interference from surface text and trademark texture feature also increases the difficulty of defect detection. In the practical production process, numerous cartridges have been found with such defect mainly due to the poor heat-sealing on the side edge. The correlations between the manifestation of the crack defect and illumination intensity are illustrated in Figure 1.

As the defects are usually found on cartridge packages during production and packaging of industrial explosive cartridges, this paper aims to provide a solution to the key problem through the introduction of a proper algorithm that can effectively extract the packaging characteristics of the cartridges and locate the defective cartridges, which must be separated from the normal ones. According to the specific conditions of the industrial explosive production line, this paper proposes a visual attention based search strategy to

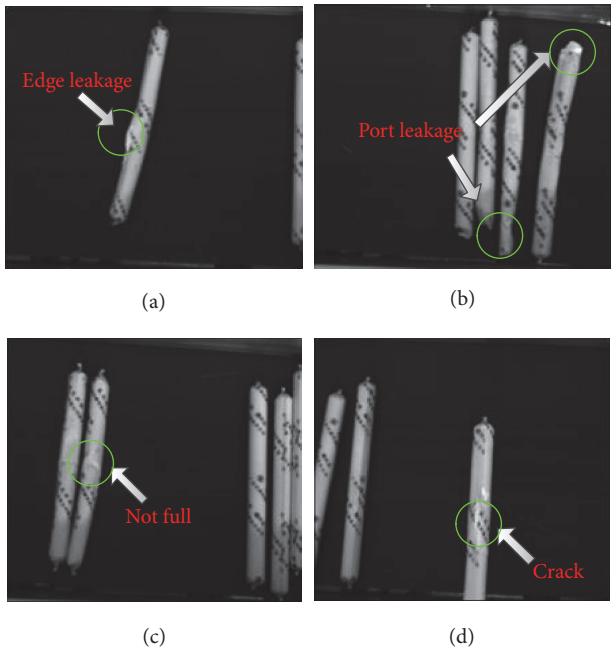


Figure 1: Classification for cartridge defect detection. (a) and (b) images are, respectively, side edge leakage cartridge and port edge leakage cartridge; (c) and (d) images are, respectively, not full cartridge and cartridge crack detection.

prevent the acquired cartridge image from being affected by the natural light and to reduce the noise interference caused by the following stages. However, due to the random distribution of the salient characteristics and interference from the surface text, there may be the existence of multiple defect targets, which are shown in Figure 1. Therefore this paper adopts an improved visual attention based search algorithm to extract the crack characteristics of the cartridge.

The paper is organized as follows: Section 2 discusses and reviews the related defect detection algorithms. In Section 3, an improved visual attention based algorithm was proposed. This method can be applied to the multitarget crack detection. Simulation experiments are described in Section 4. An analysis algorithm, called image partitioning variance-weighted eigenvalue (IPV-WEV) based algorithm, was proposed in Section 5; and Section 6 presents the conclusion of the paper.

2. The Relevant Visual Attention Algorithms

The research on the machine vision system started very early in foreign countries. Malamas et al. introduced the application of the industrial visual detection system, system composition, main approaches for industrial visual application, and main hardware and software of the system [4]. They also made a general review on the four types of detection applications, namely, dimensional quality, surface integrity, structure quality, and operation quality, according to the detection objects and the procedural features. Moganti et al. made an overview of the application of industrial detection technologies to the manufacture of printed circuit boards [5]. Recently, researches on the application of machine vision technology for the inspection of product quality are becoming increasingly popular. Moreover, due to the higher requirements on product packaging inspection and surface defect detection, positioning, and recognition, this technology has been widely applied to the production of drugs, videos, mechanical parts, electronics, textile products, and so on [6]. Their research mainly focused on feature selection and extraction and classification of feature patterns.

Li et al. proposed a rather robust algorithm that is free from the interference produced by dirt on the surface of the eggshell to extract the cracked eggshell and detect the presence of tiny cracks. Furthermore, it can be used to train the neural network according to the image pixel density histogram [7]. Razmjooy et al. proposed an appearance detection solution with scale measurement [8]. They focused on the application of mathematical methods to the automation, especially equation solving through the design, implementation, and classification of algorithms to make a simple classification of the images based on scales through the binarization processing. Jia et al. modified the image recognition method after analyzing the radiation characteristics of the heating components [9]. They filtered out the infrared radiation interference to the image information obtained with a charge-coupled device camera through a low-pass filter. In addition, Shen et al. designed a new illuminated image recognition system [10]. There are three bearings in an image, and the bearings on the left and right were used to detect the distortion defect, whereas the bearing in the middle was used to detect other defects near the deformation point. Tellaeche et al. proposed two approaches, namely, image segmentation and decision making [11]. First, they segmented the image into multiple regions with the same scale. Then, they used the support vector machine (SVM) classifier to analyze the features after the extraction of the features and attributes of every region. The results of the analysis determined the presence or absence of weeds in a certain area. In dimensional measurement and shape detection, the object scale was measured based on an image to estimate if the scale is within the permitted tolerance. They are used to detect if the shape or scale of the object conforms to the requirement. With regard to the applications of dimensional measurement and shape detection, Jiménez et al. proposed the identification of fruit on the trees based on the shape recognition algorithm [12]. He used the laser ranging method to locate the positions of the fruits for automatic harvesting. According to the model-based computer vision method, Magee and Seida designed and implemented a detection system that can be used to estimate the shape of the object in a complex industrial environment [13].

Operation quality detection was implemented to verify if an accurate operation has been performed on the tested products as per manufacturing processes or standards. To make a classification that is suitable for riding modes, Gao and Duan started by describing the main characteristics of different images according to L distance and then applied SVM [14].

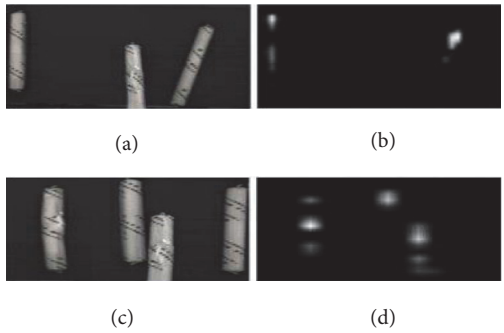


Figure 2: False detection. (a) and (c) are source images ((a) image has one defect region, whereas (c) image has two defect regions). (b) and (d) are the detection results by GBVS.

3. An Improved Visual Attention Based Algorithm

There are two processes that influence visual saliency: top-down and bottom-up processes. Visual attention based algorithm reveals the mechanisms of biological visual intelligence [15–19]. It has wide applications in various locations as reported by some computer vision researchers [20–26]. Li et al. analyzed the visual saliency in frequency domain and employed hypercomplex Fourier transform algorithm [27]. In recent years, some new computational models based on visual saliency were proposed in the welding industry, agriculture, food inspection, and so on [28–32]; however, it is fruitless by graph-based visual saliency (GBVS) [33] when the object is a gray-level image, as in Figure 2.

Feature extraction refers to the extraction of cartridge crack defect characteristics. In this section, an improved visual attention based algorithm is adopted to extract the defect characteristics of the diagnostic cartridge. This algorithm can simulate the neural structure and behaviors of the visual system of primates in their early lives. It demonstrates powerful capacity in the real-time processing of complex scenes. With the integration of multiscale image features into a topological saliency map, it can quickly select the prominent positions through an efficient computing method. Then, it can identify the crack defect on cartridge after further detailed analysis. Currently, all of the common visual attention based algorithms are mainly used to detect the defects or scratches among textural properties of products with neat textures, such as ceramic tile, cotton, or cloth. In this case, the global feature extraction algorithm was chosen as the processing algorithm to process the image before the application of visual attention based algorithm. Furthermore, this algorithm makes the calculation mainly based on the following two characteristic parameters in statistics, namely, mean value and standard deviation, according to the different window scales. The application of visual attention based algorithm, along with the specific procedures, is shown in Figure 3.

We established a feature-based model according to the following three performance indices, namely, edge, intensity, and orientation. Regarding the brightness feature, the pyramid operators were instructed to generate a nine-scale diagram, which ranged from 1 : 1 (0 scale) to 1 : 256 (8 scales) in the range of 8 octaves. Then, it was implemented through the calculation of the difference between the fine and coarse scales. Meanwhile, we used the Gabor and Roberts operators [33] separately to generate a direction feature pyramid and an edge feature pyramid for the image. The image pyramid is a simple but efficient tool used to interpret image through a multiresolution method. It was applied for image compression and machine vision in the early stages. The high-definition images were located at the bottom of the pyramid. Moreover, the pyramid level positively correlates with image resolution.

We employed Gaussian pyramid, Gabor operator, and Roberts operator to extract the intensity, orientation, and edge features, respectively. The feature template was calculated based on the center-surround operator. Center extraction can be implemented through the calculation of the difference between the fine and coarse scales. For the center-surround operator, normalization can play a role in weakening the similarity between the images and increasing the difference according to the following principle: firstly, the values of the computed image were normalized by the range of $[0, M]$, and the maximum value M and the mean of other local maximums $\overline{m}$ were calculated in an image. Then, the whole image was multiplied with $(M - \overline{m})^2$. Through the above operation, the feature image of intensity, edge, and orientation was obtained.

Through multiscale combination, the acquired feature templates were processed based on multiscale composition operators. Then, normalization was employed after the image interpolation from coarse to fine scale and a subtraction calculation on a point-to-point basis. We adopted the discrimination fusion operator to integrate the previously mentioned three conspicuous maps of edge (E), intensity (I), and orientation (O) into a saliency map SM as follows:

$$\mathrm{SM} = N\left(\mathrm{Comb}\left(I, O, E\right)\right), \tag{1}$$

where Comb(•) represents the discrimination fusion operator and N(•) denotes normalization operator.

Although it is a simple image fusion method with the adoption of linear weighted fusion operator in an Itti model, it can lead to the omission of lots of information during the fusion. Therefore, this paper proposes a defect discrimination-based fusion operator and defines the defect discrimination ξ as the degree of differentiation between the defect region and the other regions in an image. An increase in the value of ξ reflects the higher gray value of the defect region with a larger area, whereas the lower gray value denotes the nondefect region with a smaller area. Thus, an increase in the value of ξ also indicates the higher weight and the greater contribution to the image fusion process. In this case, the defect discrimination-based fusion operator ξ can be defined as

$$\mathrm{Comb}\left(N\left(I\right), N\left(O\right), N\left(E\right)\right) = \sum_{i=1}^{3} \overline{\xi}_i N\left(\bullet\right). \tag{2}$$

In different conspicuous maps, ξ is different as well, which refers to different coefficient weights. Therefore, in the fusion

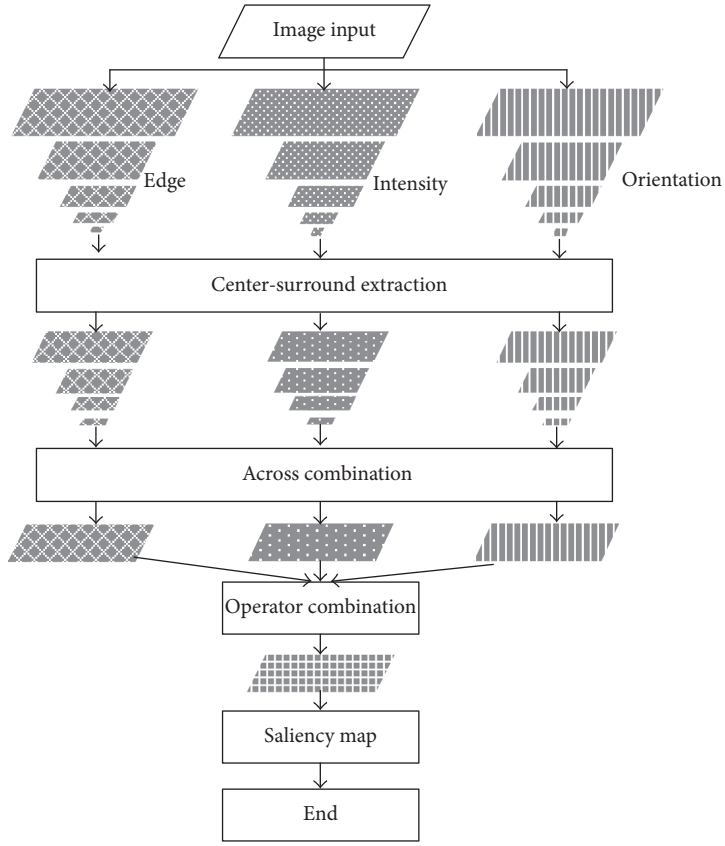


Figure 3: The proposed visual attention model.

process, different conspicuous maps provide different contributions to the saliency map SM. The specific methods for the calculation of $\overline{\xi}_i$ are shown in (3).

The image was segmented into Ω_t (t is a natural number) region sets according to the gray value. Furthermore, it was assumed that the randomly created region Ω_i ($i = 1, 2, 3, \ldots, t$) was smoothly closed. $z_i = f_i(x, y)$ is a function of the coordinates x and y of a pixel. Based on the characteristics of a defect region with large intensity area and regional edge, ξ_1 (the intensity discrimination), ξ_2 (the orientation discrimination), and ξ_3 (the edge discrimination) can be defined separately:

$$\xi_j = \frac{\max\left[\iint_{\Omega_i} f(x,y)\,dx\,dy\right]}{\sum_{i=1}^{t}\left(\iint_{\Omega_i} f(x,y)\,dx\,dy\right)}, \quad i = 1, 2, \ldots, t, \;\; j = 1, 2,$$

$$\xi_3 = \frac{\max\left[\oint_{L_i} f(x,y)\,ds\right]}{\sum_{i=1}^{t}\left(\oint_{L_i} f(x,y)\,ds\right)}, \quad i = 1, 2, \ldots, t, \tag{3}$$

$$\overline{\xi_k} = \frac{\xi_k}{\sum_{i=1}^{3}\xi_i}, \quad k = 1, 2, 3,$$

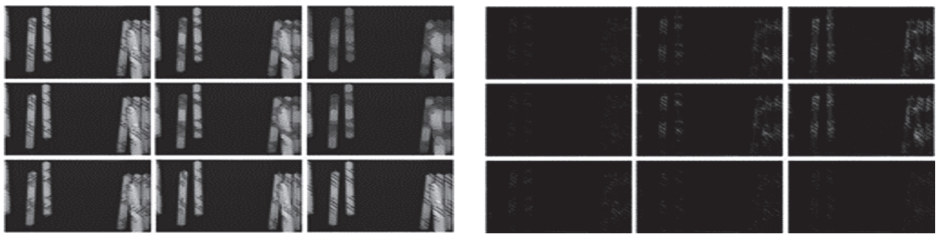
where L_i denotes the ith closed edge of Ω_i and ds denotes integral infinitesimal.

Figure 4: Input image.

4. Experiment

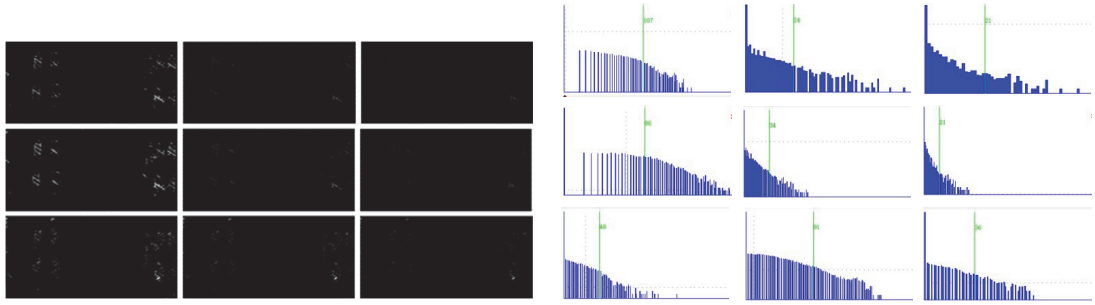
4.1. Image Preprocessing. Firstly, image preprocessing was performed on the original image, and the processing steps for background estimation, image differences, and brightness adjustment were performed as well. An opening algorithm was chosen for background estimation. The simulation results depend on the type of structural element (SE) [34]. The input image is shown in Figure 4, and the image preprocessing is presented in Figures 5–7.

4.2. Feature Measurement. Through the above-mentioned processes, most of the background interference information contained in the cartridge information was removed.



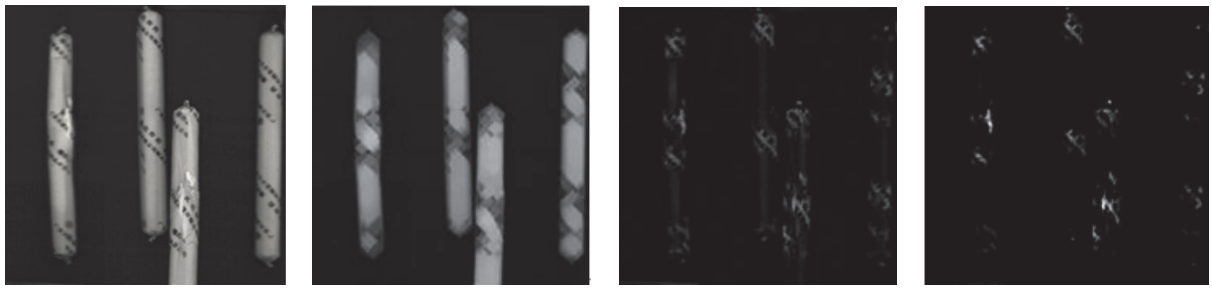
(a) Background estimation (b) Image difference

Figure 5: Image preprocessing. Step 1: the first row shows the SE type of a diamond; from left to right, the scale is 5, 10, and 15. The second row shows the SE type of a disk; from left to right, the scale is 5, 10, and 15. The last row shows the SE type of a line (scale 15); from left to right, the slope angle is 0.25π, 0.5π, and 0.75π.



(a) Brightness adjustment (b) Histogram

Figure 6: Image preprocessing. Step 2: the first row shows the SE type of a diamond (scale 10); from left to right, the adjustment γ is 0.5, 1.5, and 2.0. The second row shows the SE type of a disk (scale 10); from left to right, the adjustment γ is 0.5, 1.5, and 2.0. The last row shows the SE type of a line, with scale of 45 and slope angle of 0.25π; from left to right, the adjustment γ is 1.0, 1.5, and 2.0.



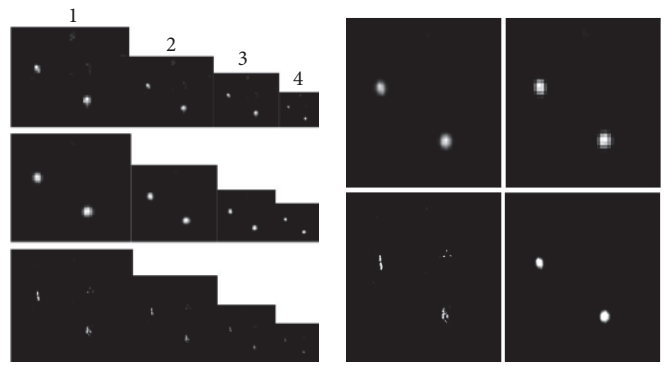
(a) Input (b) Step 1 (c) Step 2 (d) Step 3

Figure 7: Multitarget defect image preprocessing.

However, the residuary and unprocessed information still include some interference that was comparable to the defect feature. In this part, we made the complex image as the input source image, which included two defect clusters. As shown in Figure 7(a), the image preprocessing adopted the above algorithms; hence, it is still necessary to determine the specific defect features to filter out the interference factors.

To strengthen the character contrast, we built a feature-based model after the image preprocessing and created a pyramid for the three types of features, namely, brightness, direction, and edge, which were ranked in three levels, from Level 0 to Level 3, as the other hierarchies were too small to be graphically displayed. As shown in Figure 8(a), the image resolution dropped with the rise in the pyramid level.

The center-surround difference algorithm was adopted before the application of the multiscale fusion method. Then, we chose the fusion levels 1–4 to separately obtain the conspicuous maps for brightness, direction, and edge features, which are shown in Figure 8(b). The three conspicuous maps were integrated into a saliency map based on the discrimination fusion operator (3). $\overline{\xi_1}$, $\overline{\xi_2}$, and $\overline{\xi_3}$ were 0.5002, 0.2254, and 0.2744, respectively.



(a) Pyramid images (b) Across combination operator

Figure 8: Adopted visual attention algorithm. (a) From first to third row: brightness, direction, and edge. (b) Upper left shows brightness combination, while upper right shows the direction combination. Bottom left shows edge combination, and bottom right shows the operator combination of brightness, direction, and edge.

5. The Image Partitioning Variance-Weighted Eigenvalue-Based Analysis Method

5.1. Principle. This section proposes an IPV-WEV-based analysis method to realize the simultaneous recognition and positioning of multiple defect positions in the saliency map SM. The design and the implementation of the IPV-WEV method have been elaborated with the specific steps provided as follows.

(1) Image Segmentation. The $m \times n$ saliency map SM was segmented into $p \times q$ subimages:

$$\begin{pmatrix} S_{11} & \cdots & S_{1q} \\ \vdots & \ddots & \vdots \\ S_{p1} & \cdots & S_{pq} \end{pmatrix}, \quad k = 1, 2, \ldots, p;\ l = 1, 2, \ldots, q, \tag{4}$$

where S represents the saliency map SM, and every subimage S_{kl} represents an element of map S in a macro point of view. Meanwhile, S_{kl} also represents $k_u \times l_v$ matrix with $\sum_{k=1}^{p} k_u = m$ and $\sum_{l=1}^{q} l_v = n$.

(2) The Extraction of Defect Subimages. Due to the characteristic difference among the pixels at the defect and nondefect positions in the saliency map S, the variance, which was used to describe the degree of variation between the image pixel value and the mean value, can better reflect the salient features of the defect. It is clear that the variance of the image with defect is greater than that of the image without defect. Hence, the defect position can be determined through the calculation of the subimage variance and the comparison of the mean square error of the whole image according to the specific algorithm provided as follows.

(a) The calculation of the mean value and the variance of the whole saliency map S is

$$\begin{aligned} E(k,l) &= \frac{1}{m \times n} \sum_{k=1}^{m} \sum_{l=1}^{n} S(k,l), \\ \sigma^2(k,l) &= \frac{1}{m \times n} \sum_{k=1}^{m} \sum_{l=1}^{n} |S(k,l) - E(k,l)|^2, \end{aligned} \tag{5}$$

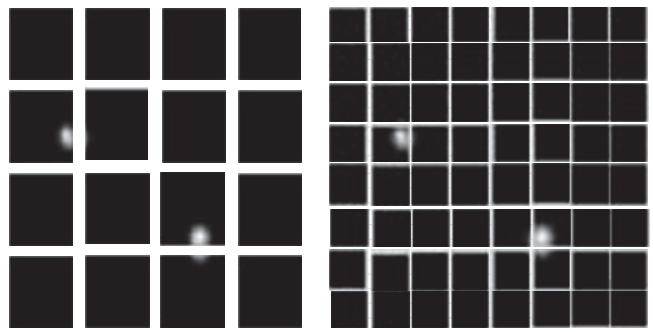
where $E(k,l)$ and $\sigma^2(k,l)$ separately represent the mean value and the variance of the saliency map S.

(b) The calculation of the mean value and the variance of the segmented subimage S_{kl} is

$$\begin{aligned} E_{S_{kl}}(k,l) &= \frac{1}{(2n+1)^2} \sum_{p=k-\omega}^{k+\omega} \sum_{q=l-\omega}^{l+\omega} S(p,q), \\ \sigma^2_{S_{kl}}(k,l) &= \frac{1}{(2n+1)^2} \sum_{p=k-\omega}^{k+\omega} \sum_{q=l-\omega}^{l+\omega} |S(p,q) - E(k,l)|^2. \end{aligned} \tag{6}$$

(c) As far as the whole image is concerned, most of the gray values of the pixels are comparable, which, however, is large only in the region with defect. Therefore, the fluctuation of a point, which is the gray value of a single pixel, was lower than that of the whole image. In other words, the variance was small for the segmented subimage, and the fluctuation was not evident if the defect information was not included. The variance of the subimage will be less than that of the whole image. However, if the subimage included some defect features, the gray value fluctuation would be significantly larger. In this case, the variance of the subimage will be greater than that of the whole image. Hence, the discrimination function can be defined as follows:

$$\begin{aligned} &\sigma^2_{S_{kl}}(k,l) - \sigma^2(k,l) \geq 0, \quad \text{YES}, \\ &\text{else}, \quad \text{NO}, \end{aligned} \tag{7}$$



(a) 4 × 4 block scale (b) 8 × 8 block scale

Figure 9: Image segmentation.

where "YES" denotes the presence of a defect cluster in the subimage. Thus, it is possible to analyze defect features on the subimage according to the discrimination function.

(3) The Calculation of the Weighted Eigenvalue. Through the construction of a weighted covariance matrix according to the principal component analysis conception, this paper calculated the weighted eigenvalue based on the gray value of every pixel to determine the defect position on the saliency map S.

(a) The center pixel $(\overline{x}, \overline{y})$ of the subimage can be defined as

$$\begin{aligned} \overline{x} &= \sum_{i=-\omega}^{\omega} \sum_{j=-\omega}^{\omega} (x+i) \cdot \frac{f(x+i, y+j)}{E_{S_{kl}}}, \\ \overline{y} &= \sum_{i=-\omega}^{\omega} \sum_{j=-\omega}^{\omega} (y+i) \cdot \frac{f(x+i, y+j)}{E_{S_{kl}}}. \end{aligned} \tag{8}$$

(b) M can be calculated as the weighted covariance matrix of the subimage:

$$M = \begin{bmatrix} m_{xx} & m_{xy} \\ m_{xy} & m_{yy} \end{bmatrix}, \tag{9}$$

where m_{xx}, m_{xy}, m_{yy} are defined as (10)–(12), respectively:

$$m_{xx} = \left[\sum_{i=-\omega}^{\omega} \sum_{j=-\omega}^{\omega} (x+i)^2 \cdot \frac{f(x+i, y+j)}{E_{S_{kl}}} \right] - \overline{x}^2, \tag{10}$$

$$m_{yy} = \left[\sum_{i=-\omega}^{\omega} \sum_{j=-\omega}^{\omega} (y+j)^2 \cdot \frac{f(x+i, y+j)}{E_{S_{kl}}} \right] - \overline{y}^2, \tag{11}$$

$$m_{xy} = \left[\sum_{i=-\omega}^{\omega} \sum_{j=-\omega}^{\omega} (x+i) \cdot (y+j) \cdot \frac{f(x+i, y+j)}{E_{S_{kl}}} \right] - \overline{x}\,\overline{y}. \tag{12}$$

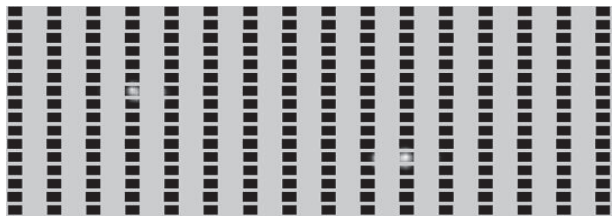


Figure 10: Image segmentation (block scale 16 × 16).

(c) According to the formula $M - \lambda I = 0$, λ_1, λ_2 can be calculated as follows:

$$\begin{aligned} \lambda_1 &= \frac{1}{2} \left[m_{xx} + m_{yy} + \sqrt{(m_{xx} - m_{yy})^2 + 4m_{xy}^2} \right], \\ \lambda_2 &= \frac{1}{2} \left[m_{xx} + m_{yy} - \sqrt{(m_{xx} - m_{yy})^2 + 4m_{xy}^2} \right]. \end{aligned} \tag{13}$$

5.2. Experiment. The experimental simulation consisted of the following two major parts: in the first part, the optimal parameters were mainly determined based on the effect of the parameters on the detection results of the IPV-WEV algorithm; and in the second part, a contrast among the IPV-WEV, region-growing, and the WTA neural network algorithms was presented.

The saliency map is the input image in which the intensity, orientation, and edge were combined. We segmented the input image into different block scales, and the image blocks can be seen in Figures 9-10.

The crack position was obtained through the calculation of the subimage variance and the comparison of the mean square error of the whole image. The experimental simulation revealed that the variance of the saliency map SM was $\sigma^2(k, l) = 14.86$. In 2 × 2 and 4 × 4 block scales, the variances of every segmented subblock are presented in Table 1.

5.3. Crack Defect Subimage Analysis

5.3.1. Algorithm Analysis. We adopted the weighted eigenvalue to quantitatively measure and analyze the defect

Table 1: Variance S_{kl} between 2×2 and 4×4 block scale.

S_{kl}	1	2	3	4
1	18.75 | 0	0.54 | 0.09	× | 1.07	× | 0
2	0 | 35.45	22.78 | 8.6	× | 0	× | 0
3	× | 0	× | 0	× | 41.75	× | 0
4	× | 0	× | 0	× | 14.75	× | 0

Note. × denotes no data; $a \mid b$ denotes variable value a in 2×2 block scale and variable value b in 4×4 block scale.

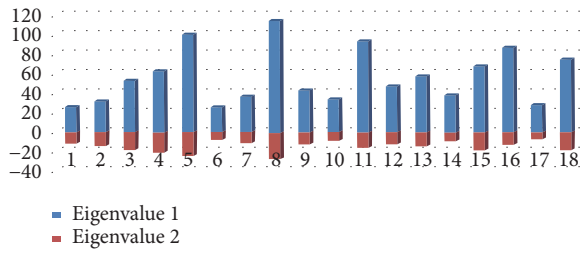


Figure 11: The detection results based on the weighted eigenvalues. Eigenvalue 1 and eigenvalue 2 denote λ_1, λ_2 respectively. *Note.* Numbers 1, 2 denote λ_1, λ_2 in subimage 2×2; numbers 3, 4 denote λ_1, λ_2 in subimage 4×4; numbers 5~9 denote λ_1, λ_2 in subimage 8×8; others denote λ_1, λ_2 in subimage 16×16.

subimage and to calculate the weighted eigenvalue of the extracted defect subimage for the construction of a weighted covariance. The weighted eigenvalues of each subimage were calculated as well to finally determine if the subimage has a defect through the comparison of λ_1 and λ_2. The defect detection results are illustrated in Figure 11.

Table 3 shows the detection time and detection rate for different block scales.

The comparison of the above simulation results with the data results revealed that (1) the defect position can be determined by calculating the eigenvalue of every subimage after the calculation of the variance of every subimage and by comparing the mean square error of the whole image; (2) block scale has an effect on defect position, and, for 2×2 and 4×4 image blocks, the detection time will be reduced due to the small number of blocks. Therefore, although it is possible to detect defect features, the defect position in the subimage can only be determined in some extent. Instead, only an approximate defect position can be detected. For the 8×8 image block, the defect position can be precisely detected (Table 2).

(3) When the defects were contained in different subimage blocks, the determination of the defect positions based on the connectivity of the cracks was possible, as the pixels in the crack defect area also constitute the locally connective region. If the detected defects of multiple subimages can be connected through combination, then all these subblocks can be determined on the defect positions. However, if there was a subimage separately located and without any connection to the other subblocks, it can be presumed that this subimage was not on the defect position.

5.3.2. Algorithm Comparison. In this experiment, 150 cartridge samples (positive samples number: 75; negative samples number: 75) were adopted to separately calculate the detection rate and defect detection time through the proposed IPV-WEV algorithm, WTA model [34], SLIC model [30], and region-growing algorithm [32]. The cartridge defect detection accuracy was denoted by the ratio between the wrongly detected samples and the correctly detected samples. The comparison of the detection precisions of the proposed algorithm, WTA model, and region-growing algorithm is presented in Table 4 and Figure 12.

The WTA algorithm was used to detect the cracks on the whole defect image through the charge-discharge method. The first region that has been quickly recharged was the defect position. However, as only one defect can be detected each time, detection of multitarget defects must be performed several times. On the other hand, the region-growing algorithm functions by determining a "seed" for region-growing and by combining all the regions that meet the growing conditions to detect the defect position. However, with this method, only one defect can be detected each time as well. For detection of multiple defects, the region-growing algorithm must be grown several times. Actually, due to the characteristic difference between the pixels at the defect and nondefect positions in the saliency image, the variance can better reflect the salient features of the defect. Moreover, as variance was used to describe the degree of variation between the pixel and mean values, an increase in the image variance also indicated a more dispersed distribution of the gray scale. In this case, the variance of the image with defect was evidently greater than that of the image without defect. Therefore, with the proposed IPV-WEV algorithm, only the variance of every image block needs to be calculated. Consequently, based on the comparison between the variance of the image block and the mean square error of the whole image, multiple defects can be simultaneously detected without individually searching for the pixels or "recharging" after the variance was taken as the discrimination function. For this reason, our proposed IPV-WEV algorithm outperformed the other two algorithms in terms of detection time.

6. Conclusion

The main objective of this paper is to propose a crack detection algorithm for industrial explosives. The proposed algorithm consisted of the following context: (1) image preprocessing was done according to the defect features of the industrial explosive cartridge, and an improved visual attention based algorithm was proposed. This algorithm features parametric analysis that can be implemented on the image according to the conspicuous maps with the introduction of the concept of defect discrimination ξ; (2) as compared with other algorithms, our proposed method can realize the real-time multitarget detection function; (3) the proposed IPV-WEN algorithm was able to analyze the cartridge defects

Table 2: Variance S_{kl} between 8 × 8 and 16 × 16 block scales.

S_{kl}	1	2	3	4	5	6	7	8	9	10	11	12	13~16
1	0 | 0	0 | 0	0 | 0	0.17 | 0	0 | 0	0 | 0	0 | 0	0 | 0	× | 0.2	× | 0	× | 0	× | 0	× | 0
2	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0.3	× | 3.1	× | 0	× | 0	× | 0	× | 0
3	0 | 0	11.3 | 0	1.63 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	× | 0.3	× | 0	× | 0	× | 0	× | 0
4	0 | 0	61.8 | 0	16.6 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	× | 0	× | 0	× | 0	× | 0	× | 0
5	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	× | 0	× | 0	× | 0	× | 0	× | 0
6	0 | 0	0 | 0	0 | 0.8	0 | 20.4	23.6 | 3.2	71.1 | 0	0 | 0	0 | 0	× | 0	× | 0	× | 0	× | 0	× | 0
7	0 | 0	0 | 0	0 | 10	0 | 53.9	7.34 | 27.8	27.4 | 0	0 | 0	0 | 0	× | 0	× | 0	× | 0	× | 0	× | 0
8	0 | 0	0 | 0	0 | 1.2	0 | 34.5	0 | 9.97	0 | 0	0 | 0	0 | 0	× | 0	× | 0	× | 0	× | 0	× | 0
9	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0
10	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0
11	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 6.1	× | 22.4	× | 1.8	× | 0
12	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 40.2	× | 50	× | 16.8	× | 0
13	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 14.1	× | 44.1	× | 5.3	× | 0
14~16	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0	× | 0

Note. × denotes no data; $a \mid b$ denotes variable value a in block scale 8 × 8 image and b in block scale 16 × 16 image.

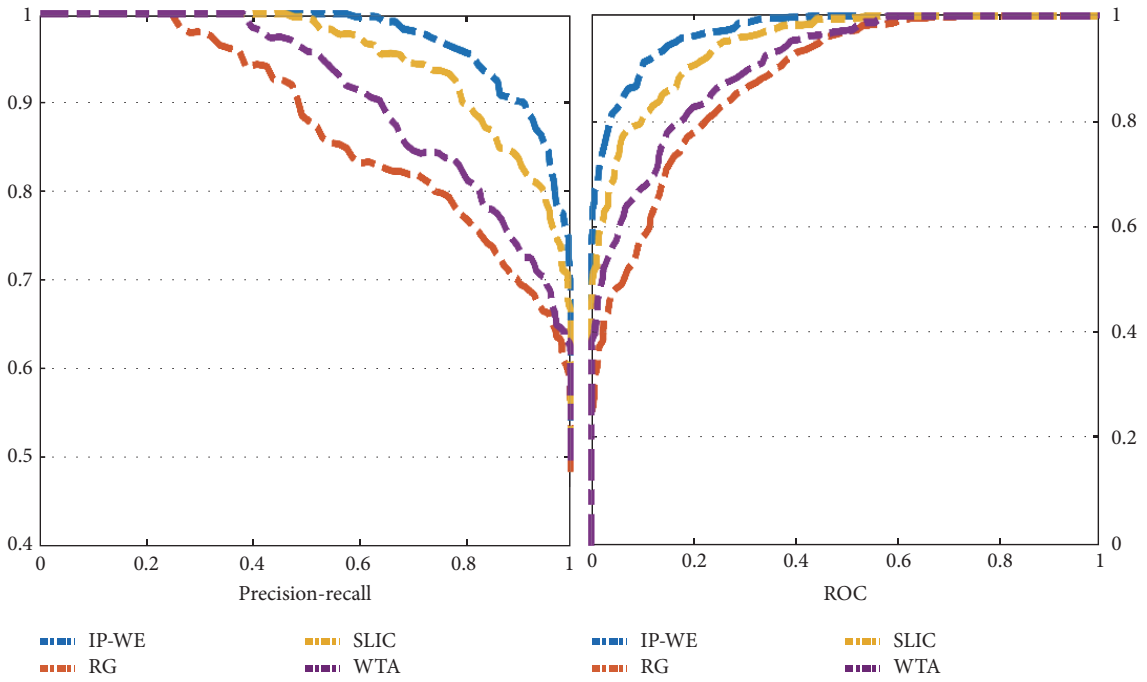


Figure 12: The precision-recall curve and ROC curve.

Table 3: Detection efficiency.

Block size	Detection time (ms)	Error rate (%)
2 × 2	22.35	0
4 × 4	24.45	0.2
8 × 8	34.51	0
16 × 16	60.13	0

Table 4: Comparison of algorithm defect results.

Model	Detection time (ms)	Error rate (%)	Accuracy rate (%)
WTA	198.6	4.7	95.3
RG	110.34	7.3	92.7
SLIC	89.7	5.2	94.8
IP-WE	60.13	2.1	97.9

based on performance indices. The comparison and experiment among algorithms revealed that the proposed method can achieve a detection accuracy of 97.9%, with the detection time of 34.51 ms, which has satisfied the requirement in the industrial explosives production.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

Haibo Xu conceived and designed the study. Buhai Shi performed the experiments. Haibo Xu and Qingming Zhang reviewed and edited the manuscript. All authors read and approved the manuscript.

References

[1] S. Z. He, "Research and development of twin screw pump in emulsified explosive packing machine," *Process Equipment & Piping*, 2005.

[2] S. U. Ming-Yang, "Image of twin-screw-type filling machine on characteristics of packaged emulsion explosives," *Engineering Blasting*, 2010.

[3] S. Li, C. Lu, Y. Cai, and J. Gui, "Study on improvement of emulsified explosive packing process," *Explosive Materials*, 2012.

[4] E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, "A survey on industrial vision systems, applications and tools," *Image and Vision Computing*, vol. 21, no. 2, pp. 171–188, 2003.

[5] M. Moganti, F. Ercal, C. H. Dagli, and S. Tsunekawa, "Automatic PCB inspection algorithms: a survey," *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 287–313, 1996.

[6] X.-F. Ding, L.-Z. Xu, X.-W. Zhang, F. Gong, A.-Y. Shi, and H.-B. Wang, "A model of saliency-based selective attention for machine vision inspection application," in *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms*, 2011.

[7] Y. Li, S. Dhakal, and Y. Peng, "A machine vision system for identification of micro-crack in egg shell," *Journal of Food Engineering*, vol. 109, no. 1, pp. 127–134, 2012.

[8] N. Razmjooy, B. S. Mousavi, and F. Soleymani, "A real-time mathematical computer method for potato inspection using machine vision," *Computers & Mathematics with Applications*, vol. 63, no. 1, pp. 268–279, 2012.

[9] Z. Jia, B. Wang, W. Liu, and Y. Sun, "An improved image acquiring method for machine vision measurement of hot formed parts," *Journal of Materials Processing Technology*, vol. 210, no. 2, pp. 267–271, 2010.

[10] H. Shen, S. X. Li, D. Y. Gu, and H. X. Chang, "Bearing defect inspection based on machine vision," *Measurement*, vol. 45, no. 4, pp. 719–733, 2012.

[11] A. Tellaeche, G. Pajares, X. P. Burgos-Artizzu, and A. Ribeiro, "A computer vision approach for weeds identification through support vector machines," *Applied Soft Computing*, vol. 11, no. 1, pp. 908–915, 2011.

[12] A. R. Jiménez, R. Ceres, and J. L. Pons, "A vision system based on a laser range-finder applied to robotic fruit harvesting," *Machine Vision and Applications*, vol. 11, no. 6, pp. 321–329, 2000.

[13] M. Magee and S. Seida, "An industrial model based computer vision system," *Journal of Manufacturing Systems*, vol. 14, no. 3, pp. 169–186, 1995.

[14] Z. Gao and L. Duan, "Vision detection of vehicle occupant classification with legendre moments and support vector machine," in *Proceedings of the 2010 3rd International Congress on Image and Signal Processing (CISP '10)*, pp. 1979–1983, October 2010.

[15] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[16] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*, pp. 545–552, December 2006.

[17] Q. Zhang, H. Liu, J. Shen, G. Gu, and H. Xiao, "An improved computational approach for salient region detection," *Journal of Computers*, vol. 5, no. 7, pp. 1011–1018, 2010.

[18] P.-J. Hsieh, J. T. Colas, and N. Kanwisher, "Pop-out without awareness: unseen feature singletons capture attention only when top-down attention is available," *Journal of Vision*, vol. 22, no. 9, pp. 1220–1226, 2011.

[19] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2015.

[20] J. Cong and Y. Yan, "Application of human visual attention mechanism in surface defect inspection of steel strip," *China Mechanical Engineering*, vol. 22, no. 10, pp. 1189–1221, 2011.

[21] G. Li, H. Luo, M. Tang, H. Mu, and Z. Zhou, "A machine vision inspection algorithm for contamination in cotton based on visual attention mechanism," *Application of Electronic Technique*, 2012.

[22] S. Liu, Z. Cao, and J. Li, "A SVD-based visual attention detection algorithm of SAR image," *Lecture Notes in Electrical Engineering*, vol. 246, pp. 479–486, 2014.

[23] Y. Liu, L. Chen, and W. Shi, "Applications of an algorithm of image preprocessing based on visual attention mechanisms in industrial inspection," *Electronic Science & Technology*, 2016.

[24] M. Mancas, C. Mancas-Thillou, B. Gosselin, and B. Macq, "A rarity-based visual attention map—application to texture description," in *Proceedings of the Image Processing, 2006 IEEE International Conference*, vol. 44, pp. 445–448, October 2006.

[25] R. Pal, "Applications of visual attention," *Innovative Research in Attention Modeling & Computer Vision Applications*, 2016.

[26] X. Wang, B. Wang, and L. Zhang, "Airport detection in remote sensing images based on visual attention," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7064, no. 3, pp. 475–484, 2011.

[27] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996–1010, 2013.

[28] Y. He, Y. Chen, Y. Xu, Y. Huang, and S. Chen, "Autonomous detection of weld seam profiles via a model of saliency-based visual attention for robotic arc welding," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3-4, pp. 395–406, 2016.

[29] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 409–416, Providence, RI, USA, June 2011.

[30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.

[31] A. Aboudib, V. Gripon, and G. Coppin, "A biologically inspired framework for visual information processing and an application on modeling bottom-up visual attention," *Cognitive Computation*, vol. 8, no. 6, pp. 1007–1026, 2016.

[32] J. K. Garner and D. M. Russell, *The Symbolic Dynamics of Visual Attention During Learning: Exploring the Application of Orbital Decomposition*, Springer International Publishing, 2016.

[33] G. E. Kalliatakis, T. Kounalakis, G. Papadourakis, and G. A. Triantafyllidis, "Image based touristic monument classification using Graph Based Visual Saliency and Scale-Invariant Feature Transform," in *Proceedings of the IASTED International Conference on Computer Graphics and Imaging (CGIM '12)*, pp. 261–266, June 2012.

[34] P. Bourgine and A. Lesne, *Morphological and Mutational Analysis: Tools for the Study of Morphogenesis*, Springer, Berlin, Germany, 2011.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 5604740, 12 pages
https://doi.org/10.1155/2018/5604740

Research Article

Water Quality Monitoring Method Based on TLD 3D Fish Tracking and XGBoost

Shuhong Cheng,[1] Shijun Zhang ,[1] Leihua Li,[1] and Dianfan Zhang[2]

[1]*School of Electrical Engineering, Yanshan University, Qinhuangdao, China*
[2]*Yanshan University Science Park, Qinhuangdao, China*

Correspondence should be addressed to Shijun Zhang; 980871977@qq.com

Received 12 June 2017; Accepted 16 January 2018; Published 18 March 2018

Academic Editor: Daniel Zaldivar

Aiming at the problem of water quality monitoring, this paper presents a method of biological water quality monitoring based on TLD (Tracking-Learning-Detection) framework and XGBoost (eXtreme Gradient Boosting). Firstly, under the framework of TLD, an independent tracking system is designed; TLD captures 3D coordinate information of fish based on video and calculates the behavior of fish movement parameters which can reflect the change of water quality via processing the coordinate information of the fish body. The data of coordinate information will be more prominent via the data processing. The integration of all built XGBoost water quality monitoring model which is based on characteristic parameters; the model was used to analyze and evaluate fish behavior parameters under unknown water quality to achieve the purpose of water quality monitoring.

1. Introduction

In the era of Industrial 4.0, highly automated and intelligent manufacturing technology will gradually occupy the field of human society in the field of industrial development. In the field of water quality pollution, an efficient, convenient, and intelligent monitoring method becomes more urgent and necessary. In order to prevent and deal with the current situation of water quality pollution, the existing water quality monitoring technologies are physical and chemical analysis technology, automatic detection technology, and biological monitoring technology [1, 2]. The physical and chemical analysis technology and automatic detection technology for long-term real-time monitoring cost are relatively large; at the same time, these two technologies can not achieve the comprehensive evaluation of water pollution degree. The biological monitoring technique has been adopted. Biological monitoring technology is comprehensive, rich, and continuous; in recent years it plays an increasingly important role and has a broad application prospect in the environmental assessment. In the monitoring of biological water quality, fish, as an important indicator organism, with its movement characteristics, physiological characteristics, and other information directly reflect the changes in the water environment and the current situation of environmental pollution. Many scholars such as Kim et al. obtain the parameters of motion behavior of fish by computer vision technology [3–8]; only a few scholars put forward the method of water quality anomaly monitoring according to these characteristic parameters. Serra-Toro et al., Lai and Chiu, and Zhangzan et al., respectively, analyze the fish swimming behavior to get the relationship between exercise behavior parameters and water quality via recursive algorithm, fuzzy reasoning method, and data evaluation, so as to achieve the purpose of monitoring the abnormal water quality [9–11].

Although these methods can be used for monitoring, there are a lot of defects in the selection of feature parameters, the length of operation, the accuracy of monitoring, and the processing of individual differences. The choice of parameters has great influence on the model; the model cannot choose effective parameters to monitor; it can only be passively chosen by human choice; because the monitoring process of these methods was divided into a plurality of independent links, from data acquisition to the final monitoring, the whole operation was too long to show the results in time and effectively; data acquisition and processing process was too complex; each link increases the error component data, resulting in low accuracy of monitoring results. It only selects

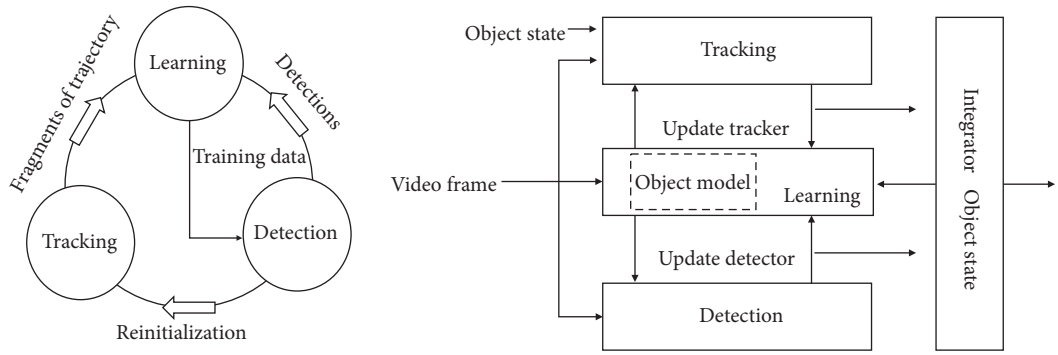


Figure 1: TLD frame structure.

features unilaterally; the sample data can not reflect the difference, the sample can not be comprehensively taken into account, and the monitoring results also are one-sided.

This paper presents the water quality monitoring methods under the framework of TLD and XGBoost; based on 3D fish tracking, we use TLD method to get the fish body coordinate in different water conditions and determine the characteristic parameters based on the calculation results of fish movement parameters. XGBoost is adopted to analyze and evaluate the feature parameter, establishing semantic mapping, and the movement of fish behavior characteristic parameters model, and finally it realized the abnormal water quality monitoring.

2. Extraction of Fish Characteristic Parameters

2.1. TLD Framework Introduction. TLD was a long-term single objective tracking algorithm proposed by Dr. Kalal et al. [12, 13]. The TLD framework mainly includes three parts: tracking part, learning part, and detection part. The relationship of the three parts and the tracking process is shown in Figure 1, for tracking part, mainly using the Forward-Backward Error method, using Lucas-Kanade optical flow tracking, the tracking results using Forward-Backward Error as feedback, the Euclidean distance for FB error compared with the original position, the distance which the longer will be abandon, this tracking method of using FB error to discard bad values was called Median Flow, which discards those larger than 50% of the European distance set; detection of TLD has three parts: the variance classifier module, set classifier module, and nearest neighbor classifier module; these three classifiers are cascaded. Each scan window of the current frame sequentially passes through the above three classifiers, both are considered to contain foreground objects; the learning part is the same as in other methods of target detection; detection module in TLD may also have errors, and the error was nothing more than the two cases being negative sample error and positive sample error. The learning module is used according to the results of the tracking module to evaluate the two kinds of errors of detection module, based on the results of the assessment to generate the target training samples and update the detection module; meanwhile, the tracking module "key points" were updated, in order to avoid similar mistakes.

2.2. Fish Tracking Framework. We put forward a more specific framework for fish tracking based on the framework of TLD. Imitating TLD settings, the process of track is also divided into three parts, each part of the connection is as shown in Figure 2; we collected video by frame rate interception and input it to the TLD tracking module.

At any time, the tracked target could be represented by its state property. The state property can be a tracking box that represents the location and size of the target, or a marker that identifies whether the tracked target was visible. The two-track box in the space domain similarity was used to measure the overlap; the calculation method is that the two track frame intersections have joint segmentation. The shape of the object was represented by image patch P, and each fish body image was sampled from the tracking frame and normalized to the same size. In the tracking module, NCC is said to be the normalized correlation coefficient; BF (brute force selection mechanism) is said to be the forced choice mechanism, tracking each pixel in a first frame in the whole fish body image sequence; FB (Forward-Backward Error) was used to estimate trajectory. Finally, through the interaction of NCC and BF, we can get the stable tracking point by taking the two intersections.

The fish body images were not only input to the tracking module, but also used in the detection module. The test needs to use the positive and negative training sample information for training in advance, so that the detection module has certain detection ability. The detection module includes three filters, respectively, Variance filter, Fern variance filter (Random-Fern), and NN filter; through the three filters the errors in image are removed and the best target fish image is obtained. The tracking module and the detection module obtain the sample information to be fused and input to the learning module. Learning module used P-N learning. The main idea was that the detector error can be identified by two types of constraints. The function of P-expert was to find the new appearance (deformation) of the target and to increase the number of positive samples. The role of N-expert was to generate negative training samples. The premise of N-expert was that the tracked object may only appear in a position in the video frame, so if the position of the foreground target is determined, all around it must be a negative sample. The learning module was used to correct the tracking module and the detection module, and the Fern structure would be

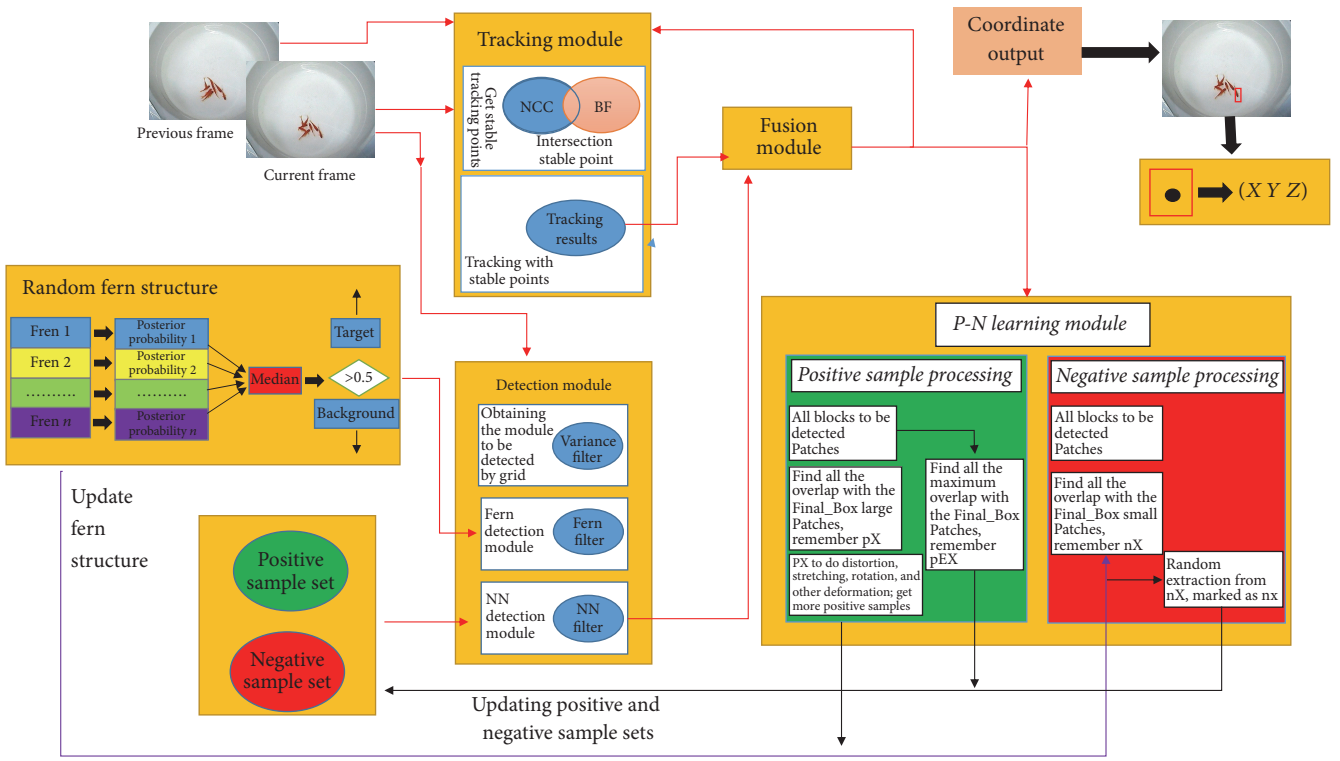


Figure 2: Fish tracking frame structure.

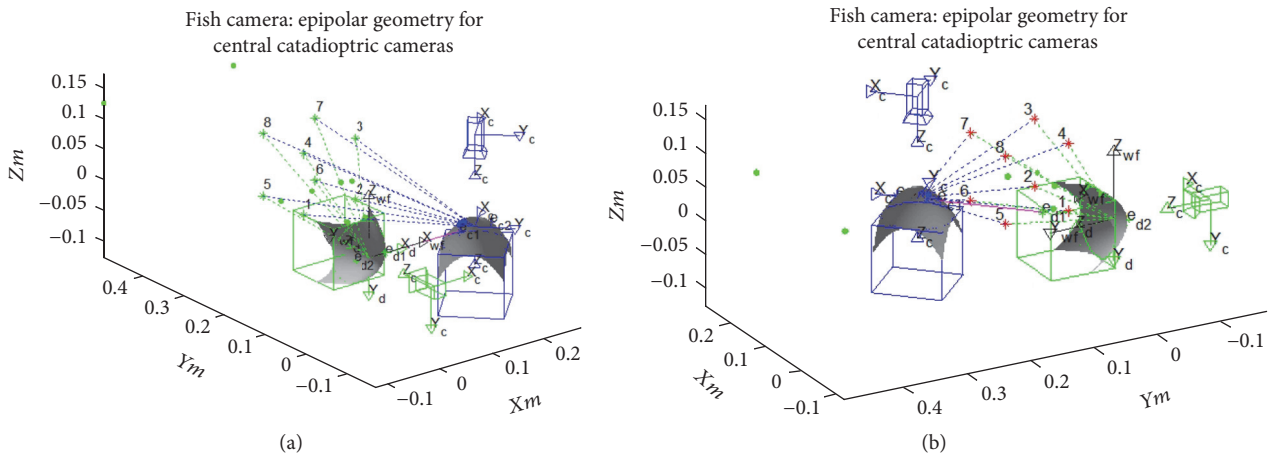


Figure 3: Camera position and fish body space coordinates.

updated, and the weight is adjusted to make the next result more reliable.

The image of the fish captured by the video was processed by the above steps, and finally the coordinate information of the fish in the three-dimensional space was obtained. Because the TLD tracking result was the two-dimensional plane coordinate information, the camera position needs to be arranged in order to collect three-dimensional coordinates. The camera was placed in the vertical direction and the horizontal direction of the fish tank, as shown in Figures 3(a) and 3(b); the two pictures show our camera positioning; in order to facilitate viewing, vertical plane and horizontal plane with a fish tank were drawn out; the blue side is said to be the vertical direction to the information acquisition of X-Y; the green part is said to be the horizontal direction to the information acquisition of X-Z.

The position of some fish in the space is as shown in Figure 4. Figures 4(a) and 4(c) show that the fish coordinate point projected onto a plane and get the scatterplot. In (b), (d) the blue oval line in the graph represents different points in different plane; elliptical is said to be in a different point in different plane. So you can see clearly the distribution of fish

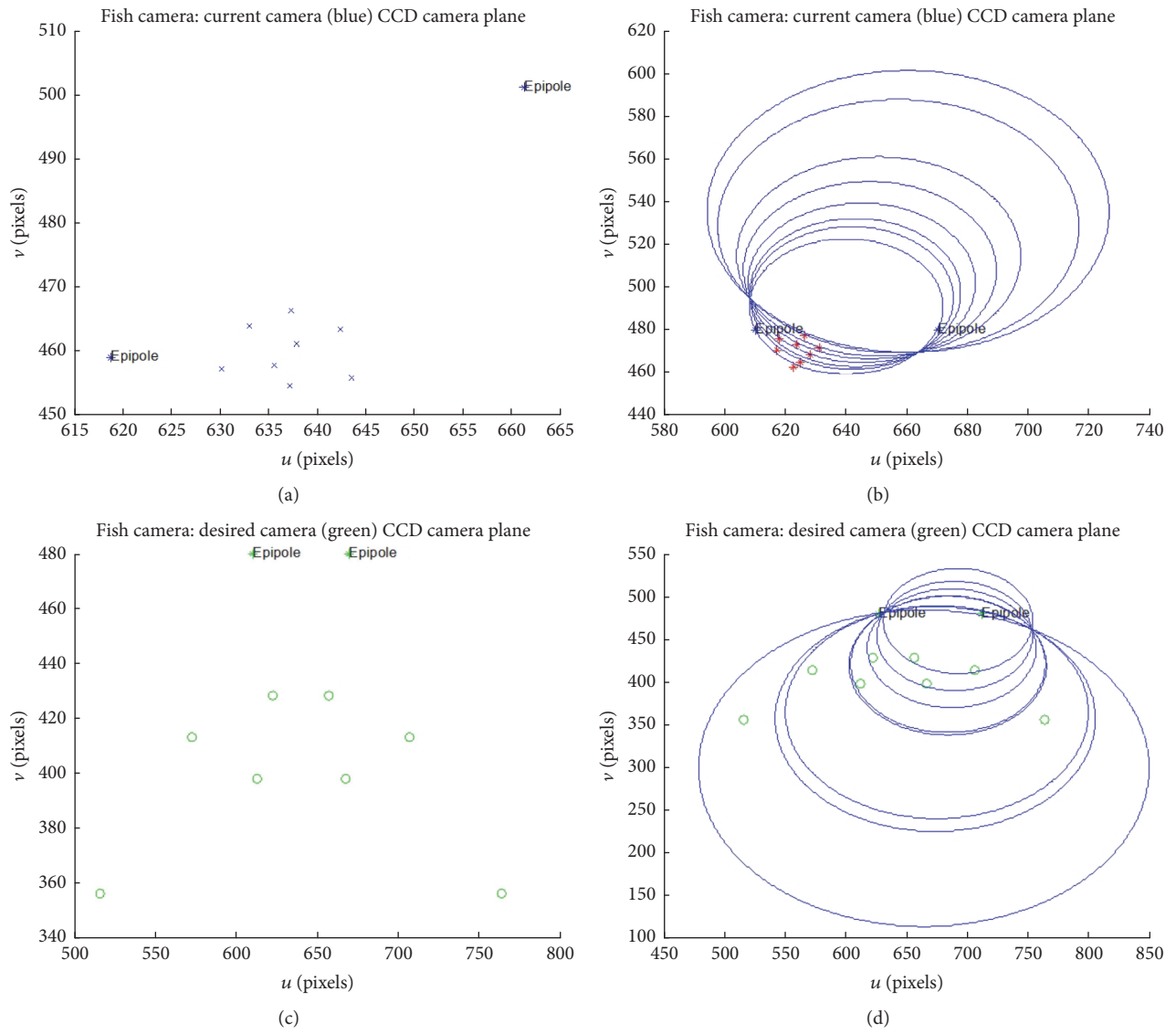


Figure 4: Camera position and fish body space coordinates.

in the three-dimensional space. The x-axis of the two cameras overlap, and the x-axis overlap information is removed to get the 3D information of the x-y-z surface.

2.3. Parameter Calculationa. The trajectory of fish can be expressed as a series of discrete discontinuous points; that is to say, the trajectory of n video sequence images of a fish can be described as

$$T = \left[x\left(t\right), y\left(t\right), z\left(t\right)\right]. \tag{1}$$

Getting fish behavior characteristic parameters which can reflect the status of the water ecological environment by using trajectory: average moving distance, speed, acceleration, X direction discrete distance, Y direction discrete distance, and distribution area, the definitions are as follows.

Average moving distance:

$$d = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2 + \left(z_1 - z_2\right)^2}. \tag{2}$$

(x_1, y_1, z_1) and (x_2, y_2, z_2) are used to represent the coordinates of the starting point and the end point of the fish target tracking trajectory in unit time. Speed V would be defined as

$$v = \frac{d}{t}. \tag{3}$$

Acceleration a would be defined as

$$a = \frac{v_1 - v_2}{t}. \tag{4}$$

v_1, v_2, respectively, expressed the speed of t_1, t_2, $\Delta t = t_2 - t_1$, $(t_2 > t_1)$.

The distribution of fish in a single plane has great difference between normal and abnormal water quality; we use measured dispersion to describe the relationship between the single fish and fish. The dispersion formula would be defined as

$$LS^2 = \begin{cases} \sum_{i=1}^{N} (x_i - \overline{x})^2 \\ \sum_{i=1}^{N} (y_i - \overline{y})^2 \\ \sum_{i=1}^{N} (z_i - \overline{z})^2 \end{cases}. \quad (5)$$

The characteristics of the dispersion of the fish body coordinates were concerned with the position of the center of gravity and the center of gravity of each individual, which integrates the characteristics of the whole and the individual. Similarly, in the X-Y plane, the distribution area of fish also has great features. Shoal area calculation formula is

$$S = \begin{cases} S_1 = \Delta x \times \Delta y & S_1 < S_2, \\ S_2 = \pi r^2 & S_1 > S_2. \end{cases} \quad (6)$$

Δx represents the maximum distance between X coordinates of all individuals in a fish at the same time, Δy is said to be the maximum distance between Y coordinate; r is said to be the fish gathered together to form the minimum circumscribed circle radius. Finally, we use TLD and especially placed cameras to collect the coordinates; the characteristic parameters are obtained through calculation to establish a large enough data set.

3. Establishment of Water Quality Monitoring Model

XGBoost (eXtreme Gradient Boosting) [14, 15] is designed by Dr. Chen Tianqi in Gradient Machine (GBDT) and the GBDT has been improved in boosting. On the basis of GBDT, XGBoost modified the objective function and loss function.

3.1. Objective Function, Obj. First of all, the objective function of GBDT is as follows:

$$\text{Objective: } \sum_{i=1}^{n} l(y_i, \widehat{y}_i) + \sum_{k} \Omega(f_k), \quad f_k \in F, \quad (7)$$

where l is the loss function, $\Omega(f_k)$ is a regular term, and f_k is tree complexity. After series of processing, we get a new objective function which is easier to use:

$$\text{Obj}^{(t)}: \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + \text{constant}. \quad (8)$$

XGBoost Taylor expansion for $f_t(x_i)$ makes it more similar to our previous goals. The process is as follows:

$$f(x + \Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2. \quad (9)$$

By applying Taylor expansion to the three terms, we can make it clear that the final objective function depends on the first derivative and the second derivative of the error function of each data point.

Definition:

$$g_i = \delta_{\widehat{y}^{(t-1)}} l\left(y_i, \widehat{y}^{(t-1)}\right), \quad h_i = \delta^2_{\widehat{y}^{(t-1)}} l\left(y_i, \widehat{y}^{(t-1)}\right). \quad (10)$$

Bring (9) and (10) back to (8):

$$\text{Obj}^{(t)} \simeq \sum_{i=1}^{n} \left[l\left(y_i, \widehat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{constant}. \quad (11)$$

Taking into account the square loss, can also continue to (10):

$$g_i = \delta_{\widehat{y}^{(t-1)}} \left(\widehat{y}^{(t-1)} - y_i\right)^2 = 2\left(\widehat{y}^{(t-1)} - y_i\right), \quad h_i = \delta^2_{\widehat{y}^{(t-1)}} \left(y_i - \widehat{y}^{(t-1)}\right)^2 = 2. \quad (12)$$

In this way, we can remove the constant term and get the new objective function:

$$\sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_t f_t^2(x_i) \right] + \Omega(f_t). \quad (13)$$

In the equation, $g_i = \delta_{\widehat{y}^{(t-1)}} l(y_i, \widehat{y}^{(t-1)})$, $h_i = \delta^2_{\widehat{y}^{(t-1)}} l(y_i, \widehat{y}^{(t-1)})$.

Then define the complexity of the tree:

$$f_t(x) = w_{q(x)}, \quad w \in R^T, \ q : R^d \longrightarrow \{1, 2, 3, \ldots, T\}, \quad (14)$$

where w is a leaf vector, q is a tree structure, which defines the complexity of the number of nodes in a tree, and L_2 is square of the output of each tree node. Under this new definition, we can rewrite the objective function as follows: I is defined as a set of samples on each leaf; $I_j = \{i \mid q(x_i) = j\}$.

$$\text{Obj}^{(t)} \simeq \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w^2_{q(x_i)} \right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2 \quad (15)$$

$$\text{Obj}^{(t)} = \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_j + \lambda \right) w_j^2 \right] + \gamma T.$$

To further simplify the formula, we define

$$G_j = \sum_{i \in I_j} g_i,$$
$$H_j = \sum_{i \in I_j} h_i, \tag{16}$$

$$\text{Obj}^{(t)} \simeq \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_j + \lambda \right) w_j^2 \right] + \gamma T, \tag{17}$$

$$\text{Obj}^{(t)} = \sum_{j=1}^{T} \left[G_j w_j + \frac{1}{2} \left(H_j + \lambda \right) w_j^2 \right] + \lambda T. \tag{18}$$

Based on the derivation of it being equal to 0, the following can be obtained:

$$w_j^* = -\frac{G_j}{H_j + \lambda}. \tag{19}$$

Bring the optimal solution w_j^* back to (18), and the final objective function is obtained:

$$\text{Obj} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T. \tag{20}$$

The above objective function, Obj, represents the maximum amount of reduction in the target when the structure of a tree is specified. You can call it a structural score. The smaller the Obj, the better the structure of the tree.

3.2. Add Scoring Function Gain

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} + \frac{\left(G_L + G_R \right)^2}{H_L + H_R + \lambda} \right] - \gamma. \tag{21}$$

$G_L^2/(H_L + \lambda)$ represents the left subtree fraction, $G_R^2/(H_R + \lambda)$ represents the right subtree fraction, $(G_L+G_R)^2/(H_L+H_R+\lambda)$ represents an undivided score, and γ represents the complexity cost of adding new leaf nodes. For each expansion, or to efficiently enumerate all possible segmentation schemes, suppose you want to enumerate all the conditions of $x < a$, for a particular split a to calculate a's left and right derivative, as shown in Figure 5.

It would be found that for all 'a', you can give all the gradients and GL and GR as long as a scan is done from left to right. And then the above formula can be used to calculate the score of each partition.

3.3. XGBoost Biological Water Quality Monitoring Model.

XGBoost joined the regularization; regularized boosting was a great help in reducing overfitting. Compared with GBDT the XGBoost can achieve parallel processing; the speed has been greatly improved; comparing SVM classification

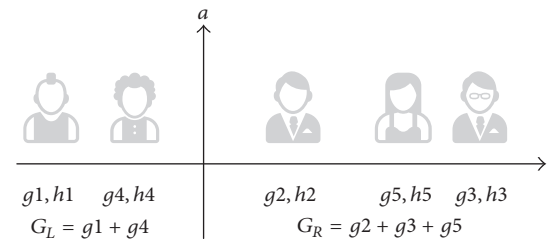


Figure 5: Sketch map of segmentation.

with XGBoost, XGBoost allows the user to define custom optimization objectives and evaluation criteria; it will add a new dimension in the model, so it is not subject to any restrictions on the data processing; in the process of data collection, because various artificial or experimental defects exist, it will inevitably lead to the phenomenon of data loss; while the XGBoost was built to deal with the missing value rules, the user needs to provide a different values sample and then take it as a parameter input; so this value is taken as missing values; XGBoost has different treatment methods in different nodes encountered with missing values and will learn when encountering the missing value in the future how to deal with it. The method provides a good solution for the deletion of the fish characteristic parameters, and the XGBoost allows the use of cross validation in each round of boosting iteration. Therefore, the optimal number of boosting iterations can be easily obtained; XGBoost can continue to train on the results of the previous round. This feature was a great advantage in the application of the classification of fish characteristic parameters and can realize the continuity of the model.

Therefore, we try to use XGBoost to replace the previous classifier to establish water quality classification model. Prior to the establishment of the model, we need to preprocess the collected feature data to improve the training speed and accuracy of the model. Firstly, Smoothing was performed to create a feature sample set. XGBoost was an excellent decision tree classifier, which can use the objective function and scoring function as the model's performance.

XGBoost water quality classification model:

$$\text{Obj} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T,$$
$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} + \frac{\left(G_L + G_R \right)^2}{H_L + H_R + \lambda} \right] - \gamma. \tag{22}$$

In this paper, the steps of establishing water quality monitoring model were as follows:

(1) We use the TLD technology to preprocess the fish motion parameters and set up a set of characteristic parameters.

(2) The feature parameter set was used as the input of XGBoost for training.

(3) Cross validation method was used to obtain the optimal number of boosting iterations.

Input:
Part.1
Image patches (positive and negative image patches), I_i;
Target(x, y, z), Target$_n$;
Output:
Tracking results (x_i, y_i, z_i), at time t;
(1) $t \Leftarrow 1 \cdots T$
(2) **if** $T = 1$ **then**
(3) Marker the target
(4) **else**
(5) Stage 1: using NCC and BF get the stable points.
(6) Stage 2: I_i through Variance filter, Fern filter, NN filter get the best patches and staable point.
(7) Compare stage 1 with stage 2 get the best points.
(8) Update the Random Fren and positive, negative sample set.
(9) **end if**
Input:
Part.2
Fish characteristic parameter: F_i as data, F_i;
Model parameter: P_i
include (base-score = 0.5, colsample-bylevel = 1, colsample-bytree = 1, gamma = 0, learning-rate = 0.1, max-delta-step = 0, max-depth = 3, min-child-weight = 1, missing = None, n-estimators = 100, $n$thread = −1, objective = binary: logistic, reg-alpha = 0, reg-lambda = 1, scale-pos-weight = 1, seed = 0, silent = True, subsample = 1)
Output:
Water Quality degree: Q_i; model; Classification Accuracy
(10) Load data
(11) Split data into train and test sets by train-test-split( ).
(12) Load XGBClassifier and model.predict.
(13) Calculating Classification Accuracy

Algorithm 1: TLD + XGBoost for water quality.

(4) The tree structure and the characteristic score were used to analyze and evaluate the model.

The algorithm flow was shown in Algorithm 1.

4. Experimental Results and Analysis

In the fish red carp, body length was about 3 cm; the camera is BNT shadow (HD720) and using tank; normal water quality is common water; abnormal water is chemical reagent which is added to copper or chromium.

In order to verify the validity and feasibility of the method in this paper, the following experiments are designed. Get the fish motion video images of normal and abnormal water quality by using the digital camera in the course of the experiment; each frame size was $480 * 640$, the frame of 25 f/s.

4.1. Select Characteristic Parameters. In order to improve the speed of data processing and the classification accuracy of XGBoost classifier, we set the parameters selection and choose normal water quality data set; the abnormal water quality data was intercepted according to certain sampling frequency. The characteristic parameters of the six groups were selected as follows: near distance, d, speed, V, acceleration, a, X, dispersion direction, and Y, dispersion direction and distribution area S, as shown in Figure 6.

In Figure 6 the color changes from blue to red, showing the water quality change from normal water quality to abnormal water quality. Left with blue is said to be the healthy state; right with red is said to be the unhealthy water quality. It can be observed from Figure 6 that characteristic parameters have obvious trend; in normal water, acceleration a, speed V, near distance d, X dispersion direction, and Y dispersion direction in the direction of the image on the left were less active; in each image the number of difference changes was low; and the right was risk quality, comparing characteristic parameter changes with the left; there was a significant difference.

4.2. Feature Parameter Preprocessing and Sample Set. The preprocessing of the characteristic parameters was to remove the gross errors in the data by Rajda (3σ) criterion.

Due to the normalization of XGBoost own data normalization, here were no longer repeat operation.

The concrete steps to build a sample set were as follows:

(1) The original data of fish motion parameters were obtained by randomly selecting 2000 frames of images under normal and abnormal water quality.

(2) Set up the original data sample set, including training samples 2000×6 (normal and abnormal water quality of each 1000×6) and test samples 2000×6 (normal and abnormal water quality of each 1000×6).

(3) Preprocess the original data.

4.3. Experimental Result. A total of 4 sets of samples were established in this experiment; we use these data for testing; test results were shown in Table 1. As can be seen from

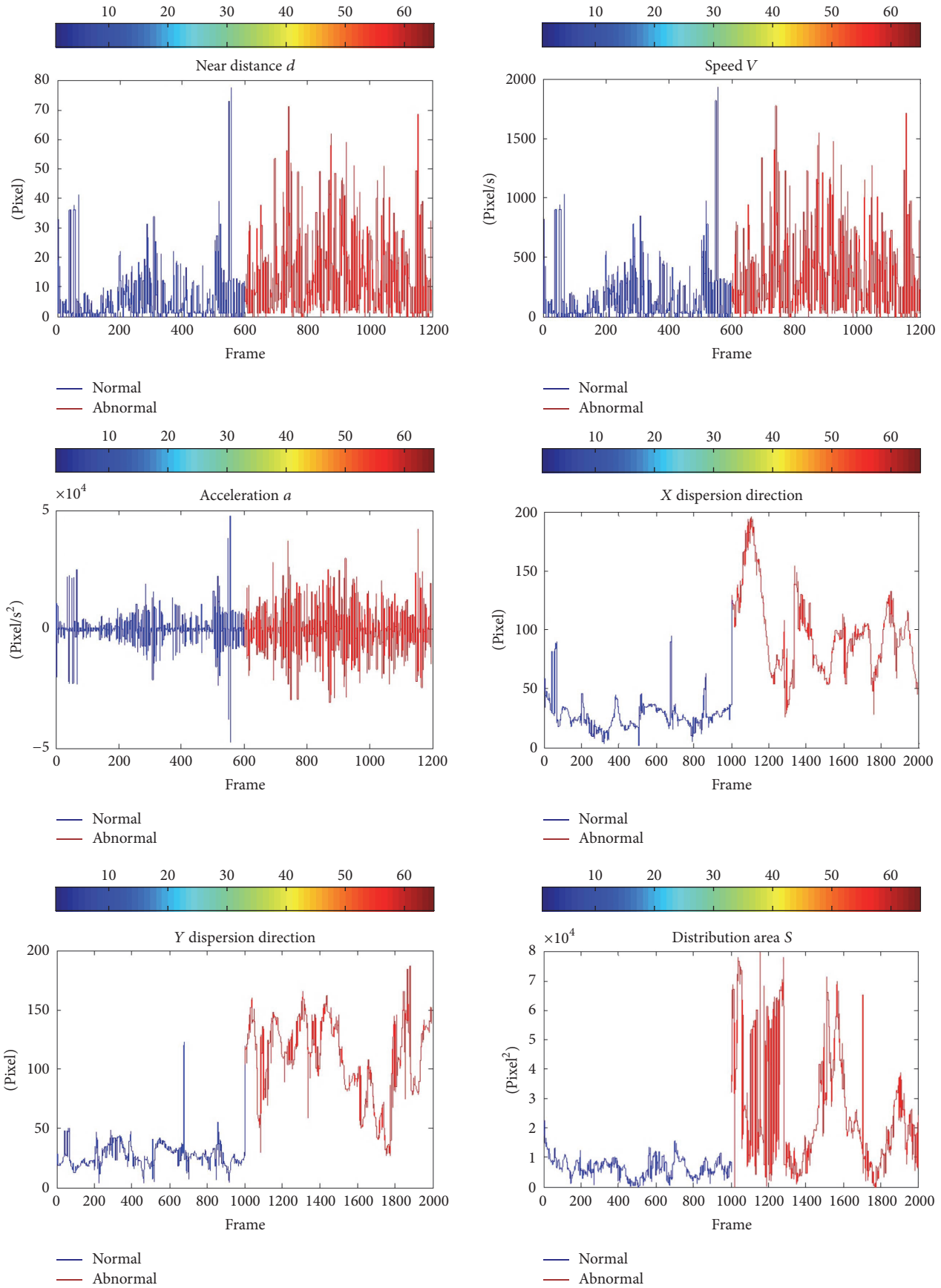


Figure 6: Schematic diagram of characteristic parameters.

Table 1: XGBoost classification performance.

Data types	Accuracy%	Time/s
Original data 1	98.66	0.0620
Original data 2	99.40	0.0590
Original data 3	98.81	0.0659
Original data 4	97.92	0.0650
Original data 5	98.56	0.0630
Original data 6	99.32	0.0645

Table 2: Characteristic parameters.

Characteristic parameter	Code
X dispersion direction–X-LS	f0
Y dispersion direction–Y-LS	f1
Distribution area–S	f2
Speed–V	f3
Acceleration–a	f4
Near distance–d	f5

Table 3: Accuracy of XGBoost classification.

Feature	Thresh	Feature	Accuracy%
X-LS	0.274	f0	93.33
S	0.250	f2	96.97
Y-LS	0.215	f1	98.18
V	0.130	f3	97.88
a	0.130	f4	97.88
d	0.000	f5	98.94

the table, the classification accuracy and time all have an excellent performance. Because the data were too much, here we choose the last set of data: original data 4 for detailed instructions. Using XGBoost the output of the classification model of original data 4 is obtained; the model of the code in the name of the characteristics of the parameters is shown in Table 2. The score of each characteristic parameter in the model was shown in Figure 7.

The specific weight of each characteristic parameter for the overall sample set and their respective classification accuracy were shown in Table 3. It can be seen from the chart during the classification processing, that the model selects the most obvious characteristic parameters; for those weak characteristic parameters, model was used to adjust the weight of its classification, especially particularly bad data directly discarded. The model iterates over each feature parameter and can be used to segment the training set by calculating the eigenvalues of each feature parameter and then using tree structured Figure 8 to view the decision-making process (3rd boosted tree). In Figure 8, the feature and the feature values for each split were shown as well as the output leaf nodes. The split decisions with each node and the different colors for left and right splits (blue and red) were also shown.

Table 4: K-fold cross validation results.

Number	Time/s	Accuracy%	Error%
(2)	0.1040	91.75	1.55
(3)	0.1749	92.55	5.39
(4)	0.2650	93.85	4.41
(5)	0.3439	92.60	5.36
(6)	0.4140	94.04	6.99
(7)	0.4830	95.15	5.52
(8)	0.5660	94.05	7.94
(9)	0.6390	94.06	6.85
(10)	0.7380	93.70	8.23

4.4. Model Evaluation and Optimization

4.4.1. k-Fold Cross Validation. This time the cross validation process is to repeat the experiment for K times, each time from the K sections to choose a different part as test data (ensure that the K part of the data was tested separately) and the rest of the $K - 1$ test data as training data for experiment. Finally, the obtained K experimental results are average. The verification results were shown in Table 4. It includes both the mean (Accuracy%) and standard deviation (Error%) classification accuracy. As can be seen from the table, XGBoost as a classifier can still maintain a high accuracy after multiple cross validation, and the time used for each classification was very small.

4.4.2. Loss Function and Classification Error. Using XGBoost as a decision tree classifier, we can also use the loss function and classification error to evaluate our water quality classification model, as shown in Figure 9. X-axis abscissa represents the number of data.

The Log Loss function is very common as the evaluation criteria; in Figure 9, you can clearly see the model loss decreases and reaching the ideal state. The classification error is divided into two kinds: one is the training error (Train), another is test error (Test); the training error stabilized at around 0.010 and the test error finally remains at a very low value and can fully meet the application of our water quality classification model.

4.5. Optimization. We optimize the processing speed of the model and use the computer multicore processor (CPU) to improve the speed of the model. From a cost point of view, the price of a single core CPU was cheaper, but the speed of computing compared with multicore CPU comparison has great gap. From Figure 10 we can see that when the number of processors in the core model selection gradually increased, the processing speed of the model has great changes too, because the test data from the scale is still relatively small, not enough to fully display the processing speed of the model, but the advantages of multicore processor in speed optimization still can be seen.

4.6. Comparison between XGBoost and SVM. SVM is one of the current mainstream classifiers that has always been a very

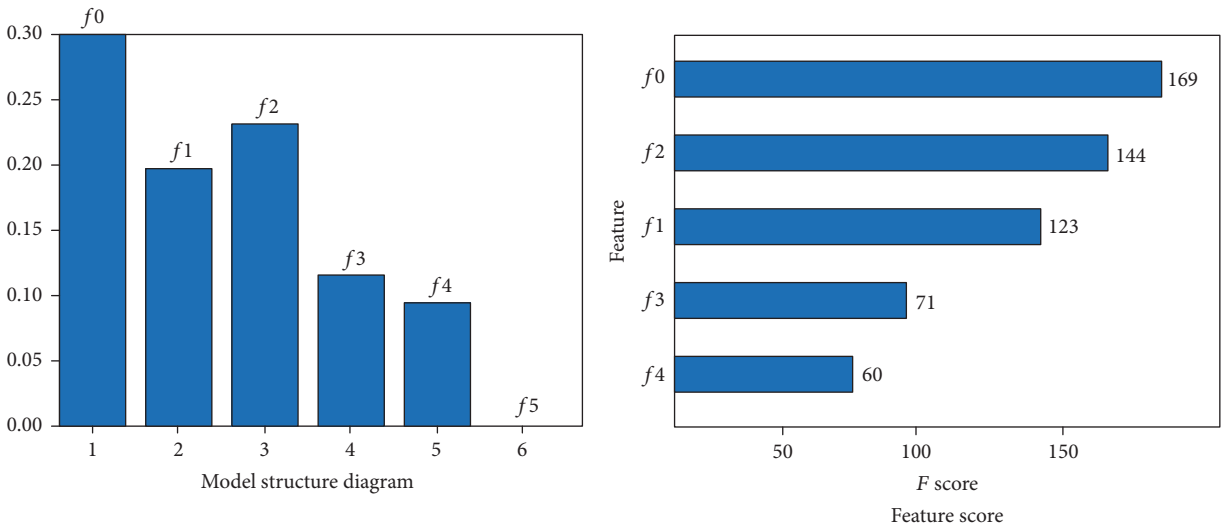


Figure 7: Schematic diagram of the model.

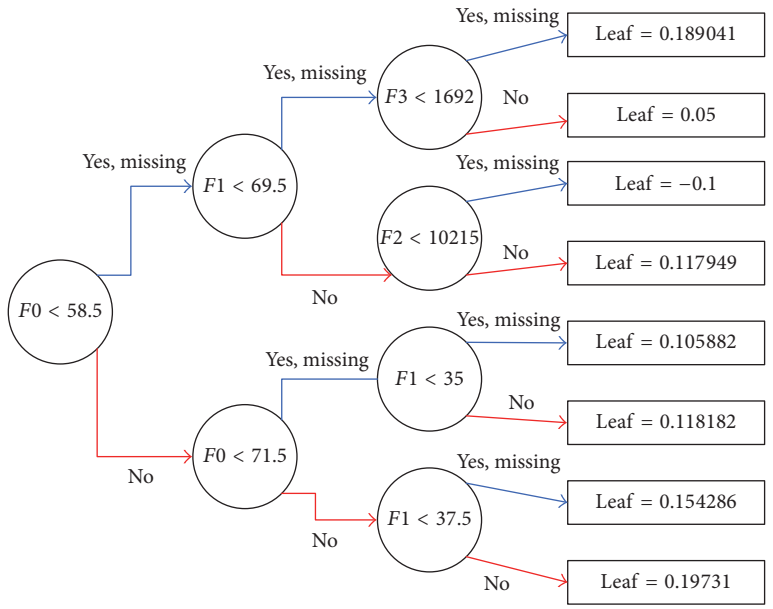


Figure 8: Tree structure.

good performance in the classification problem; in order to demonstrate the performance of the XGBoost classifier, we made a comparison between SVM and XGBoost. Because the kernel function of SVM will affect the classification results, in order to allow the SVM classifier to achieve the best working condition, we adopted the previous laboratory work [16]; using RBF kernel parameters, the penalty factor C is given in Table 5. XGBoost uses the data set for original data 4; in order to play the best working condition, SVM needs to be further normalized on the basis of original data 4.

The final SVM classification effect is shown in Figure 11.

As shown in Figure 12, compared with the accuracy of the classification and the time used in the classification, XGBoost has a huge advantage, especially the advantages of classification time; XGBoost is faster than SVM to speed up to two orders of magnitude. Details are shown in Table 5.

5. Conclusion

We obtain 3D coordinates by tracking fish in the water. A series of motion characteristic parameters which can be used to represent the water quality were calculated; in the processing of parameters, we found that some features can be used to distinguish between normal and abnormal water quality. According to this idea, water quality monitoring model was set up based on XGBoost classifiers. After a large number of

Table 5: Comparison of XGBoost and SVM classification.

Types	$f0$	$f1$	$f2$	$f3$	$f4$	$f5$
SVM-time/s	10.016	13.330	15.241	15.730	16.052	16.412
XGBoost-time/s	*0.060*	*0.066*	*0.073*	*0.075*	*0.089*	*0.094*
SVM-accuracy%	93.65	94.50	65.60	66.90	50.00	50.00
XGBoost-accuracy%	*93.45*	*96.88*	*98.07*	*98.07*	*97.92*	*97.92*
SVM-σ	0.12	0.058	0.058	0.058	0.012	0.012
SVM-C	7.46	3.73	7.46	7.46	7.46	7.46

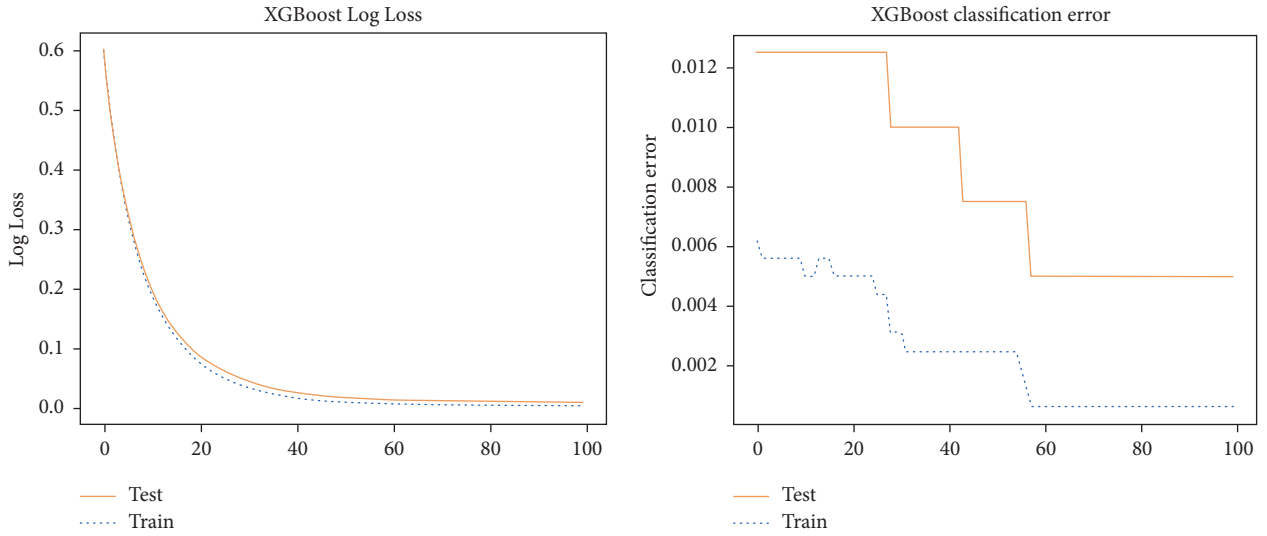


Figure 9: Loss function and classification error.

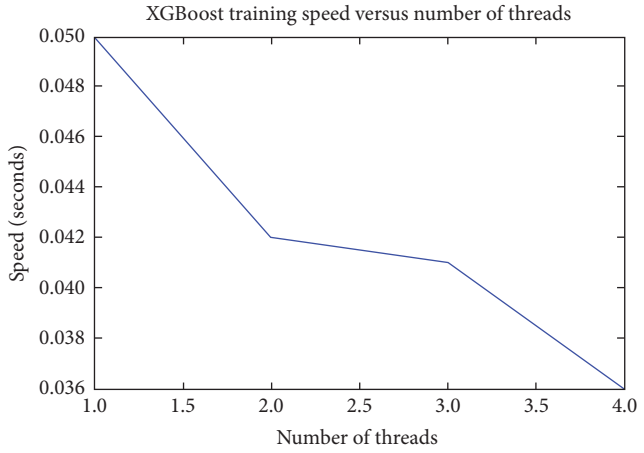


Figure 10: Multicore computing speed optimization.

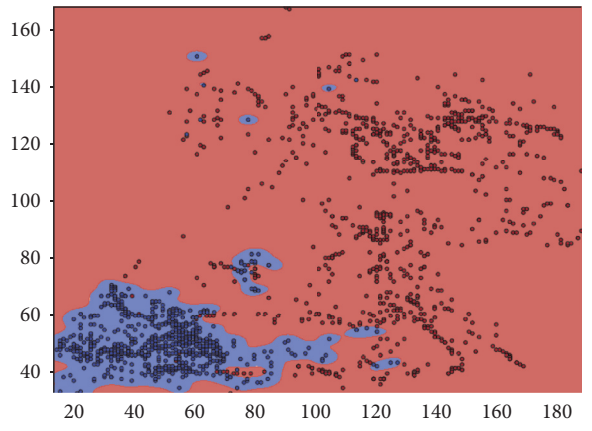


Figure 11: SVM classification effect.

experiments, this model can realize the water quality classification quickly, accurately, and conveniently. Compared with the previous classifier, XGBoost was more outstanding. However, the whole process of water quality monitoring can not achieve closure, so we must rely on line training and human error processing; the whole system can not achieve real-time monitoring. We will further strengthen the model's ability to distinguish. In the analysis of the feature image, the processing of the intermediate state can be transformed into the frequency domain by the way of signal processing, removing interference information, and improving the accuracy of early warning. And as a follow-up we can deteriorate the experimental environment and enhance the robustness of the system.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

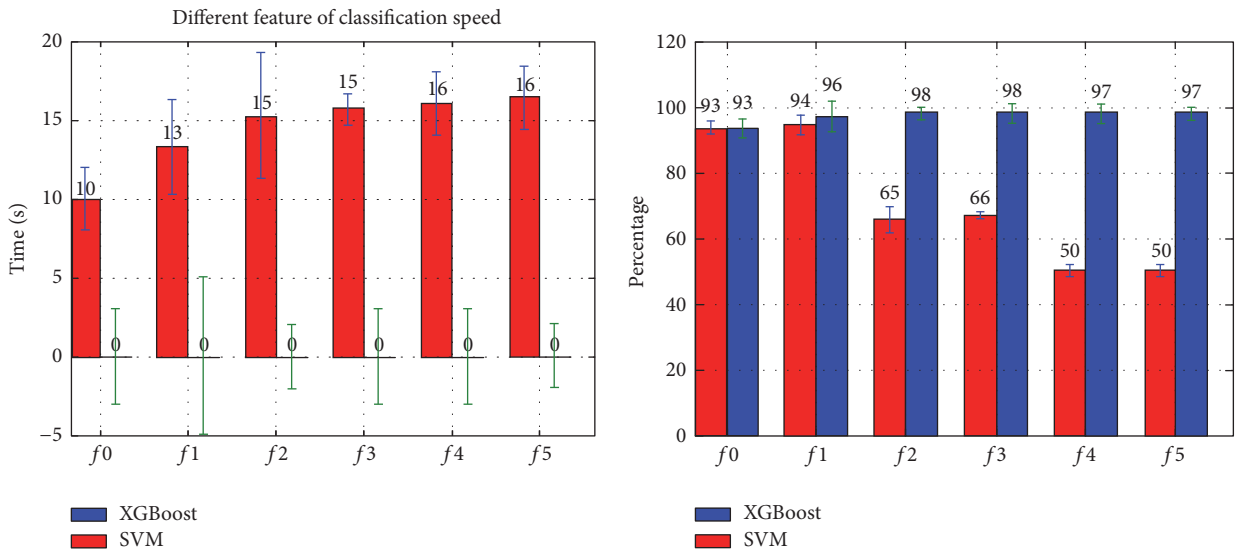


Figure 12: Classification speed and accuracy of XGBoost and SVM.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61601400) and the Postdoctoral Scientific Research Project of Hebei Province (Grant no. B2016003027).

References

[1] V. Simeonov, P. Simeonova, S. Tsakovski, and V. Lovchinov, "Lake water monitoring data assessment by multivariate statistics," *Journal of Water Resource and Protection*, vol. 2, no. 4, pp. 353–361, 2010.

[2] A. Jang, Z. Zou, K. K. Lee, C. H. Ahn, and P. L. Bishop, "State-of-the-art lab chip sensors for environmental water monitoring," *Measurement Science and Technology*, vol. 22, no. 3, Article ID 032001, 2011.

[3] M. C. Kim, W. M. Shin, M. S. Jeong et al., "Real-time motion generating method for artifical fish," *Computer Science and Network Security*, vol. 10, no. 7, pp. 52–61, 2007.

[4] K. Nimkerdphol and M. Nakagawa, "Effect of sodium hypochlorite on zebrafish swimming behavior estimated by fractal dimension analysis," *Journal of Bioscience and Bioengineering*, vol. 105, no. 5, pp. 486–492, 2008.

[5] H. Ma, T.-F. Tsai, and C.-C. Liu, "Real-time monitoring of water quality using temporal trajectory of live fish," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5158–5171, 2010.

[6] C. Jiujun, X. Gang, Y. Xiaofang et al., "Fish activity model based on tail swing frequency," *Journal of Image and Graphics*, vol. 14, no. 10, pp. 2177–2180, 2009.

[7] H. Jianglong, F. Jinglong, and W. Daquan, "Water quality monitoring using multi-object tracking algorithm," *Journal of Mechanical & Electrical Engineering*, vol. 29, no. 5, pp. 613–615, 2012.

[8] S.-H. Cheng, J. Cai, and C.-H. Hu, "Fish motion tracking research based on video algorithm," *Guangdian Gongcheng/Opto-Electronic Engineering*, vol. 38, no. 2, pp. 14–18, 2011.

[9] C. Serra-Toro, R. Montoliu, V. J. Traver, I. M. Hurtado-Melgar, M. Núñez-Redó, and P. Cascales, "Assessing water quality by video monitoring fish swimming behavior," in *Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR 2010*, pp. 428–431, August 2010.

[10] C.-L. Lai and C.-L. Chiu, "Using image processing technology for water quality monitoring system," in *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics, ICMLC 2011*, pp. 1856–1861, Guilin,China, July 2011.

[11] J. Zhangzan, X. Gang, C. Jiujun et al., "Anomaly detection of water quality based on visual perception and V-detector," *Information and Control*, vol. 40, no. 1, pp. 130–136, 2011.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: tracking-learning-detection applied to faces," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 3789–3792, Hong Kong, China, September 2010.

[14] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 785–794, August 2016.

[15] T. Chen, T. He, and M. Benesty, "xgboost: Extreme Gradient Boosting," 2017.

[16] S. Cheng and J. Liu, "A method of water quality monitoring based on computer vision and SVM," *Opto-Electronic Engineering*, vol. 5, pp. 28–33, 2014.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 1259703, 10 pages
https://doi.org/10.1155/2018/1259703

Research Article

Image Denoising Algorithm Combined with SGK Dictionary Learning and Principal Component Analysis Noise Estimation

Wenjing Zhao, Yue Chi, Yatong Zhou, and Cheng Zhang

Tianjin Key Laboratory of Electronic Materials and Devices, Hebei University of Technology, Tianjin 300401, China

Correspondence should be addressed to Yatong Zhou; zhouyatong_zw@126.com

Received 21 August 2017; Revised 31 January 2018; Accepted 13 February 2018; Published 14 March 2018

Academic Editor: Daniel Zaldivar

SGK (sequential generalization of K-means) dictionary learning denoising algorithm has the characteristics of fast denoising speed and excellent denoising performance. However, the noise standard deviation must be known in advance when using SGK algorithm to process the image. This paper presents a denoising algorithm combined with SGK dictionary learning and the principal component analysis (PCA) noise estimation. At first, the noise standard deviation of the image is estimated by using the PCA noise estimation algorithm. And then it is used for SGK dictionary learning algorithm. Experimental results show the following: (1) The SGK algorithm has the best denoising performance compared with the other three dictionary learning algorithms. (2) The SGK algorithm combined with PCA is superior to the SGK algorithm combined with other noise estimation algorithms. (3) Compared with the original SGK algorithm, the proposed algorithm has higher PSNR and better denoising performance.

1. Introduction

In the image acquisition and transmission, noise is inevitably carried, which will reduce image quality, so image denoising has a very important significance. Image denoising algorithms can be divided into space domain denoising and frequency domain denoising. The former includes the mean filtering, median filtering, and Wiener filtering. The latter includes Fourier transform [1], Laplace transform [2], and wavelet transform [3]. A series of postwavelet multiscale tools have been developed based on the wavelet theory to filter noise effectively such as curvelet [4], directionlet [5], bandelet [6], and shearlet [7].

In recent years, there are some novel denoising algorithms such as nonlocal mean [8] denoising, Gaussian mixture model denoising [9], and dictionary learning denoising [10] based on sparse representation [11]. An image denoising method based on wavelet and SVD transforms improves denoising performance [12]. Moreover, K-singular value decomposition (K-SVD) [13] based on overcomplete sparse representation has recently been the subject of intense research activity within the denoising community [14, 15]. However, K-SVD increases the iteration number when dealing with large data. So Sujit proposed the SGK [13] dictionary learning algorithm in 2013, which not only overcomes the drawbacks of ordinary dictionary learning that breaks the sparse coefficient structure but also can be applied to a variety of sparse representations, with the low complexity and fast calculation ability [10].

At present, many image denoising algorithms need to foreknow the noise standard deviation [16], but it is usually unknown in practice. So the noise estimation has been developed in the image denoising community. The classic image filtering in [17] estimates the noise standard deviation by the convolution of image and filter. The DCT of the image patch [18] concentrates the image structure in the low frequency coefficient region, so that the noise estimation can be performed by the high frequency coefficient. It is also common to estimate noise level by the grayscale value of the image [19]. Patch-based local variance [20] generally estimates noise level by robust statistical algorithms. The Bayesian contraction algorithm [21] is used to denoise the image and analyze the autocorrelation of residuals in the range of noise standard deviation to find the true value. The distribution of the sideband filter response [22] can be divided into two parts according to the difference of the image and noise, which is calculated by the expected

maximization [23]. The kurtosis of the edge sideband filter response distribution [24] is constant for the noisy image, and a kurtosis model can be established and the noise standard deviation can be evaluated by finding the best parameters of the model. However, the above algorithms mostly assume that the image is uniform. For images with abundant textures, Pyatykh et al. [25] proposed PCA noise estimation based on the data patch, where the noise standard deviation can be estimated as the minimum eigenvalue of the image patch covariance matrix.

Based on the above considerations, a denoising algorithm combined with SGK dictionary learning and PCA noise estimation is proposed. Firstly, the image with additive Gaussian white noise is segmented, and the noise level is estimated by calculating the minimum eigenvalue of the image patch covariance matrix. Then the estimated noise standard deviation is entered into SGK dictionary learning algorithm to denoise the image. During the denoising process, each image patch is sparse and the sparse representation coefficient is calculated by pursuit algorithm. The dictionary atom is updated with the sparse representation coefficient; therefore a more accurate approximation of the image patch is obtained. The experimental results show that the proposed algorithm is superior to other algorithms in noise level estimation and has better denoising performance.

2. SGK Dictionary Learning Denoising Algorithm

2.1. Image Denoising Problem and SGK Dictionary Learning. SGK dictionary learning algorithm is a generalization of the K-means clustering. It mainly consists of two stages: sparse coding stage and dictionary update stage when using SGK dictionary learning algorithm to perform denoising [13], and the flow chart is shown in Figure 1. SGK algorithm firstly processes image through the original DCT dictionary and then updates dictionary with the sparse representation coefficient. Each local patch extracted in the image is sparse-coded by new training dictionary to achieve the denoising performance.

2.2. Sparse Coding Stage. For an image $\mathbf{A}$ of size $\sqrt{T} \times \sqrt{T}$ added to additive white Gaussian noise $\mathbf{W} \in R^{\sqrt{T}\times\sqrt{T}}$, it constitutes a noisy image $\mathbf{B}$:

$$\mathbf{B} = \mathbf{A} + \mathbf{W}. \tag{1}$$

Assume that the dictionary $\mathbf{D} \in R^{t\times l}$ consists of image atoms $\mathbf{d}_l \in R^t$, where $l = 1, 2, \ldots, L$. $\mathbf{Q}_{ij}$ represents a $t \times T$ matrix that extracts patches of size $\sqrt{t} \times \sqrt{t}$ in image $\mathbf{A}$, which is $\forall_{ij}\ \{\mathbf{Q}_{ij}\mathbf{A} \in R^t\}$. For each local patch, the sparse representation $\mathbf{a} = \mathbf{Q}_{ij}\mathbf{A}$ can be represented by a dictionary $\mathbf{D}$:

$$\widehat{\beta} = \arg\min_{\beta} \quad \left\{\eta \|\beta\|_0 + \|\mathbf{D}\beta - \mathbf{a}\|_2^2\right\}. \tag{2}$$

For any patch in the image,

$$\widehat{\beta}_{ij} = \arg\min_{\beta_{ij}} \quad \left\{\eta_{ij} \left\|\beta_{ij}\right\|_0 + \left\|\mathbf{D}\beta_{ij} - \mathbf{Q}_{ij}\mathbf{A}\right\|_2^2\right\} \quad \forall_{ij}. \tag{3}$$

Therefore the global image representation is shown as

$$\left\{\widehat{\mathbf{A}}, \widehat{\beta}_{ij}\right\} = \arg\min_{\mathbf{A},\beta_{ij}} \quad \left\{\rho \|\mathbf{B} - \mathbf{A}\|_2 + \sum_{ij} \eta_{ij} \left\|\beta_{ij}\right\|_0 + \sum_{ij} \left\|\mathbf{D}\beta_{ij} - \mathbf{Q}_{ij}\mathbf{A}\right\|_2\right\}. \tag{4}$$

For the solution of (4), β_{ij} can be obtained by (3), and then $\mathbf{B}$ is represented as sparse approximation of $\mathbf{A}$ by choosing appropriate η_{ij}, so it can be obtained as

$$\begin{aligned} \widehat{\beta}_{ij} = \arg\min_{\beta_{ij}} \quad & \left\|\beta_{ij}\right\|_0 \\ \text{s.t.} \quad & \left\|\mathbf{Q_{ij}B} - \mathbf{D}\beta_{ij}\right\|_2^2 \le (C\sigma)^2 \\ & \forall_{ij}. \end{aligned} \tag{5}$$

2.3. Dictionary Update Stage. In the dictionary update stage, updating each image's atoms sequentially can minimize sparse representation error, which is denoted as

$$\mathbf{g}_{ij} = \mathbf{Q}_{ij}\mathbf{A} - \mathbf{D}\beta_{ij} = \mathbf{Q}_{ij}\mathbf{A} - \sum_r \mathbf{d}_m \beta_{ij}(r). \tag{6}$$

In (6), $\beta_{ij}(r)$ is the rth component of β_{ij}. And the error matrix $\mathbf{G}$ is composed of all these elements $\{\mathbf{g}_{ij}\}$. So the error of the image patch $\mathbf{d}_l$ can be expressed as

$$\mathbf{g}_{ij}^l = \mathbf{Q}_{ij}\mathbf{A} - \sum_{r\neq l} \mathbf{d}_m \alpha_{ij}(r) = \mathbf{g}_{ij} + \mathbf{d}_l \alpha_{ij}(l). \tag{7}$$

All these $\{\mathbf{g}_{ij}^l\}$ form the error matrix $\mathbf{G}^l$ and also form vector $\boldsymbol{\beta}_l$ containing corresponding $\{\beta_{ij}(l)\}$, so it has

$$\begin{aligned} \mathbf{G}^l = \mathbf{E} + \mathbf{d}_l\boldsymbol{\beta}_l \Longrightarrow \\ \|\mathbf{G}\|_F^2 = \left\|\mathbf{G}^l - \mathbf{d}_l\boldsymbol{\beta}_l\right\|_F^2. \end{aligned} \tag{8}$$

And $\|\cdot\|_F$ is Frobenius norm in (8). According to the sequential generalization of K-means [12], the solution of (8) is

$$\mathbf{d}_l^{(t+1)} = \arg\min_{\mathbf{d}_l} \quad \|\mathbf{G}\|_F^2 = \arg\min_{\mathbf{d}_l} \left\|\mathbf{G}^l - \mathbf{d}_l\boldsymbol{\beta}_l\right\|_F^2. \tag{9}$$

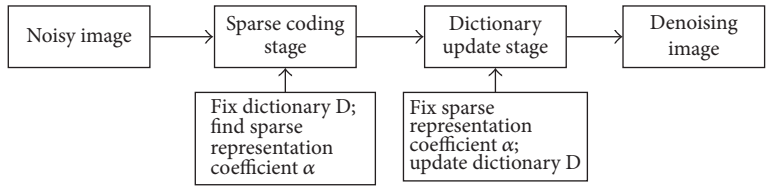


Figure 1: SGK algorithm denoising flow chart.

The closed-form solution of (9) is

$$\widehat{\mathbf{d}}_l = \mathbf{G}^l \boldsymbol{\beta}_l^T \left(\boldsymbol{\beta}_l \boldsymbol{\beta}_l^T\right)^{-1}. \tag{10}$$

It replaces all atoms with $\mathbf{d} = \widehat{\mathbf{d}}_l$ and SGK training dictionary $\widehat{\mathbf{D}}$ is used to obtain the final sparse representation of the component $\widehat{\beta}_{ij}$ for each extracted local patch. $\widehat{\mathbf{A}}$ is obtained as

$$\widehat{\mathbf{A}} = \arg\min_{\mathbf{A}} \left\{ \rho \|\mathbf{B} - \mathbf{A}\|_2 + \sum_{ij} \left\| \mathbf{D}\widehat{\beta}_{ij} - \mathbf{Q}_{ij}\mathbf{A} \right\|_2 \right\}. \tag{11}$$

The final solution of the sparse representation error minimization problem is

$$\widehat{\mathbf{A}} = \left(\lambda \mathbf{I}_{\mathbf{N}} + \sum_{ij} \mathbf{Q}_{ij}^T \mathbf{Q}_{ij}\right)^{-1} \left(\lambda \mathbf{B} + \sum_{ij} \mathbf{Q}_{ij}^T \mathbf{D}\widehat{\beta}_{ij}\right). \tag{12}$$

3. Noise Estimation Theory

3.1. Noise Estimation Theory Based on PCA. Suppose that $\mathbf{A}$ is a clean image with size of $\sqrt{T}\times\sqrt{T}$, and $\mathbf{B}$ represents an image with additive white Gaussian noise $\mathbf{W}$. The noise variance is unknown, so it needs to be estimated. For $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$, each image contains $H = (\sqrt{T} - \sqrt{M} + 1)(\sqrt{T} - \sqrt{M} + 1)$ patches with size $M = \sqrt{M}\times\sqrt{M}$. Since $\mathbf{W}$ is the independent additive white Gaussian noise, there is $\mathbf{W} \sim N_M(0, \sigma^2\mathbf{I})$ and $\mathrm{cov}(\mathbf{A}, \mathbf{W}) = 0$.

Suppose that $\mathbf{J}_\mathbf{A}$ and $\mathbf{J}_\mathbf{B}$ are, respectively, the sample covariance matrices of $\mathbf{A}$ and $\mathbf{B}$. Meanwhile, $\widetilde{\mu}_{\mathbf{A},1} \geq \widetilde{\mu}_{\mathbf{A},2} \geq \cdots \geq \widetilde{\mu}_{\mathbf{A},M}$ are the eigenvalues of $\mathbf{J}_\mathbf{A}$, and the corresponding eigenvectors are $\widetilde{\mathbf{W}}_{\mathbf{A},1}, \ldots, \widetilde{\mathbf{W}}_{\mathbf{A},M}$. Similarly, $\widetilde{\mu}_{\mathbf{B},1} \geq \widetilde{\mu}_{\mathbf{B},2} \geq \cdots \geq \widetilde{\mu}_{\mathbf{B},M}$ are the eigenvalues of $\mathbf{J}_\mathbf{B}$, and corresponding eigenvectors are $\widetilde{\mathbf{W}}_{\mathbf{B},1}, \ldots, \widetilde{\mathbf{W}}_{\mathbf{B},M}$. $\widetilde{\mathbf{W}}_{\mathbf{B},1}^T\mathbf{B}, \ldots, \widetilde{\mathbf{W}}_{\mathbf{B},M}^T\mathbf{B}$ represent the sample principal component of $\mathbf{B}$ [26], and S^2 represents the sample variance, so it is shown as follows:

$$S^2\left(\widetilde{\mathbf{W}}_{\mathbf{B},k}^T\mathbf{B}\right) = \widetilde{\mu}_{\mathbf{B},k}, \quad k = 1, 2, \ldots, M, \tag{13}$$

where S^2 represents the sample variance.

In order to apply PCA to noise variance estimation, it defines positive integer c. The clean image $\mathbf{A}$ satisfies $\mathbf{A}_i \in \mathbf{W}_{M-c} \subset R^M$, and its dimension $M - c$ is less than the number of coordinates M. So there is

$$\mathbf{E}\left(\left|\widetilde{\mu}_{\mathbf{B},h} - \sigma^2\right|\right) = O\left(\frac{\sigma^2}{\sqrt{H}}\right), \quad H \longrightarrow \infty. \tag{14}$$

And it is held for all $h = M - c + 1, \ldots, M$. When considering the overall principal component, $\mathrm{cov}(\mathbf{A}, \mathbf{W}) = 0$ represents $\sum_\mathbf{B} = \sum_\mathbf{A} + \sum_\mathbf{W}$, where $\sum_\mathbf{B}$, $\sum_\mathbf{A}$, and $\sum_\mathbf{W}$, respectively, represent the overall covariance matrices of $\mathbf{B}$, $\mathbf{A}$, and $\mathbf{W}$. Meanwhile, the minimum eigenvalues of $\sum_\mathbf{W} = \sigma^2\mathbf{I}$ and $\sum_\mathbf{A}$ are zero, so the minimum eigenvalue of $\sum_\mathbf{B}$ is σ^2. With the sample size N tending to infinity, it meets

$$\lim_{N\to\infty} \mathbf{E}\left(\left|\widetilde{\mu}_{\mathbf{B},M} - \sigma^2\right|\right) = 0, \tag{15}$$

which represents the fact that $\widetilde{\mu}_{\mathbf{B},M}$ converges to σ^2, so the noise variance can be estimated as $\widetilde{\mu}_{\mathbf{B},M}$ and it is a consistent estimation of the noise level.

If the above assumptions hold, the expected values of $\widetilde{\mu}_{\mathbf{B},M-c+1} - \widetilde{\mu}_{\mathbf{B},M}$ can be calculated from the trigonometric inequality and (15):

$$\mathbf{E}\left(\widetilde{\mu}_{\mathbf{B},M-c+1} - \widetilde{\mu}_{\mathbf{B},M}\right) = \mathrm{O}\left(\frac{\sigma^2}{\sqrt{H}}\right). \tag{16}$$

The condition of (16) is

$$\widetilde{\mu}_{\mathbf{B},M-c+1} - \widetilde{\mu}_{\mathbf{B},M} < \frac{T\sigma^2}{\sqrt{H}}, \tag{17}$$

where T is a fixed value and it satisfies $T > 0$.

For the estimation of noise variance σ_{est}^2, it can be verified by (17). If (17) holds, σ_{est}^2 is the final estimation. But if (17) cannot hold, it is necessary to extract a subset of the image patches with a small standard deviation. Performing noise estimation again to satisfy (17) is satisfied until the final noise estimation is obtained.

3.2. Estimate the Noise Level Based on PCA: An Example. Figure 2 is a PCA noise estimation example of the house image. Figure 2(a) is the house image, and Figure 2(b) shows the noise estimation results under the different noise standard deviation. It can be seen that the PCA noise estimation value is very close to the true value.

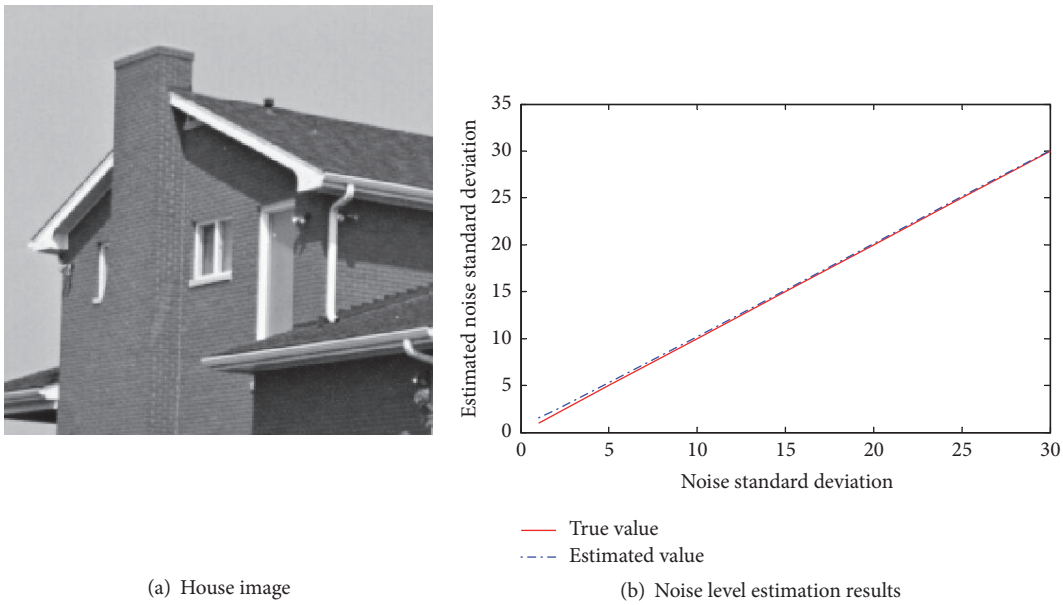
4. Experimental Results' Analysis

The standard Kodak Photo CD benchmark was used to evaluate the performance of denoising algorithm. The size of some images is $256 * 256$, and the size of other images is $512 * 512$. The patches sizes of all images are 8×8.

4.1. Comparison of Four Dictionary Learning Denoising Algorithms. We conduct experiments for image denoising by using SGK, DCT, Global, K-SVD dictionary learning

Table 1: Image denoising results of five dictionary learning algorithms.

Algorithms	Time/s	PSNR/dB	MSE
SGK	17.378	32.077	77.470
DCT	83.884	31.079	113.873
Global	100.479	31.734	89.435
K-SVD	284.181	32.171	72.383
BM3D	12.256	30.717	119.478



(a) House image

(b) Noise level estimation results

Figure 2: Noise estimation based on principal component analysis.

algorithms, and BM3D algorithms. The similarity of the first four algorithms is to build a dictionary and then use the dictionary to denoise. DCT algorithm denoises an image by sparsely representing each block with the overcomplete DCT dictionary, thus averaging the represented parts [11]. Global algorithm denoises an image by training a dictionary on patches from the noisy image, sparsely representing each block with this dictionary and averaging the represented parts [11]. K-SVD algorithm uses DCT dictionary to initialize and then uses singular value decomposition for dictionary updating [11]. BM3D algorithm is an image signal denoising method based on transform domain enhancement sparse representation [26].

Throughout this experiment, we use SGK, DCT, Global, K-SVD, and BM3D algorithms to denoise the Barbara image with $\sigma = 25$ as an example. As shown in Figures 3(b)–3(f), the denoising results of five different algorithms are basically the same, and the image's details are basically well preserved. In the following, we do quantitative comparisons between five algorithms, and the experimental data is shown in Table 1. The PSNR of the SGK is similar to that of K-SVD, and it is superior to Global, DCT, and BM3D algorithms. Encouragingly, we see that SGK runs much faster than K-SVD. As to the value of MSE, SGK is smaller than Global, DCT, and BM3D algorithms. With all the above-mentioned results, the denoising supremacy of SGK over the rest algorithms is demonstrated.

4.2. Comparison of SGK Combined with Different Noise Estimation Algorithms. In order to analyze the sensitivity of SGK algorithm to the noise level, the noise standard deviation is set as $\sigma = 5, 10, 15, 20, 25$, respectively. SGK algorithm is used to denoise the Peppers image with different noise standard deviation.

Figure 4 shows the denoising experiment's results using the SGK algorithm with different offsets. The five different colors curves, respectively, represent the case where the PSNR varies with the noise offsets if the noise standard deviation is given. It can be found in Figure 4 that the PSNR of denoised image is basically invariant when the noise standard deviation has negative offset of 0~−5% and the forward offset of 0~+5%. When the offset of noise standard deviation continues to increase, it shows a significant downward trend in −5%~−25% of the negative offset and 5% to 25% of forward offset, which indicates that the image's PSNR is significantly reduced. Therefore the PSNR would be changed when the noise standard deviation has negative or forward offset, which shows that the SGK algorithm is sensitive to the offsets of the noise standard deviation. So it is necessary to estimate noise level before image denoising. If the estimated noise level is close to the true noise, the denoising results will be more accurate.

In order to analyze the performances of SGK dictionary denoising combined with different noise estimation

(a) Original Barbara image (b) SGK denoised image (c) DCT denoised image

(d) Global denoised image (e) K-SVD denoised image (f) BM3D denoised image

Figure 3: Using five kinds of dictionary learning algorithm for image denoising.

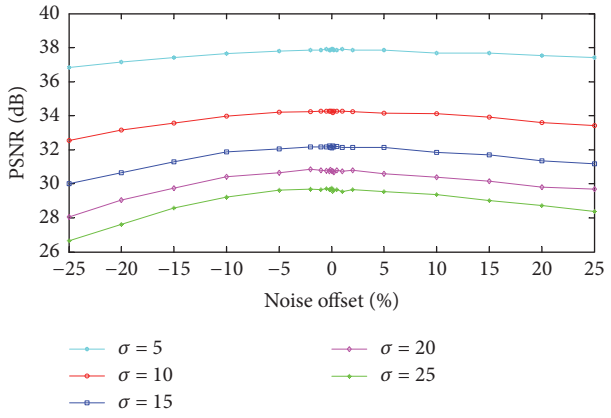


Figure 4: SGK denoising performance with different noise offset.

algorithms, this paper also introduces four noise estimation algorithms: Kurtosis [25], local standard deviation distribution mode (Mode) [27], local standard deviation (Med) [27], and local standard deviation minimum (Min) [27].

Kurtosis assumes that the corresponding distribution of the kurtosis edge bandpass filter should be a constant for the noise-free images. However, the kurtosis at the entire scale of a noisy image may vary. Under this assumption, the noise standard deviation can be estimated by the kurtosis model.

Mode can divide image and the noise standard deviation is estimated according to the distribution pattern of the image's local standard deviation. As the variance of noise is constant throughout the picture, it will affect every local variance value equally. As a result, the maximum of the bell-shaped distribution will reflect the local variance of the degraded image within homogeneous areas. This value is the mode of the distribution, and it is very close to the mode, so we can use the mode to estimate it.

Med estimates noise standard deviation based on the median of the image's local standard deviation. If practical difficulties to properly estimate the mode arise, it may be useful to use the median operator instead, due to its greater simplicity and the fact that both parameters, albeit different, are not far apart in practice. So an alternative estimation procedure can be done.

Min estimates the noise standard deviation based on the minimum value of the local standard deviation of the images. Within a uniform area, the variance of the degraded image equals the variance of noise. According to the previous statement, one straightforward way to estimate standard deviation is to calculate the variance within homogeneous regions, where the variance of the original image is close to zero.

The above noise estimation algorithms are all combined with the SGK dictionary learning algorithm in this paper. Now we combine the five noise estimation algorithms mentioned above with SGK, respectively. The combined algorithms are referred to as PCA + SGK, Kurtosis + SGK, Mode + SGK, Med + SGK, and Min + SGK. Figure 5 shows the

(a) Original Lena image (b) PCA + SGK (c) Kurtosis + SGK

(d) Mode + SGK (e) Med + SGK (f) Min + SGK

Figure 5: Comparison of five kinds of algorithms.

denoising results of the Lena image, where the noise standard deviation is set to $\sigma = 15$. Analysis of Figure 5 leads to the following observation: the denoising performance of Mode + SGK in Figure 5(d) is slightly the worst, while the remaining four algorithms are basically the same. They all retain the details of the image in the denoising process.

In order to analyze the denoising performance of five algorithms quantitatively, additive Gaussian white noise images with $\sigma = 2, 5, 10, 20, 30, 40$ are, respectively, denoised and compared in Table 2. The best results are shown in boldface. Experiments performed on noisy Lena image indicate that the proposed algorithm outperforms, in terms of estimation accuracy $|\sigma_{\text{est}} - \sigma|$, estimation time, PSNR, and MSE, the four existing algorithms.

Figure 6 illustrates the performance of algorithms more intuitively varying with standard deviation. Figure 6(a) shows variation of noise estimation absolute error with the noise standard deviation. It can be seen that PCA + SGK has the least value, which is superior to the other four algorithms. So the estimation of PCA + SGK is the most accurate. Figure 6(b) shows the variation of the noise estimation time with the noise standard deviation. For the noise estimation time, Mode + SGK, Med + SGK, and Min + SGK are the least accurate, followed by PCA + SGK and Kurtosis + SGK. Figure 6(c) shows the variation of PSNR with the noise standard deviation. With the increase of standard deviation, PCA + SGK and Kurtosis + SGK keep the PSNR higher, followed by Med + SGK and Mode + SGK, and Min + SGK has the lowest PSNR. Figure 6(d) shows the variation of MSE with the noise standard deviation. In this experiment, PCA + SGK owns the lowest value of MSE, which means that the denoising performance is better than any of the other four algorithms.

4.3. Denoising Experiment of Noisy Image with Unknown Standard Deviation. The above experiments presumably assumed that the standard deviation of the noise contained in the image is known. In order to demonstrate the advantage of the proposed PCA + SGK algorithm, it is used to denoise the noisy image with unknown standard deviation and compare it with the original SGK algorithm. Twelve classic original images are shown in Figure 7. These images are mixed into additive white Gaussian noise with unknown standard deviation. After that, we do PCA + SGK denoising and SGK denoising, respectively. When SGK is used for denoising, the standard deviation can only be guessed based on the noisy image or given a random value because the noise standard deviation is unknown. When PCA + SGK is used for denoising, the noise standard deviation is first estimated by PCA and thus is entered into SGK for denoising. Denoising results of the Cameraman image using these two algorithms are shown in Figure 8. Figure 8(a) shows the noise image with $\sigma = 10$. Figure 8(b) shows the denoised image by SGK and Figure 8(c) shows the denoised image by PCA + SGK. The experiment testifies for the good performance of our approach. It can be seen that the denoising performance of

Table 2: Denoising indicators on SGK combined with five noise estimation algorithms.

σ	Algorithms	σ_{est}	$\sigma_{est} - \sigma$	$\lvert\sigma_{est} - \sigma\rvert$	Time/s	PSNR/dB	MSE
$\sigma = 2$	PCA + SGK	**2.733**	**+0.733**	**0.733**	1.799	**43.385**	**2.983**
	Kurtosis + SGK	2.841	+0.841	0.841	4.699	43.378	2.987
	Mode + SGK	2.768	+0.768	0.768	0.070	43.222	3.010
	Med + SGK	4.063	+2.063	2.063	0.051	41.811	4.286
	Min + SGK	0.496	−1.504	1.504	**0.037**	42.119	3.993
$\sigma = 5$	PCA + SGK	**5.510**	**+0.510**	**0.510**	1.829	**38.383**	**9.436**
	Kurtosis + SGK	5.166	+0.166	0.166	5.865	38.302	9.614
	Mode + SGK	4.480	−0.520	0.520	0.091	38.139	9.981
	Med + SGK	6.194	+1.194	1.194	0.047	38.251	9.727
	Min + SGK	0.674	−4.326	4.326	**0.035**	34.175	24.865
$\sigma = 10$	PCA + SGK	**10.342**	**+0.342**	**0.342**	1.604	**35.177**	**19.201**
	Kurtosis + SGK	9.576	+0.424	0.424	4.848	34.795	21.559
	Mode + SGK	8.809	+1.191	1.191	0.090	34.074	25.447
	Med + SGK	10.463	+0.463	0.463	0.051	35.149	19.869
	Min + SGK	1.640	−18.360	18.360	**0.037**	28.244	97.417
$\sigma = 15$	PCA + SGK	**15.153**	**+0.147**	**0.147**	1.995	**34.269**	**30.632**
	Kurtosis + SGK	14.196	+0.804	0.804	5.208	33.899	30.760
	Mode + SGK	12.494	+2.506	2.506	0.084	31.381	47.311
	Med + SGK	14.838	−0.162	0.162	0.082	33.221	30.971
	Min + SGK	1.807	−13.193	13.193	**0.029**	28.274	96.766
$\sigma = 20$	PCA + SGK	**20.071**	**+0.071**	**0.071**	1.618	**31.911**	**41.876**
	Kurtosis + SGK	18.943	−1.057	1.057	4.818	31.574	45.256
	Mode + SGK	17.038	−2.963	2.963	0.094	30.186	62.300
	Med + SGK	19.319	−0.681	0.681	0.055	31.777	43.193
	Min + SGK	1.877	−18.123	18.123	**0.028**	22.152	396.153
$\sigma = 30$	PCA+SGK	**29.672**	**−0.328**	**0.328**	1.393	**29.917**	**66.286**
	Kurtosis + SGK	28.480	−1.520	1.520	5.126	29.724	69.292
	Mode + SGK	23.356	−6.644	6.644	0.084	29.044	161.709
	Med + SGK	28.199	−1.801	1.801	0.054	29.614	71.074
	Min + SGK	3.292	−26.708	26.708	**0.029**	18.675	882.180
$\sigma = 40$	PCA + SGK	**39.546**	**−0.454**	**0.454**	1.865	**28.371**	**94.630**
	Kurtosis + SGK	38.238	−1.762	1.762	5.134	28.305	96.062
	Mode + SGK	33.463	−6.537	6.537	0.087	25.813	170.526
	Med + SGK	37.068	−2.932	2.932	0.053	27.910	105.212
	Min + SGK	5.385	−34.615	34.615	**0.034**	16.222	1152.049

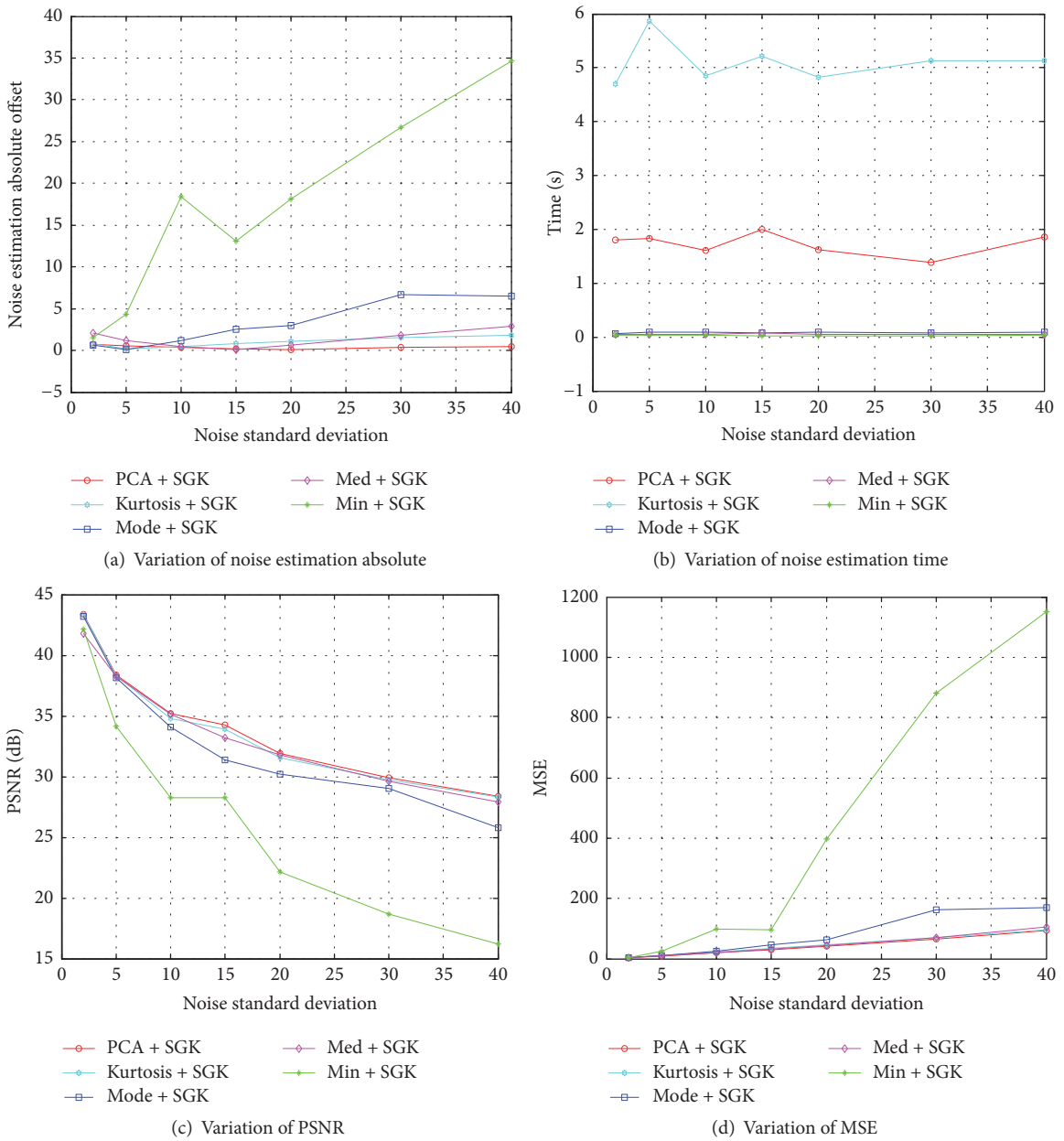
PCA + SGK is better than SGK, which retains more details of the original image.

The denoising results of 12 images are shown in Table 3. It is seen that the PSNR of SGK is less than that of PCA + SGK; that is, PCA + SGK has better denoised performance than PCA. Because the standard deviation of noisy image is not given when using SGK, the noise level can only be guessed and entered into SGK for denoising. While using PCA + SGK to deal with noisy image, PCA is first used to estimate the standard deviation, and then the estimated value is entered into SGK for denoising, so the denoising performance is better. Quantitative comparisons with traditional SGK illustrate the benefits of PCA + SGK.

5. Conclusions

In this paper, the algorithm of PCA noise estimation combined with SGK dictionary learning was proposed to denoise image. The noisy image is first divided into patches, and the noise standard deviation is estimated by calculating the minimum eigenvalue of the image patch covariance matrix. After that, the estimated noise standard deviation is entered into SGK dictionary learning algorithm. The sparse representation of each training sample is obtained by sparse coding, and the dictionary atoms are updated by dictionary updating to denoise the image. This algorithm effectively solves the problem that the SGK algorithm requires a prior noise standard deviation for image denoising. This paper has the following three conclusions.

Firstly, the SGK dictionary learning algorithm is compared with K-SVD, DCT, Global, and BM3D algorithms. The PSNR of SGK algorithm and those of the other four algorithms do not have much difference, and the MSE of SGK algorithm is only higher than K-SVD algorithm. SGK algorithm owns great advantage in denoising time, which is much faster than K-SVD, DCT, and Global algorithms. Therefore, the SGK algorithm has the best denoising performance.



(a) Variation of noise estimation absolute

(b) Variation of noise estimation time

(c) Variation of PSNR

(d) Variation of MSE

Figure 6: Variation of denoising performance with noise standard deviation on five algorithms.

Figure 7: Twelve classic images.

Table 3: Denoising results of two algorithms for 12 images.

Image	Guessed noise value σ'	PSNR of SGK	PCA estimated value σ_{est}	PSNR of PCA + SGK
Barbara	12.15	31.5826	3.25	40.8875
Boat	20.02	31.0877	4.96	36.9938
Bridge	29.78	24.7522	8.67	32.1713
Cameraman	35.11	27.5935	10.22	33.7171
Couple	30.45	26.1943	12.30	32.2721
Fingerprint	20.10	28.0063	15.06	29.7811
Flintstones	4.98	23.4492	19.10	29.0998
Hill	5.23	22.5119	20.21	29.9451
House	10.00	22.8280	22.58	32.4558
Lena	15.15	23.7830	24.34	30.8314
Man	15.50	21.8376	28.04	28.3323
Peppers	18.24	22.4973	30.38	28.7696

(a) Noisy Cameraman image (b) SGK denoised image (c) PCA + SGK denoised image

Figure 8: Comparison of two denoising algorithms on the Cameraman image.

Secondly, the PCA algorithm is compared with the other four noise estimation algorithms: Kurtosis, Mode, Mad, and Min. The five algorithms are, respectively, combined with the SGK algorithm to denoise the additive Gaussian white noise images with different standard deviation. The absolute deviation of the noise estimated by PCA + SGK is the smallest, and it is better than the other four algorithms; that is, the noise standard deviation estimation of this algorithm is the most accurate. For the noise estimation time, Min + SGK, Med + SGK, and Mode + SGK all have a faster estimation and then it is PCA + SGK algorithm proposed in this paper and Kurtosis + SGK is the slowest. On the other hand, PCA + SGK and Kurtosis + SGK keep the high PSNR, followed by Med + SGK and Mode + SGK. The lowest PSNR is that of Min + SGK. At the same time, the MSE value of PCA + SGK is the lowest. So the denoising performance of proposed algorithm is better than the other four algorithms. It is found that the proposed algorithm is more accurate to estimate the noise standard deviation with faster denoising speed and good denoising performance.

Thirdly, PCA + SGK and SGK are, respectively, used to denoise the image with different standard deviation. Experiments show that PSNR of PCA + SGK is much higher than that of SGK. When using SGK for denoising, the noise standard is unclear, so the denoising performance is not good, while PCA + SGK firstly uses the PCA to estimate the noise standard deviation, which is close to the true value of the noise level, so the denoising performance is more ideal and the image's details are better preserved. While performance improvement is different for different images, the results nonetheless indicate the potential of proposed algorithm over original SGK algorithms.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Hebei Province (no. E2016202341) and Humanity and Social Science Foundation of Ministry of Education of China (15YJA630108).

References

[1] A. M. Scarfone, "Deformed Fourier transform," *Physica A: Statistical Mechanics and its Applications*, vol. 480, pp. 63–78, 2017.

[2] N. Yalcin, E. Celik, and A. Gokdogan, "Multiplicative Laplace transform and its applications," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 20, pp. 9984–9995, 2016.

[3] Y. Hel-Or and D. Shaked, "A discriminative approach for wavelet denoising," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 443–457, 2008.

[4] J.-L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.

[5] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: anisotropic multidirectional representation with separable filtering," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1916–1933, 2006.

[6] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.

[7] B. G. Bodmann, G. Kutyniok, and X. Zhuang, "Gabor shearlets," *Applied and Computational Harmonic Analysis*, vol. 38, no. 1, pp. 87–114, 2015.

[8] A. Ben Said, R. Hadjidj, K. Eddine Melkemi, and S. Foufou, "Multispectral image denoising with optimized vector non-local mean filter," *Digital Signal Processing*, vol. 58, pp. 115–126, 2016.

[9] X. Cong-Hua, C. Jin-Yi, and X. Wen-Bin, "Medical image denoising by generalised Gaussian mixture modelling with edge information," *IET Image Processing*, vol. 8, no. 8, pp. 464–476, 2014.

[10] S. K. Sahoo and A. Makur, "Dictionary training for sparse representation as generalization of K-means clustering," *IEEE Signal Processing Letters*, vol. 20, no. 6, pp. 587–590, 2013.

[11] S. Liu, L. Li, Y. Peng, G. Qiu, and T. Lei, "Improved sparse representation method for image classification," *IET Computer Vision*, vol. 11, no. 4, pp. 319–330, 2017.

[12] M. Wang, Z. Li, X. Duan, and W. Li, "An image denoising method with enhancement of the directional features based on wavelet and SVD transforms," *Mathematical Problems in Engineering*, vol. 2015, Article ID 469350, 9 pages, 2015.

[13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[15] B. Dumitrescu and P. Irofti, "Regularized K-SVD," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 309–313, 2017.

[16] P. Jiang and J.-Z. Zhang, "Fast and reliable noise level estimation based on local statistic," *Pattern Recognition Letters*, vol. 78, pp. 8–13, 2016.

[17] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, "Block-based noise estimation using adaptive gaussian filtering," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 218–226, 2005.

[18] N. N. Ponomarenko, V. V. Lukin, M. S. Zriakhov, A. Kaarna, and J. Astola, "An automatic approach to lossy compression of AVIRIS images," in *Proceedings of the 2007 IEEE International Geoscience and Remote Sensing Symposium, (IGARSS '07)*, pp. 472–475, Spain, June 2007.

[19] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, 2013.

[20] B. R. Corner, R. M. Narayanan, and S. E. Reichenbach, "Noise estimation in remote sensing imagery using data masking," *International Journal of Remote Sensing*, vol. 24, no. 4, pp. 689–702, 2003.

[21] P. Wyatt and H. Nakai, "Developing nonstationary noise estimation for application in edge and corner detection," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1840–1853, 2007.

[22] A. Barducci, D. Guzzi, P. Marcoionni, and I. Pippi, "Assessing noise amplitude in remotely sensed images using bit-plane and scatterplot approaches," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, pp. 2665–2675, 2007.

[23] W. Yao, "A note on EM algorithm for mixture models," *Statistics & Probability Letters*, vol. 83, no. 2, pp. 519–526, 2013.

[24] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2209–2216, October 2009.

[25] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 687–699, 2013.

[26] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Proceedings of the Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, vol. 6064:, pp. 354–365, USA, January 2006.

[27] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and C. Alberola-López, "Automatic noise estimation in images using local statistics. Additive and multiplicative cases," *Image and Vision Computing*, vol. 27, no. 6, pp. 756–770, 2009.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 2134395, 13 pages
https://doi.org/10.1155/2018/2134395

Research Article

A Novel Technique Based on Visual Words Fusion Analysis of Sparse Features for Effective Content-Based Image Retrieval

Muhammad Yousuf,[1] **Zahid Mehmood**,[1] **Hafiz Adnan Habib**,[2] **Toqeer Mahmood**,[2] **Tanzila Saba**,[3] **Amjad Rehman**,[4] **and Muhammad Rashid**[5]

[1]*Department of Software Engineering, University of Engineering and Technology, Taxila 47050, Pakistan*
[2]*Department of Computer Science, University of Engineering and Technology, Taxila 47050, Pakistan*
[3]*College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia*
[4]*College of Computer and Information Systems, Al-Yamamah University, Riyadh 11512, Saudi Arabia*
[5]*Department of Computer Engineering, Umm Al-Qura University, Makkah 21421, Saudi Arabia*

Correspondence should be addressed to Zahid Mehmood; zahid.mehmood@uettaxila.edu.pk

Received 16 July 2017; Accepted 4 February 2018; Published 6 March 2018

Academic Editor: Marco Perez-Cisneros

Content-based image retrieval (CBIR) is a mechanism that is used to retrieve similar images from an image collection. In this paper, an effective novel technique is introduced to improve the performance of CBIR on the basis of visual words fusion of scale-invariant feature transform (SIFT) and local intensity order pattern (LIOP) descriptors. SIFT performs better on scale changes and on invariant rotations. However, SIFT does not perform better in the case of low contrast and illumination changes within an image, while LIOP performs better in such circumstances. SIFT performs better even at large rotation and scale changes, while LIOP does not perform well in such circumstances. Moreover, SIFT features are invariant to slight distortion as compared to LIOP. The proposed technique is based on the visual words fusion of SIFT and LIOP descriptors which overcomes the aforementioned issues and significantly improves the performance of CBIR. The experimental results of the proposed technique are compared with another proposed novel features fusion technique based on SIFT-LIOP descriptors as well as with the state-of-the-art CBIR techniques. The qualitative and quantitative analysis carried out on three image collections, namely, Corel-A, Corel-B, and Caltech-256, demonstrate the robustness of the proposed technique based on visual words fusion as compared to features fusion and the state-of-the-art CBIR techniques.

1. Introduction

Image retrieval on the basis of image contents has been a vigorous area of research in the last three decades [1]. Many approaches have been introduced regarding image retrieval on the basis of image contents [2, 3]. A text-based image retrieval system has two issues. Firstly, the annotation task takes a longer time, which makes it unfeasible for huge databases. Secondly, assigning keywords for image annotation is subjective. These two drawbacks led to the development of a new system, which is CBIR [2]. CBIR aims to develop techniques which can be used for extracting similar images from image archives. Current CBIR methods are further categorized as global and local features [1, 4, 5]. Low-level features such as color, texture, shape, and spatial layout form the basis of CBIR [3, 6–10]. The main problem with CBIR is the issue of the semantic gap [3, 11] prevailing among high-level image concepts and low-level image features. The bag-of-visual-words (BoVW) model is a standard way to scramble local features into a vector of fixed length. It is one of the most widely used image feature representation methods [12]. The BoVW framework was suggested for the first time in the text retrieval domain for the analysis of text documents. It has subsequently been used in applications of computer vision [12–17]. In this model, feature vectors are quantized into visual words to formulate a dictionary or codebook. Visual words are formulated by clustering the local features [18].

Figure 1: Images of two different semantic categories with close visual appearance and semantic layout.

Human eyes discriminate images based on their visual contents. When we apply a feature extraction technique to the images that have a similar visual appearance, it may produce close feature vectors values that reduce the performance of the CBIR. The images shown in Figure 1 belong to two different semantic categories. These images are visually as well as semantically similar to each other. When a machine learning technique like support vector machine (SVM) classifies such type of images, it is possible that some images may be wrongly classified due to their similar semantic or visual appearance, which reduces the performance of the CBIR system.

SIFT performs better in the case of scale changes and on invariant rotations. However, SIFT does not perform better when there are low contrast and illumination changes within an image [19]. LIOP performs better in cases of low contrast and illumination changes within an image [20]. SIFT even performs better when there is large rotation and scale changes, while LIOP does not perform well in such cases [20].

In this article, we propose a novel technique based on visual words fusion as well as features fusion of the SIFT and LIOP feature descriptors based on the bag-of-visual-words (BoVW) methodology in order to deal with the aforementioned issues. For each image collection, the images are categorized into training and test sets, and SIFT and LIOP features are extracted separately from each image in the sets. After that, k-means clustering algorithm [21] is applied to the extracted features that represent image features in the form of clusters. Each cluster is specified as a visual word, and the combination of visual words constitutes a dictionary. For the proposed technique based on visual words fusion of SIFT and LIOP descriptors, clustering is applied individually to the extracted SIFT and LIOP features that have produced two dictionaries. After that, both dictionaries are fused or integrated together which results in the fusion of SIFT and LIOP visual words. For the proposed technique based on features fusion of SIFT and LIOP descriptors, both extracted features are fused together. Subsequently, clustering is applied to the fused features that constitute a single dictionary. These visual words are used to formulate a histogram from each image in the training set. Following this, these histograms are used to train the SVM classifier. At the end, images are retrieved from an image collection by applying the similarity measure technique based on the Euclidean distance between the query image and the images stored in an image collection.

The main contributions of this research article are as follows:

(1) A novel image representation in the form of the visual words fusion of SIFT and LIOP feature descriptors based on the BoVW methodology

(2) A novel image representation in the form of the features fusion of SIFT and LIOP feature descriptors based on the BoVW methodology

(3) Reduction of the semantic gap between low-level features of an image and high-level semantic concepts

The remaining sections of this article are organized as follows: the relevant state-of-the-art CBIR techniques are briefly described in Section 2 entitled as "Related Work." The detailed methodology of the proposed technique is discussed in Section 3 entitled as "Proposed Methodology." Section 4 presents the details of the experiments and performance analysis on three image collections. Section 5 concludes the proposed technique.

2. Related Work

CBIR has been an active research area for the last three decades due to its wide range of applications in image retrieval techniques [22]. The term "content-based" refers to the fact that the search technique evaluates the actual contents of an image rather than using traditional image annotation techniques for image retrieval. The term "content" in this framework refers to texture, color, shape, or any other information that can be derived from the image itself. There are various types of image retrieval techniques which are based on texture, shape, color, and spatial layout [23, 24]. Different interest points based detectors and descriptors have been proposed for feature extraction in image retrieval techniques [25–30].

Liu et al. [7] propose a novel descriptor known as microstructure descriptor (MSD). MSD is determined by

underlying colors and edge orientation which perfectly depicts the image features. To retrieve the images effectively, the method assimilates color, texture, shape, and spatial layout information. However, this approach is inadequate for global properties of the image and is unable to exploit relations among positions of dissimilar entities in the proposed design. Mansoori et al. [2] also propose a CBIR technique based on a SIFT descriptor, a hue descriptor, and soft assignment. The SIFT is used for extracting keypoints, while local patches around them are described by applying SIFT and hue descriptors. The distinct vocabulary is created for each descriptor which is then quantized by applying a k-means clustering algorithm. In this model, the soft assignment is used instead of a hard assignment in order to overcome the forfeiture in quantization that can reduce retrieval performance. The proposed technique reveals enhanced performance in comparison with other comparable CBIR techniques. Chang et al. [6] present a novel framework for content-based image retrieval by investigating the particle swarm optimization algorithm (PSOA). The proposed technique extracts three kinds of features from each image, namely, color, texture, and shape features, to find the similarities between the query image and images from the catalog. It employs appropriate distance measure for each kind of feature utilized. The PSOA is incorporated to elevate the proposed technique via finding out close prime combinations among features and their corresponding similarity measurements. Shen and Wu [4] develop an innovative method for CBIR by merging color, spatial, and texture features of the image. A feature vector is formed by utilizing all three of these features. The CENsus transform hISTogram (CENTRIST) feature is used for spatial structure and a principle component analysis (PCA) is applied on CENTRIST for dimension reduction. This algorithm incorporates diverse density (DD) and multiple instance learning (MIL) to achieve objective occurrences. This technique produces better results when compared to the state-of-the-art CBIR techniques. However, a few limitations of this method have been found, leading to the conclusion that more research is needed in some aspects. Talib et al. [5] introduce a framework for CBIR by constructing a weighted dominant color (DC) descriptor. In order to extract semantic features, the descriptor assigns weights to each DC in the image. This technique overcomes the shortcomings of dominant color descriptor (DCD) and diminishes the consequence of image background during the image matching decision. The technique tends to increase the performance. Pedronette et al. [31] exploit the reranking technique for retrieving images based on their visual contents. The proposed technique improves the effectiveness of CBIR. The reranking method does not entail distance information among complete ranked lists or images of a given collection. The proposed technique counts on the ranked list that was generated by efficient indexing structures and it is considered appropriate for large image collections as it scales up very well.

Zheng et al. [32] embed multiple binary features at the indexing level for large scale image retrieval. The multi-IDF scheme models correlation between features. The Hamming embedding method is used as a matching verification method. In order to lessen the effect of incorrect detection and boost the accuracy of visual matching, SIFT visual words are integrated with binary features. Karakasis et al. [33] propose a CBIR technique that uses an affine moment in order to describe the invariants lying in the local areas of the image for the sake of image retrieval. The produced moments are incorporated into the BoVW model in order to produce detailed feature vectors. A setup of three different design elements is used. Firstly, affine moments are computed. Secondly, invariants are calculated over the results of the real image. In the last phase, the process of normalization is executed in order to increase the range of invariants. The second phase intends to improve the first phase, while the third phase improves the results of the second phase. Rahimi and Moghaddam [34] introduce a CBIR technique based on intraclass and interclass features. Intraclass features are called the distribution of color tone, whereas singular value decomposition (SVD) and complex wavelet transform produce interclass features. A self-organizing map (SOM) is given by these features based on the artificial neural network (ANN) in order to improve the performance of the CBIR. Rashno et al. [35] introduce a novel CBIR technique in which feature extraction is done through wavelet transform and color feature selection. In this scheme, each image in the image collection is represented using a feature vector which is comprised of texture features from wavelet transform and color features from RGB and HSV domains. For texture features based on wavelet transform, images are decomposed into four subbands and then a low-frequency subband is used as texture features. For color features, DCD is used for the quantization of the image, while color statistics and histogram features are calculated. The ant colony optimization technique is used for selecting relevant and unique features from the entire feature set which contains both color and texture features. Mehmood et al. [36] present a CBIR technique that utilizes local and global histograms of visual words from the image. Both histograms contain the information regarding the semantics of an image. The global histogram is constructed by utilizing the visual information of the whole image, whereas the local histogram is constructed by extracting visual information from a local rectangular region of the image. The local histogram contains the spatial information of the salient objects within the image. The proposed technique has significantly improved the performance of the CBIR.

Zhao et al. [38] propose a CBIR technique which integrates three image descriptors for identifying visual contents of the image. These features are based on color, texture, and shape. The association in the distribution of color range in an image is taken by color distribution entropy. The color level cooccurrence algorithm makes use of the texture level matrix in order to seize the recurrence of textures as descriptors. The shape, rotation, and rescaling are done by the use of invariant moments. Euclidean distance is used to compute the similarity measure. de Ves et al. [39] put forward a subjective methodology in order to reduce the semantic gap while incorporating concerned users' interests and their relative responses. The main intention is to achieve the objective of reducing the semantic gap using the PCA and regression

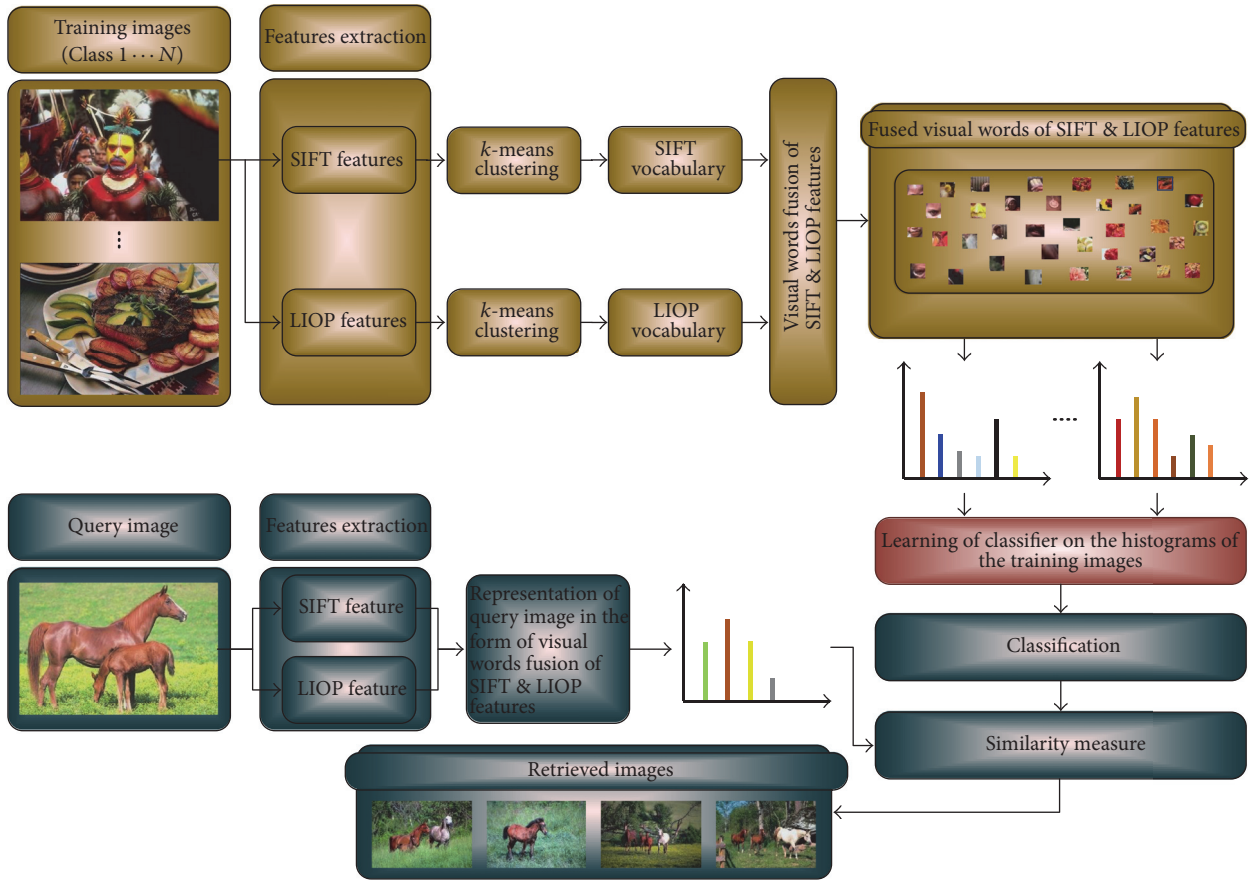


Figure 2: Block diagram of the proposed technique based on visual words fusion of SIFT and LIOP descriptors.

model. The former approach is responsible for rescaling the feature vectors, thereby reducing their dimensions, whereas the latter one is adjusted by the use of groups of nonoverlapped principal components. The local and dynamic nature of the proposed algorithm helps to achieve the intended results semantically. Xia et al. [40] present a CBIR technique to preserve the privacy of images in the cloud. While the cloud has solved the problem of low storage, at the same time, the privacy of users is highly concerned while outsourcing the images. The proposed technique exploits KNN in order to encode the visual features. These features are then utilized to compute the relevance, which in turn is utilized in the reranking procedure. In order to prevent the illegal copying and dissemination of retrieved images, the water-based protocol is exploited. Significant improvement has been observed in image search. The drawback of this technique lies in the lack of strength of the watermarking method.

3. Proposed Methodology

This section describes the detailed procedure of the proposed technique based on visual words fusion as well as features fusion of SIFT and LIOP descriptors based on the BoVW methodology for an effective CBIR. The block diagram of the proposed technique based on visual words fusion of SIFT and LIOP descriptors is shown in Figure 2.

The detailed procedure of the proposed technique is given as follows:

(1) For each image in the training and test sets, SIFT and LIOP features are computed.

(2) The SIFT features [48] are computed from each image over dense grid by applying the following mathematical equations:

$$h(t,i,j) = \left(k_i k_j * \overline{J}_t\right)\left(T + m\sigma\left[\frac{x_i}{y_i}\right]\right),$$
$$\overline{J}_t(x) = \omega_{\text{ang}} < \left(j(x) - \theta_t \left| j(x)\right|\right), \tag{1}$$

where σ is scale, θ is orientation, T is the center of the detected keypoint of the SIFT descriptor, m is descriptor magnification factor, J is gradient, h is the histogram of descriptors, ω_{ang} represents the angular velocity, and (x_i, y_j) represent the coordinate points of the (ith, jth) position. The kernels k_i and k_j are defined for a sample coordinate point (x, y) by the following mathematical equations:

$$k_i(x) = \frac{1}{\sqrt{2\pi}\sigma_{\text{win}}}\exp\left(\frac{-1(x - x_i)^2}{2\sigma_{\text{win}}^2}\right)\omega\left(\frac{x}{m\sigma}\right),$$
$$k_j(y) = \frac{1}{\sqrt{2\pi}\sigma_{\text{win}}}\exp\left(\frac{-1(y - y_i)^2}{2\sigma_{\text{win}}^2}\right)\omega\left(\frac{y}{m\sigma}\right), \tag{2}$$

where the side of the flat window is represented by σ_{win}.

(3) The LIOP features [20] are also computed from each image by applying the following mathematical equation:

$$\text{LIOP descriptor} = \left(\text{des}_1, \text{des}_2, \ldots, \text{des}_l\right),$$
$$\text{des}_l = \sum_{x \in \text{bin}_l} w(x)\,\text{LIOP}(x),$$
$$\text{where LIOP}(x) = \varphi\left(\gamma\left(P(x)\right)\right), \quad (3)$$
$$P(x) = \left(I(x_1), I(x_2), \ldots, I(x_n)\right) \in P^n,$$
$$w(x) = \sum_{i,j} \text{sgn}\left(\left|I(x_i) - I\left(x_j\right)\right| - T_{lp}\right) + 1.$$

In the above equation, for a sample point x_n, $I(x_n)$ represents the intensity of the nth neighboring sample, $P(x)$ is the N-dimensional feature vector of the intensities which represents the N neighboring sample points of a point x in the local patch, the mapping γ sorts the elements of the N-dimensional feature vector, preset threshold is represented by T_{lp}, sign function is represented by sgn, $w(x)$ represents the weighted function of the LIOP descriptor, the feature mapping function is represented by φ, and i, j represent the coordinate position of the nth sample point x_n.

(4) For the proposed technique based on visual words fusion of SIFT and LIOP descriptors, k-means [21] clustering technique is applied to the extracted features of SIFT and LIOP descriptors that produced two dictionaries. The resultant SIFT-based dictionary contains visual words of SIFT-based features, while LIOP-based dictionary contains visual words of LIOP-based features. Both dictionaries are fused together in order to perform visual words fusion of SIFT and LIOP features. The dictionary of each descriptor is formulated by applying the following mathematical equation on the extracted features of each descriptor:

$$R = \sum_{i=1}^{k} \sum_{x_l \in s_i} \left(x_l - u_i\right)^2, \quad (4)$$

where R represents the dictionary, u_i is the mean of all the points in the cluster s_i, and x_l represents the lth cluster or visual word.

After applying the clustering technique to extracted features of SIFT and LIOP descriptors, it produces two dictionaries that are represented by the following mathematical equations:

$$D_{\text{SIFT}} = \left\{v_{s1}, v_{s2}, v_{s3}, \ldots, v_{sn}\right\}$$
$$D_{\text{LIOP}} = \left\{v_{l1}, v_{l2}, v_{l3}, \ldots, v_{ln}\right\}, \quad (5)$$

where D_{SIFT} and D_{LIOP} are the resultant dictionaries that contain n visual words (i.e., $\{v_{s1}, v_{s2}, v_{s3}, \ldots, v_{sn}\}$ and $\{v_{l1}, v_{l2}, v_{l3}, \ldots, v_{ln}\}$) of SIFT and LIOP-based features, respectively.

After computing dictionaries for SIFT and LIOP feature descriptors, both dictionaries are concatenated which results in visual words fusion of both descriptors, represented mathematically as follows:

$$D_R = \left\{D_{\text{SIFT}}; D_{\text{LIOP}}\right\}, \quad (6)$$

where D_R is the resultant dictionary that contains SIFT and LIOP features in the form of fused visual words for more compact representation of image visual contents.

(5) For the proposed technique based on features fusion of SIFT and LIOP descriptors, SIFT and LIOP features are computed from each image, fused or integrated together, and at the end, k-means clustering technique [21] is applied to the fused features which produces a single dictionary.

The proposed technique based on visual words fusion of SIFT and LIOP descriptors results in better performance compared to the proposed technique based on features fusion of the SIFT and LIOP descriptors and the state-of-the-art CBIR techniques because the size of the dictionary representing visual contents of the images is twice as large compared to features fusion technique, which represents visual contents of the images by formulating a single dictionary.

(6) After applying the k-means [21] clustering technique, the visual contents of each image are now in the form of visual words. These visual words are used to build a histogram for each image.

(7) For image classification, the SVM classifier is selected along with Hellinger kernel [49] instead of the linear kernel. The learning of the SVM classifier is performed using histograms that are formulated from each image in the training set. The Hellinger kernel function is used with the SVM classifier because it explicitly computes the features map instead of computing the kernel values, while the classifier still remains linear. The mathematical representation of the Hellinger kernel function of the SVM on the normalized histograms is as follows:

$$K\left(n, n'\right) = \sum_i \sqrt{n(j)\,n'(j)}, \quad (7)$$

where n and n' represent the normalized histograms of each image.

(8) After training the proposed CBIR model, the testing of the proposed technique is performed by taking an image from the test set and applying the same aforementioned process to compute the histogram from the test image. The images are retrieved by measuring the similarity between the test image representation and training images stored in an image collection by applying the Euclidean distance formula.

4. Evaluation Metrics, Experimental Results, and Discussions

This section presents the performance measurements of the proposed technique. The performance is evaluated using precision, recall, and precision-recall (PR) curve parameters on Corel-A/1000 [50, 51], Corel-B/1500 [30], and Caltech-256 [52] image collections and the results are compared with the state-of-the-art CBIR techniques. All the results of the experiments are reported by performing each experiment 10 times. The dictionary size and features percentages per image are two important parameters that affect the performance of the proposed technique. Increasing the size of the dictionary at some certain level for compact representation of the visual contents of the images increases the performance

of the image retrieval, while larger sizes of the dictionary result in overfitting problem of CBIR. Similarly, in order to reduce the computational cost of the proposed technique that is slightly increased due to visual words fusion as well as the features fusion of SIFT and LIOP feature descriptors, performance analysis is carried out using different features percentages per image as reported in the subsequent sections.

The precision measures the specificity or accuracy while recall measures the sensitivity or robustness of the CBIR techniques. Both are mathematically represented by the following equations:

$$P = \frac{I_r}{I_t}, \qquad R = \frac{I_r}{I_s}, \tag{8}$$

where I_r represents the number of correctly retrieved images, I_t represents the total number of retrieved images, and I_s represents the total number of the images in a particular semantic category.

4.1. Analysis of the Evaluation Metrics on the Corel-A Image Collection. The Corel-A image collection is a subset of the WANG image collection. It contains 1000 images that are categorized into 10 semantic categories and the resolution of each image in this image collection is either 256 × 384 or 384 × 256. Each semantic category in this image collection contains 100 images. For a performance analysis of this image collection, images are divided into two sets known as training (70% images) and test (30% images) sets. The images in the training set are used to train the proposed model, while images in the test set are used to test the performance of the proposed model. In order to find the best performance of the proposed technique based on visual words fusion of SIFT and LIOP feature descriptors, different sizes of the dictionary (i.e., 20, 50, 100, 200, 400, 600, 800, 1000, and 1200) using different features percentages (i.e., 10%, 25%, 50%, 75%, and 100%) per image are formulated. The reason for selecting different features percentages per image is to reduce the computational cost that is slightly increased due to the visual words fusion as well as features fusions of SIFT and LIOP feature descriptors without affecting the performance of the proposed techniques.

The performance analysis in terms of the mean average precision (MAP) versus different sizes of the dictionary of the proposed technique based on features fusion of SIFT and LIOP descriptor that is compared with the MAP performance of the standalone SIFT and standalone LIOP techniques based on the BoVW methodology is presented in Figure 3. According to the experimental details shown in Figure 3, the best MAP performance of 82.90% is achieved on a dictionary size of 800 visual words using 75% feature per image. The proposed technique based on features fusion of SIFT and LIOP descriptors outperform in terms of the MAP performance as compared to the MAP performance of the standalone SIFT and standalone LIOP techniques on all the reported dictionary sizes.

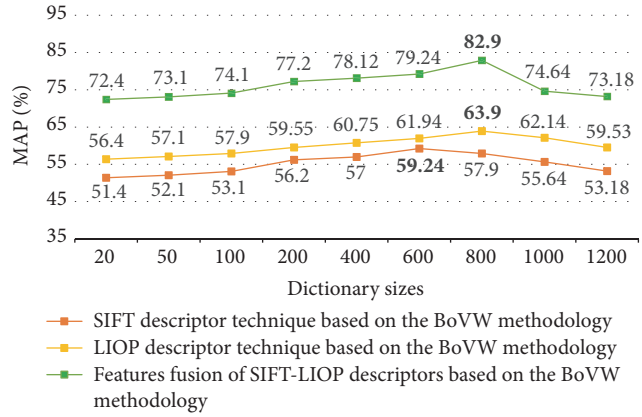


Figure 3: Performance comparison in terms of MAP performance between the proposed techniques based on features fusion, standalone SIFT, and standalone LIOP features on different sizes of the dictionary on the Corel-A image collection.

Table 1 presents the experimental details of the proposed technique based on visual words fusion of SIFT and LIOP descriptors on different reported sizes of the dictionary using different features percentages per image. The best MAP performance of 87.30% is achieved with a dictionary size of 800 visual words using 50% features per image. In order to verify the statistical significance of the experimental results of the proposed technique based on visual words fusion, the results of the statistical analysis are also reported in Table 1. The statistical results of the nonparametric Wilcoxon matched-pairs signed-rank test are also reported by comparing obtained MAP performance on dictionary size of 800 visual words with other reported dictionary sizes (20, 50, 100, 200, 400, 600, 800, 1000, and 1200) as well as with [36] using standard 95% confidence interval value. According to the statistical results of the nonparametric Wilcoxon matched-pairs signed-rank test, the proposed technique based on visual words fusion is statistically more effective because the value of P is less than the level of the significance (i.e., $\propto \leq 0.05$) for all the reported dictionary sizes.

In order to demonstrate the robustness of the proposed technique based on visual words fusion of SIFT and LIOP descriptors, its MAP performance is also compared with the MAP performance of the proposed technique based on features fusion as well as with the state-of-the-art CBIR techniques [36, 41–44], whose experimental details are shown in Figure 4 and Table 2. According to the experimental details, the proposed technique based on visual words fusion significantly outperforms in terms of the performance analysis as compared to its competitor CBIR techniques. The performance analysis in terms of the precision-recall (PR) curve as shown in Figure 5 is also carried with the state-of-the-art CBIR techniques [36, 37] which also demonstrate the robustness of the proposed technique based on visual words fusion of SIFT and LIOP descriptors on the Corel-A image collection.

The image retrieval results of the proposed technique based on visual words fusion of SIFT and LIOP descriptors for the semantic category "Beach" of the Corel-A image

Table 1: Statistical analysis and MAP performance of the proposed technique based on visual words fusion on different dictionary sizes and features percentages per image (bold values indicate the best performance).

Features percentages per image	Performance analysis in terms of the MAP performance (in %) on the different sizes of the dictionary								
	20	50	100	200	400	600	800	1000	1200
10%	74.30	74.60	76.10	78.50	79.30	80.70	84.50	76.60	75.30
25%	75.30	75.60	76.20	79.20	79.60	81.60	84.60	77.40	75.60
50%	75.60	76.00	76.50	80.60	81.70	82.30	**87.30**	77.50	76.30
75%	75.80	76.30	77.50	81.30	82.10	83.01	85.60	78.20	76.40
100%	76.00	77.60	79.10	81.70	82.30	83.60	85.70	78.50	77.30
MAP	75.40	76.10	77.10	80.20	81.00	82.24	**85.90**	77.64	76.18
Std. error	0.29	0.48	0.56	0.61	0.64	0.51	0.50	0.33	0.34
Std. deviation	0.66	1.09	1.25	1.36	1.43	1.14	1.12	0.74	0.77
Conf. interval	74.50–76.20	74.60–77.30	75.50–78.60	78.50–81.90	79.20–82.70	80.80–83.60	84.10–86.90	76.70–78.50	75.20–77.10
Statistical analysis using nonparametric Wilcoxon matched-pairs signed-rank test									
P value	0.043	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Z-value	2.023	2.03	2.02	2.02	2.02	2.03	2.02	2.02	2.02

Table 2: Performance analysis of the proposed technique based on visual words fusion on the Corel-A image collection which is reported using dictionary size of 800 visual words and features percentage of 50% per image (bold values indicate the best performance).

Semantic category	Proposed technique based on the visual words fusion	SIFT-LBP [41]	LGH-BoVW [36]	Color SIFT-EODH [42]	Poursistani et al. [43]	Yildizer et al. [44]
Africa	73.20	57.0	73.03	**74.60**	70.24	50.00
Beach	**75.00**	58.0	74.58	37.80	44.44	70.00
Buildings	**80.30**	43.0	80.24	53.90	70.80	20.00
Buses	95.50	93.0	95.84	**96.70**	76.30	80.00
Dinosaurs	**100**	98.0	97.95	99.00	**100**	90.60
Elephants	87.40	58.0	**87.64**	66.00	63.80	60.00
Flowers	98.30	83.0	85.13	92.00	92.40	**100.00**
Horses	**97.10**	68.0	86.29	87.00	94.70	80.00
Mountains	**83.80**	46.0	82.43	58.50	56.20	50.00
Food	**82.40**	53.0	78.96	62.20	74.50	20.00
MAP	**87.30**	65.7	84.21	72.77	74.34	62.00

collection and semantic categories "Sunset" and "Postcards" of the Corel-B image collection are shown in Figures 6, 9, and 10, respectively. The numeric value shown at the top of each image is the score of the respective image. The image shown at the top of each figure is the query image, while the rest of the images are the retrieved images that are obtained by applying the Euclidean distance formula between a score of the query image and scores of the retrieved images. The images whose numeric values are more close to the score of the query image are more identical to the query image which shows reduction of the semantic gap between low-level features of the image and high-level image semantic concepts and vice versa.

According to the experimental results shown in Table 2, the proposed techniques based on the visual words fusion of the SIFT and LIOP descriptors outperform in terms of the MAP performance as compared to the LGH-BoVW [36] technique as well as the state-of-the-art CBIR techniques [41–44] based on the BoVW methodology. For a dictionary size of N number of visual words, the proposed technique of this article represents visual contents of the images by assigning $2 \times N$ visual words due to the feature extraction from each image by applying two feature descriptors (i.e., SIFT and LIOP that formulate two dictionaries) as well as visual words of the resultant dictionary which contains the features of the SIFT and LIOP descriptors due to visual words fusion, while in case of the LGH-BoVW [36] technique, visual contents of the images are represented by assigning N number of visual words because single feature descriptor is applied on each image as well as visual words of the resultant dictionary which also contains the feature of the single descriptor.

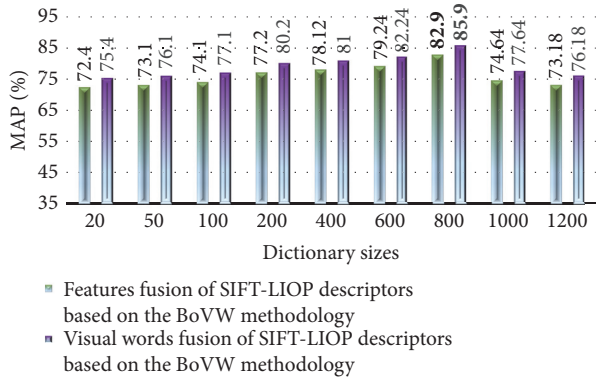


Figure 4: Performance comparison in terms of MAP performance between the proposed technique based on visual words fusion versus features fusion of SIFT and LIOP features techniques on different sizes of the dictionary on the Corel-A image collection.

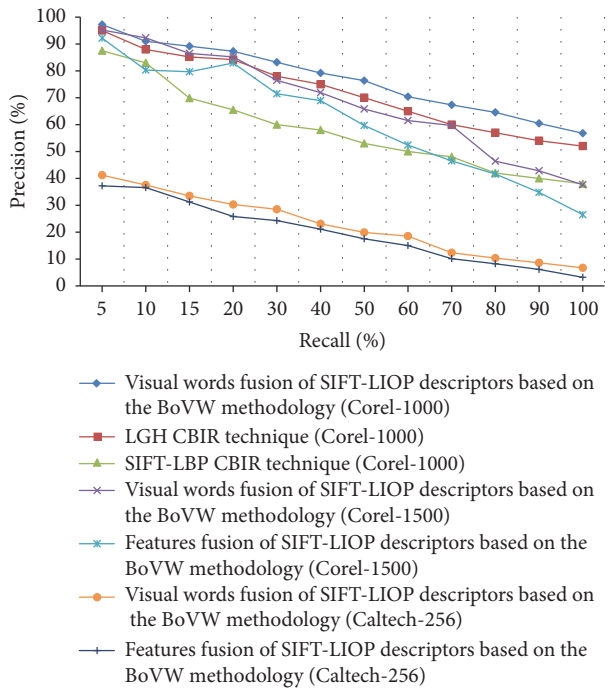


Figure 5: PR-curve comparison of the proposed technique based on visual words fusion versus features fusion of SIFT-LIOP descriptors as well as with the state-of-the-art CBIR techniques [36, 37] on the Corel-A, Corel-B, and Caltech-256 image collections.

4.2. Analysis of the Evaluation Metrics on the Corel-B Image Collection. The Corel-B image collection is a subset of the WANG image collection that contains images of different resolutions (i.e., 256 × 384, 384 × 256, 128 × 192, and 192 × 128). The total number of images in the Corel-B image collection is 1500; these are categorized into 15 semantic categories known as "Women," "Tigers," "Sunsets," "Postcards," "Caves," "Food," "Horses," "Mountains," "Flowers," "Elephants," "Dinosaurs," "Buses," "Buildings," and "Africa." The images are divided into two sets known as training (50% images) and test (50% images) sets for training and testing purposes. The performance analysis in terms of the MAP performance on different sizes of the dictionary is shown in Figures 7 and 8 and Table 3 for the proposed techniques based on visual words fusion, feature fusion, standalone SIFT, and standalone LIOP features based on the BoVW methodology. In the case of the proposed technique based on visual words fusion of SIFT and LIOP features, the best MAP performance of 85.20% is obtained with a dictionary size of 1000 visual words and using 50% features per image. The best MAP performance is achieved using the proposed technique based on features fusion of SIFT and LIOP features which is 82.96% with a dictionary size of 1000 visual words and using 75% features per image. According to the experimental details shown in Figures 7 and 8 and Table 3, the proposed technique based on visual words fusion outperforms as compared to the proposed technique based on features fusion, standalone SIFT, standalone LIOP, and the state-of-the-art CBIR techniques [3, 45] on a dictionary of all the reported sizes.

Table 3: Performance analysis of the proposed technique based on visual words fusion on the Corel-B image collection which is reported using dictionary size of 1000 visual words and features percentage of 50% per image (bold values indicate the best performance).

Performance measures	Proposed technique based on visual words fusion	GMM + mSpatiogram [45]	SQ + spatiogram [3]
MAP	**85.20**	74.10	63.95
Average recall	**17.00**	13.80	12.79

According to the experimental details shown in Figure 5 (experimental details provided earlier in Section 4.1), the performance measurement using PR-curve also demonstrates the robustness of the proposed technique based on visual words fusion that is compared with PR-curve of the proposed technique based on features fusion of SIFT and LIOP feature descriptors.

The results of image retrieval for the semantic categories "Sunset" and "Postcards" of the Corel-B image collection are shown in Figures 9 and 10.

4.3. Analysis of the Evaluation Metrics on the Caltech-256 Image Collection. We have also examined the performance analysis of the proposed technique on the Caltech-256 image collection [52]. The dimensions of each image in this collection are 300 × 200. There are 256 image semantic categories and each semantic category includes a minimum of 80 images. The total number of images in this collection is 30,607.

The performance analysis in terms of the MAP performance of the proposed technique based on features fusion, standalone SIFT, and standalone LIOP features techniques on different sizes of the dictionary is shown in Figure 11. According to the experimental details shown in Figure 11, the proposed technique based on features fusion of SIFT and LIOP descriptors performs better than the standalone SIFT and standalone LIOP features techniques based on the

Figure 6: Semantic category "Beach" of the Corel-A image collection shows a reduction of the semantic gap between retrieved images according to the query image.

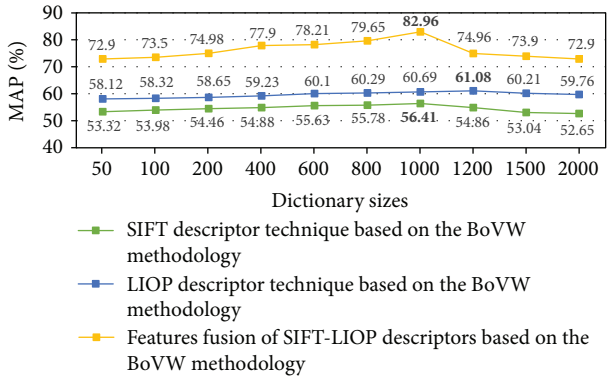


Figure 7: Performance comparison in terms of MAP performance between the proposed techniques based on features fusion, standalone SIFT, and standalone LIOP features on different sizes of the dictionary on the Corel-B image collection.

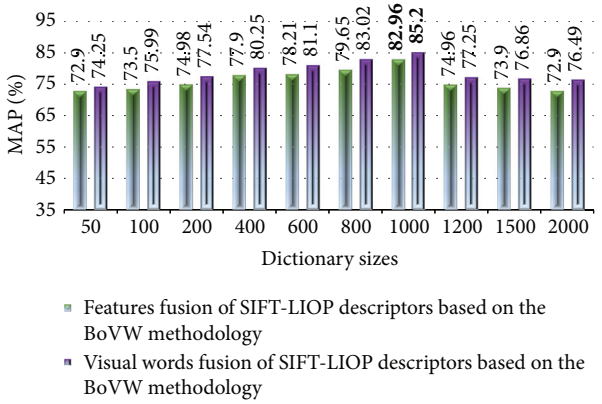


Figure 8: Performance comparison in terms of MAP performance between the proposed techniques based on visual words fusion versus features fusion of SIFT and LIOP features on different sizes of the dictionary on the Corel-B image collection.

BoVW methodology on a dictionary of all the reported sizes. According to the experimental details shown in Figure 12 and Table 4, the proposed technique based on visual words fusion of SIFT and LIOP descriptors outperforms in terms of MAP performance as compared to the features fusion technique and the state-of-the-art CBIR techniques [7, 46] on a dictionary of all the reported sizes. In the case of the proposed technique based on visual words fusion, the best MAP performance is achieved on a dictionary size of 1200 visual words that is 30.30%. The best MAP performance in case of features fusion technique is 25.82%, which is achieved on a dictionary size of 1500 visual words.

Figure 5 (experimental details provided earlier in Section 4.1) shows a comparison of performance analysis in terms of MAP performance using PR-curve between the proposed techniques based on visual words fusion versus

Table 4: Performance analysis of the proposed technique based on visual words fusion on the Caltech-256 image collection which is reported using dictionary size of 1200 visual words and features percentage of 75% per image.

Performance measures	Proposed technique based on visual words fusion	MN-ARM [7]	DCT [46]
MAP	30.30	28.21	23.91
Average recall	06.06	05.64	04.78

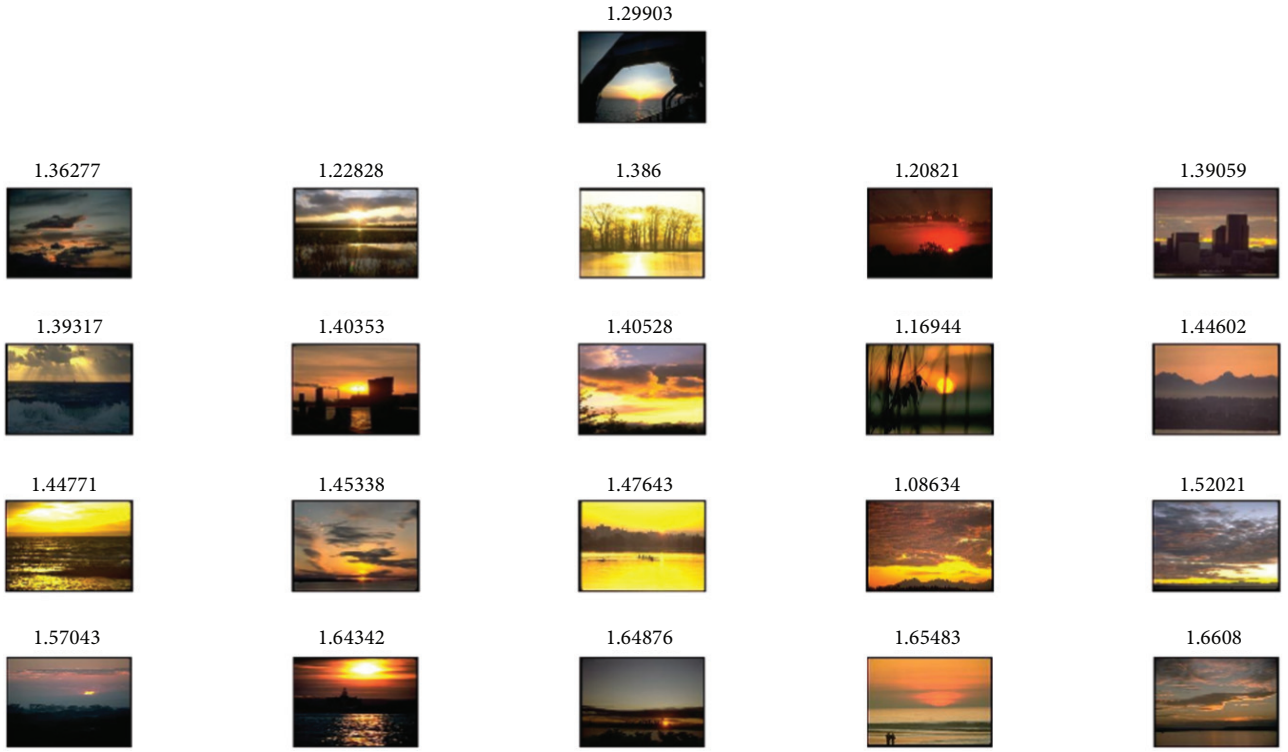


Figure 9: Semantic category "Sunset" of the Corel-B image collection shows a reduction of the semantic gap between retrieved images according to the query image.

Table 5: Performance analysis in terms of the computational complexity of complete framework.

Retrieved images	Proposed technique based on the visual words fusion of SIFT-LIOP	LGH technique [36]	FBWN technique [47]
Foremost-20	**0.7761**	0.7837	0.87

features fusion. According to the experimental details shown in Figure 5, the performance analysis using PR-curve also demonstrates the robustness of the proposed technique based on visual words fusion as compared to the proposed technique based on features fusion of SIFT and LIOP descriptors.

4.4. Performance Analysis in Terms of the Computational Complexity. All the experiments are performed on a Dell laptop with the following specifications: Intel (R) Pentium CPU B950 @ 2.10 GHz, 2.00 GB RAM, external SSD hard drive with a capacity of 120 GB, and Windows 7 64 bit operating system. The proposed technique is implemented in MATLAB R2015b and the dictionary is formulated offline by taking all the images of a training set. The performance is tested at runtime by taking a sample image from the test set using Corel-A image collection. The computational complexity (in seconds) of the complete framework from features computation to retrieved images is shown in Table 5 which is a proof of the robustness of the proposed technique in terms of the computational complexity as compared to the state-of-the-art CBIR techniques [36, 47].

5. Conclusions

The semantic gap between the low-level features of an image and high-level semantic concepts is an important issue that affects the performance of the CBIR. Increasing the size of the dictionary to represent visual contents of the images at some certain level increases the performance of the image retrieval, while larger sizes of dictionary tend to overfit. In this article, the proposed technique based on visual words fusion of SIFT and LIOP feature descriptors significantly improves the performance of the image retrieval by reducing the semantic gap issue of CBIR and assigning more visual words per image. The performance of the proposed technique based on visual words fusion is significantly improved as compared to the features fusion technique and the state-of-the-art CBIR techniques because the size of the dictionary to represent visual contents of the images is twice as large compared to the feature fusion technique. Additionally, the resultant dictionary contains features of the SIFT and LIOP descriptors in the form of visual words as compared to the state-of-the-art CBIR techniques. In order to reduce the computational cost of the proposed technique, which is slightly increased due to the fusion of SIFT and LIOP feature descriptors, different feature percentages per image are suggested without affecting the performance of the proposed technique.

Figure 10: Semantic category "Postcards" of the Corel-B image collection shows a reduction of the semantic gap between retrieved images according to the query image.

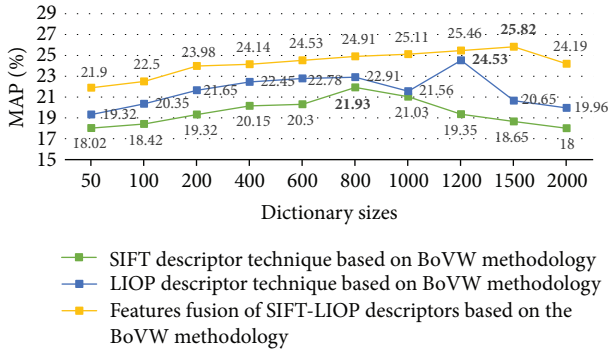


Figure 11: Performance comparison in terms of MAP performance between the proposed techniques based on features fusion, standalone SIFT, and standalone LIOP features on different sizes of the dictionary on the Caltech-256 image collection.

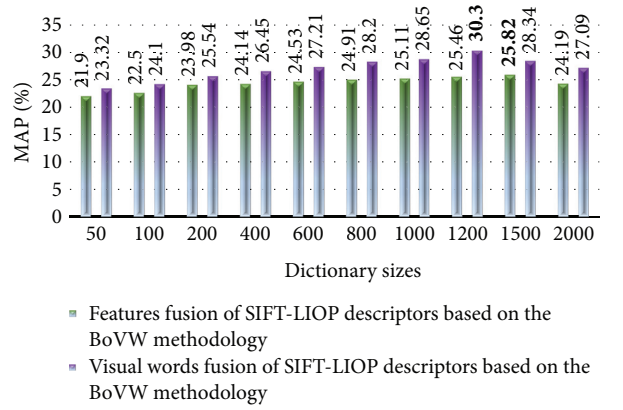


Figure 12: Performance comparison in terms of MAP performance between the proposed techniques based on visual words fusion versus features fusion of SIFT and LIOP features on different sizes of the dictionary on the Caltech-256 image collection.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All the authors contributed equally to this work.

Acknowledgments

This work was partially supported by the Machine Learning Research Group, Prince Sultan University Riyadh, Saudi Arabia [RG-CCIS-2017-06-02]. The authors are grateful for this financial support.

References

[1] N. Shrivastava and V. Tyagi, "Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching," *Information Sciences*, vol. 259, pp. 212–224, 2014.

[2] N. S. Mansoori, M. Nejati, P. Razzaghi, and S. Samavi, "Bag of visual words approach for image retrieval using color information," in *Proceedings of the 2013 21st Iranian Conference on Electrical Engineering, ICEE 2013*, Mashhad, Iran, May 2013.

[3] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188–198, 2013.

[4] G.-L. Shen and X.-J. Wu, "Content based image retrieval by combining color texture and CENTRIST," in *Proceedings of the Constantinides International Workshop on Signal Processing (CIWSP '13)*, pp. 1–4, January 2013.

[5] A. Talib, M. Mahmuddin, H. Husni, and L. E. George, "A weighted dominant color descriptor for content-based image retrieval," *Journal of Visual Communication and Image Representation*, vol. 24, no. 3, pp. 345–360, 2013.

[6] B.-M. Chang, H.-H. Tsai, and W.-L. Chou, "Using visual features to design a content-based image retrieval method optimized by particle swarm optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2372–2382, 2013.

[7] G.-H. Liu, Z.-Y. Li, L. Zhang, and Y. Xu, "Image retrieval based on micro-structure descriptor," *Pattern Recognition*, vol. 44, no. 9, pp. 2123–2133, 2011.

[8] M. E. Elalami, "A new matching strategy for content based image retrieval system," *Applied Soft Computing*, vol. 14, pp. 407–418, 2014.

[9] G. W. Jiji and P. J. Durairaj, "Content-based image retrieval techniques for the analysis of dermatological lesions using particle swarm optimization technique," *Applied Soft Computing*, vol. 30, pp. 650–662, 2015.

[10] X.-Y. Wang, Y.-J. Yu, and H.-Y. Yang, "An effective image retrieval scheme using color, texture and shape features," *Computer Standards & Interfaces*, vol. 33, no. 1, pp. 59–68, 2011.

[11] X.-Y. Wang, Y.-W. Li, H.-Y. Yang, and J.-W. Chen, "An image retrieval scheme with relevance feedback using feature reconstruction and SVM reclassification," *Neurocomputing*, vol. 127, pp. 214–230, 2014.

[12] C. Tsai, "Bag-of-words representation in image annotation: A Review," *ISRN Artificial Intelligence*, vol. 2012, pp. 1–19, 2012.

[13] Z. Mehmood, T. Mahmood, and M. A. Javid, "Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support vector machine," *Applied Intelligence*, vol. 48, no. 1, pp. 166–181, 2017.

[14] Z. Mehmood, S. M. Anwar, and M. Altaf, "A novel image retrieval based on rectangular spatial histograms of visual words," *Kuwait Journal of Science*, vol. 45, no. 1, pp. 54–69, 2018.

[15] N. Ali, K. B. Bajwa, R. Sablatnig, and Z. Mehmood, "Image retrieval by addition of spatial information based on histograms of triangular regions," *Computers & Electrical Engineering*, pp. 539–550, 2016.

[16] T. Mahmood, A. Irtaza, Z. Mehmood, and M. Tariq Mahmood, "Copy–move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images," *Forensic Science International*, vol. 279, pp. 8–21, 2017.

[17] T. Mahmood, Z. Mehmood, M. Shah, and Z. Khan, "An efficient forensic technique for exposing region duplication forgery in digital images," *Applied Intelligence*, pp. 1–11, 2017.

[18] Z. Liu, H. Li, L. Zhang, W. Zhou, and Q. Tian, "Cross-indexing of binary SIFT codes for large-scale image search," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2047–2057, 2014.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, IEEE, Kerkyra, Greece, September 1999.

[20] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 603–610, Barcelona, Spain, November 2011.

[21] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society*, vol. 28, no. 1, pp. 100–108, 1979, Series C (Applied Statistics).

[22] J.-M. Guo, H. Prasetyo, and J.-H. Chen, "Content-based image retrieval using error diffusion block truncation coding features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 466–481, 2015.

[23] T. Mahmood, T. Nawaz, R. Ashraf et al., "A survey on block based copy move image forgery detection techniques," in *Proceedings of the International Conference on Emerging Technologies (ICET '15)*, pp. 1–6, Peshawar, Pakistan, December 2015.

[24] T. Mahmood, T. Nawaz, Z. Mehmood, Z. Khan, M. Shah, and R. Ashraf, "Forensic analysis of copy-move forgery in digital images using the stationary wavelets," in *Proceedings of the 6th International Conference on Innovative Computing Technology, INTECH 2016*, pp. 578–583, Dublin, Ireland, August 2016.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[26] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, IEEE, San Diego, Calif, USA, June 2005.

[28] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: speeded up robust features," in *European conference on computer vision*, Lecture Notes in Computer Science, pp. 404–417, Springer, Berlin, Germany, 2006.

[29] D. J. Mankowitz and S. Ramamoorthy, "BRISK-based visual feature extraction for resource constrained robots," in *RoboCup 2013: Robot World Cup XVII*, S. Behnke, M. Veloso, A. Visser, and R. Xiong, Eds., vol. 8371 of *Lecture Notes in Computer Science*, pp. 195–206, Springer, Berlin, Germany, 2014.

[30] Z. Mehmood, F. Abbas, T. Mahmood, M. A. Javid, A. Rehman, and T. Nawaz, "Content-based image retrieval based on visual words fusion versus features fusion of local and global features," *Arabian Journal for Science and Engineering*, pp. 1–20, 2018.

[31] D. C. G. Pedronette, J. Almeida, and R. D. S. Torres, "A scalable re-ranking method for content-based image retrieval," *Information Sciences*, vol. 265, pp. 91–104, 2014.

[32] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3368–3380, 2014.

[33] E. G. Karakasis, A. Amanatiadis, A. Gasteratos, and S. A. Chatzichristofis, "Image moment invariants as local features for content based image retrieval using the Bag-of-Visual-Words model," *Pattern Recognition Letters*, vol. 55, pp. 22–27, 2015.

[34] M. Rahimi and M. E. Moghaddam, "A content-based image retrieval system based on color ton distribution descriptors," *Signal, Image and Video Processing*, pp. 691–704, 2015.

[35] A. Rashno, S. Sadri, and H. Sadeghiannejad, "An efficient content-based image retrieval with ant colony optimization feature selection schema based on wavelet and color features," in *Proceedings of the 2015 International Symposium on Artificial Intelligence and Signal Processing, AISP 2015*, pp. 59–64, Mashhad, Iran, March 2015.

[36] Z. Mehmood, S. M. Anwar, N. Ali, H. A. Habib, and M. Rashid, "A Novel image retrieval based on a combination of local and global histograms of visual words," *Mathematical Problems in Engineering*, vol. 2016, Article ID 8217250, 2016.

[37] X. Yuan, J. Z. Yu, Qin. Z., and T. Wan, "A SIFT-LBP image retrieval model based on bag of features," in *Proceedings of the IEEE International Conference on Image Processing*, IEEE, Brussels, Belgium, 2011.

[38] Z. Zhao, Q. Tian, H. Sun, X. Jin, and J. Guo, "Content Based Image Retrieval Scheme using Color, Texture and Shape Features," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 1, pp. 203–212, 2016.

[39] E. de Ves, X. Benavent, I. Coma, and G. Ayala, "A novel dynamic multi-model relevance feedback procedure for content-based image retrieval," *Neurocomputing*, vol. 208, pp. 99–107, 2016.

[40] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[41] J. Yu, Z. Qin, T. Wan, and X. Zhang, "Feature integration analysis of bag-of-features model for image retrieval," *Neurocomputing*, vol. 120, pp. 355–364, 2013.

[42] X. Tian, L. Jiao, X. Liu, and X. Zhang, "Feature integration of EODH and Color-SIFT: application to image retrieval based on codebook," *Signal Processing: Image Communication*, vol. 29, no. 4, pp. 530–545, 2014.

[43] P. Poursistani, H. Nezamabadi-pour, R. Askari Moghadam, and M. Saeed, "Image indexing and retrieval in JPEG compressed domain based on vector quantization," *Mathematical and Computer Modelling*, vol. 57, no. 5-6, pp. 1005–1017, 2013.

[44] E. Yildizer, A. M. Balci, M. Hassan, and R. Alhajj, " Efficient contentbased image retrieval using multiple support vector machines ensemble," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2385–2396, 2012.

[45] S. Zeng, R. Huang, H. Wang, and Z. Kang, "Image retrieval using spatiograms of colors quantized by gaussian mixture models," *Neurocomputing*, vol. 171, pp. 673–684, 2016.

[46] D. Zhong and I. Defée, "DCT histogram optimization for image database retrieval," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2272–2281, 2005.

[47] A. ElAdel, R. Ejbali, M. Zaied, and C. B. Amar, "A hybrid approach for content-based image retrieval based on Fast Beta Wavelet network and fuzzy decision support system," *Machine Vision and Applications*, vol. 27, no. 6, pp. 781–799, 2016.

[48] A. Vedaldi and B. Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *Proceedings of the International Conference on Multimedia (MM '10)*, pp. 1469–1472, October 2010.

[49] A. Vedaldi and A. Zisserman, "Sparse kernel approximations for efficient classification and detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2320–2327, June 2012.

[50] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[51] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003.

[52] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*, 2007.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 3742048, 16 pages
https://doi.org/10.1155/2018/3742048

Research Article

A k-Deviation Density Based Clustering Algorithm

Chen Jungan,[1,2] **Chen Jinyin**,[1] **Yang Dongyong**,[1] **and Li Jun**[2]

[1]*College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China*
[2]*College of Electronic Information, Zhejiang Wanli University, Ningbo 315100, China*

Correspondence should be addressed to Yang Dongyong; yangdy@zjut.edu.cn

Received 2 October 2017; Revised 29 December 2017; Accepted 17 January 2018; Published 26 February 2018

Academic Editor: Erik Cuevas

Due to the adoption of global parameters, DBSCAN fails to identify clusters with different and varied densities. To solve the problem, this paper extends DBSCAN by exploiting a new density definition and proposes a novel algorithm called k-deviation density based DBSCAN (kDDBSCAN). Various datasets containing clusters with arbitrary shapes and different or varied densities are used to demonstrate the performance and investigate the feasibility and practicality of kDDBSCAN. The results show that kDDBSCAN performs better than DBSCAN.

1. Introduction

DBSCAN is a classical density based clustering method [1] and has many desirable features including good robustness to noise and outliers. However, due to the adoption of global parameters, especially the introduction of neighborhood radius Eps, DBSCAN fails to identify clusters with different and varied densities. To solve this problem, two main methods have been proposed as follows.

(1) Adaptive Local Density or Eps. GRIDBSCAN [2] and GMDBSCAN [3] use the grid technique to calculate the local density (Eps, MinPts), where MinPts is defined as the minimum neighbors of a point when considering the point as the core point. APSCAN [4] uses the Affinity Propagation (AP) algorithm to partition a dataset into some patches and calculate the local density of each patch. VDBSCAN [5] uses a k-dist plot to select several Eps values for different densities. Multi-DBSCAN [6] uses the must-link constraint and k-nearest distance to calculate Eps values for different densities. DBSCAN-DLP [7] partitions a dataset into many subsets with different density levels by analyzing the statistical characteristics of its density variation and then estimates the Eps value for each subset. DSets-DBSCAN regards the data in the dominant set as core points and those from extrapolation as border ones, so Eps can be determined automatically based on the dominant set [8]. After the local density or Eps is estimated, all these algorithms apply DBSCAN to merge those data with similar density.

EDBSCAN [9] assigns varied values for Eps according to the local density based on the k-nearest neighbors, and the clustering process starts from the highest local density point towards the lowest local density one. DDSC [10] uses the Homogeneity test to detect the density difference between different regions; if their density difference is less than α, those regions will be merged into the same cluster.

(2) Redefinition of the Density with No Parameter Eps. H-density [11] estimates the local density of the nonnormalized probability distribution according to the neighborhood of radius R, and the hierarchical agglomerative strategy is used to merge clusters according to the di-similarity measures. In the multidensity DBSCAN, two adjacent spatial regions are separated into two clusters when the difference between DST and AVGDST violates a threshold, where DST is the average distance between one point and its k-nearest neighbors and AVGDST is the average distance between any point in one cluster and its k neighbors [12]. In K-DBSCAN [13], the K-means clustering algorithm is employed to divide all points into K-level groups based on their l-density values (here, l-density value is the average distance of the point P and its l-nearest neighbors), and then DBSCAN is used to merge similar data according to the density levels.

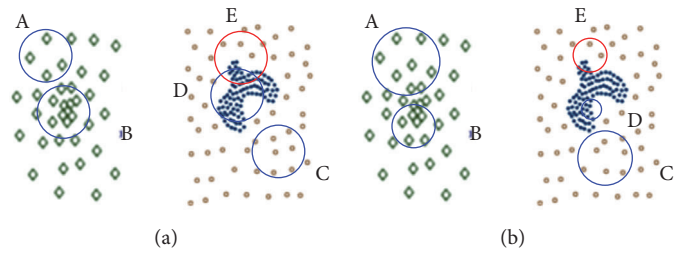


Figure 1: Dataset with varied densities.

Among these methods, Eps is automatically calculated according to different densities in the first method, and the definitions of different densities are proposed in the second method. Hereinto, one kind of definition is based on the k-nearest neighborhood method such as the local density of the nonnormalized probability distribution [11], the average distance between one point and its k-nearest neighbors [12], l-density [13], and k neighborhood density [7]. Based on these definitions, the varied densities can be represented separately by di-similarity measures [11], the difference between DST and AVGDST [12], and the density variation or density level [7].

It is known that the main objective of defining the density concept is to cluster the objects with similar density into the same cluster. For example, in Figure 1, the blue circles of A, B, C, and D can be viewed as one normal cluster, while the red circle E is an abnormal one. However, it is difficult to describe the density with Eps and MinPts because of different and varied densities. For example, in Figure 1(a), the fixed Eps have different densities in all these circle regions and the difference between circle E and another circle is not apparent.

Inspired by the above discussion, redefinition of the density based on the k-nearest neighbors seems a better solution to solve the problem of different and varied densities. Unfortunately, the density based on the average distance between one point and its k-nearest neighbors [7, 12, 13] is not good enough to describe the varied densities. For example, in Figure 1(b), if $k = 7$, the average distance is different in all these circle regions and circle E cannot be distinguished from other circles.

In this work, a new density definition called k-deviation density is proposed to describe the different and varied densities. The k-deviation density is defined as the proportion between the maximum distance and average distance. Given $k = 7$ in Figure 1(b), the densities of the circle regions of A, B, C, and D are similar, while the density of the circle region E is different from that of A, B, C, or D. Thus a k-deviation density based clustering algorithm (kDDBSCAN) can be put forward.

2. The kDDBSCAN Algorithm

The basic idea of this paper is that the objects in the same cluster have the similar and small k-deviation density which can reflect the deviation of an object from others. Based on the k-deviation density and DBSCAN, kDDBSCAN is proposed. In this section, some basic concepts or definitions are given, and the process of the proposed algorithm is described in detail.

The points used to calculate the k-deviation are sampled through the k-nearest neighborhood (KNN) method. According to [14], the shared nearest neighborhood (SNN) method can be used due to its robustness in high dimension dataset. However, SNN is not efficient because of its complexity. In view of this, the k neighborhood method is proposed to combine KNN and SNN.

When k points are sampled through the k neighborhood method, the k-deviation density can be calculated. The smaller k-deviation density means that it is more likely that these k samples are in the same cluster. Hence, a deviation factor is proposed as the given threshold to decide whether the given datasets are in the same cluster. If the k-deviation is greater than the deviation factor, the given datasets are not in the same cluster. This process is called directly density-reachable (DDR) and can be used to identify core points. Furthermore, density-reachable is proposed to decide whether two core points belong to the same cluster.

2.1. Basic Definitions

Definition 1 (k-nearest neighborhood). The k-nearest neighborhood (KNN) of a point x is denoted by $N_k(x)$.

Definition 2 (mutual k-nearest neighborhood). Given two points x_i and x_j, if $x_i \in N_k(x_j)$ and $x_j \in N_k(x_i)$, then x_i and x_j are mutual k-nearest neighbors. The mutual k-nearest neighborhood (mKNN) of a point x is denoted by $M_k(x)$.

Definition 3 (k neighborhood). The k neighborhood of a point x is denoted by $NM_k(x, n)$, where n is the minimal number of the mutual neighborhood:

$$\begin{aligned} NM_k(x,0) &= N_k(x) \quad n = 0 \\ NM_k(x,n) &= M_k(x) \bigwedge \left| M_k(x) \right| \geq n \quad n > 0. \end{aligned} \tag{1}$$

Definition 4 (k-deviation density). Let $x_i \in NM_k(x, n)$ and $d(x, x_i)$ be the distance from a point x to its ith nearest neighbor. Then, the k-deviation density is defined as

$$\mathrm{Dev}_k(x,n) = \frac{\max_{x_i \in NM_k(x,n)} \left(d(x,x_i)\right)}{\mathrm{avg}_{i \notin \mathrm{argmax}(d(x,x_i))} \left(d(x,x_i)\right)}. \tag{2}$$

Definition 5 (directly density-reachable). A point p is directly density-reachable from a point q if $p \in NM_k(q, n)$ and

Input: DataSt D, k, the minimal number of the mutual neighborhood n, the deviation factor α
Procedure kDDBSCAN(D, k, n, α)
(1) $C = 0$ /*C is cluster id */
(2) For each point Pi in D do
(3) If $Pi \cdot C\text{id}$ = UNCLASSIFIED Then
(4) Calculate the k-deviation density $\text{Dev}_k(Pi, n)$ /* see Definition 4 */
(5) If $\text{Dev}_k(Pi, n) \leq \alpha$ Then
(6) **ExpandCluster**(Pi, C)
(7) $C = C + 1$
(8) Else
(9) $Pi \cdot \text{Cid} = \text{OUTLINE}$
(10) End If
(11) End If
(12) End For
End Procedure

Algorithm 1: The implementation of kDDBSCAN.

Procedure ExpandCluster(P0, C)
(1) CorePoints = CorePoints $\cup$ $P0$
(2) For each point Pi in CorePoints do
(3) $Pi \cdot \text{Cid} = C$ /* Assign Pi to Cluster C */
(4) For each point Pj in $NM_k(Pi, n)$ do
(5) If $Pj \cdot \text{Cid}$ = UNCLASSIFIED $\|$ OUTLINE Then
(6) $Pj \cdot \text{Cid} = C$ /* Assign Pj to Cluster C */
(7) End If
(8) Calculate k-deviation density $\text{Dev}_k(Pj, n)$ /* see Definition 4 */
(9) If $\text{Dev}_k(Pj, n) \leq \alpha \&\& Pi$ and Pj is density-reachable Then /* see Definition 6 */
(10) CorePoints = CorePoints $\cup$ Pj
(11) Else
(12) $Pj \cdot \text{Cid} = \text{OUTLINE}$
(13) End If
(14) End For
(15) End For
End Procedure

Algorithm 2: The procedure of ExpandCluster.

$\text{Dev}_k(x) \leq \alpha$ (core point condition), where α is the deviation factor.

Definition 6 (density-reachable). A point p is density-reachable from a point q if the following conditions are satisfied:

(1) $\max(\max_{x_i \in NM_k(x,n)}(d(p, x_i))/\max_{x_i \in NM_k(x,n)}(d(q, x_i)), \max_{x_i \in NM_k(x,n)}(d(q, x_i))/\max_{x_i \in NM_k(x,n)}(d(p, x_i))) \leq \alpha$

(2) $\max(\text{avg}_{x_i \in NM_k(x,n)}(d(p, x_i))/\text{avg}_{x_i \in NM_k(x,n)}(d(q, x_i)), \text{avg}_{x_i \in NM_k(x,n)}(d(q, x_i))/\text{avg}_{x_i \in NM_k(x,n)}(d(p, x_i))) \leq \alpha$

(3) $\max(\max_{x_i \in NM_k(x,n)}(d(p, x_i))/\text{avg}_{x_i \in NM_k(x,n)}(d(q, x_i)), \max_{x_i \in NM_k(x,n)}(d(q, x_i))/\text{avg}_{x_i \in NM_k(x,n)}(d(p, x_i))) \leq \alpha$

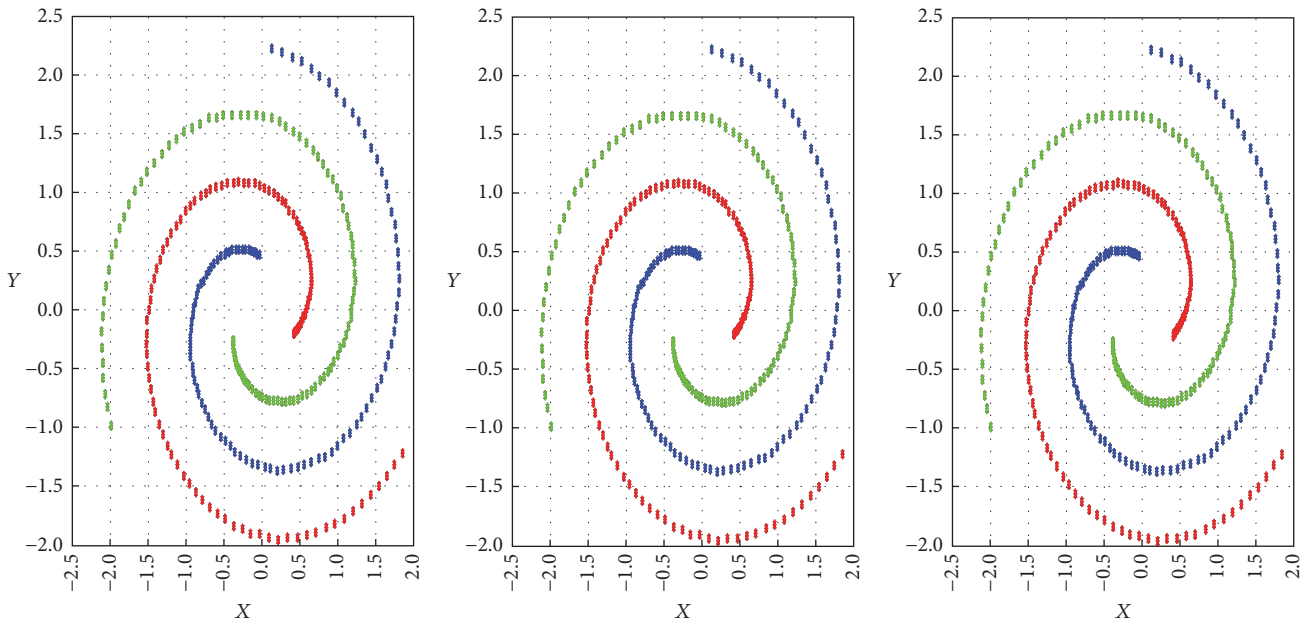
(4) $\max(d(p, q)/\max_{x_i \in NM_k(x,n)}(d(p, x_i)), \max_{x_i \in NM_k(x,n)}(d(p, x_i))/d(p, q)) \leq \alpha$

(5) $\max(d(p, q)/\max_{x_i \in NM_k(x,n)}(d(q, x_i)), \max_{x_i \in NM_k(x,n)}(d(q, x_i))/d(p, q)) \leq \alpha$.

2.2. The Process of kDDBSCAN. The procedure of ExpandCluster in Algorithm 1 is shown in Algorithm 2. The proposed kDDBSCAN, which is an extension of DBSCAN, is depicted as Algorithm 1.

3. Experiments

To thoroughly evaluate the effectiveness of kDDBSCAN, this section covers various types of datasets including clusters with arbitrary shape, uniform density, or varied densities. Since clusters in 2D datasets are easy to be visualized and compared by different algorithms, the performance comparison of kDDBSCAN with DBSCAN and the evaluation on parameter sensitivity are firstly conducted in the two-dimensional space. Then we demonstrate that the proposed algorithm is applicable to multidimensional datasets. UCI



(a) DBSCAN (4, 0.3) (b) kDDBSCAN (5, default, default) (c) kDDBSCAN (4, 2.8, 0)

Figure 2: Spiral dataset.

datasets with ground truth are used to quantify the performance according to the clustering results from the perspectives of Homogeneity, Completeness, V-measure, ARI, FMI, and NMI. Finally, BSDS500, Olivetti Face dataset, and MNIST are used to investigate the feasibility and practicality of kDDBSCAN. In kDDBSCAN, the default values are $n = 1$ and $\alpha = 999999$.

3.1. Clustering on the Two-Dimensional Dataset. Two-dimensional datasets are chosen according to their different characteristics. Spiral dataset represents the dataset with well-separated and nonspherical cluster. Aggregation and Flame datasets have adjacent regions with uniform density. D1 and Path based datasets represent the dataset containing embedded and adjacent clusters with different densities. Jain dataset contains sparse data regions with different densities. Compound dataset contains adjacent, embedded regions with varied and different densities.

3.1.1. Comparison with DBSCAN. Figures 2–8 show the results from DBSCAN and kDDBSCAN on those two-dimensional datasets, respectively, and the corresponding parameters for each algorithm on each kind of dataset are also given. Since the k neighborhood method is different for $n = 0$ and $n > 0$, the results when $n = 0$ are given separately. When the parameters n and α are set as default values, kDDBSCAN has only one parameter k.

(1) Spiral dataset contains well-separated and nonspherical cluster. Figure 2 shows that both DBSCAN and kDDBSCAN can obtain correct results.

(2) Aggregation and Flame datasets contain adjacent regions with uniform density. Figure 4 shows that the KNN method performs worse than DBSCAN or mKNN method.

(3) D1 and Path based datasets contain embedded and adjacent clusters with different densities.

In Figure 5, kDDBSCAN performs better than DBSCAN because the k-deviation density has considered the deviation factor of the dataset with different densities. Figure 6 shows that mKNN performs better than DBSCAN and KNN, indicating that mKNN is effective for adjacent clusters.

(4) Jain dataset contains sparse data regions with different densities.

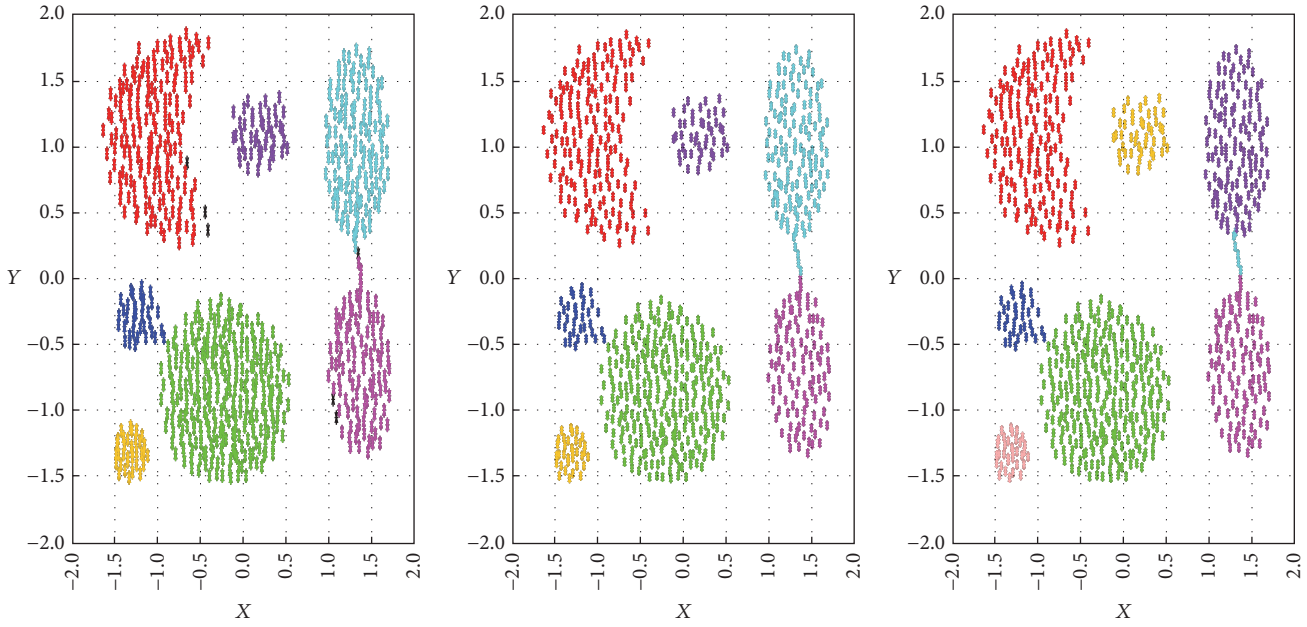
In Figure 7, DBSCAN cannot obtain correct results because of data sparsity, whereas kDDBSCAN performs well because the k-deviation density has considered the deviation factor of the dataset with sparse data.

(5) Compound dataset contains adjacent, embedded regions with varied and different densities. In Figure 8, DBSCAN takes the sparse data as noise while kDDBSCAN does not.

Through the above discussion, it can be seen that the k-deviation density can handle the dataset with varied densities or sparse data. The parameter α is effective in clustering the adjacent region according to the k-deviation density, as illustrated in Figures 3–5. The method of mKNN ($n = 1$) has the ability to cluster the well-separated data, as illustrated in Figures 2, 7, and 8.

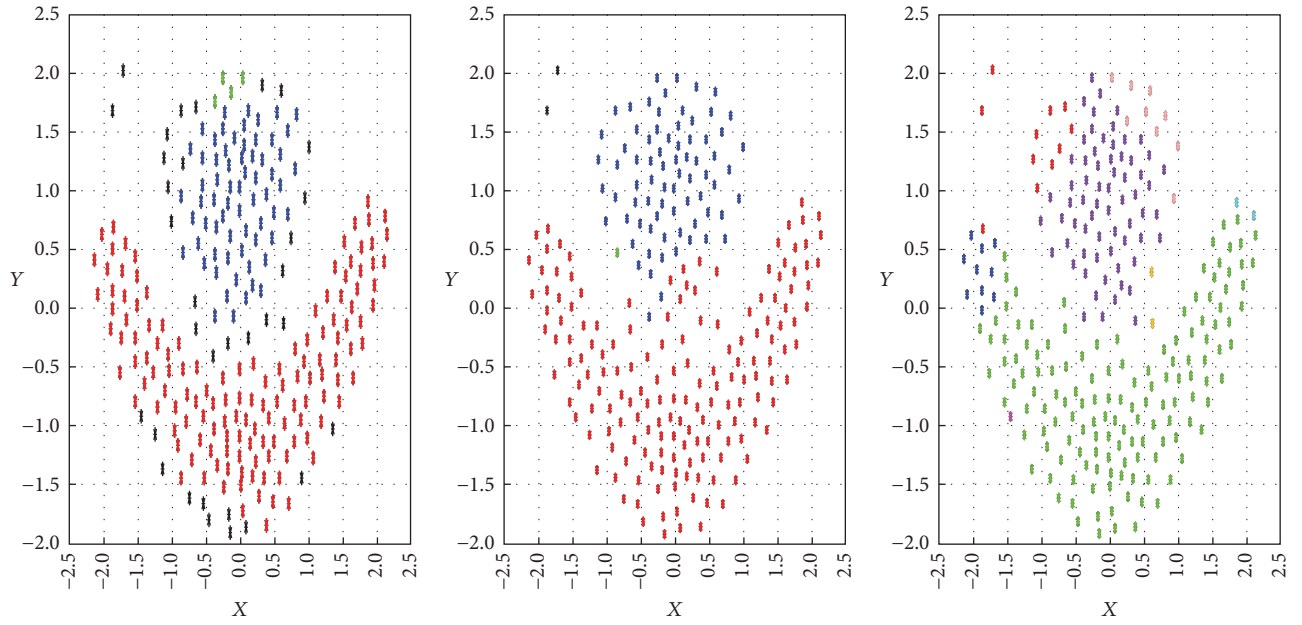
3.1.2. Parameter Sensitivity. In Figure 9, k is sensitive to different densities in sparse data regions as the k value decides the nearest neighborhood.

In Figure 10, n is sensitive to varied and different densities when adjacent regions exist.



(a) DBSCAN (5, 0.14) (b) kDDBSCAN (5, 2.2, default) (c) kDDBSCAN (5, 2, 0)

Figure 3: Aggregation dataset.

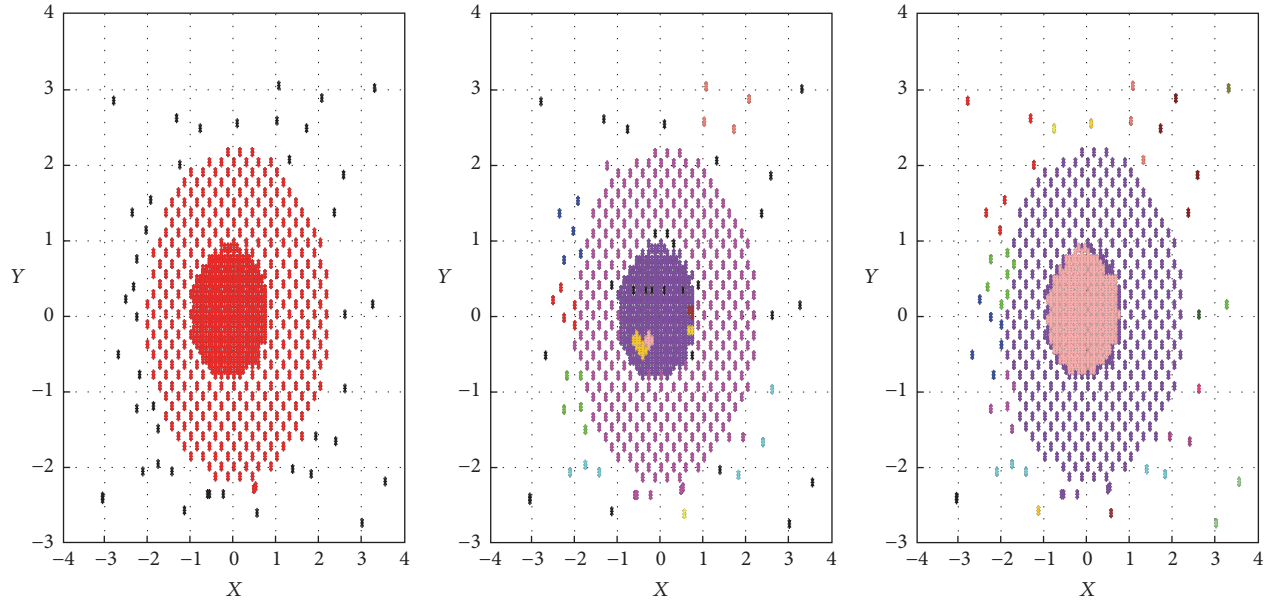


(a) DBSCAN (4, 0.25) (b) kDDBSCAN (11, 1.59, default) (c) kDDBSCAN (10, 1.58, 0)

Figure 4: Flame dataset.

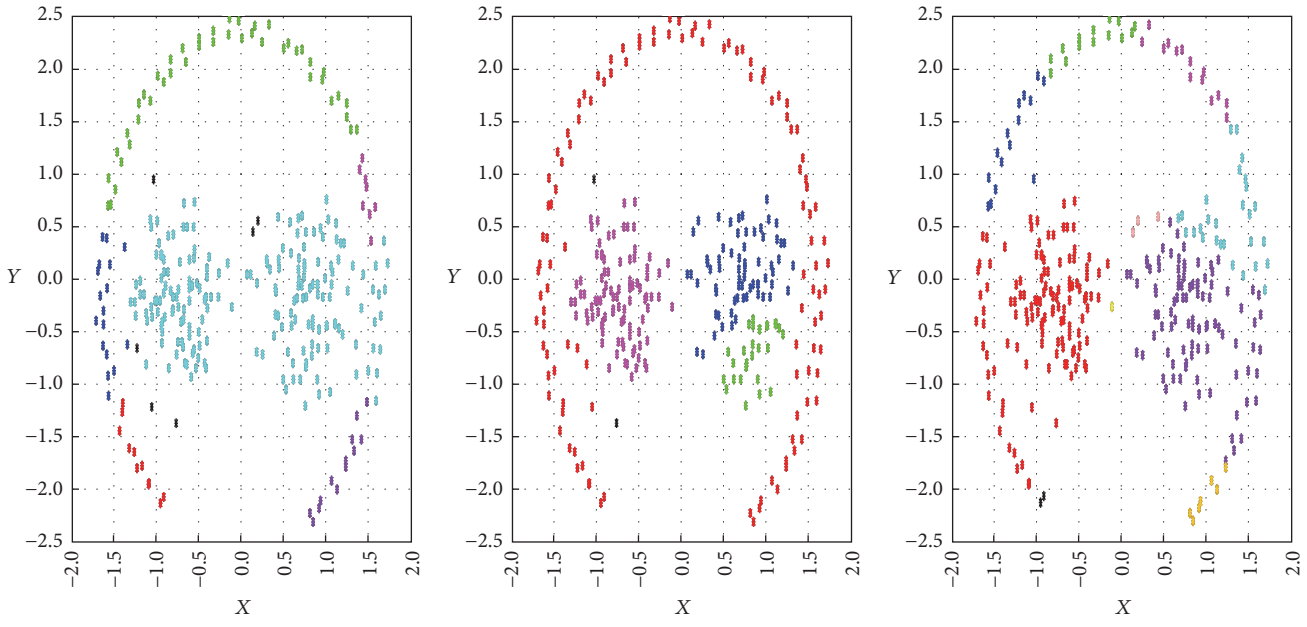
In Figure 11, as the parameter α is effective in clustering the adjacent regions according to the k-deviation density, it is sensitive when clustering adjacent regions. However, n is not sensitive to uniform density.

3.2. Clustering on Multidimensional Datasets. To demonstrate the applicability of kDDBSCAN to multidimensional dataset, UCI datasets with ground truth are used to evaluate its performance according to the clustering results from the perspectives of Homogeneity, Completeness, V-measure, ARI, FMI, and NMI. The input parameters are listed as Table 1, and the corresponding results are shown in Table 2. As can be seen, both algorithms can get the same results with Iris, but kDDBSCAN is more effective for other datasets.



(a) DBSCAN (4, 0.2) (b) kDDBSCAN (5, 1.6, default) (c) kDDBSCAN (5, 1.6, 0)

Figure 5: D1 dataset.

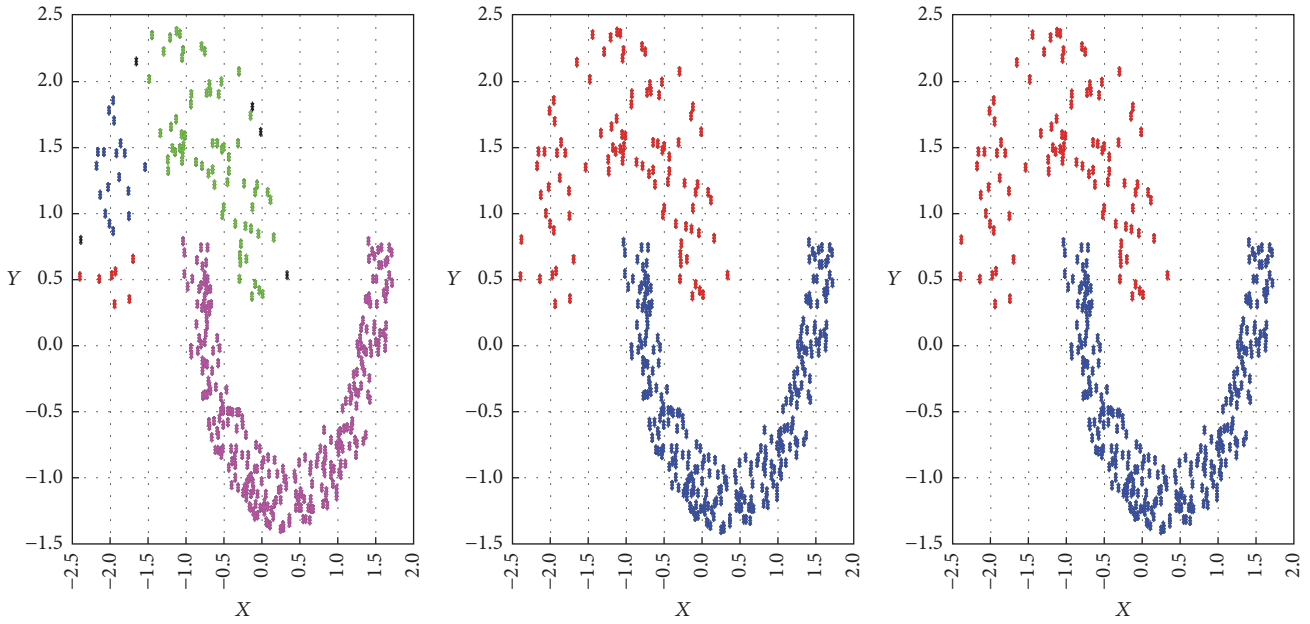


(a) DBSCAN (4, 0.28) (b) kDDBSCAN (9, default, 4) (c) kDDBSCAN (9, 1.8, 0)

Figure 6: Path based dataset.

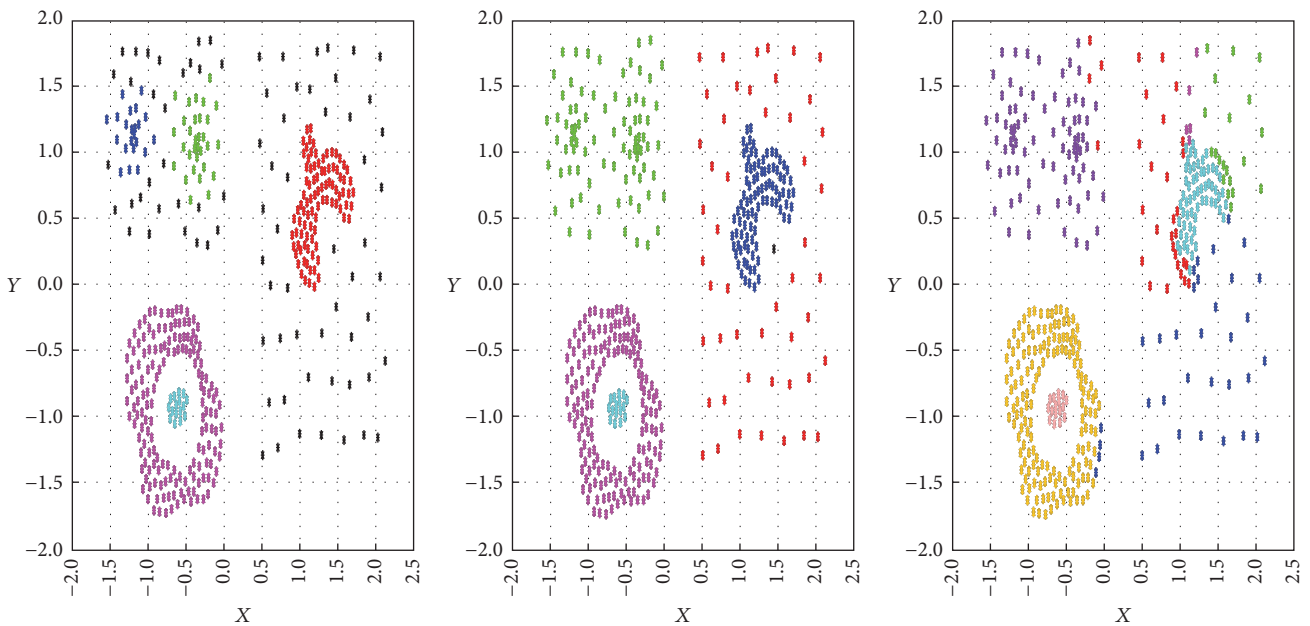
3.3. Application of kDDBSCAN to Image Segmentation. Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) consist of 500 natural images with ground-truth human annotations [15]. Among these images, three are used to demonstrate the effectiveness of kDDBSCAN relative to DBSCAN, and the results are shown in Table 3 and Figure 12. Here, the data point is composed of the pixel location and the corresponding RGB value. In addition, the parameter sensitivity of kDDBSCAN is analyzed in this section and the results are shown in Figures 13–18.

3.3.1. Comparison with DBSCAN. As shown in Figure 12, the segmentation and boundary detection is basically matched with the ground truth although some noise is required to



(a) DBSCAN (4, 0.3) (b) kDDBSCAN (8, default, default) (c) kDDBSCAN (8, 2.4, 0)

Figure 7: Jain dataset.

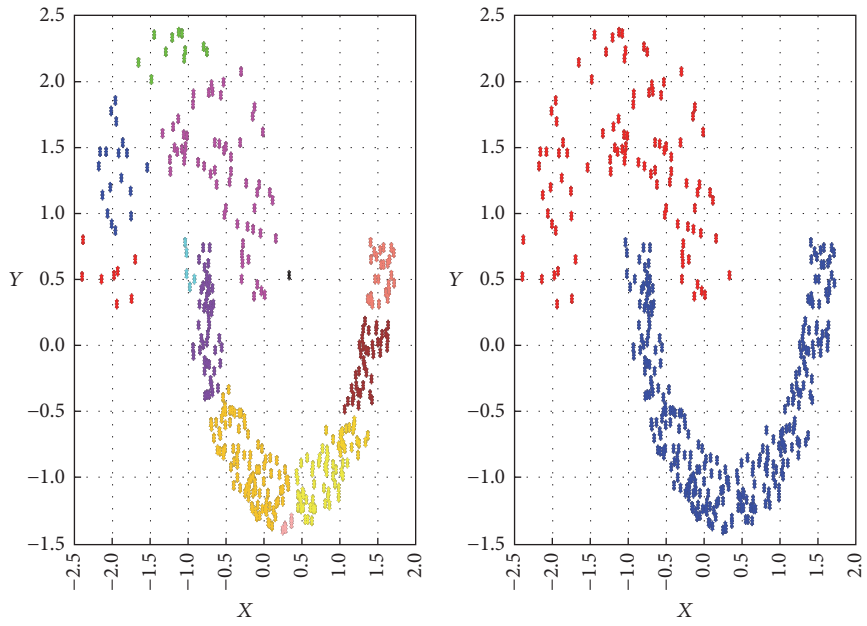


(a) DBSCAN (5, 0.2) (b) kDDBSCAN (8, default, default) (c) kDDBSCAN (8, 1.9, 0)

Figure 8: Compound dataset.

Table 1: The input parameters of different datasets for DBSCAN and kDDBSCAN.

	Iris	Glass	Wdbc	Segment
DBSCAN	(1.6, 15)	(1.2, 5)	(2.3, 5.0)	(2.0, 10)
kDDBSCAN	(20, default, default)	(12, default, default)	(7, default, default)	(8, default, default)



(a) kDDBSCAN (5, default, default) (b) kDDBSCAN (8, default, default)

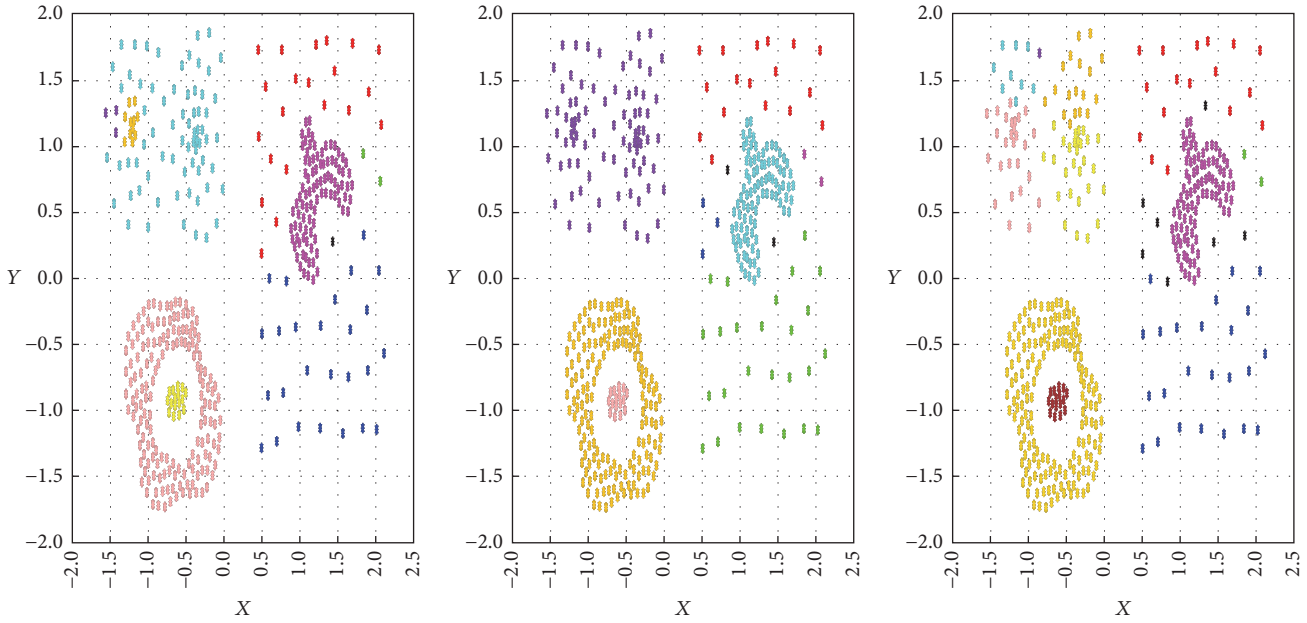
Figure 9: Jain dataset.

Table 2: The corresponding results from DBSCAN and kDDBSCAN.

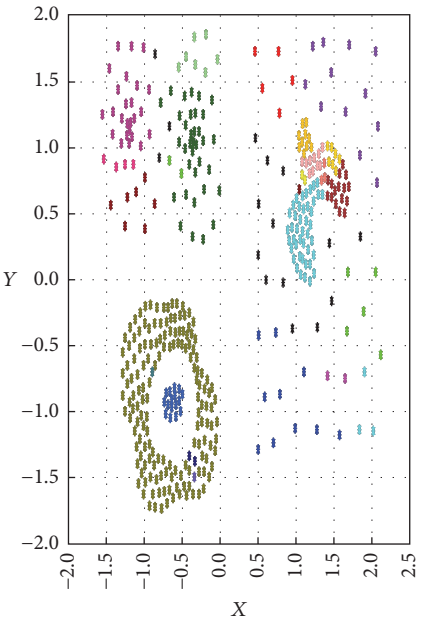
		Iris	Glass	Wdbc	Segment
Homogeneity	DBSCAN	0.579	0.306	0.305	0.416
	kDDBSCAN	0.579	0.361	0.385	0.451
Completeness	DBSCAN	1	0.502	0.220	0.582
	kDDBSCAN	1	0.540	0.336	0.589
V-measure	DBSCAN	0.733	0.380	0.256	0.485
	kDDBSCAN	0.733	0.433	0.359	0.511
ARI	DBSCAN	0.568	0.245	0.230	0.222
	kDDBSCAN	0.568	0.289	0.409	0.247
FMI	DBSCAN	0.771	0.527	0.607	0.430
	kDDBSCAN	0.771	0.558	0.743	0.445
NMI	DBSCAN	0.761	0.392	0.259	0.492
	kDDBSCAN	0.761	0.442	0.360	0.515

Table 3: The corresponding results with BSDS500 from DBSCAN and kDDBSCAN.

Index	Algorithm	Homogeneity	Completeness	V-measure	ARI	FMI	NMI
35070	kDDBSCAN	**0.775**	0.409	**0.536**	**0.598**	**0.787**	**0.563**
	DBSCAN	0.591	**0.431**	0.498	0.507	0.729	0.504
35008	kDDBSCAN	**0.749**	0.417	0.536	0.687	0.883	0.559
	DBSCAN	0.535	**0.597**	**0.564**	**0.707**	**0.915**	**0.565**
22090	kDDBSCAN	**0.738**	**0.626**	**0.677**	**0.765**	**0.808**	**0.680**
	DBSCAN	0.575	0.612	0.593	0.550	0.655	0.594



(a) kDDBSCAN (8, 1.7, default) (b) kDDBSCAN (8, 1.7, 2) (c) kDDBSCAN (8, 1.7, 3)



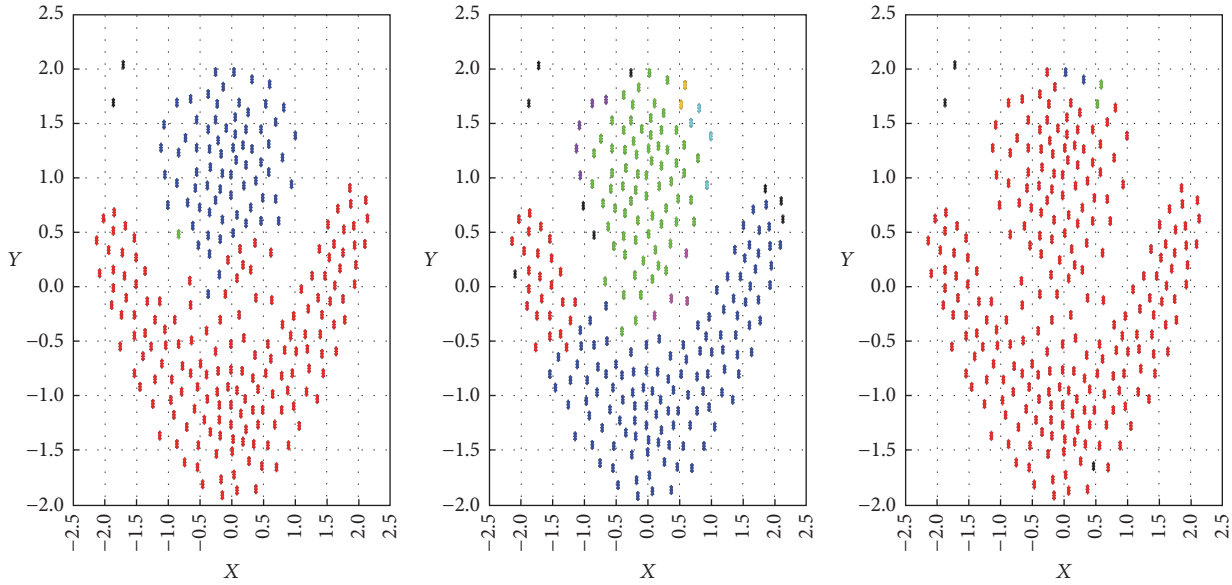
(d) kDDBSCAN (8, 1.7, 4)

Figure 10: Compound dataset.

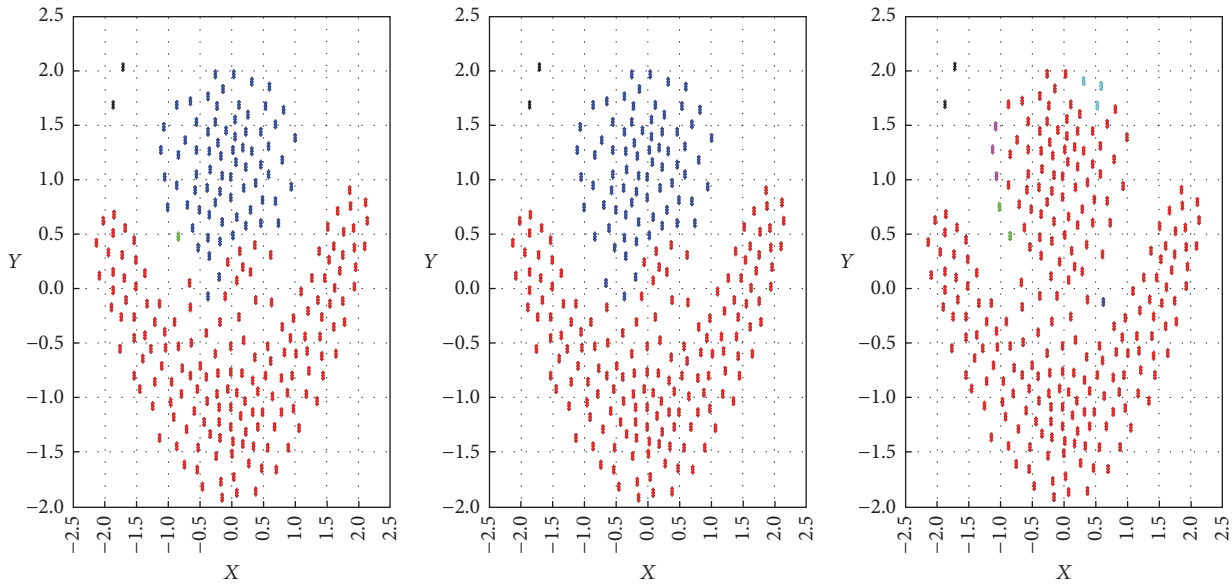
be further processed. In kDDBSCAN, more noise can be found. Nevertheless, kDDBSCAN can separate the flower of BSD35008 and the hill of BSD22090 more clearly than DBSCAN does. The results in Table 3 indicate that kDDBSCAN can achieve better cluster performance than DDBSCAN. In particular, the values of Homogeneity and FMI for kDDBSCAN are both above 0.7.

3.3.2. Parameter Sensitivity of kDDBSCAN. In this part, the visual image segmentation results by kDDBSCAN are discussed, and then the clustering results with different parameter values using kDDBSCAN are evaluated.

(1) Results of the Segmentation with Different Parameter Values. In Figure 13, with the increase of α value, the likelihood that similar pixels are clustered into the same cluster also increases, because higher α value means larger tolerance to the difference between pixels. Similarly, in Figure 14, with the increase of n value, similar pixels have less likelihood to be clustered into the same cluster. In general, higher α value and



(a) kDDBSCAN (11, 1.59, default) (b) kDDBSCAN (11, 1.5, default) (c) kDDBSCAN (11, 1.6, default)



(d) kDDBSCAN (11, 1.59, 2) (e) kDDBSCAN (11, 1.59, 3) (f) kDDBSCAN (11, 1.59, 4)

Figure 11: Flame dataset.

lower n value lead to larger likelihood for similar pixels to be clustered into the same cluster.

As the parameter k decides the number of sampled points, the deviation density of every core point does not vary monotonically when k increases. Consequently, there must be multiple optimum k values to achieve better performance when α and n are constant. This is in accordance with the different results of the last images with different k values in Figure 15.

(2) Clustering Evaluation. Figures 16–18 show the results with different parameter values using kDDBSCAN for different evaluation indexes. Based on the analysis of Figures 13 and 14, higher α value and lower n value can lead to larger likelihood for similar pixels to be clustered into the same cluster. As can be seen from Figures 16 and 17, most evaluation indexes vary monotonically with α and n. By contrast, in Figure 18, the Homogeneity value does not vary monotonically with k, so there is an optimum value k required to be set manually.

3.4. Application of kDDBSCAN to Olivetti Face Dataset and MNIST. kDDBSCAN and DBSCAN are applied to the Face dataset to group the images for the same person to a cluster without any previous training. Olivetti Face dataset is a widespread benchmark for machine learning algorithms. There are ten different images for each of 40 distinct persons.

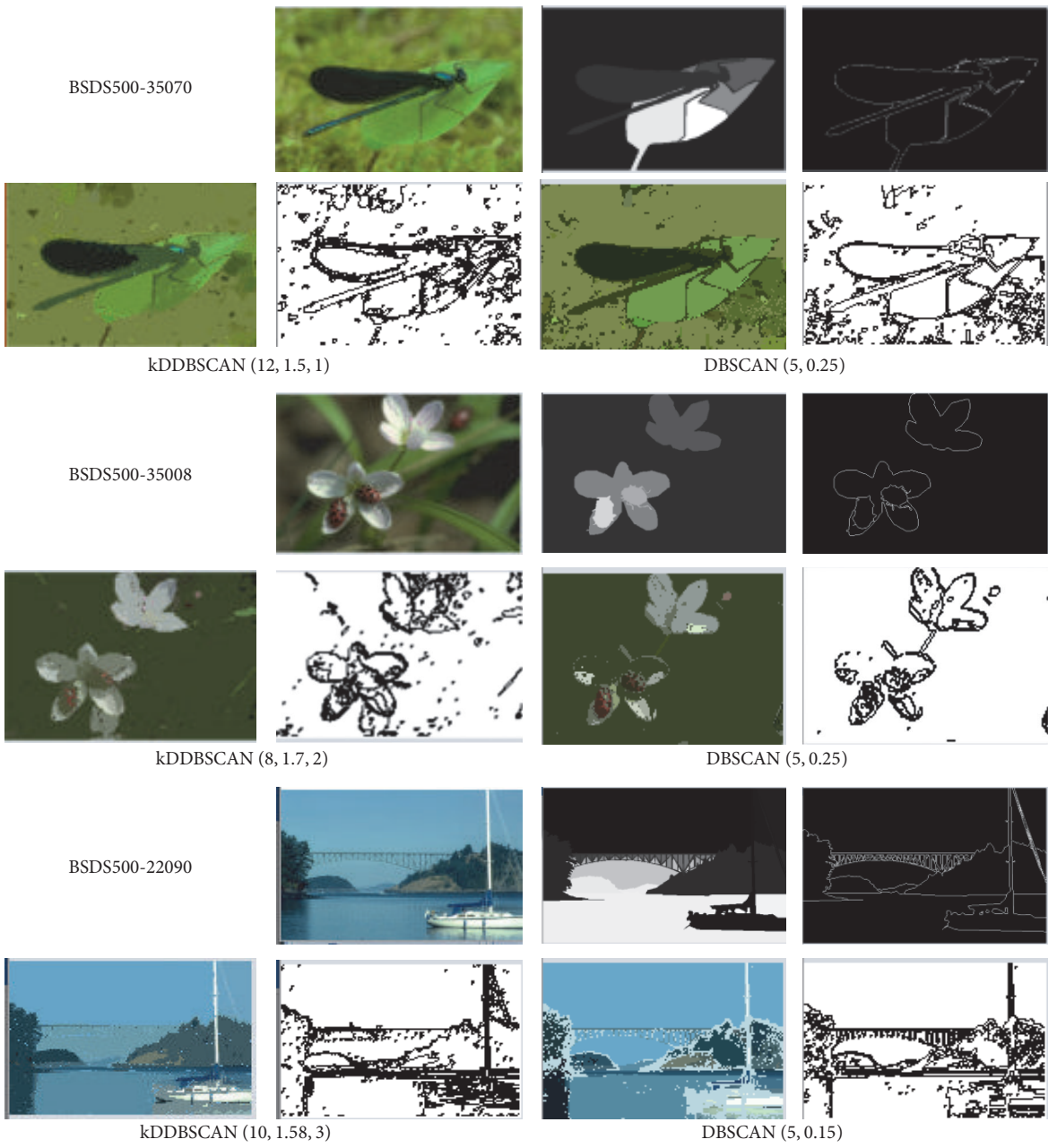


Figure 12: Image segmentation results with BSDS500 from kDDBSCAN and DBSCAN (the images in the first row are the original images and ground-truth images, and other images are the segmentation and boundary detection results).

Figure 13: Results of the segmentation with different α values ($k = 12, n = 1$).

Figure 14: Results of the segmentation with different n values ($k = 10, \alpha = 1.5$).

Figure 15: Results of the segmentation with different k values ($n = 1, \alpha = 1.5$).

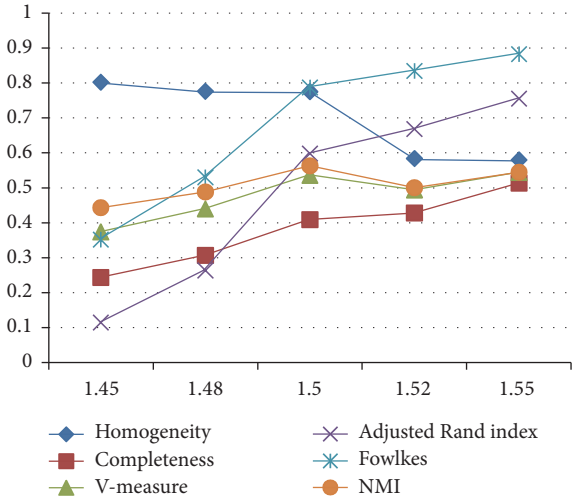


Figure 16: Results with different α values using kDDBSCAN ($k = 12, n = 1$).

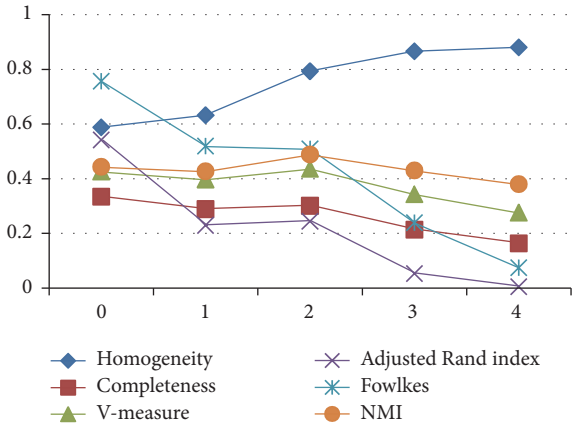


Figure 17: Results with different n values using kDDBSCAN ($k = 10, \alpha = 1.5$).

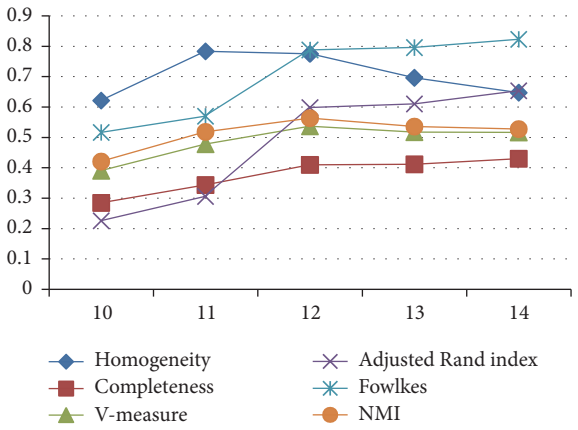


Figure 18: Results with different k values using kDDBSCAN ($n = 1, \alpha = 1.5$).

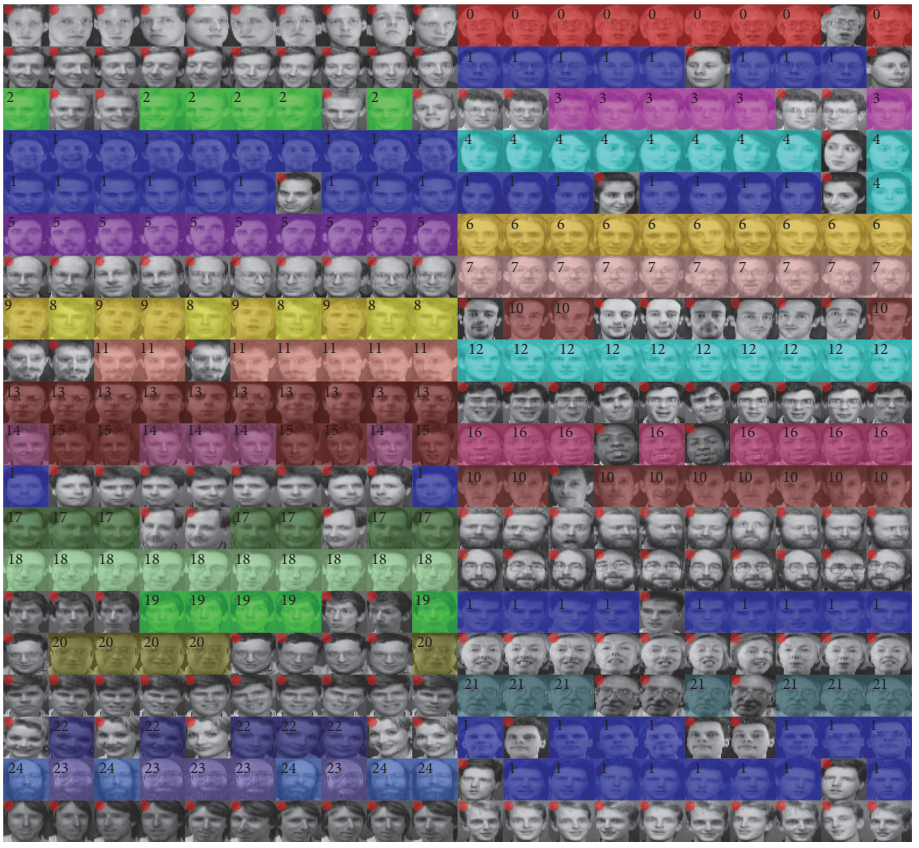


Figure 19: DBSCAN (4, 0.13) with Olivetti Face dataset.

Similar to [16], the similarity between two images is calculated according to the method in [17]. Figures 19 and 20 show the clustering results, where the images of the same color correspond to the same cluster and the images with red point are taken as noise. It can be seen that DBSCAN identifies six persons as the same person "1," whereas kDDBSCAN only identifies two persons as the same person "4." The values of those clustering evaluation indexes such as Homogeneity, Completeness, V-measure, ARI, FMI, and NMI are given in Table 4. It is clear that kDDBSCAN performs better than DBSCAN.

The MNIST database of handwritten digits has a training set of 60,000 examples and a testing set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image [18]. 500 examples of every digit are randomly chosen and put into kDDBSCAN. Table 5 provides the results of clustering evaluation, and Figure 21 displays the visual clustering results. As can be seen from Table 5, the maximum

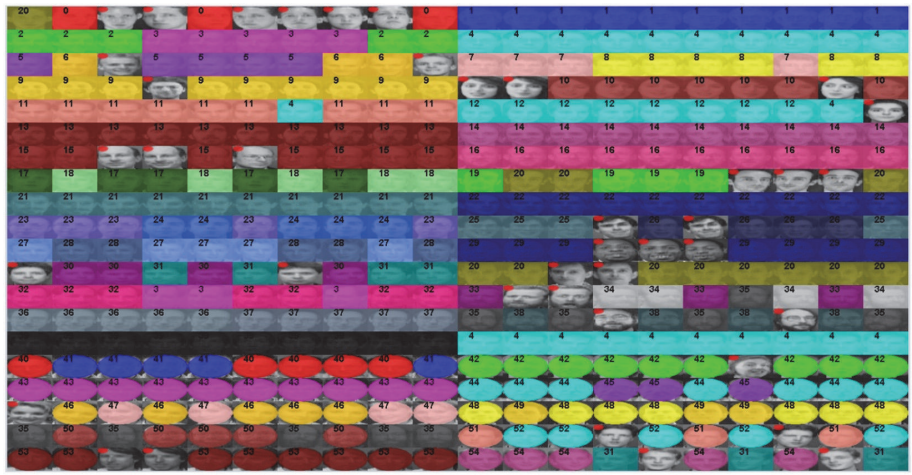


Figure 20: kDDBSCAN (4, default, default) with Olivetti Face dataset.

Table 4: The corresponding results of clustering evaluation with Olivetti Face dataset.

Index	Homogeneity	Completeness	V-measure	ARI	FMI	NMI
kDDBSCAN (4, default, default)	0.818	0.787	0.802	0.434	0.449	0.802
DBSCAN (4, 0.13)	0.502	0.775	0.609	0.095	0.216	0.624

Table 5: The corresponding results of clustering evaluation with MNIST dataset.

K	n	Alpha	Homogeneity	Completeness	V-measure	ARI	FMI	NMI
10	3	1.15	0.720	0.374	0.492	0.096	0.173	0.519
		1.17	0.700	0.397	0.506	0.147	0.22	0.527
		1.2	0.608	0.45	0.517	0.261	0.337	0.523
12	4	1.15	0.719	0.375	0.493	0.087	0.165	0.519
		1.17	0.698	0.405	0.513	0.153	0.228	0.532
		1.2	0.649	0.476	0.549	0.294	0.366	0.556
15	4	1.15	0.582	0.469	0.519	0.281	0.376	0.522
		1.17	0.561	0.532	0.546	0.323	0.427	0.546
		1.2	0.288	0.471	0.358	0.094	0.317	0.368

Homogeneity can reach 0.72, whereas the maximum ARI is only 0.323. The higher Homogeneity value in Table 5 means the larger maximum number of the clusters in the top-left of Figure 21, while the higher ARI value means that more similar examples can be clustered into the same cluster. Obviously, these two clustering indexes are contradictory. Hence, which clustering index is better depends on the actual application circumstance. From Figure 21, it can be concluded that the digits (4, 7, 9) and (5, 8) are easy to be grouped into the same cluster.

4. Conclusions

In summary, the basic idea of this paper is that the objects in the same cluster have the similar and small k-deviation density which can reflect the deviation of an object from others. On this basis, kDDBSCAN is proposed based on the k-deviation density and DBSCAN to identify clusters with different and varied densities, and various datasets containing clusters with arbitrary shapes, uniform density, and different or varied densities are used to demonstrate the performance of kDDBSCAN. The results show that kDDBSCAN can achieve better results than DBSCAN.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61502423), Zhejiang Provincial Natural Science Foundation (nos. Y14F020092,

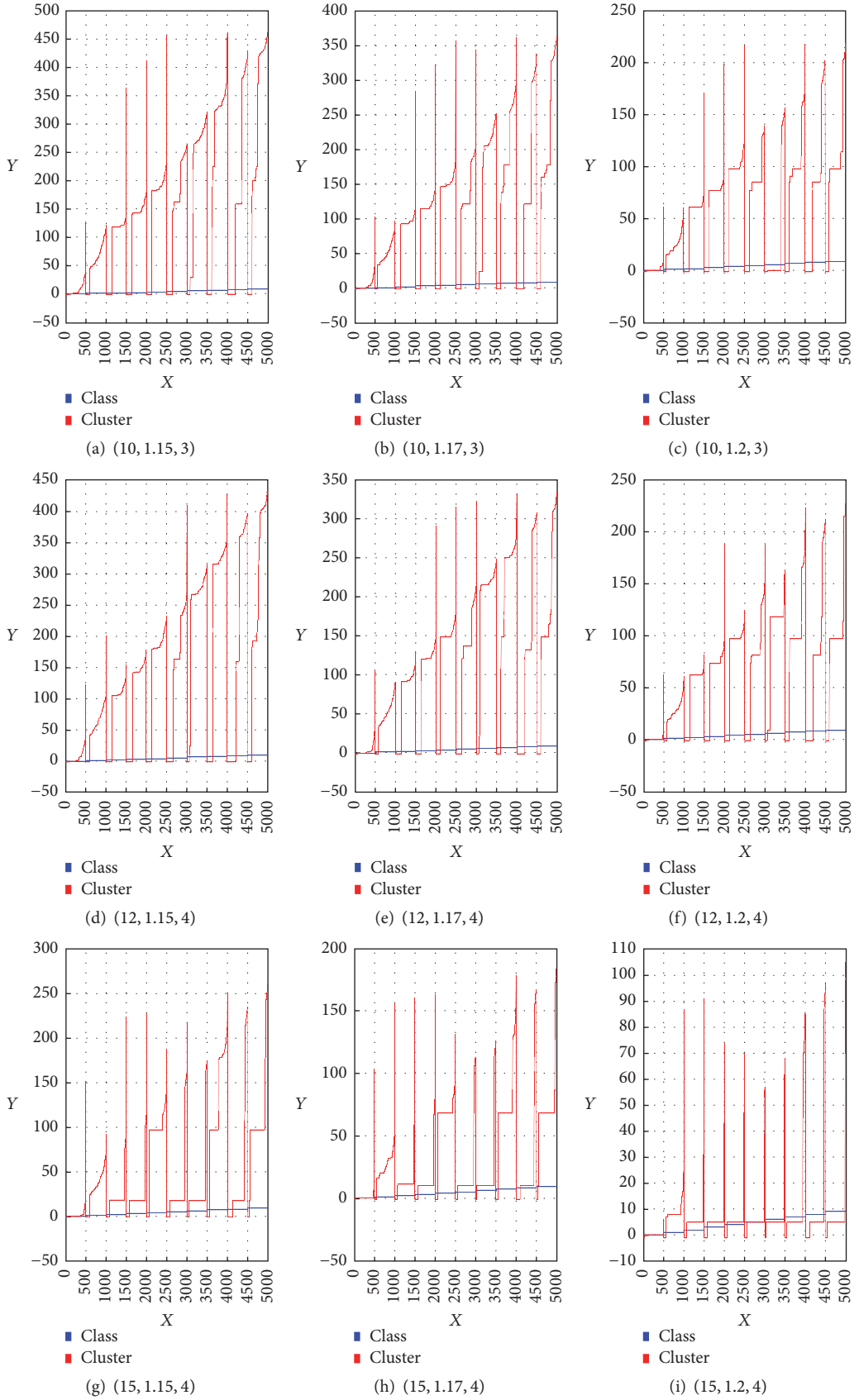


Figure 21: kDDBSCAN with MNIST dataset (in each figure, the blue line represents the ground truth, the red line represents the clustering index value.

LY14F040002, and LY16G020012), Zhejiang Public Welfare Technology Research Project Foundation (no. 2017C31040), and Ningbo Natural Science Foundation (nos. 2015A610130 and 2013A610006).

References

[1] M. Ester, H. P. Krigel, J. Sander, and X. Xu, "A Density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the International Conference on Knowledge Discovery and DataMining*, pp. 226–231, 1996.

[2] O. Uncu, W. A. Gruver, D. B. Kotak, D. Sabaz, Z. Alibhai, and C. Ng, "GRIDBSCAN: GRId density-based spatial clustering of applications with noise," in *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2976–2981, Taiwan, October 2006.

[3] C. Xiaoyun, M. Yufang, Z. Yan, and W. Ping, "GMDBSCAN: Multi-density DBSCAN cluster based on grid," in *Proceedings of the IEEE International Conference on e-Business Engineering, ICEBE'08*, pp. 780–783, China, October 2008.

[4] X. Chen, W. Liu, H. Qiu, and J. Lai, "APSCAN: A parameter free algorithm for clustering," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 973–986, 2011.

[5] L. Peng, Z. Dong, and W. Naijun, "VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise," in *Proceedings of the ICSSSM'07: 2007 International Conference on Service Systems and Service Management*, China, June 2007.

[6] T.-Q. Huang, Y.-Q. Yu, K. Li, and W.-F. Zeng, "Reckon the parameter of DBSCAN for multi-density data sets with constraints," in *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009*, pp. 375–379, China, November 2009.

[7] Z. Xiong, R. Chen, Y. Zhang, and X. Zhang, "Multi-density DBSCAN algorithm based on density levels partitioning," *Journal of Information and Computational Science*, vol. 9, no. 10, pp. 2739–2749, 2012.

[8] J. Hou, H. Gao, and X. Li, "DSets-DBSCAN: a parameter-free clustering algorithm," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3182–3193, 2016.

[9] A. Ram, A. Sharma, A. S. Jalal, R. Singh, and A. Agrawal, "An enhanced density based spatial clustering of applications with noise," in *Proceedings of the 2009 IEEE International Advance Computing Conference, IACC 2009*, pp. 1475–1478, India, March 2009.

[10] B. Borah and D. K. Bhattacharyya, "DDSC: A density differentiated spatial clustering technique," *Journal of Computers*, vol. 3, no. 2, pp. 72–79, 2008.

[11] D. Pascual, F. Pla, and J. Sanchez, "Non parametric local density-based clustering for multimodal overlapping distributions," *Intelligent Data Engineering and Automated Learning IDEAL*, pp. 671–678, 2006.

[12] W. Ashour and S. Sunoallah, "Multi density DBSCAN," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6936, pp. 446–453, 2011.

[13] M. Debnath, P. K. Tripathi, and R. Elmasri, "K-DBSCAN: Identifying spatial clusters with differing density levels," in *Proceedings of the 2015 International Workshop on Data Mining with Industrial Applications, DMIA 2015*, pp. 51–60, Paraguay, September 2015.

[14] M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6187, pp. 482–500, 2010.

[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, pp. 416–423, July 2001.

[16] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[17] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey, "Complex wavelet structural similarity: a new image similarity index," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2385–2401, 2009.

[18] Y. LeCun and C. Cortes, "The mnist database of handwritten digits," Tech. Rep., Available electronically at http://yann.lecun.com/exdb/mnist, 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 3125879, 8 pages
https://doi.org/10.1155/2018/3125879

Research Article

Multimodal Feature Learning for Video Captioning

Sujin Lee and Incheol Kim

Department of Computer Science, Kyonggi University, San 94-6, Yiui-dong, Youngtong-gu, Suwon-si 443-760, Republic of Korea

Correspondence should be addressed to Incheol Kim; kic@kgu.ac.kr

Received 6 October 2017; Revised 16 January 2018; Accepted 24 January 2018; Published 19 February 2018

Academic Editor: Daniel Zaldivar

Video captioning refers to the task of generating a natural language sentence that explains the content of the input video clips. This study proposes a deep neural network model for effective video captioning. Apart from visual features, the proposed model learns additionally semantic features that describe the video content effectively. In our model, visual features of the input video are extracted using convolutional neural networks such as C3D and ResNet, while semantic features are obtained using recurrent neural networks such as LSTM. In addition, our model includes an attention-based caption generation network to generate the correct natural language captions based on the multimodal video feature sequences. Various experiments, conducted with the two large benchmark datasets, Microsoft Video Description (MSVD) and Microsoft Research Video-to-Text (MSR-VTT), demonstrate the performance of the proposed model.

1. Introduction

As video data increases, there has been a recent surge of interest in automatic video content analysis. Furthermore, technological advancement in computer vision, natural language processing, and machine learning has resulted in an increase of interest in complex intelligence problems relating to the simultaneous understanding of natural language and video clips. Video-based complex intelligence problems typically include video captioning and video question answering. As illustrated by the example shown in Figure 1, video captioning refers to the task of generating a natural language sentence that explains the content of the input video clip.

Video captioning process generally comprises feature extraction from input video clips and caption generation based on the extracted features. In many related works, video captioning was addressed using an encoder-decoder framework [1–3]. In these frameworks, features are first extracted by the encoder, followed by caption generation using the decoder. A convolutional neural network (CNN) like ResNet [4], VGG [5], and C3D [6] is selected as an encoder for such frameworks, whereas a recurrent neural network (RNN) like LSTM [7] is chosen as a decoder. However, they considered frame features of the video equally, without any particular focus. Some subsequent works have attempted to make use of an attention-based mechanism to learn where to focus in the image/video during captioning [8–10]. On the other hand, they still ignore the gap between low-level video feature and sentence descriptions, without clearly representing high-level video concepts. In order to address the above-mentioned problems, recent works add explicit high-level semantic concepts of the input image/video [11–13]. Although significant performance improvements were achieved, integration of semantic concepts into the LSTM-based caption generation process is still constrained in these ways: semantic features are used only (1) for initialization of the first step of the LSTM or (2) for implementing a soft attention mechanism to the LSTM-based caption generation process.

This study proposes a deep neural network model, SeFLA (SEmantic Feature Learning and Attention-Based Caption Generation), for effective video captioning by utilizing both visual and semantic features that describe the video content. In the proposed model, visual features are extracted using ResNet CNN, while semantic features are obtained using LSTM RNN. Moreover, the proposed model adopts an attention-based mechanism that determines which semantic feature to focus on at every time step to generate correct captions effectively based on the multimodal video features. To assess the performance of the suggested model, various

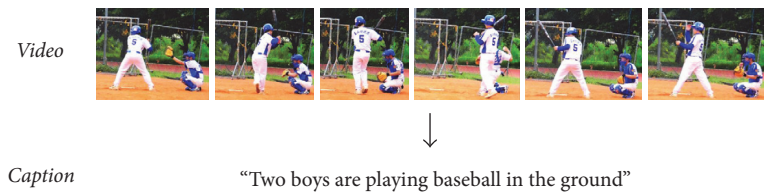


Figure 1: Example of video captioning.

Figure 2: Examples of dynamic and static semantic features.

experiments are run using the Microsoft Video Description (MSVD) [14] and Microsoft Research Video-to-Text (MSR-VTT) [15] datasets, following which the results are discussed.

2. Related Work

Previously, visual content understanding and natural language processing were not correlated with each other. Integrating visual content with natural language learning to generate descriptions for images/videos has been regarded as a challenging task [16, 17]. Video captioning is a critical step towards machine intelligence and many applications such as video retrieval, video understanding, blind navigation, and automatic video subtitling. Inspired by the successful use of the encoder-decoder framework employed in machine translation, many existing works on video captioning employ a convolutional neural network (CNN) as an encoder, obtaining a fixed-length vector representation of a given video. On the other hand, they adopt a recurrent neural network (RNN), typically implemented with long short-term memory (LSTM) [7] as a decoder to generate a natural language caption [1–3]. However, although there is salient part of the video that contribute more to captioning, they considered frame features of the video equally, without any particular focus.

Some recent works attempted to make use of an attention-based mechanism to learn where to focus in the image/video during caption generation [8–10]. Attention mechanism is a standard part of the deep learning toolkit, contributing to impressive results in neural machine translation, visual captioning, and question answering. Attention mechanism applicable to a video clip can be categorized into temporal attention, which indicates the frames to focus on in a video frame sequence and spatial attention, which specifies the key regions in a frame. In a recent work, an adjusted temporal attention mechanism is employed to avoid focusing on non-visual words (e.g., "the" and "a") during caption generation [10]. Although the attention-based approaches mentioned above have achieved excellent results, they still ignore the gap between low-level video feature and sentence descriptions, without clearly representing high-level video concepts.

Furthermore, recent works show that adding explicit high-level semantic concepts of the input image/video can further improve visual captioning [11–13]. In these works, detecting explicit semantic concepts encoded in an image/video and adding this high-level semantic information into the CNN-LSTM framework have improved performance significantly. Specifically, [16, 17] proposed to discover and integrate the rich semantic description, such as objects, scenes, and actions, to benefit the video caption task. Their models jointly learn the dynamics within both visual and textual modalities for video captioning. Although significant performance improvements were achieved, integration of semantic concepts into the LSTM-based caption generation process is still constrained in these ways: semantic features are used only (1) for initialization of the first step of the LSTM or (2) for implementing a soft attention mechanism to the LSTM-based caption generation process. Also, unlike our SeFLA model, previous works using semantic features [11, 12] are limited in that they do not distinguish the dynamic semantic features from the static semantic features. Moreover, they use a relatively simple LSTM model for generating captions.

3. Video Captioning Model

3.1. Model Outline. This study proposes a video captioning model that utilizes semantic features along with visual features that describe video clips for more effective video captioning. Direct linking of visual features extracted by a convolutional neural network (CNN), such as ResNet and VGG, to LSTM-based textual caption generation may ignore the rich intermediate/high-level description, such as objects, scenes, and actions. To address the issue, this study employs additionally two different types of semantic features: dynamic and static semantic features. As shown in Figure 2(a), dynamic semantic feature corresponds to the action taking place within the input video. In contrast, static semantic feature refers to the object, person, and background present in the video, as illustrated in Figure 2(b). In other words, verbs in caption sentence correspond to dynamic semantic feature and nouns to static semantic features.

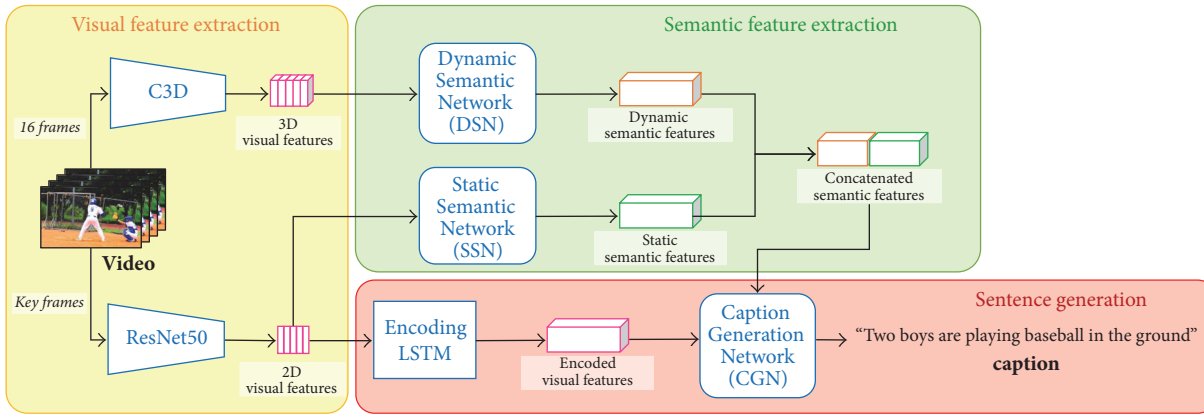


Figure 3: Overall framework of the proposed video captioning model.

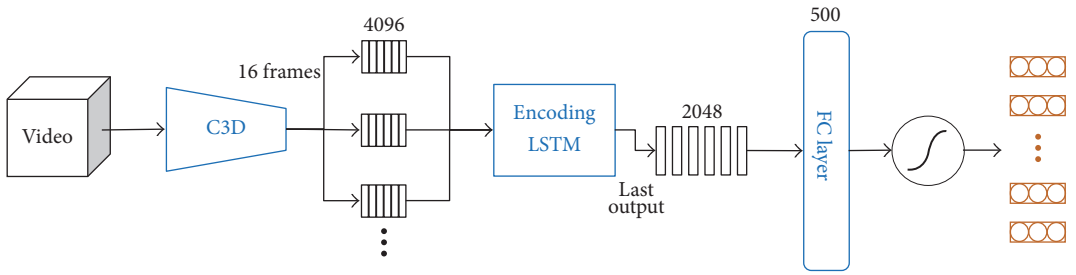


Figure 4: Dynamic semantic feature network (DSN).

The overall framework of the proposed SeFLA model is illustrated in Figure 3. It consists of three main parts: visual feature extraction, semantic feature extraction, and sentence generation. First, visual features required for caption generation are extracted using pretrained ResNet and C3D. The extracted visual features then serve as inputs to the Dynamic Semantic Network (DSN) and static semantic network (SSN), which will be introduced in Section 3.2. In particular, DSN uses visual features of C3D which effectively represents dynamic feature of the video, whereas SSN uses ResNet which represents static feature. Dynamic semantic features and static semantic features are then extracted from each network, which are subsequently concatenated and utilized as inputs to the caption generation network (CGN) introduced in Section 3.3, at each time step. Moreover, CGN applies the attention mechanism on the concatenated semantic features to treat each semantic feature differently at each time step. Visual features extracted via ResNet serve as inputs not only to the SSN, but also to the LSTM that encodes visual features. The final output from the encoding LSTM is given to the initialization step of the CGN. At every time step, the CGN determines the specific semantic feature to focus on and computes the probability distribution of the words. Afterwards, the caption is generated based on the probability distribution of the output words.

3.2. Semantic Feature Learning. To implement caption generation using semantic features, they must first be identified from the input video. As explained previously, semantic features can be categorized into dynamic semantics that illustrate actions and static semantics that denote objects, persons, and backgrounds; clear-cut differences exist between these. Identification of a dynamic semantic feature based on a single frame is hardly possible and requires observation of the video clip for a certain period. On the other hand, a static semantic feature corresponds to an object, person, or background present in a particular moment and, thus, can be identified using a single frame. Hence, extraction of dynamic and static semantic features was carried out separately and treated as a matter of multilabel classification in this study. Dynamic semantic features were extracted based on visual features that effectively illustrated temporal and spatial features of the video, while static semantic features were extracted based on visual features that effectively described the spatial features.

The DSN suggested in this research is shown in Figure 4. First, visual features were extracted in clips, intervals of 16 frames, using a pretrained C3D CNN (see (1)) to exploit the visual features that effectively described the temporal and spatial features of the video. $v_1^i, \ldots, v_{16}^i$ in (1) denotes each single frame in the ith clip and n_v the total number of frames. The extracted visual features (c_i) are then encoded (e) using the LSTM RNN model, as shown in (2). c_t refers to the visual feature corresponding to a single clip encoded at the current time step (t), while h_{t-1} denotes the previous hidden state of the LSTM.

$$c_i = \mathrm{C3D}\left(v_1^i, \ldots, v_{16}^i\right), \quad i \in \left\{0, 1, \ldots, \frac{n_v}{16}\right\} \tag{1}$$

$$e = \mathrm{LSTM}\left(c_t, h_{t-1}\right). \tag{2}$$

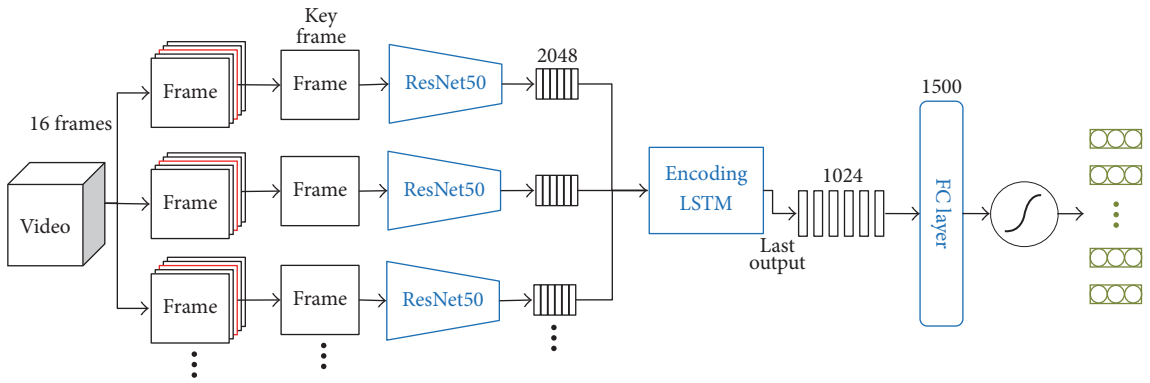


Figure 5: Static semantic feature network (SSN).

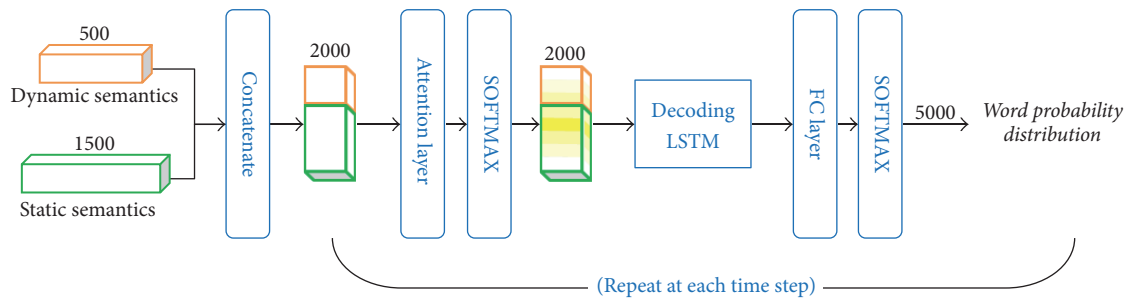


Figure 6: Caption generation network (CGN).

Next, the probability distribution of the dynamic semantic feature (p_d) can be determined from the encoded visual feature (e), fully connected layer, and sigmoid activation function, as shown in (3), where W_d denotes the weight values to be trained and b_d the bias.

$$p_d = \text{sigmoid}\left(W_d \cdot e + b_d\right). \tag{3}$$

The proposed SSN is shown in Figure 5. First, temporal features are extracted from the pretrained ResNet CNN to utilize visual features that effectively describe the spatial features of the video. The video is then divided into clips, intervals of 16 frames each, as expressed by (4), and the visual features (r_i) extracted from the 8th frame (v_8^i) in the ith clip are encoded (e) by LSTM in the manner shown in (5). Subsequently, as shown in (6), fully connected layer and sigmoid activation function are used to determine the probability distribution (p_s) of the SSN.

$$r_i = \text{ResNet}\left(v_8^i\right), \quad i \in \left\{0, 1, \ldots, \frac{n_v}{16}\right\} \tag{4}$$

$$e = \text{LSTM}\left(r_t, h_{t-1}\right) \tag{5}$$

$$p_s = \text{sigmoid}\left(W_s \cdot e + b_s\right). \tag{6}$$

3.3. Attention-Based Caption Generation. This research proposes an attention-based caption generation network (CGN) for effective caption generation using multimodal features, as illustrated in Figure 6.

CGN receives dynamic semantic features and static semantic features as inputs at every time step and identifies the probability distribution. Both dynamic and static semantic features are concatenated and serve as inputs for the attention layer. Conventionally, it is advisable to direct attention to an object within the video if the word to be generated is a noun, and similarly the focus should be on a behavior observed in the video if the word is a verb. In this paper, the attention layer is used to determine the type of semantic feature to focus on at the current time step when implementing a CGN. At the attention layer, a weight value (W_a) that reflects the semantic feature to focus on at a current time step (t) is applied to compute semantic features. The weighted semantic feature (a_t) can be calculated using (7), where s_t refers to the semantic feature given as input and b_a denotes the bias.

$$a_t = \text{softmax}\left(W_a \cdot s_t + b_a\right). \tag{7}$$

The converted semantic features serve as inputs to the decoding LSTM. The decoding LSTM learns sentence structures based on the input semantic features (a_t) and output a status value (h_t) that indicates the word to be generated at the current time step (t), as expressed in (8). The initial hidden state ($h_{t=0}$) of the decoding LSTM is initialized as the final hidden status value of the encoding LSTM that encodes the temporal features.

$$h_t = \text{LSTM}\left(a_t, h_{t-1}\right). \tag{8}$$

The outputs from the decoding LSTM are given as inputs to the fully connected layer. The probability distribution (p_t),

which indicates appropriate words at the current time step (t), is computed in the fully connected layer according to (9), where W_p refers to the weight value to be trained, h_t the inputs given from the decoding LSTM, and b_p the bias.

$$p_t = \text{softmax}\left(W_p \cdot h_t + b_p\right). \tag{9}$$

At each time step, attention values for input semantic features are computed, and the probability distribution of words is output via decoding LSTM and fully connected layer. Then, the output words are strung in order from the first to the keyword denoting the end of statement "⟨EOS⟩" to generate a caption.

4. Performance Evaluation

4.1. Dataset. To train and assess the performance of the CGN suggested in the study, the MSVD dataset and a video caption dataset collected from YouTube videos were used. The MSVD dataset consisted of 1970 YouTube video clips and 80,000 caption statements corresponding to such clips. The sizes of training, cross-validation, and test sets were 1200, 100, and 670, respectively.

On the other hand, the MSR-VTT (Video-to-Text) dataset consists of around 10,000 web video clips. The video clips are classified into 20 categories: music, people, gaming, sports/actions, news/events/politics, education, TV shows, movie/comedy, animation, vehicles/autos, how-to, travel, science/technology, animals/pets, kids/family, documentary, food/drink, cooking, beauty/fashion, and advertisement. They are divided into 6513, 497, and 2990 videos for training, validation, and test sets, respectively. Each video has around 20 natural language captions.

To train the semantic feature networks suggested in the study, training datasets were required. To collect datasets for training, MSVD video caption datasets were used. First, the Part-Of-Speech (POS) tag function in Natural Language Toolkit (NLTK) was used to separate nouns and verbs, while plural nouns and tenses of verbs, past, continuous, and so on, were converted back to their root forms using the lemmatize function in NLTK. Among the extracted verbs, the 500 most frequently appearing words were selected as labelled data for the dynamic semantic features, while 1500 most frequent nouns were chosen as labelled data for static semantic features. A video was labelled with 1 if its caption contained one of the verbs designated as labelled data for dynamic semantic feature, and 0 otherwise. The static dataset was compiled in a similar fashion. Each video contained approximately 7 nouns and 3 verbs present in the datasets. The semantic feature datasets comprised 1200, 100, and 670 examples for training, cross-validation, and test, respectively, like the MSVD caption dataset.

4.2. Model Training. For this research, Keras, a deep learning library in Python, was run in Ubuntu 14.04 LTS environment to implement the proposed models. The hardware specifications for the experiments are as follows: CPU: Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz, RAM: 32 GB, and GPU: GeForce GTX 1080. Input videos were tailored with uniform sampling such that each video contained 40 clips, and each clip consisted of 16 frames. For the semantic feature networks (SSN and DSN), Adam was used as the model optimization algorithm, and the binary cross-entropy cost function in (10) was used for the loss function. Here, y denotes the actual value, while $\tilde{y}$ indicates the expected value.

$$L_{\text{binary}} = -\left[y \log \tilde{y} + (1 - y) \log (1 - \tilde{y})\right]. \tag{10}$$

Once the semantic feature networks were fully trained, semantic features were extracted from all videos in the caption dataset, which were then used as inputs for the caption generation network (CGN). For the CGN, RMSprop was used as the model optimization algorithm, and the categorical cross-entropy cost function in (11) was selected as the loss function.

$$L_{\text{categorical}} = -\frac{1}{n}\sum_{x}\left[y \log \tilde{y} + (1 - y) \log (1 - \tilde{y})\right]. \tag{11}$$

The batch size and the epoch for learning semantic feature networks (SSN and DSN) were set at 32 and 500, while those for the caption generation network (CGN) were 25 and 50, respectively.

4.3. Experiments. The first experiment was conducted to assess the performance of the semantic feature extraction network suggested in this study. The accuracy for each semantic feature extraction network was calculated with Mean Square Error (MSE) as shown in (12). In (12), n represents the output dimension, y_i the actual value, and $\tilde{y}_i$ the expected value.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \tilde{y}_i\right)^2. \tag{12}$$

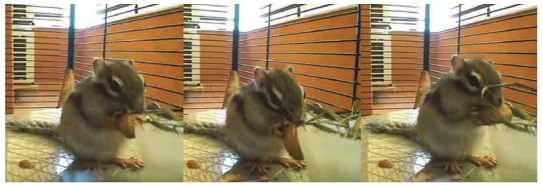
The level of performance of each network evaluated using MSE is tabulated in Table 1, where DSN and SSN denote the dynamic and static semantic network, respectively. The recorded values in the table indicate a high accuracy of semantic feature extraction in both networks.

Table 1: Performance of semantic feature networks on MSVD dataset.

Networks	Val-accuracy	Test-accuracy
DSN	99.42%	99.43%
SSN	99.61%	99.64%

Figure 7 shows the results for the qualitative assessment of both semantic feature networks. As illustrated in the figure, the SSN extracts words that indicate that the subjects are carrying out certain behaviors, whereas the DSN extracts words that describe the behaviors displayed by the subjects.

The aim of the second experiment was to investigate the effects of each semantic feature on caption generation performance. The CGN used in this experiment was kept the same as the selective attention CGN suggested in this study, while the input features were varied. BLEU@N [18] and CIDEr-D [19], which are typical caption generation evaluation metrics, were selected as measures for the performance of CGN. All



Static semantics: squirrel, peanut, chipmunk, nut, rabbit, shell, animal
Dynamic semantics: eat, play, chew

Static semantics: piano, boy, kid, song keyboard, room, music
Dynamic semantics: play, sit, sing

Figure 7: Some examples of semantic features.

Table 2: Comparison of different feature sets on MSVD dataset.

Feature sets	B@1	B@2	B@3	B@4	CIDEr
CGN	66.1	47.8	37.1	26.5	26.4
DSN + CGN	76.0	58.1	45.7	35.8	50.0
SSN + CGN	78.8	63.4	51.4	41.4	77.8
DSN + SSN + CGN	**84.8**	**70.8**	**60.0**	**50.0**	**94.3**

Table 3: Performance comparison with other state-of-the-art models on MSVD dataset.

Models	B@1	B@2	B@3	B@4	CIDEr
SCN [11]	-	-	-	51.1	77.7
LSTM-TSA [12]	82.8	72.0	62.8	52.8	74.0
hLSTMat [10]	82.9	**72.2**	**63.0**	**53.0**	73.8
SeFLA	**84.8**	70.8	60.0	50.0	**94.3**

Table 4: Performance comparison with other state-of-the-art models on MSR-VTT dataset.

Models	BLEU@4
MP-LSTM (V) [1]	34.8
MP-LSTM (C) [1]	35.4
MP-LSTM (V + C) [1]	35.8
SA (V) [2]	35.6
SA (C) [2]	36.1
SA (V + C) [2]	36.6
hLSTMt [10]	37.4
hLSTMat [10]	38.3
SeFLA	**41.8**

evaluation metrics were computed using codes provided by Microsoft COCO evaluation server. CGN in Table 2 depicts the case when captions were generated using solely the visual features, DSN + CGN the case when only DSN was used, SSN + CGN the case when only SSN was used, and finally DSN + SSN + CGN the case when both DSN and SSN were utilized in tandem.

The results in Table 2 indicate that models that utilized semantic feature networks were more effective than the case that only used the CGN. A noteworthy observation is that the DSN + CGN model performed better than the SSN + CGN model. This may be attributed to the effect of the dynamic semantic feature that indicates activity present in the video unlike static semantic feature that can only illustrate objects, persons, and backgrounds. Also, this may be caused by the fact that, in a given caption for a video, there are usually one verb (activity) and multiple nouns (objects). Furthermore, the model incorporating both the DSN and SSN proved to be the most effective, implying that the two semantic feature networks contribute to the caption generation performance independently.

The third experiment was conducted on MSVD dataset for a comparative assessment of the SeFLA caption generation model that was proposed in this study. Table 3 records the performance of SeFLA in comparison with the other models proposed in previous studies. SCN [11] in Table 3 was suggested by Gan et al., while LSTM-TSA [12] and hLSTMat [10] were proposed by Song et al., respectively. SCN and LSTM-TSA incorporate semantic feature networks, while hLSTMat employs an attention-based layered LSTM as the RNN for caption generation. Specifically, both SCN and LSTM-TSA use semantic features as well as visual features. However, unlike our SeFLA, they are limited in that they do not distinguish the dynamic semantic features from the static semantic features.

From Table 3, the performance achieved by SeFLA is observed to be 84.8% and 94.3% on BLEU@1 and CIDEr, respectively. This indicates that SeFLA is more effective by 1.9% and 16.6% than the other models for the respective metrics. However, SeFLA recorded subpar performance in BLEU@2, BLEU@3, and BLEU@4, illustrating that SeFLA, although effective in predicting word by word, is relatively inefficient when consecutively predicting a few words. This observation is also reflective of SeFLA's ineffectiveness in generating prepositional and postpositional particles, in contrast to its superiority in generating nouns or verbs with the help of semantic features. Such a problem might arise due to the lack of datasets to train the CGN on the sentence structures of LSTM. However, in general standards, the caption generating capability of SeFLA using semantic features, as proposed by this study, can be considered efficient.

Table 4 shows the performance comparisons between the SeFLA model and other models on MSR-VTT dataset. (V) denotes that the model uses VGGnet as a CNN model for video encoding, (C) denotes C3D, and (V + C) denotes that the model use both CNN models.

Table 4 shows that the proposed SeFLA model achieved 41.8% BLEU@4 score, that is, 3.5%, better performance than

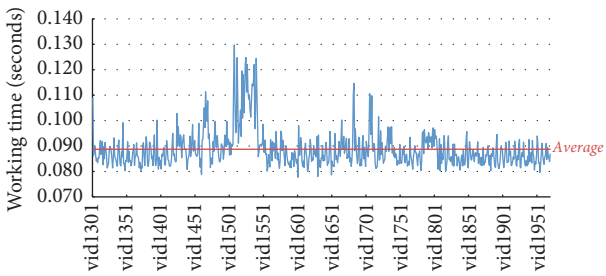


Figure 8: Working time of the SeFLA model on each MSVD test video.

previous studies on MSR-VTT. The result indicates that the SeFLA model has better caption generation performance than previous models with the help of semantic features.

In the fifth experiment, the working time of the SeFLA model, which is the caption generation time, was measured on MSVD test dataset. Note that the feature extraction time is not included in the working time.

Figure 8 shows the results of working time measurement, and the average working time was 0.89 sec. Each working time was affected by the number of words in the generated caption and the length of the input video.

5. Conclusion

This study proposed a deep neural network model capable of effective video captioning. Apart from visual features, the proposed model learns additionally semantic features that describe the video content effectively. In our model, visual features of the input video are extracted using convolutional neural networks such as C3D and ResNet, while semantic features are obtained using recurrent neural networks such as LSTM. In addition, our model includes an attention-based caption generation network to generate the correct natural language captions based on the multimodal video feature sequences. Various experiments, conducted with the two large benchmark datasets: Microsoft Video Description (MSVD) and Microsoft Research Video-to-Text (MSR-VTT), demonstrate the performance of the proposed model. Our future works are as follows. First, a more sophisticated attention mechanism will be incorporated into our SeFLA model for further boosting video captioning. Second, we will investigate how to leverage multimodal features for multiple sentence generation for videos.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Technology Innovation Program or Industrial Strategic Technology Development Program (10077538, Development of Manipulation Technologies in Social Contexts for Human-Care Service Robots) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

[1] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - Video to text," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4534–4542, December 2015.

[2] L. Yao, A. Torabi, K. Cho et al., "Describing videos by exploiting temporal structure," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4507–4515, December 2015.

[3] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly Modeling Embedding and Translation to Bridge Video and Language," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4594–4602, Las Vegas, NV, USA, June 2016.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 770–778, Las Vegas, Nev, USA, June 2016.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the in Proceedings of the International Conference on Learning Representations (ICLR15)*, 2015.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4489–4497, December 2015.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] K. Xu, J. Ba, and R. Kiros, "attend and tell: neural image caption generation with visual attention," in *Proceedings of the in Proceedings of International Conference on Machine Learning (ICML15)*, 2015.

[9] M. Zanfir, E. Marinoiu, and C. Sminchisescu, "Spatio-temporal attention models for grounded video captioning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 10114, pp. 104–119, 2017.

[10] J. Song, L. Gao, Z. Guo, W. Liu, D. Zhang, and H. T. Shen, "Hierarchical LSTM with Adjusted Temporal Attention for Video Captioning," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 2737–2743, Melbourne, Australia, August 2017.

[11] Z. Gan, C. Gan, X. He et al., "Semantic Compositional Networks for Visual Captioning," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1141–1150, July 2017.

[12] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 984–992, Honolulu, Hawaii, USA, July 2017.

[13] Y. Yu, H. Ko, J. Choi, and G. Kim, "End-to-end concept word detection for video captioning, retrieval, and question answering," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3261–3269, July 2017.

[14] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar et al., "Youtube2text: recognizing and describing arbitrary activities

using semantic hierarchies and zero-shot recognition," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, (ICCV '13)*, pp. 2712–2719, December 2013.

[15] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 5288–5296, July 2016.

[16] F. Nian, T. Li, Y. Wang, X. Wu, B. Ni, and C. Xu, "Learning explicit video attributes from mid-level representation for video captioning," *Computer Vision and Image Understanding*, 2017.

[17] A. Liu, N. Xu, and Y. Wong, "Hierarchical & multimodal video captioning: discovering and transferring multimodal knowledge for vision to language," *Computer Vision and Image Understanding*, 2017.

[18] K. A. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*, pp. 311–318, July 2002.

[19] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '15)*, pp. 4566–4575, June 2015.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 2786952, 23 pages
https://doi.org/10.1155/2018/2786952

Research Article

Improved Unsupervised Color Segmentation Using a Modified *HSV* Color Model and a Bagging Procedure in *K*-Means++ Algorithm

Edgar Chavolla, Arturo Valdivia, Primitivo Diaz, Daniel Zaldivar, Erik Cuevas, and Marco A. Perez

Electronics Department, CUCEI, University of Guadalajara, Avenida Revolución 1500, 44430 Guadalajara, JAL, Mexico

Correspondence should be addressed to Edgar Chavolla; chavolla@gmail.com

Received 6 October 2017; Revised 24 November 2017; Accepted 2 January 2018; Published 14 February 2018

Academic Editor: Qin Yuming

Accurate color image segmentation has stayed as a relevant topic between the researches/scientific community due to the wide range of application areas such as medicine and agriculture. A major issue is the presence of illumination variations that obstruct precise segmentation. On the other hand, the machine learning unsupervised techniques have become attractive principally for the easy implementations. However, there is not an easy way to verify or ensure the accuracy of the unsupervised techniques; so these techniques could lead to an unknown result. This paper proposes an algorithm and a modification to the *HSV* color model in order to improve the accuracy of the results obtained from the color segmentation using the *K*-means++ algorithm. The proposal gives better segmentation and less erroneous color detections due to illumination conditions. This is achieved shifting the hue and rearranging the *H* equation in order to avoid undefined conditions and increase robustness in the color model.

1. Introduction

Machine learning application area is growing every day, so it is possible to find applications using machine learning in health areas [1–5], system behavior predictions [6–10], image and video analysis [11–15], and speech and writing recognition [16–20], just to mention some of the most notable and recent applications. Some of the results of these advances in machine learning can be appreciated in several applications that are widely and freely available. As a result, the usage of machine learning has become a common component in the daily modern life.

Despite the huge improvements done in machine learning in the recent years, it still requires more work and research. This can be stated from the fact that a total accuracy is not yet achieved by machine learning, and sometimes the results are still not usable. This is the reason for the present proposal, which is developed in the spirit of improving a common process performed in image analysis using machine learning.

This paper gives an overview of the color models by stating their advantages and problems found when they are implemented or are used as input in other processes. A second topic discussed in the color models overview is the concept of chromatic and achromatic separation by the exclusion or mitigation of the illumination component. The latter topic has significant relevance, since many of the issues found in color detection and segmentation come from the fact that the illumination can change the perception from a color ranging from bright white to black, visiting several tones of the same base color. The overview will set the ground on which some changes are made to the *HSV* perceptual color model.

Also a section regarding machine leaning algorithms is included, where the relevance of unsupervised learning is explained. Also in this section what issues can be found in the popular *K*-means algorithms is explained, as well as of course some existing techniques used to improve the classification results obtained from the usage of this algorithm (like *K*-means++). This section is used to support some minor extra changes applied in combination with some commonly used techniques that can improve the outcome from the algorithm.

The changes applied to the color model and to the classification algorithm are implemented in a way that they aid the resulting classification process. The resulting process is compared against other color models and some variants. As a testing set, the Berkley Segmentation Dataset and Benchmark BSDS500 is mainly used [21], which provides several testing images and ground truth segmentations done by different subjects. The BSDS500 dataset has been used as a testing environment by other segmentation works [22–27]. Using the BSDS500 dataset gives a more reliable ground in the testing cases.

2. Commonly Used Colors Models

Color is a property that is usually addressed in computer vision, because it becomes useful to distinguish and recognize objects or characteristics in an image. Due to its importance, several ways to describe and explain the color hue have been developed. All these developed methods can be named as color models and color spaces. A color model is a set of equations and procedures used to calculate a specific color, while a color space is the set of all the possible colors generated by a color model.

The additive, the subtractive, the perceptive, and the CIE models are among the most common color models that can be found. Other models were made especially for video and image transmission (like television broadcasting).

The *additive color model* consists of the mixture of two or more colors known as primary colors. The most representative model in this type is the Red-Green-Blue model or *RGB*. This model is widely spread, since it is the base of many electronic devices that display color (television, computers, mobile phones, etc.). This model is the simplest to implement, since it only required an amount of each primary color added over a black surface in order to obtain a given color hue [28].

The *subtractive color model* is similar to the additive color model; it uses a set of primary colors to obtain a color hue. The difference between additive and subtractive color models is that the subtractive model subtracts or blocks a certain amount of primary colors over a white surface instead of adding an amount of the primary colors over a black surface. The most representative subtractive color model is the Cyan-Magenta-Yellow model or *CMY*. This model is mostly used in printing processes.

The idea behind the *perceptual color models* is to create a similar process to the one that occurs when the brain processes an image. It is also referred to as a psychological interpretation of the colors. Basically this type of model splits the color in a hue component, a saturation component, and a light component. The most common models in this category are *HSV*, *HSL*, and *HSI*. These models have the characteristic of being represented by geometric figures, usually a cone, bicone, or cylinder. This type of geometric representation allows an easy manipulation of the color [29].

The *CIE* color models are those models created in the International Commission on Illumination (CIE). The CIE is a global nonprofit organization that gathers and shares information related to the science and art of light, color, vision, photobiology, and image technology [30].

This organization was the first to propose the creation of standardized color models. The most famous are CIE-*RGB*, CIE-*XYZ*, and CIE-*Lab* or *Lab*. The *Lab* color model is used in several image edition software tools, since it offers a robust gamut. The *Lab* color model has an illumination component "*L*" and two chromatic components "*a*" and "*b*."

In the video and image transmissions, different color models are used. These models do not belong to a specific type and are related to the *Lab* color model. These models also split the color into an illumination component and two chromatic components but differ from *Lab* model in the way of the calculation of each component. The main purpose of these models is to adapt to the color image to the transmission processes (television broadcasting). Most of the calculations of the components in these models are meant to be used directly in analog television sets or cameras. The most common models in this category are YCbCr, YUV, and YDbDr.

2.1. Problems with the Color Models. A reason for the existence of many colors models is that none of the color models is perfect. Any color model has failure points or is sometimes hard to manipulate. Due to the advantages and disadvantages, each color model has its own niche.

The additive and subtractive color models are easy to implement and understand, but they do not have a linear behavior. Also they are highly susceptible to illumination changes.

The CIE color models have robust gamut and the illumination component is separated from the chromatic components. The problems with the CIE color models are related to the nonlinearity behavior and the difficulty in the implementation of these models.

The models used for video and image transmissions are designed to be used in digital and analogic transmissions, so the implementation of these models for other purposes is complex.

The perceptual color models have the illumination component isolated and have a linear behavior. These models do not offer a robust gamut as the CIE models, since the perceptual color models only have one component for the chromatic information. Another issue with the perceptual color models comes from the equations used to calculate the hue and the saturation component. Hence, in case of having a white, black, or gray color, these two components could become undefined.

Equations (1), (2), and (3) are used for the *HSV* color model. Using these equations as example, it can be seen that in case of white or black or a gray tone the maximum and the minimum have the same value. In this case the H component (see (3)) becomes undefined. A usual workaround implemented in the most popular image processing libraries is to assign the value of zero when H is undefined. In the *HSV* color model, red hue has an H value of zero. So, it produces erroneous detections assigning the same hue to red, black, and gray tones.

$$V = \max(R, G, B) \tag{1}$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{1 - |\max(R,G,B) + \min(R,G,B) - 1|} \tag{2}$$

$$H = \begin{cases} 60 * \left(\dfrac{G-B}{\max(R,G,B) - \min(R,G,B)} \right) & R = \max(R,G,B) \\ 60 * \left(\dfrac{2 + (B-R)}{\max(R,G,B) - \min(R,G,B)} \right) & G = \max(R,G,B) \\ 60 * \left(\dfrac{4 + (R-G)}{\max(R,G,B) - \min(R,G,B)} \right) & B = \max(R,G,B). \end{cases} \tag{3}$$

2.2. Alternative Color Models. Due to the issues present in color models, some proposals have arrived in order to alleviate the problems found. Some of these alternative color models are variants from the existing color models, and the creation is meant to address a specific issue or to create an easier implementation of the model.

The normalized *RGB* or n-*RGB* was created specifically to help the *RGB* color model to deal with the illumination changes. Illumination is one of the most serious issues when color detection is performed. So the main idea behind the normalization is to use a percentage of the primary color instead of an amount. Theoretically, illumination modifies proportionality each color component, so the *RBG* color (50, 100, and 150) should have the same color hue as the color (5, 10, and 15) but with different illumination.

$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B} \tag{4}$$

$$b = \frac{B}{R+G+B}.$$

Equations (4) are used to calculate the n-*RGB* color space. The n-*RGB* space mitigates the effect of shadows and shines but also it could reduce the detection precision [31].

Another technique to improve detection processes and avoid the effect of illumination is to ignore the illumination component. This is usually done in perceptual color models and *Lab*-like color models, where the illumination component can be split. By applying this partial selection of components from the color models in color segmentation, some interference coming from unnecessary data like illumination or saturation can be avoided. Also as the information input from the color model is reduced, the segmentation and identification process is accelerated in the classification algorithms.

The most common cases are from the perceptual models, where the H and S components [32–34] or the H and V components [35] or only the H component [36, 37] or a mixture between the components [38] is used.

Another case is the partial usage of the *Lab* color model, where the L component is excluded, using only the chromatic components to perform the color detection [39].

3. Adapting *HSV* Color Model to *K*-Means

K-Means is an algorithm classified under the unsupervised learning category. Unsupervised learning algorithms are capable of discovering structures and relationships by themselves just using the input data [40].

The *K*-mean algorithm is commonly used in clustering processes. The algorithm was introduced by MacQueen in 1967 [41], even though the idea was conceived in 1957. The public disclosure of the algorithm was not done until 1982 [42]. The *K*-means algorithm is an iterative method that selects k random clusters centroids. In every iteration, the centroids are adjusted using the closest data points to each centroid. The algorithm ends when a defined iteration has been executed or a desired minimum data-centroid distance has been found. This behavior makes the *K*-means be referred to as an expectation maximization algorithm variant.

K-Means algorithm has some variations that are meant to improve the quality of the resulting segmentation; some popular examples are fuzzy *C*-means and *K*-means++.

The *K*-means algorithm does not always generate good results, mostly due to the random cluster centroid initialization. This random initialization generates problems like the case of having two cluster centroids being defined too close to each other. This would result in the misclassification of one group of related items in two different clusters. Another case is when a cluster centroid is defined far from the real data related group centroid. The random defined cluster centroid could never reach the real centroid in the amount of defined iterations.

These kinds of problem motivate researchers to propose improvements to the original *K*-means algorithm. Arthur and Vassilvitskii proposed an improvement to the *K*-means algorithm, focusing on the initialization process; they called their algorithm *K*-means++ [43]. Basically *K*-means++ uses a simple probabilistic approach to calculate the initial cluster centroids by obtaining the probability of how well a given point performs as a possible centroid.

Due to the advantages and the ease of implementation, this paper uses *K*-means++ in order to create more accurate results in the clustering process.

Image segmentation by using machine learning has been developed in many papers and works. We can find some recent works using neural networks [44–46], Gaussian mixture model [47, 48], support vector machine [49–51], and support vector machine with *K*-means family based training [52, 53]. Even though *K*-means algorithm is old, it is still used in image segmentation due to its ease of implementation [39, 52–54].

As it was mentioned, *HSV* produces undefined values when a black or white or gray tone is present in the image. This would discourage the usage of this model or using it under the premise that sometimes the color detection will fail under the previously mentioned circumstances.

An additional issue comes into account when a distance-based algorithm like *K*-means is used. This comes from the fact that the H component is measured like the angle of a circumference. The usage of this angle representation implies that the next H value for 359 is 0. An algorithm like *K*-means

detects that 359 and 0 are far from each other and they should be classified in different clusters.

The previous issue could be solved by adding additional logic in the distance measurement method by adding rules to avoid the miscalculation of the distance in the H component. Using this approach could produce an excessive increase in the computational work.

The implementation of K-means++ does not guarantee the correct classification of the input items. K-Means++ improves the general outcome by providing a better start. A better start helps to find the best solution faster and/or to reduce the amount of erroneous clusters definition.

The present paper proposes an adaptation to the *HSV* model in order to overcome the previously mentioned issues while providing basis to improve the result in the K-means++ algorithm.

Most of the image libraries use the 1-byte per component representation, which implies that the value for the H component from the *HSV* color model must be adapted to fit in the given space. The preferred approach is to take the half of the H component (divided by two), so the H component goes from 0 to 179. Regarding the S and V components, each has a range from 0 to 255.

This approach is preferred, since if the 2-byte representation is used, the amount of memory required to process the image increases significantly.

3.1. Modified HSV Color Model Calculation. The proposed adaptation applied to *HSV* color model addressed two important issues: the undefined values produced by the H component equation [see (3)] and the discontinuity in this component when it changes from 359 to 0.

The proposed change consists in modifying the way H is defined, especially when it becomes undefined. The idea is to take advantage of the unassigned values in the H byte. The H byte covers a range only from 0 to 179, so 180 to 255 are unassigned empty values. Basically, instead of assigning H to zero when white, black, and grayscale colors are detected, these colors are assigned to a range of the empty values.

The selected range in this work for the black, white, and gray tones is from 200 to 255. The starting point was selected in a way that the separation from the last H value (179) is easily detected by K-means. Lower starting points can be chosen, but 200 was selected in order to remark the separation between the possible clusters. The two areas defined in the H component match the definition of chromatic and achromatic regions. In this case, the chromatic region is defined from 0 to 179 and the achromatic region from 200 to 255. Using the achromatic and chromatic definitions [55] adapted for *HSV* color space [Figure 1], the following can be stated:

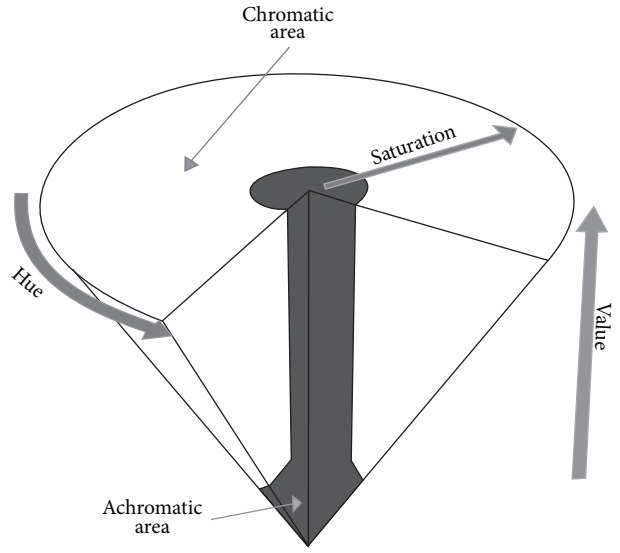


Figure 1: Achromatic and chromatic areas for *HSV* color space.

(1) Color hue (H) is meaningless when the illumination (V) is very low (turns to black).

(2) Color hue (H) is unstable when the saturation (S) is very low (turns to gray).

(3) When saturation (S) is low and illumination (V) is high, the color hue (H) is meaningless (turns to white).

In all the cases when H becomes unstable or meaningless, the achromatic zone is considered; otherwise the chromatic zone is considered. The procedure marks as achromatic an *HSV* value when the saturation is low or the illumination is low. The previous statement requires the definition of a threshold (th) that indicates when the *HSV* value is achromatic. Using the definition, this threshold must be applied to the S and V components and it will indicate when S or V values are low enough to consider H meaningless.

Using the previous concepts in the H equation [see (3)] results in the following equation:

$$H = \begin{cases} (S \text{ and } V \text{ above th}) & \begin{cases} 60 * \left(\dfrac{G - B}{\max(R,G,B) - \min(R,G,B)} \right) & R = \max(R,G,B) \\ 60 * \left(\dfrac{2 + (B - R)}{\max(R,G,B) - \min(R,G,B)} \right) & G = \max(R,G,B) \\ 60 * \left(\dfrac{4 + (R - G)}{\max(R,G,B) - \min(R,G,B)} \right) & B = \max(R,G,B) \end{cases} \\ (S \text{ or } V \text{ below th}) & \left\{ 200 + \left(\left(\dfrac{V}{255} \right) * 55 \right). \right. \end{cases} \tag{5}$$

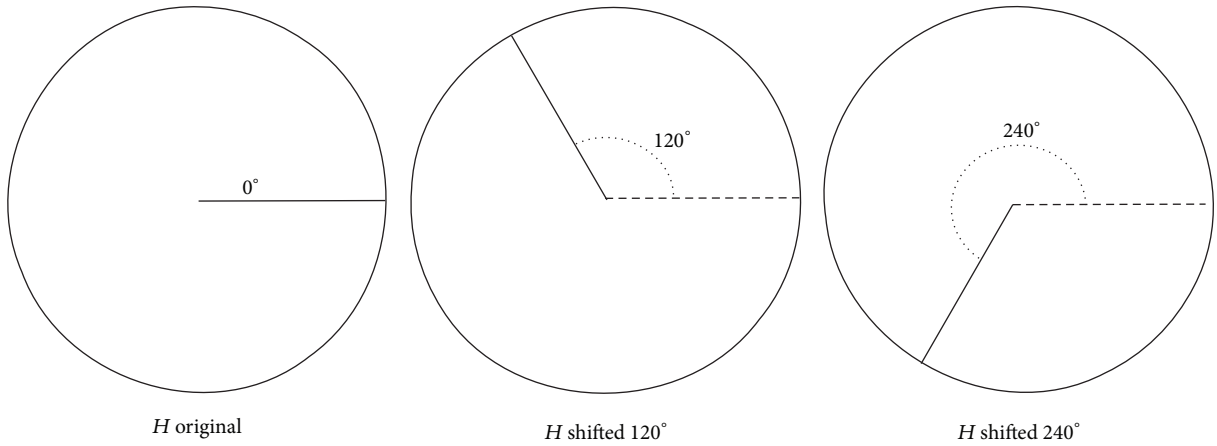


Figure 2: Original H and the two shifted H representations.

As it was explained in the previous sections, H is the angle of a circumference, so the next value from 359 is zero. In the case of the 1 byte per component representation, the next value from 179 is zero. The color hue corresponding to this discontinuity area is the red tone. This issue produces the creation of two separate clusters for the red color, even if the hues are almost the same.

Some approaches can be implemented in order to correct the possible erroneous creation of clusters. Rules in the distance measurement in the K-means++ algorithm when the H value is close to the discontinuity region can be implemented. But this generates an important load in the computer work.

This paper proposes the usage of shifted angles for the H component. This means that the discontinuity can be placed in another color hue. The creation of two shifted angles representations for H is proposed, so they can be combined and eliminate the discontinuity issue. The original H has the discontinuity in the red hue, the first shifted H ($H120$) has the discontinuity in the green hue (120°), and the second shifted H ($H240$) has the discontinuity in the blue hue (240°). As can be seen, the shift operation is done in evenly defined amounts (120° from each H component) [Figure 2].

In the case of the 1-byte representation, the shift amount is 60 for $H120$ and 120 for the $H240$ (half of the original values).

The original and the two shifted H components are meant to be processed by K-means++. This would seem to generate significant extra computational work, but this process is meant to solve the discontinuity issue in the HSV and also improve the classification performed by the K-means++ algorithm. This is explained in detail in the next section where the complete improvement process is exposed.

The work done in this paper uses the HSV partial model approach to eliminate the effect of illumination in component V from the color process. It also excludes the S component, since the main purpose is the segmentation or classification by color hue. The selection of only one component speeds up the process done by K-means++ by reducing the complexity in the input.

```
set threshold_v, threshold_s //predefined thresholds
set H_Entry, H120_Entry, H240_Entry
for each pixel in RGB do
  set V with Eq. (1)
  set S with Eq. (2)
  if V > threshold_v and S > threshold_s then
    set H with the first part of Eq. (5)
    if H > = 120 then
      set H120 equals H − 120
    else
      set H120 equals H + 60
    if H > = 60 then
      set H240 equals H − 60
    else
      set H240 equals H + 120
  else
    set H with the second part of Eq. (5)
    set H120 equals H
    set H240 equals H
  add H, H120, H240 in H_Entry, H120_Entry, H240_Entry
for each entry in [H_Entry, H120_Entry, H240_Entry] do
    execute K-Means with entry
```

Pseudocode 1: Modified HSV color model pseudocode for K-means++.

The complete process to create the input for the K-means++ is as in Pseudocode 1 in order to calculate H, $H120$, and $H240$ for each pixel in the RGB input image.

Since the pseudocode is set to operate in a 1 byte per channel model, the H values are in the range of 0–179. After obtaining the K-means++ clusters, a matching and grouping operation is performed. The reason for the matching and grouping operation is to detect similar clusters and group them together. The idea is that if a cluster group has two or more members, this cluster group has more probability to be a real cluster. This approach makes those cluster groups that were affected by the discontinuity be detected and ignored as

```
set groups, result
for each clusters in [clusters_H, clusters_H120, clusters_H240] do
  for each cluster in clusters do
    if length(groups) == 0 then
      set new_group
      add cluster in new_group
      add new_group in groups
      continue
    for each group in groups do
     if distance(cluster, group) < threshold then
      add cluster in group
     else
      set new_group
      add cluster in new_group
      add new_group in groups
for each group in groups do
 if length(group) == 1 then
  delete group
 else
  add merge(group) in result
```

Pseudocode 2: Cluster grouping.

they usually contain only one member. The shift operation forces the discontinuity to affect a different hue, so the other tones are not affected.

For instance, the original H component is affected in the red hue by the discontinuity, so the K-means++ algorithm would produce a split cluster in this affected hue area. But $H120$ and $H240$ are not affected in the red hue by the discontinuity, so the K-means++ algorithm would produce the correct cluster for a red hue.

Another reason to apply K-means++ to three versions of the same information is to improve the cluster quality. Even though K-means++ is an improvement over K-means, still certain amount of the process relies on randomness, producing sometimes a not so accurate initial centroid. Performing the same classification several times helps to enforce the results by taking those groups of similar clusters with more members as the most probable real clusters. The process for the K-means++ clustering and grouping can be described as in Pseudocode 2.

The purpose of the matching and grouping is to take those clusters with a high similarity and group them together. This could seem to be a trivial task, but its implications make it a complex procedure. The simplest approach is to only use the Euclidean distance between the cluster centroids and group the clusters with the lowest distance [54].

The previous approach could not always produce the best result, due to the variance of the elements in the clusters or the cases of missing clusters or the case where a cluster is divided. An algorithm proposed to match clusters alleviating the possible issues found is the Mixed Edge Cover (MEC) [56]. The MEC algorithm calculates the similarities and dissimilarities between the clusters using a distance measurement that eliminates the variance issues. The Mahalanobis distance between each cluster element can be used for this purpose [57]. So this paper uses the Mahalanobis distance as the similarity measurement between the clusters.

Bagging is a technique used in machine learning, where several versions of a predictor algorithm or a classifier algorithm are used to generate a new predictor or classifiers. Usually this is done by averaging the results in predictors, and, in the case of the classifiers, a voting process is performed [58]. The proposed procedure creates groups of similar clusters and then eliminates the groups with fewer members using a voting system.

After the voting is finished in the first bagging process, a second bagging process is executed in order to create a unified cluster from each selected cluster group. The voting in the second bagging process creates a cluster from those cluster items common in two or more clusters. So if a cluster item appears in just one cluster, this is considered as noise data or a misclassified pixel.

3.2. Proposed Method's Theoretical Ground. The issues found in color spaces are related to discontinuities and nonlinear behaviors. Classification methods based on distances like K-means cannot handle these issues correctly when a color classification is required. The HSV model has a linear behavior in the color hue component H but suffers from a discontinuity when it changes from 359 to 0.

The proposed change moves the discontinuity to different values. It creates two additional versions of the H component ($H120$ and $H240$), where the discontinuity occurs in different colors hues. Performing a clustering operation on one of the components, H, $H120$, or $H240$, produces clusters, where the discontinuity could be manifested in the form of real cluster divided into two clusters. This issue does not exist in the clusters coming from the other two components.

Performing cluster matching and grouping over all the clusters coming from all the components generates groups, where if it contains 2 or more elements it can be considered like a real cluster; otherwise the group can be ignored. So, this process alleviates the discontinuity issue found in the H component.

Additionally splitting the chromatic and achromatic values allows reducing the effect of shines and reflections that can lead to incorrect classification. Also it avoids the issues happening in the HSV when the pixel is a shade of gray (H becomes undefined using (3)). Instead of setting the H value to 0 in this case, the proposed improvement uses an unassigned value range in the H component. This facilitates the clustering process by having a specific region for the chromatic tone and a separated region for the achromatic ones.

All the changes performed in the proposed improvement eliminate the discontinuity and provide a more linear input data for the K-means algorithm. Additionally the changes allow mitigating shadow, shines, and reflection which alter the perception of color tones. This has a positive effect in the classification performed by K-means compared to classification performed using other color models, producing more accurate results.

4. Testing and Experimentation

In the testing process, the proposed HSV model is tested against other color models and the original HSV. The testing dataset comes from two sources, mainly the BSDS500 and a couple of images from the Free Images website [59]. From the first dataset, the ground truth is taken from the files inside the dataset, while in the second test some ground truth images were created. All the color models are processed by the K-means++ algorithm which is set to find 4 or 5 clusters (usually the amount of segmented objects found in the BSDS500 dataset).

Once the clusters are obtained for each tested color model, they will be evaluated using statistical measurements. Usually measurements like specificity [see (6)], sensitivity [see (7)], and accuracy [see (8)] are used in segmentation tests. These measurements use parameters like True Positive (TP, number of pixels included in the segmented object which are correctly classified), True Negative (TN, number of pixels not included in the segmented object which are correctly classified), False Positive (FP, number of pixels included in the segmented object which are incorrectly classified), and False Negative (FN, number of pixels not included in the segmented object which are incorrectly classified). This work uses balanced accuracy [see (9)] [60] in order to use the accuracy as an overall measurement, in which the specificity and sensitivity are added in certain proportion by applying the adjustment parameters α and β (usually these parameters are set to 0.5).

$$\mathrm{SPC} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

$$\mathrm{SEN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{ACC} = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN})} \tag{8}$$

$$\mathrm{Bacc} = \alpha * \mathrm{SEN} + \beta * \mathrm{SPC}. \tag{9}$$

Balanced accuracy measurement should give an overview of how well the test is performing, but unfortunately this is not always possible. Basically, since the parameters depend on the amount of pixels in the segmented object or outside of it, it could lead to a high balanced accuracy if a high value in specificity or sensitivity is calculated. In order to avoid this case, the parameters α [see (10)] and β [see (11)] are calculated considering the number of pixels in the segmented object (VP) and the pixels outside the segmented object or background (BP) [61].

$$\alpha = \frac{\mathrm{VP}}{(\mathrm{VP} + \mathrm{BP})} \tag{10}$$

$$\beta = \frac{\mathrm{BP}}{(\mathrm{VP} + \mathrm{BP})}. \tag{11}$$

The selected color models used for the comparison are those that appear commonly in the literature:

(i) *RGB*
(ii) n*RGB*
(iii) *HSV*
(iv) *HS* Original
(v) *H* Original (*H* Orig)
(vi) *H* Modified (*H* Mod)
(vii) *Lab*
(viii) *ab*
(ix) YCbCr
(x) CbCr

The test for each color model is executed 20 times, taking the best result and the average. So a more reliable statistical comparison can be made among the color models using K-means++ algorithm.

In order to apply the modified HSV model, it is necessary to define a threshold value, th [see (5)], so the chromatic and achromatic regions can be placed in the H component. After performing some tests over a group of images, it was observed that setting the threshold around 30% of the value for the V and S components produced the best result in the segmentation, so this threshold will be used in the tests.

Also a set of images from the selected datasets sources is selected to perform the comparison. The BSDS500 dataset is intended mainly to perform object segmentation. Color segmentation algorithms can solve the segmentation task in some of the proposed scenarios in the BSDS500 dataset. So, taking that in account, a subset of images, where the ground truth is close to color segmentation, was selected.

In order to provide more comparison data regarding the behavior of the proposed improvement, another clustering algorithm is used in the tests. Gaussian mixture model performs in a similar way to K-means, so implementation of

(a) (b) (c) (d) (e) (f)

Figure 3: Original images for segmentation. Images (a), (c), (d), and (e) are from BSDS500. Images (b) and (f) are from Free Images website.

the GMM using the expectation maximization (EM) method is used to provide a comparison with a different algorithm.

For both algorithms, the conditions are similar; both perform 200 iterations. Regarding the starting point for the Gaussians in GMM, the K-means++ initialization algorithm is used to set the initial mean and standard deviation. It creates a scenario where a fair comparison can be made.

4.1. Test Results. A few images from the selected test dataset are exposed in order to demonstrate how every color performs in the segmentation done by K-means++.

The images in Figure 3 are selected to show visually the segmentation done by each of the selected color models and some metrics showing the performance.

In Tables 1–12 in the first row the ground truth clusters coming from each of the images from Figure 3 are given. The following rows contain the results from each of the segmentations produced using each of the selected color models. In the last columns, some metrics measuring the performance are given:

(i) Mean BAcc: the average balanced accuracy using all the data from all the clusters and all the iterations

(ii) Best BAcc: the best individual balanced accuracy for one cluster occurring in the iterations

(iii) Mean sen.: the average sensitivity

(iv) Mean spe.: the average specificity

(v) Avg. time: the running time for the algorithm given in seconds. This is used to measure the CPU time needed. For the proposal, the time measurement is divided into 2 phases: one for the clustering time (C) and another for the bagging time (B)

From the results in Tables 1–12, it can be seen that the proposed improvement is most of the time in the first place.

Table 1: Segmentation performance results for image "a" using K-means++.

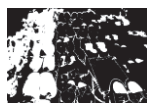
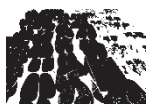
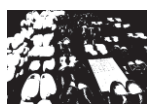
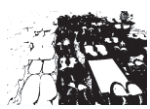
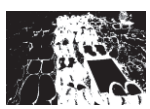
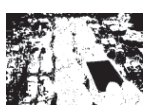
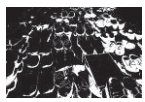
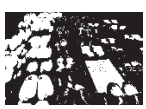
K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
RGB					Best BAcc.: *0.85028* Mean BAcc: *0.74197* Mean sen.: *0.62917* Mean spe.: *0.81776* *Avg. time: 3.8598*
n*RGB*					Best BAcc.: *0.93637* Mean BAcc: *0.82969* Mean sen.: *0.53766* Mean spe.: *0.88299* *Avg. time: 1.5083*
Lab					Best BAcc.: *0.86040* Mean BAcc: *0.7717* Mean sen.: *0.58555* Mean spe.: *0.85133* *Avg. time: 3.5174*
ab					Best BAcc.: *0.95228* Mean BAcc: *0.84635* Mean sen.: *0.71681* Mean spe.: *0.90009* *Avg. time: 1.9523*
YCrCb					Best BAcc.: *0.79657* Mean BAcc: *0.75988* Mean sen.: *0.64811* Mean spe.: *0.83186* *Avg. time: 3.28*
CrCb					Best BAcc.: *0.94294* Mean BAcc: *0.81556* Mean sen.: *0.61425* Mean spe.: *0.87505* *Avg. time: 1.5142*
HSV					Best BAcc.: *0.91797* Mean BAcc: *0.85888* Mean sen.: *0.72094* Mean spe.: *0.90462* *Avg. time: 1.4304*
HS					Best BAcc.: *0.91816* Mean BAcc: *0.83095* Mean sen.: *0.50944* Mean spe.: *0.89749* *Avg. time: 1.333*
H Orig					Best BAcc.: *0.91808* Mean BAcc: *0.76817* Mean sen.: *0.22684* Mean spe.: *0.96200* *Avg. time: 0.7393*
H Mod					Best BAcc.: ***0.96551*** Mean BAcc: ***0.87931*** Mean sen.: *0.77377* Mean spe.: *0.90182* *Avg. time (C): 1.4847* *Avg. time (B): 2.3388*

Table 2: Segmentation performance results for image "a" using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
RGB					Best BAcc.: *0.87485* Mean BAcc: *0.76002* Mean sen.: *0.65702* Mean spe.: *0.82868* *Avg. time: 7.4548*
n*RGB*					Best BAcc.: *0.93637* Mean BAcc: *0.82969* Mean sen.: *0.53766* Mean spe.: *0.88299* *Avg. time: 5.14159*
Lab					Best BAcc.: *0.86040* Mean BAcc: *0.7717* Mean sen.: *0.58555* Mean spe.: *0.85133* *Avg. time: 4.5137*
ab					Best BAcc.: *0.95228* Mean BAcc: *0.84635* Mean sen.: *0.71681* Mean spe.: *0.90009* *Avg. time: 3.1992*
YCrCb					Best BAcc.: *0.79657* Mean BAcc: *0.75988* Mean sen.: *0.64811* Mean spe.: *0.83186* *Avg. time: 4.7854*
CrCb					Best BAcc.: *0.94294* Mean BAcc: *0.81556* Mean sen.: *0.61425* Mean spe.: *0.87505* *Avg. time: 3.1574*
HSV					Best BAcc.: *0.91797* Mean BAcc: *0.85888* Mean sen.: *0.72094* Mean spe.: *0.90462* *Avg. time: 4.2984*
HS					Best BAcc.: *0.91816* Mean BAcc: *0.83095* Mean sen.: *0.50944* Mean spe.: *0.89749* *Avg. time: 3.4293*
H Orig					Best BAcc.: *0.91808* Mean BAcc: *0.76817* Mean sen.: *0.22684* Mean spe.: *0.96200* *Avg. time: 2.82860*

And when it is not in the first place it is really close to the first place. Visually it can be seen also that the closest result to the ground truth images is the modified model.

Aside from the previous tests, some additional tests over other images were executed. In Tables 13 and 14, the means of the results from all the tests are displayed. In addition to the measures exposed in Tables 13 and 14, two new measures were added:

(i) Best mean BAcc: the best average of an iteration

(ii) Worst mean BAcc: the worst average from an iteration

After summarizing all the measurements, it can be noted that the proposed modification has a positive effect across the tests in K-means++. Performing a comparison against the results coming from GMM, the test performed over *ab* (partial model from *Lab*) has a better performance in the

Table 3: Segmentation performance results for image "b" using K-means++.

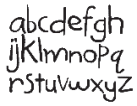
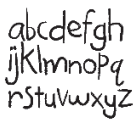
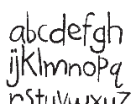
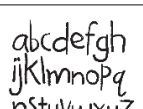
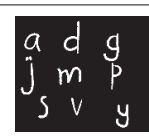
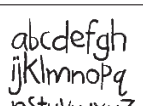
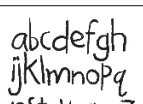
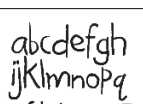
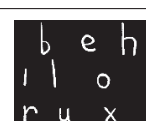
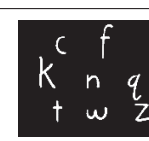
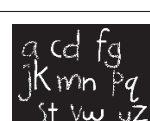
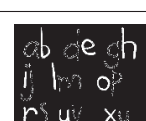
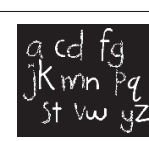
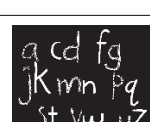
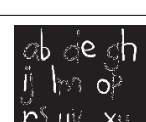
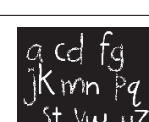
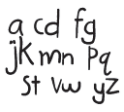
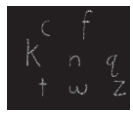
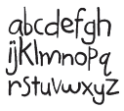
K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
RGB					Best BAcc.: *0.99587* Mean BAcc: *0.98778* Mean sen.: *0.93760* Mean spe.: *0.97967* *Avg. time: 1.75579*
n*RGB*					Best BAcc.: *0.99518* Mean BAcc: *0.95903* Mean sen.: *0.61221* Mean spe.: *0.90010* *Avg. time: 1.8469*
Lab					Best BAcc.: *0.99575* Mean BAcc: *0.99005* Mean sen.: *0.92404* Mean spe.: *0.98184* *Avg. time: 1.113*
ab					Best BAcc.: *0.99780* Mean BAcc: *0.96863* Mean sen.: *0.72131* Mean spe.: *0.92257* *Avg. time: 1.5184*
YCrCb					Best BAcc.: *0.99567* Mean BAcc: *0.98458* Mean sen.: *0.71123* Mean spe.: *0.97607* *Avg. time: 1.84769*
CrCb					Best BAcc.: *0.99688* Mean BAcc: *0.96830* Mean sen.: *0.61425* Mean spe.: *0.92157* *Avg. time: 1.286*
HSV					Best BAcc.: *0.97252* Mean BAcc: *0.90427* Mean sen.: *0.82520* Mean spe.: *0.94918* *Avg. time: 1.41410*
HS					Best BAcc.: *0.97236* Mean BAcc: *0.90964* Mean sen.: *0.81605* Mean spe.: *0.95333* *Avg. time: 1.1394*
H Orig					Best BAcc.: *0.92121* Mean BAcc: *0.84475* Mean sen.: *0.75755* Mean spe.: *0.86620* *Avg. time: 1.3613*
H Mod					Best BAcc.: ***0.99589*** Mean BAcc: ***0.99259*** Mean sen.: *0.94847* Mean spe.: *0.98287* *Avg. time (C): 1.7464* *Avg. time (B): 3.166*

Table 4: Segmentation performance results for image "b" using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
RGB					Best BAcc.: *0.99833* Mean BAcc: *0.98109* Mean sen.: *0.96164* Mean spe.: *0.98576* *Avg. time: 4.6552*
n*RGB*					Best BAcc.: *0.99846* Mean BAcc: *0.95547* Mean sen.: *0.61632* Mean spe.: *0.90544* *Avg. time: 6.5616*
Lab					Best BAcc.: *0.99733* Mean BAcc: *0.98261* Mean sen.: *0.88530* Mean spe.: *0.98876* *Avg. time: 3.3779*
ab					Best BAcc.: *0.99838* Mean BAcc: *0.97665* Mean sen.: *0.84241* Mean spe.: *0.95545* *Avg. time: 3.10470*
YCrCb					Best BAcc.: *0.99833* Mean BAcc: *0.98288* Mean sen.: *0.91770* Mean spe.: *0.98878* *Avg. time: 3.8567*
CrCb					Best BAcc.: *0.99829* Mean BAcc: *0.96058* Mean sen.: *0.69106* Mean spe.: *0.91341* *Avg. time: 4.0068*
HSV					Best BAcc.: *0.99818* Mean BAcc: *0.91195* Mean sen.: *0.84421* Mean spe.: *0.96268* *Avg. time: 3.47339*
HS					Best BAcc.: *0.97405* Mean BAcc: *0.91007* Mean sen.: *0.83561* Mean spe.: *0.96637* *Avg. time: 3.71440*
H Orig					Best BAcc.: *0.92188* Mean BAcc: *0.83645* Mean sen.: *0.75261* Mean spe.: *0.86185* *Avg. time: 2.5643*

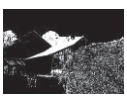
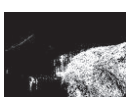
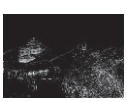
worst BAcc and the best BAcc measurements, but in the average (mean BAcc) the proposed method has a better score.

In order to correctly validate the experimental results, a statistical test is performed over the balanced accuracy observed in the comparison results. In this case, the Wilcoxon test is conducted. The Wilcoxon test is a nonparametric test used when a normal distribution cannot be guaranteed in the data. Its null hypothesis, over two different results, considers that the two compared populations come from the same distribution [62]. The Wilcoxon method has been commonly used to compare algorithms behaviors in order to verify which one has a better performance using normalized values (from 0 to 1) [63]. The Wilcoxon signed-rank sum is set to use the right tail. Under such conditions, the alternative hypothesis is that the first population data has a higher median than the second population data. Therefore, the first

Table 5: Segmentation performance results for image "c" using K-means++.

K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Metrics
Ground truth						
RGB						Best BAcc.: *0.95833* Mean BAcc: *0.84450* Mean sen.: *0.58841* Mean spe.: *0.90370* *Avg. time: 3.72890*
n*RGB*						Best BAcc.: *0.97927* Mean BAcc: *0.90491* Mean sen.: *0.64566* Mean spe.: *0.94043* *Avg. time: 2.6793*
Lab						Best BAcc.: *0.95947* Mean BAcc: *0.84665* Mean sen.: *0.58651* Mean spe.: *0.90504* *Avg. time: 2.82420*
ab						Best BAcc.: *0.984993* Mean BAcc: ***0.96148*** Mean sen.: *0.88592* Mean spe.: *0.97132* *Avg. time: 1.0364*
YCrCb						Best BAcc.: *0.96193* Mean BAcc: *0.84934* Mean sen.: *0.64811* Mean spe.: *0.90670* *Avg. time: 3.2729*
CrCb						Best BAcc.: *0.98338* Mean BAcc: *0.93794* Mean sen.: *0.86600* Mean spe.: *0.95268* *Avg. time: 1.0914*
HSV						Best BAcc.: *0.97474* Mean BAcc: *0.88877* Mean sen.: *0.49019* Mean spe.: *0.95910* *Avg. time: 3.01899*
HS						Best BAcc.: *0.97121* Mean BAcc: *0.86365* Mean sen.: *0.42246* Mean spe.: *0.94520* *Avg. time: 3.0209*
H Orig						Best BAcc.: *0.96433* Mean BAcc: *0.92863* Mean sen.: *0.73995* Mean spe.: *0.94777* *Avg. time: 0.66040*
H Mod						Best BAcc.: ***0.98993*** Mean BAcc: *0.94321* Mean sen.: *0.76518* Mean spe.: *0.96808* *Avg. time (C): 1.9414* *Avg. time (B): 2.5229*

Table 6: Segmentation performance results for image "c" using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Metrics
RGB						Best BAcc.: *0.97231* Mean BAcc: *0.86137* Mean sen.: *0.64875* Mean spe.: *0.91404* *Avg. time: 7.4137*
n*RGB*						Best BAcc.: *0.95474* Mean BAcc: *0.89090* Mean sen.: *0.50531* Mean spe.: *0.94607* *Avg. time: 5.039*
Lab						Best BAcc.: *0.97013* Mean BAcc: *0.86512* Mean sen.: *0.65040* Mean spe.: *0.91608* *Avg. time: 4.7831*
ab						Best BAcc.: *0.98512* Mean BAcc: *0.96448* Mean sen.: *0.90266* Mean spe.: *0.97358* *Avg. time: 3.16470*
YCrCb						Best BAcc.: *0.97427* Mean BAcc: *0.86622* Mean sen.: *0.66280* Mean spe.: *0.91694* *Avg. time: 5.13540*
CrCb						Best BAcc.: *0.98513* Mean BAcc: *0.94658* Mean sen.: *0.88602* Mean spe.: *0.95928* *Avg. time: 1.9721*
HSV						Best BAcc.: *0.98307* Mean BAcc: *0.88245* Mean sen.: *0.52690* Mean spe.: *0.94644* *Avg. time: 5.60439*
HS						Best BAcc.: *0.98099* Mean BAcc: *0.87426* Mean sen.: *0.55223* Mean spe.: *0.92978* *Avg. time: 4.122*
H Orig						Best BAcc.: *0.96963* Mean BAcc: *0.92605* Mean sen.: *0.73088* Mean spe.: *0.94629* *Avg. time: 1.90460*

population data has the balanced accuracy obtained by the proposed approach, while the second has the results obtained by the other color models using the K-means and the GMM algorithms (Tables 15 and 16).

As can be observed in Tables 15 and 16, all the tests reject the null hypothesis about the two data populations being the same using an alpha value of 0.05. The alternative hypothesis stating that the improved color model has a better outcome is selected.

Regarding the time measurements, the proposed method has a similar time to any 3-channel color model used in the tests for K-means++. But the total time used for the proposed improvements is higher when it is considered together with the bagging process. This is expected since the bagging

Table 7: Segmentation performance results for image "d" using K-means++.

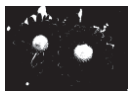
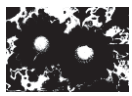
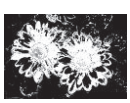
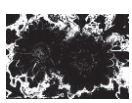
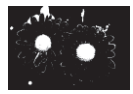
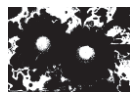
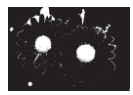
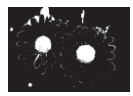
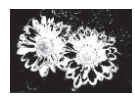
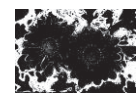
K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
RGB					Best BAcc.: *0.95994* Mean BAcc: *0.86935* Mean sen.: *0.75575* Mean spe.: *0.90860* *Avg. time: 1.76310*
n*RGB*					Best BAcc.: *0.90987* Mean BAcc: *0.81386* Mean sen.: *0.54885* Mean spe.: *0.87593* *Avg. time: 3.72300*
Lab					Best BAcc.: *0.95932* Mean BAcc: ***0.88656*** Mean sen.: *0.78536* Mean spe.: *0.91907* *Avg. time: 1.2676*
ab					Best BAcc.: *0.98018* Mean BAcc: *0.81514* Mean sen.: *0.85177* Mean spe.: *0.80258* *Avg. time: 0.8469*
YCrCb					Best BAcc.: *0.96415* Mean BAcc: *0.86543* Mean sen.: *0.74899* Mean spe.: *0.90738* *Avg. time: 1.87110*
CrCb					Best BAcc.: ***0.98062*** Mean BAcc: *0.81118* Mean sen.: *0.84099* Mean spe.: *0.80067* *Avg. time: 0.9742*
HSV					Best BAcc.: *0.90939* Mean BAcc: *0.78772* Mean sen.: *0.45141* Mean spe.: *0.85449* *Avg. time: 1.25549*
HS					Best BAcc.: *0.92218* Mean BAcc: *0.78140* Mean sen.: *0.38454* Mean spe.: *0.87325* *Avg. time: 1.144*
H Orig					Best BAcc.: *0.91348* Mean BAcc: *0.80163* Mean sen.: *0.48664* Mean spe.: *0.86596* *Avg. time: 0.5639*
H Mod					Best BAcc.: *0.97414* Mean BAcc: *0.87874* Mean sen.: *0.64856* Mean spe.: *0.92610* *Avg. time (C): 1.9026* *Avg. time (B): 2.6005*

Table 8: Segmentation performance results for image "d" using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
RGB					Best BAcc.: *0.96009* Mean BAcc: *0.88859* Mean sen.: *0.80937* Mean spe.: *0.92209* *Avg. time: 3.97900*
n*RGB*					Best BAcc.: *0.94253* Mean BAcc: *0.75979* Mean sen.: *0.05137* Mean spe.: *0.98592* *Avg. time: 5.96510*
Lab					Best BAcc.: *0.96159* Mean BAcc: *0.89942* Mean sen.: *0.82844* Mean spe.: *0.92873* *Avg. time: 3.47550*
ab					Best BAcc.: *0.97867* Mean BAcc: *0.82215* Mean sen.: *0.85910* Mean spe.: *0.81874* *Avg. time: 1.8572*
YCrCb					Best BAcc.: *0.96228* Mean BAcc: *0.88119* Mean sen.: *0.79477* Mean spe.: *0.91842* *Avg. time: 2.36320*
CrCb					Best BAcc.: *0.97951* Mean BAcc: *0.81375* Mean sen.: *0.85651* Mean spe.: *0.80762* *Avg. time: 1.94140*
HSV					Best BAcc.: *0.94228* Mean BAcc: *0.78670* Mean sen.: *0.41777* Mean spe.: *0.86373* *Avg. time: 2.6595*
HS					Best BAcc.: *0.91606* Mean BAcc: *0.78668* Mean sen.: *0.41516* Mean spe.: *0.86913* *Avg. time: 2.21520*
H Orig					Best BAcc.: *0.94253* Mean BAcc: *0.78293* Mean sen.: *0.27400* Mean spe.: *0.93489* *Avg. time: 2.0863*

operations involve a significant amount of computation, especially when the clusters need to be matched as mentioned in Section 3. It is worth mentioning that, using different cluster matching procedures, the total time can decrease, but this study is out of scope of the present work.

Performing a comparison against the time observed for GMM, it can be seen that the execution time is usually higher than any K-means tests executed (including the time needed for the proposed method). This can be explained considering that the EM algorithm used in GMM involves several operations that require a considerable amount of time.

5. Conclusions

Even though nowadays deep learning techniques and complex machine learning algorithms are being used with significant success, the unsupervised learning algorithms

Table 9: Segmentation performance results for image "e" using K-means++.

K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
RGB					Best BAcc.: *0.99148* Mean BAcc: *0.85533* Mean sen.: *0.55745* Mean spe.: *0.92899* *Avg. time: 1.89000*
n*RGB*					Best BAcc.: *0.99373* Mean BAcc: *0.94526* Mean sen.: *0.85293* Mean spe.: *0.95088* *Avg. time: 0.93870*
Lab					Best BAcc.: *0.98950* Mean BAcc: *0.85747* Mean sen.: *0.61804* Mean spe.: *0.92534* *Avg. time: 2.7201*
ab					Best BAcc.: *0.99226* Mean BAcc: *0.9104* Mean sen.: *0.78740* Mean spe.: *0.91014* *Avg. time: 1.02559*
YCrCb					Best BAcc.: *0.99224* Mean BAcc: *0.86661* Mean sen.: *0.67054* Mean spe.: *0.92761* *Avg. time: 1.31560*
CrCb					Best BAcc.: *0.99250* Mean BAcc: *0.89663* Mean sen.: *0.80508* Mean spe.: *0.92476* *Avg. time: 1.0642*
HSV					Best BAcc.: *0.99258* Mean BAcc: *0.88251* Mean sen.: *0.64855* Mean spe.: *0.92107* *Avg. time: 1.71349*
HS					Best BAcc.: *0.99258* Mean BAcc: *0.92937* Mean sen.: *0.80617* Mean spe.: *0.93624* *Avg. time: 1.1482*
H Orig					Best BAcc.: *0.99241* Mean BAcc: *0.84233* Mean sen.: *0.58515* Mean spe.: *0.87480* *Avg. time: 0.58769*
H Mod					Best BAcc.: ***0.99601*** Mean BAcc: ***0.95133*** Mean sen.: *0.78958* Mean spe.: *0.94632* *Avg. time (C): 1.3908* *Avg. time (B): 2.28200*

Table 10: Segmentation performance results for image "e" using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
RGB					Best BAcc.: *0.98543* Mean BAcc: *0.87761* Mean sen.: *0.56276* Mean spe.: *0.94142* *Avg. time: 4.03769*
n*RGB*					Best BAcc.: *0.99400* Mean BAcc: *0.95521* Mean sen.: *0.89530* Mean spe.: *0.96174* *Avg. time: 3.09780*
Lab					Best BAcc.: *0.97908* Mean BAcc: *0.86823* Mean sen.: *0.65741* Mean spe.: *0.92706* *Avg. time: 4.4489*
ab					Best BAcc.: *0.99123* Mean BAcc: *0.90806* Mean sen.: *0.79366* Mean spe.: *0.94410* *Avg. time: 1.9395*
YCrCb					Best BAcc.: *0.98606* Mean BAcc: *0.86326* Mean sen.: *0.68289* Mean spe.: *0.92291* *Avg. time: 2.966*
CrCb					Best BAcc.: *0.99088* Mean BAcc: *0.90607* Mean sen.: *0.84163* Mean spe.: *0.93257* *Avg. time: 2.29639*
HSV					Best BAcc.: *0.99261* Mean BAcc: *0.89050* Mean sen.: *0.69485* Mean spe.: *0.93121* *Avg. time: 2.5547*
HS					Best BAcc.: *0.99259* Mean BAcc: *0.93409* Mean sen.: *0.81547* Mean spe.: *0.94257* *Avg. time: 2.22500*
H Orig					Best BAcc.: *0.99252* Mean BAcc: *0.85474* Mean sen.: *0.63664* Mean spe.: *0.86330* *Avg. time: 2.3193*

are still an attractive option. The advantage of unsupervised techniques like K-means resides in the fact that they require no training; the implementation is relatively simple and they do not require excessive computational resources.

The exposed paper presents a modified version of the *HSV* model, which is merely a different presentation of the *H* component applying small changes. It is important to mention that these changes are meant to be used to help the K-means algorithm and not to be used as a new color model or for different purposes where the effect of the changes could lead to unexpected results. A proper study should be performed before using the changes in other cases or applications.

Table 11: Segmentation performance results for image "f" using K-means++.

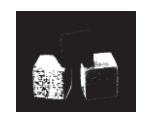
K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
RGB					Best BAcc.: *0.99323* Mean BAcc: *0.94312* Mean sen.: *0.81053* Mean spe.: *0.94599* *Avg. time: 2.2399*
n*RGB*					Best BAcc.: *0.98536* Mean BAcc: *0.89529* Mean sen.: *0.59818* Mean spe.: *0.88083* *Avg. time: 1.34869*
Lab					Best BAcc.: *0.99398* Mean BAcc: *0.95669* Mean sen.: *0.85852* Mean spe.: *0.95288* *Avg. time: 1.33930*
ab					Best BAcc.: ***0.99830*** Mean BAcc: *0.94054* Mean sen.: *0.78114* Mean spe.: *0.92947* *Avg. time: 1.3225*
YCrCb					Best BAcc.: *0.99438* Mean BAcc: *0.94116* Mean sen.: *0.80296* Mean spe.: *0.94756* *Avg. time: 1.143*
CrCb					Best BAcc.: *0.99645* Mean BAcc: *0.91684* Mean sen.: *0.70957* Mean spe.: *0.90729* *Avg. time: 1.19459*
HSV					Best BAcc.: *0.96321* Mean BAcc: *0.90030* Mean sen.: *0.67543* Mean spe.: *0.94132* *Avg. time: 2.8174*
HS					Best BAcc.: *0.99215* Mean BAcc: *0.89926* Mean sen.: *0.65391* Mean spe.: *0.94095* *Avg. time: 1.1508*
H Orig					Best BAcc.: *0.98262* Mean BAcc: *0.89173* Mean sen.: *0.50620* Mean spe.: *0.93380* *Avg. time: 0.68540*
H Mod					Best BAcc.: *0.99826* Mean BAcc: ***0.98679*** Mean sen.: *0.91741* Mean spe.: *0.99612* *Avg. time (C): 1.84471* *Avg. time (B): 3.11008*

Table 12: Segmentation performance results for image "f" using GMM.

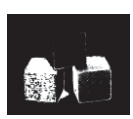
GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
RGB					Best BAcc.: *0.99634* Mean BAcc: *0.93741* Mean sen.: *0.80034* Mean spe.: *0.95122* *Avg. time: 5.87*
n*RGB*					Best BAcc.: *0.98807* Mean BAcc: *0.86814* Mean sen.: *0.49632* Mean spe.: *0.84997* *Avg. time: 3.437*
Lab					Best BAcc.: *0.99766* Mean BAcc: *0.95731* Mean sen.: *0.86890* Mean spe.: *0.95453* *Avg. time: 3.38859*
ab					Best BAcc.: *0.99832* Mean BAcc: *0.91859* Mean sen.: *0.69350* Mean spe.: *0.90564* *Avg. time: 2.8533*
YCrCb					Best BAcc.: *0.99738* Mean BAcc: *0.94940* Mean sen.: *0.84076* Mean spe.: *0.95318* *Avg. time: 3.4557*
CrCb					Best BAcc.: *0.99872* Mean BAcc: *0.92172* Mean sen.: *0.76444* Mean spe.: *0.91761* *Avg. time: 2.2925*
HSV					Best BAcc.: *0.93728* Mean BAcc: *0.88837* Mean sen.: *0.68992* Mean spe.: *0.93274* *Avg. time: 4.0777*
HS					Best BAcc.: *0.97601* Mean BAcc: *0.89289* Mean sen.: *0.59888* Mean spe.: *0.94690* *Avg. time: 2.65219*
H Orig					Best BAcc.: *0.98304* Mean BAcc: *0.91677* Mean sen.: *0.75014* Mean spe.: *0.927527* *Avg. time: 1.69729*

Table 13: Final averages from all the tests for all the color spaces for K-means++.

Color model	Mean BAcc	Best mean BAcc	Worst mean BAcc	Mean sen.	Mean spe.
RGB	0.92801	0.85683	0.77453	0.71207	0.88839
n*RGB*	0.94225	0.87095	0.77021	0.64817	0.87594
Lab	0.93251	0.87306	0.78774	0.74322	0.90118
ab	0.96864	0.90492	0.82715	0.8005	0.90480
YCrCb	0.9260	0.86462	0.76710	0.73808	0.89486
CrCb	0.95407	0.87833	0.77657	0.75341	0.87851
HSV	0.95070	0.87095	0.76136	0.67054	0.91871
HS	0.94426	0.85733	0.76829	0.64076	0.90818
H Orig	0.93414	0.84049	0.70521	0.5838	0.89377
H Mod	**0.97581**	**0.93520**	**0.82823**	**0.83216**	**0.94343**

Table 14: Final averages from all the tests for all the color spaces for GMM.

Color model	Mean BAcc	Best mean BAcc	Worst mean BAcc	Mean sen.	Mean spe.
RGB	0.88575	0.90196	0.85548	0.74395	0.92508
n*RGB*	0.87772	0.89974	0.95050	0.53360	0.9254
Lab	0.89226	0.90416	0.86582	0.74341	**0.92888**
ab	**0.90574**	**0.93919**	**0.90347**	**0.77589**	0.91127
YCrCb	0.88432	0.90150	0.86464	0.74329	0.92262
CrCb	0.89592	0.91717	0.85903	0.77130	0.90246
HSV	0.87142	0.89272	0.83729	0.65110	0.92527
HS	0.86954	0.87888	0.81174	0.62142	0.92431
H	0.86954	0.87888	0.81174	0.62142	0.62142

Table 15: Wilcoxon tests between the proposed improvement and the K-means and GMM algorithms using *ab*, CrCb, *H*, *HS*, and *HSV* color models.

	Versus *ab*	Versus CrCb	Versus *H*	Versus *HS*	Versus *HSV*
K-Means	P val.: $2.2507e-07$	P val.: $7.3529e-10$	P val.: $1.1304e-11$	P val.: $8.7888e-12$	P val.: $1.3146e-11$
GMM	P val.: $7.7979e-07$	P val.: $7.3529e-10$	P val.: $1.1887e-11$	P val.: $1.0752e-11$	P val.: $2.5107e-11$

Table 16: Wilcoxon tests between the proposed improvement and the K-means and GMM algorithms using *Lab*, n*RGB*, *RGB*, and YCrCb color models.

	Versus *Lab*	Versus n*RGB*	Versus *RGB*	Versus YCrCb
K-Means	P val.: $1.2802e-08$	P val.: $5.5030e-11$	P val.: $1.6585e-09$	P val.: $1.0108e-09$
GMM	P val.: $4.2636e-08$	P val.: $8.9224e-11$	P val.: $2.8298e-08$	P val.: $1.9830e-09$

The changes in the H component allow and force using bagging in the resulting K-means++ clusters. So, the usage of bagging procedure and the chromatic/achromatic separation in the H component improve the outcome from the color segmentation.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] D. Bone, C.-C. Lee, T. Chaspari, J. Gibson, and S. Narayanan, "Signal processing and machine learning for mental health research and clinical applications [Perspectives]," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 195-196, 2017.

[2] Q. Zhang, X. Zeng, W. Hu, and D. Zhou, "A machine learning-empowered system for long-term motion-tolerant wearable monitoring of blood pressure and heart rate with Ear-ECG/PPG," *IEEE Access*, vol. 5, pp. 10547–10561, 2017.

[3] J. Zhang, R. L. Lafta, X. Tao et al., "Coupling a fast fourier transformation with a machine learning ensemble model to support recommendations for heart disease patients in a telehealth environment," *IEEE Access*, vol. 5, pp. 10674–10685, 2017.

[4] J. Wu, Y. Xiao, C. Xia et al., "Identification of biomarkers for predicting lymph node metastasis of stomach cancer using clinical DNA methylation data," *Disease Markers*, vol. 2017, Article ID 5745724, 7 pages, 2017.

[5] J. K. Kim and S. Kang, "Neural network-based coronary heart disease risk prediction using feature correlation analysis," *Journal of Healthcare Engineering*, vol. 2017, Article ID 2780501, pp. 1–13, 2017.

[6] J. Zhang, J. Xiao, J. Wan et al., "A parallel strategy for convolutional neural network based on heterogeneous cluster for mobile information system," *Mobile Information Systems*, vol. 2017, Article ID 3824765, 12 pages, 2017.

[7] H. Chen, B. Jiang, and N. Lu, "Data-driven incipient sensor fault estimation with application in inverter of high-speed railway," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8937356, 13 pages, 2017.

[8] Y. Liu, Y. Liu, J. Liu et al., "A MapReduce based high performance neural network in enabling fast stability assessment of power systems," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4030146, 12 pages, 2017.

[9] M. Shafiq, X. Yu, A. A. Laghari, and D. Wang, "Effective feature selection for 5G IM applications traffic classification," *Mobile Information Systems*, vol. 2017, Article ID 6805056, pp. 1–12, 2017.

[10] R. Eskandarpour and A. Khodaei, "Machine learning based power grid outage prediction in response to extreme events," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3315-3316, 2017.

[11] Y. Yang, M. Yang, S. Huang, Y. Que, M. Ding, and J. Sun, "Multifocus image fusion based on extreme learning machine and human visual system," *IEEE Access*, vol. 5, pp. 6989–7000, 2017.

[12] J. Kremer, K. Stensbo-Smidt, F. Gieseke, K. S. Pedersen, and C. Igel, "Big universe, big data: machine learning and image analysis for astronomy," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 16–22, 2017.

[13] R. M. Mehmood, R. Du, and H. J. Lee, "Optimal feature selection and deep learning ensembles method for emotion

recognition from human brain EEG sensors," *IEEE Access*, vol. 5, pp. 14797–14806, 2017.

[14] Y. Xia, Z. Ji, A. Krylov, H. Chang, and W. Cai, "Machine learning in multimodal medical imaging," *BioMed Research International*, vol. 2017, Article ID 1278329, 2 pages, 2017.

[15] H. Liu, C. Zhang, and D. Huang, "Extreme learning machine and moving least square regression based solar panel vision inspection," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 7406568, 10 pages, 2017.

[16] G. Wen, H. Li, J. Huang, D. Li, and E. Xun, "Random deep belief networks for recognizing emotions from speech signals," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 1945630, 9 pages, 2017.

[17] R. Narayan, V. P. Singh, and S. Chakraverty, "Quantum neural network based machine translator for hindi to english," *The Scientific World Journal*, vol. 2014, Article ID 485737, 8 pages, 2014.

[18] S. Rosenblum and G. Dror, "Identifying developmental dysgraphia characteristics utilizing handwriting classification methods," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 2, pp. 293–298, 2017.

[19] L. Likforman-Sulem, A. Esposito, M. Faundez-Zanuy, S. Clemencon, and G. Cordasco, "EMOTHAW: A novel database for emotional state recognition from handwriting and drawing," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 2, pp. 273–284, 2017.

[20] B. Zhou, "Statistical machine translation for speech: A perspective on structures, learning, and decoding," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1180–1202, 2013.

[21] "The Berkeley Segmentation Dataset and Benchmark, BSDS500," https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[22] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[23] A. Sironi, E. Turetken, V. Lepetit, and P. Fua, "Multiscale centerline detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1327–1341, 2016.

[24] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1558–1570, 2015.

[25] S. Zhu, D. Cao, S. Jiang, Y. Wu, and P. Hu, "Fast superpixel segmentation by iterative edge refinement," *IEEE Electronics Letters*, vol. 51, no. 3, pp. 230–232, 2015.

[26] J. Sigut, F. Fumero, O. Nuñez, and M. Sigut, "Automatic marker generation for watershed segmentation of natural images," *IEEE Electronics Letters*, vol. 50, no. 18, pp. 1281–1283, 2014.

[27] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 128–140, 2017.

[28] L. Velho, A. Frery, and J. Gomes, *Image Processing for computer graphics and vision*, Spring-er, Second edition, 2009.

[29] A. Hanbury and J. Serra, "A 3D-polar coordinate colour representation suitable for image analysis. pattern recognition and image processing Group Technical Report 77," Tech. Rep., Vienna University of Technology, Vienna, Austria, 2003.

[30] CIE, *Commission internationale de l'Eclairage proceedings*, Cambridge University Press, Cambridge, UK, 1931.

[31] G. Finlayson and R. Xu, "Illuminant and gamma comprehensive normalisation in log RGB space," *Pattern Recognition Letters*, vol. 24, no. 11, pp. 1679–1690, 2003.

[32] T. Kuremoto, Y. Kinoshita, L.-B. Feng, S. Watanabe, K. Kobayashi, and M. Obayashi, "A gesture recognition system with retina-V1 model and one-pass dynamic programming," *Neurocomputing*, vol. 116, pp. 291–300, 2013.

[33] E. Blanco, M. Mazo, L. M. Bergasa, S. Palazuelos, and M. Marrón, "A method to increase class separation in the HS plane for color segmentation applications," in *Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP '07)*, Spain, 2007.

[34] H. Yang, X. Wang, Q. Wang, and X. Zhang, "LS-SVM based image segmentation using color and texture information," *Journal of Visual Communication and Image Representation*, vol. 23, no. 7, pp. 1095–1112, 2012.

[35] S. Pharadornpanitchakul, A. Duangchit, and R. Chaisricharoen, "Enhanced danger detection of headlight through vision estimation and vector magnitude," in *Proceedings of the 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE '14)*, Thailand, March 2014.

[36] W.-M. Liu, L.-H. Wang, and Z.-F. Yang, "Application of self adapts to RGB threshold value for robot soccer," in *Proceedings of the 2010 International Conference on Machine Learning and Cybernetics (ICMLC '10)*, pp. 704–707, China, July 2010.

[37] S. M. Khaled, M. S. Islam, M. G. Rabbani et al., "Combinatorial color space models for skin detection in sub-continental human images," in *Proceedings of the Visual Ibnformatics, First International Visual Informatics Conference (IVIC '09)*, pp. 532–542, 2009.

[38] A. Vadivel, S. Suralb, and A. Majumdar, "An integrated color and intensity co-occurrence matrix," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 974–983, 2007.

[39] R. Mente, B. V. Dhandra, and G. Mukarambi, "Color image segmentation and recognition based on shape and color features," *International Journal of Computer Science Engineering (IJCSE)*, vol. 3, no. 1, pp. 2319–7323, 2014.

[40] K. Murphy, *Machine Learning A probabilistic Perspective*, MIT Press, Cambridge Massa-chusetts, 2012.

[41] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[42] S. P. Lloyd, "Least squares quantization in PCM," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[43] D. Arthur and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

[44] S. Arumugadevi and V. Seenivasagam, "Color image segmentation using feedforward neural networks with FCM," *International Journal of Automation and Computing*, vol. 13, no. 5, pp. 491–500, 2016.

[45] C. Pan, D. S. Park, Y. Yang, and H. M. Yoo, "Leukocyte image segmentation by visual attention and extreme learning machine," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1217–1227, 2012.

[46] S. W. Oh and S. J. Kim, "Approaching the computational color constancy as a classification problem through deep learning," *Pattern Recognition*, vol. 61, pp. 405–416, 2017.

[47] Q. Sang, Z. Lin, and S. T. Acton, "Learning automata for image segmentation," *Pattern Recognition Letters*, vol. 74, pp. 46–52, 2016.

[48] M. Sridharan and P. Stone, "Structure-based color learning on a mobile robot under changing illumination," *Autonomous Robots*, vol. 23, no. 3, pp. 161–182, 2007.

[49] K. Kim, C. Oh, and K. Sohn, "Non-parametric human segmentation using support vector machine," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 2, pp. 150–158, 2016.

[50] A. Lucchi, P. Marquez-Neila, C. Becker et al., "Learning structured models for segmentation of 2-D and 3-D imagery," *IEEE Transactions on Medical Imaging*, vol. 34, no. 5, pp. 1096–1110, 2015.

[51] M. Gong, Y. Qian, and L. Cheng, "Integrated foreground segmentation and boundary matting for live videos," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1356–1370, 2015.

[52] A. Pratondo, C. Chui, and S. Ong, "Integrating machine learning with region-based active contour models in medical image segmentation," *Journal of Visual Communication and Image Representation*, 2016.

[53] X.-Y. Wang, Q.-Y. Wang, H.-Y. Yang, and J. Bu, "Color image segmentation using automatic pixel classification with support vector machine," *Neurocomputing*, vol. 74, no. 18, pp. 3898–3911, 2011.

[54] H. G. Li, G. Q. Wu, X. G. Hu, J. Zhang, L. Li, and X. Wu, "K-Means Clustering with Bagging and MapReduce," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, pp. 1–8, 2011.

[55] D.-C. Tseng and C.-H. Chang, "Color segmentation using perceptual attributes," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition (IAPR '92)*, pp. 228–231, Netherlands, September 1992.

[56] A. Azad, S. Pyne, and A. Pothen, "Matching phosphorylation response patterns of antigen-receptor-stimulated T cells via flow cytometry," *BMC Bioinformatics*, vol. 13, p. S10, 2012.

[57] A. Azad, B. Rajwa, and A. Pothen, "Immunophenotype discovery, hierarchical organization, and template-based classification of flow cytometry samples," *Frontiers in Oncology*, 2016.

[58] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[59] FreeImages, A web based free photography stock site, http://www.freeimages.co.uk.

[60] K. Brodersen, C. S. Ong, K. Stephan, and J. Buhmann, "The balanced accuracy and its posterior distribution," in *Proceedings of the In 20th international conference on pattern recognition (ICPR)*, pp. 3121–3124, 2010.

[61] L. C. Neto, G. Ramalho, J. F. S. Neto, R. Veras, and F. N. Medeiros, "An unsupervised coarse-to-fine algorithm for blood vessel segmentation in fundus images," *Expert Systems with Applications*, vol. 78, 2017.

[62] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.

[63] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 1524286, 19 pages
https://doi.org/10.1155/2018/1524286

Research Article

Segmentation of Melanoma Skin Lesion Using Perceptual Color Difference Saliency with Morphological Analysis

Oludayo O. Olugbara, Tunmike B. Taiwo, and Delene Heukelman

ICT and Society Research Group, Durban University of Technology, P.O. Box 1334, Durban 4000, South Africa

Correspondence should be addressed to Oludayo O. Olugbara; oludayoo@dut.ac.za

Received 29 September 2017; Accepted 2 January 2018; Published 13 February 2018

Academic Editor: Erik Cuevas

The prevalence of melanoma skin cancer disease is rapidly increasing as recorded death cases of its patients continue to annually escalate. Reliable segmentation of skin lesion is one essential requirement of an efficient noninvasive computer aided diagnosis tool for accelerating the identification process of melanoma. This paper presents a new algorithm based on perceptual color difference saliency along with binary morphological analysis for segmentation of melanoma skin lesion in dermoscopic images. The new algorithm is compared with existing image segmentation algorithms on benchmark dermoscopic images acquired from public corpora. Results of both qualitative and quantitative evaluations of the new algorithm are encouraging as the algorithm performs excellently in comparison with the existing image segmentation algorithms.

1. Introduction

The purpose of this study is to test the performance of perceptual color difference saliency algorithm for segmentation of melanoma skin lesion in dermoscopic images. Melanoma is a cancer of pigment that produces melanocytes and is one of the most serious, complex, aggressive, and fatal forms of all skin cancer related diseases [1]. It is a cancerous skin disease that typically results from environmental factors such as exposure to sunlight [2]. It originates from the parts of the body such as skin, eyes, brain, spinal cord, and mucous membrane containing melanocytes. The ability to spread widely to other parts of the body is a unique characteristic that makes melanoma one of the deadliest skin cancer diseases. Its prevalence is rapidly increasing across the world, as recorded death cases of its patients continue to annually escalate [1, 3, 4]. Prevention which is better than cure and early detection are recommended as the best strategies for improving outcomes in melanoma and to reduce the induced mortality rate of the disease because treatment at later stages can be hard [1, 4–6].

Digital dermoscopy is a widely used noninvasive tool that combines optical magnification and special illumination techniques to render an improved dermoscopic image for clinical diagnosis of melanoma. Dermatologists have regularly applied this tool for several decades to analyze the surface structure of human skin that is invisible to the naked eyes [7, 8]. However, this diagnostic process is time-consuming and highly subjective and it requires a great deal of experience from a dermatologist [4, 8]. Due to the complexity of melanoma treatment at later stages, researchers are attempting to develop an efficient noninvasive automated computer aided system to make its diagnosis faster and easily accessible to nonexpert practitioners [4, 8–11]. Such an automated system relies heavily on reliable segmentation of skin lesion, pertinent extraction of skin lesion features, and effective classification of skin lesion using the extracted features [11].

This study focuses on segmentation of melanoma skin lesion in dermoscopic images because other subsequent diagnostic stages heavily depend on its output [7, 8, 12]. Moreover, segmentation is one of the central stages for computer aided diagnosis of melanoma with dermoscopic images [13]. The automatic segmentation of skin lesion is particularly challenging because of the possible presence of undesirable factors in the form of skin hairs, specular

reflections, variegated coloring, weak edges, low contrast, irregular and fuzzy borders, marker ink, color chart, ruler marks, dark corners, skin lines, blood vessels, and air or oil bubbles [10, 14–16].

The method of saliency based segmentation has emerged as an important tool for medical image analysis because of its capability to identify salient objects in images [17, 18]. Its application in computer vision is largely inspired by the findings that human vision perception has a higher probability to focus on the part of an image that carries useful information [17, 19]. The cognitive properties of visual saliency incorporated into the conventional saliency segmentation methods are based upon local or global visual rarity such as contrast prior, color prior, brightness prior, and center prior. Contrast prior is one of the frequently used visual rarities which assumes that color contrast between object and background is usually high to detect visual saliency. Color prior assumes that background has uniform color while salient object colors are variegated. Brightness prior assumes that the brightness of background is higher than that of the salient object [13]. However, while methods based on these cognitive properties have performed well on certain images, they can fail to accurately detect salient objects that share uniform homogeneity with the background and for salient objects that touch the image border slightly [20]. The center prior assumes that images are acquired such that a salient object is often framed near the image center while background is distributed in the borders. However, salient objects in many images often appear off the image centers which makes the center prior map incorrectly suppress salient objects far off the image centers and highlight certain background regions near the image center [21, 22].

The methodology of the perceptual color difference saliency segmentation algorithm reported in this paper consists of four essential stages. They are color image transformation, luminance image enhancement, salient pixel computation, and image artifact filtering. The main contributions of this paper are as follows:

(a) The new saliency algorithm effectively segment melanoma skin lesion in dermoscopic images through the aggregation of color feature of a background pixel and color feature of an object pixel.

(b) The new saliency algorithm uses a simple decision rule that does not follow the conventional thresholding methods for binary segmentation of melanoma skin lesion in grayscale dermoscopic saliency map.

(c) The outputs computed by the new saliency algorithm are qualitatively evaluated using test images acquired from public medical corpora and quantitatively evaluated in terms of precision, recall, accuracy, and dice which are widely used statistical metrics for evaluating binary segmentation results.

(d) A detailed evaluation against other existing saliency and nonsaliency benchmark algorithms is performed that provided a fair comparison to demonstrate the performance of the new saliency algorithm.

2. Related Study

The discussion of related studies is organized in four dimensions in order to show currency, originality, relevance, and relatedness of this study to the previous research and to justify the suitability of the study methodology. These dimensions are nonsaliency based segmentation, saliency based segmentation, color image models, and perceptual color difference.

2.1. Nonsaliency Based Segmentation. Many image segmentation algorithms have been developed to deal with the complex problem of segmenting skin lesion from the healthy skin. They can be appositely categorized into region, edge, and pixel based methods [8, 23]. Region based methods such as the modified JSEG [12], region growing [24], modified watershed [25], and statistical region merging [26] group image pixels into clusters and maintain connectivity between cluster pixels. Edge based methods such as zero-crossing of Laplacian-of-Gaussian [27] and geodesic active contour [28] are aimed at detecting discontinuities in image pixel intensity values [29]. Pixel based methods group similar pixels as belonging to a homogenous cluster that corresponds to an object or part of an object [30] and are widely applied because of their inherent simplicity and robustness [31, 32]. Thresholding and clustering algorithms are archetypes of the pixel based methods that have been applied for segmentation of skin lesion [9, 33]. Research has revealed that existing segmentation algorithms achieve good results when dermoscopic images exhibit good contrast and in the absence of undesirable factors. However, they often lack robustness for low contrast images and may not perform well on complex images that exhibit significant volume of undesirable artifacts [4, 7].

2.2. Saliency Based Segmentation. Saliency based methods have received a great deal of attention in cognitive science, computer vision, and image processing [34] and they have been applied to image segmentation [34–36]. However, the application of saliency methods for segmentation of skin lesion is relatively new [13, 17, 37]. Saliency segmentation computes the most informative region in an image based on human vision perception such that salient and nonsalient parts become foreground region (skin lesion) and background region (healthy skin), respectively. It has been alluded that a good saliency segmentation model should satisfy three essential criteria of good segmentation, high resolution, and computational efficiency [38]. Good segmentation means that the probability of missing real salient regions and falsely marking background regions as salient regions should be low. High resolution means that saliency maps should possess high resolution to accurately locate salient objects and retain original image information. Computational efficiency means that saliency based segmentation methods should rapidly detect salient regions with less complexity. This paper reports a less complicated saliency based image segmentation algorithm that achieves good performance and generates a high resolution saliency map containing much salient pixels.

Many saliency based segmentation algorithms reported in literature are based on color feature of an input image,

but they significantly differ in their computational strategies. Color is one of the most important cues that people use extensively to identify real world objects. It is widely used in medical image analysis for screening dermoscopic images in order to discriminate between healthy skin and unhealthy skin regions [39]. The basic assumption in most cases of dermoscopic image analysis is that the lighter shade of color corresponds to healthy skin, while unhealthy skin possesses different color distribution that differs from the healthy skin [40]. Itti et al. [41] introduced a computational saliency segmentation model based on color, intensity, and texture features for rapid scene analysis. A method to segment salient regions in video sequences based on the application of luminance information has been discussed by Zhai and Shah [42]. The spectral residual approach based on Fourier transform has been reported [43] with several improvements to segment salient objects in images [34, 44]. However, many of the improved saliency segmentation algorithms still face difficulty when salient objects share similar color features with the background pixels. These algorithms often lack the ability to effectively handle complicated images with low contrast [18, 20, 37]. Complementing the methods of saliency computation with other useful analysis methods such as the morphological analysis can significantly improve image segmentation results. The hybrid segmentation of skin lesion in dermoscopic images using wavelet transform along with morphological analysis has been reported [1], while segmentation using saliency combined with Otsu threshold has been discussed [13].

2.3. Color Image Models. The importance of selecting a suitable color model for color image segmentation has been emphasized in the literature [43–45]. Since the appearance of skin in an image is illumination dependent, different color models are widely used for skin lesion analysis with the objective of finding a color model where the color of skin lesion is invariant to illumination conditions. Researchers have attempted to identify the most discriminating and effective color models for processing skin lesion in dermoscopic images. The decomposition of a color image into constituent components is a good analysis technique for medical diagnosis because essential information is conveyed in the color of an image [46].

The segmentation of skin lesion in dermoscopic images using wavelet networks considers the *R*, *G*, and *B* channels of the *RGB* color model as the network inputs and network structure formation [47]. The segmentation of skin lesion in dermoscopic images based on wavelet transform along with morphological analysis found the *B* channel of the *RGB* color model to give better performance than grayscale conversion [1]. The segmentation of skin lesion based on the *RGB*, normalized *RGB*, YIQ, and $I_1I_2I_3$ color models has been reported to give good results for *Q* channel of YIQ and I_3 channel of $I_1I_2I_3$ [48]. It has been found by experimental comparison of HSI, CMY, YCbCr, and CIE $L^*a^*b^*$ color models that the "*H*" channel of the HSI and "a^*" channel of CIE $L^*a^*b^*$ gave good results for segmentation of skin lesion [49].

This study applies the CIE $L^*a^*b^*$ color model instead of the widely used *RGB* color model for segmentation of melanoma skin lesion. The color model is perceptually uniform; it separates luminance and chrominance information and comes with different intrinsic human visual perception based color difference formulae that are useful for saliency computation [11]. However, the *RGB* color model is not perceptually uniform and it does not separate luminance and chrominance information because of the high correlated nature of its channels [50]. The information in all the channels of the color image is utilized in this study to ensure that no useful color information is otherwise discarded.

2.4. Perceptual Color Difference. Color analysis is an important topic in different studies such as prosthodontics, aesthetics, and dental materials science where color quantification is used to gain the understanding of scientific data [51]. The clinical relevance of these studies is highly dependent on how much color change is considered perceptible. The determination of color difference has been proposed in the literature to improve the correlation between color measurement and human vision perception. The measurement of color difference is considered an important problem for color analysis. The practical application of color difference is mostly found in clinical dentistry, where the ability to reproduce the exact shade of natural teeth using restorative dental material is considered a challenging problem [51–56]. The other useful applications of color difference include content-based retrieval [57], quality inspection of food [58, 59], and video compression [60], but it has not been well explored for saliency based segmentation of melanoma skin lesion in dermoscopic images.

There are diverse color difference formulae which are designed to provide a quantification of the correlation between the computed and perceived color differences. The most widely used formula of them includes the CIELAB and CIELUV recommended by the Commission Internationale de l'Eclairage (CIE). The CIEDE2000 color difference formula is applied in this study because it is the recent CIE recommendation with more consistent trends in lightness and hue angle dependencies [54]. It was designed to improve the earlier color difference formulae and correction between the computed and perceived color differences. It incorporates a term that accounts for the interaction between Chroma and hue differences, a modification of the coordinate that affects colors with low Chroma and parameters that account for the influence of illumination and vision conditions in color difference [54]. In addition, it reflects the color differences perceived by the human eye and is generally recommended for evaluating color difference thresholds in dental research and in vivo instrumental color analysis [56].

3. Material and Methods

The discussion of the experimental images, perceptual color difference saliency, and algorithm implementation are presented in this section.

3.1. Experimental Images. Dermoscopic images used for experimentation in this study are acquired from the International Symposium on Biomedical Imaging (ISBI 2016)

challenge [61] and Pedro Hispano Hospital (PH2) corpora [62]. These corpora particularly inspired us because they contain numerous challenging dermoscopic images and support the development of automated algorithms for the analysis of skin lesion. A dermoscopic image is considered to be "challenging" if one or more undesirable factors are present in the image. These challenging images are usually excluded from test images in the previous research in order to ensure accurate border segmentation [12, 63].

3.2. Perceptual Color Difference Saliency. The essential stages of the methodology of perceptual color difference saliency are color image transformation, luminance image enhancement, salient pixel computation, and image artifact filtering.

3.2.1. Color Image Transformation. The input RGB color image of $M \times N \times 3$ dimensions has values in the range $[0, 1]$, where M and N are the number of rows and columns, respectively. The *RGB* image is transformed into CIE $L^*a^*b^*$ color image to achieve perceptual color image for saliency computation. The process of transforming an Adobe *RGB* color image to CIE $L^*a^*b^*$ color image is usually performed in two steps. The first step converts the Adobe *RGB* image into CIE XYZ image according to the following equation [64, 65]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.5767 & 0.2973 & 0.0270 \\ 0.1855 & 0.6273 & 0.0706 \\ 0.1882 & 0.0752 & 0.9912 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}, \tag{1}$$

where r, g, and b are defined in terms of the constant gamma value which in this study is $\gamma_c = 2.4$. The parameters $\alpha_1 = 0.055$ and $\alpha_2 = 1.055$ in (2) are added to correct the *RGB* values obtained from digital cameras to obtain the best possible calibration of the transformation model [64–66]:

$$\begin{aligned} r &= \left(\frac{R + \alpha_1}{\alpha_2}\right)^{\gamma_c}, \\ g &= \left(\frac{G + \alpha_1}{\alpha_2}\right)^{\gamma_c}, \\ b &= \left(\frac{B + \alpha_1}{\alpha_2}\right)^{\gamma_c}. \end{aligned} \tag{2}$$

In the second step of the transformation process, the CIE XYZ image is transformed to the CIE $L^*a^*b^*$ image following the ITU-R BT.709 recommendation. The transformed image serves as input to the luminance image enhancement function. The D65 illuminant is used in this study where $X_n = 0.95047$, $Y_n = 1.00000$, and $Z_n = 1.08255$ are the CIE XYZ tristimulus values of standard light source [64, 67]:

$$\begin{aligned} L^* &= 116 * \left[f\left(\frac{Y}{Y_n}\right) - \frac{16}{116}\right], \\ a^* &= 500 * \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right], \\ b^* &= 200 * \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right], \end{aligned} \tag{3}$$

where

$$f(s) = \begin{cases} s^{1/3}, & \text{if } s > \left(\frac{6}{29}\right)^3 \\ \left(\frac{841}{108}\right) * s + \frac{4}{29}, & \text{if } s \le \left(\frac{6}{29}\right)^3. \end{cases} \tag{4}$$

3.2.2. Luminance Image Enhancement. The transformation of *RGB* color image alone does not alleviate the adverse effect of illumination or low contrast. This is because an absolute separation between luminance and chrominance channels is not achievable due to high correlation between the image channels [68, 69]. It is therefore desirable to enhance luminance channel of the input image which does not change the original color of a pixel [69]. The adaptive gamma correction function has been recommended for this purpose because a fixed gamma correction function is not always desirable for all types of images. The following adaptive gamma correction function is applied in this study to enhance the luminance channel of the transformed input image [69]:

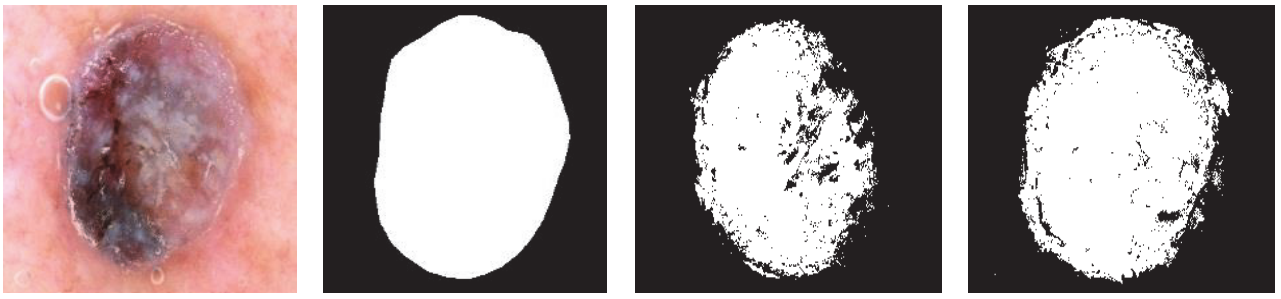
$$L_{\text{out}} = \frac{L_{\text{in}}^{\gamma_a}}{1 + H(0.5 - \mu)\left(\mu^{\gamma_a} - 1\right)\left(1 - L_{\text{in}}^{\gamma_a}\right)}. \tag{5}$$

The images L_{in} and L_{out} are input luminance and output luminance, respectively, and γ_a is the adaptive gamma correction value that controls the slope of the transformation function. The Heaviside function $H(x)$ returns a value of 1 if its argument is greater than 0; otherwise it returns a value of 0. Rahman et al. [69] gave logarithm and exponential adaptive gamma correction functions to, respectively, enhance low contrast and high contrast images. The functions gave impressive segmentation results for a number of images. However, for some high contrast images such as an image with a mean value of 0.7097, standard deviation of 0.1513, and gamma value of 1.0720, the image enhancement needs further improvement as shown in Figure 1(c). The segmentation result can be seen to improve as shown in Figure 1(d) with an increase in the gamma value from 1.0720 to 2.9212 using the product of logarithm and exponential functions introduced by Rahman et al. [69] as the gamma correction function:

$$\gamma_a = -\log_2(\sigma) \exp\left(\frac{(1 - \mu - \sigma)}{2}\right), \tag{6}$$

where σ and μ are the global standard deviation and global mean of the luminance image, respectively. The enhanced luminance image together with the chrominance images serve as input to the salient pixel computation function.

3.2.3. Salient Pixel Computation. Pixel saliency can be computed in terms of the difference of color feature with the global mean of this color feature [35, 38]. However, this method has difficulty indistinguishing similar color feature



(a) (b) (c) (d)

Figure 1: Enhancement of luminance channel using adaptive gamma correction. (a) Original image, (b) ground truth, (c) exponential gamma function enhanced image, and (d) product of logarithmic and exponential gamma functions enhanced image.

(a) (b) (c) (d)

Figure 2: Estimation of background and object pixels. (a) Input image. (b) Yellow ellipsoidal patch predominantly contains background pixels and red rectangular patch predominantly contains object pixels. (c) Input image. (d) Yellow ellipsoidal patch contains all background pixels and red rectangular patch contains all object pixels.

in background and object regions in an input image [70]. In this study, the mean of background color feature and mean of object color feature are computed instead of the global mean to correct this deficiency. The mean of background color feature can be estimated by the mean of pixel values on an ellipsoidal patch drawn close to image borders. Similarly, the mean of object color feature can be estimated by the mean of pixel values within a rectangular patch drawn close to the image center. This design principle follows the assumption of center prior [6, 10, 21, 33, 71]. However, this study applies a different computational strategy to cater for the identification of skin lesion pixels not necessarily framed near the image center.

The applied computational strategy is compactly described as follows. The background mean (m_{bl}, m_{ba}, and m_{bb}) and background standard deviation (σ_{bl}, σ_{ba}, and σ_{bb}) are computed from values of pixels on an ellipsoidal patch traced by the midpoint ellipse algorithm to achieve computational efficiency [72, 73]. Moreover, the object mean (m_{ol}, m_{oa}, and m_{ob}) is computed from values of pixels within a rectangular patch whenever the inequalities $p(x, y) < m_{bi} - \eta\sigma_{bi}$ and $p(x, y) > m_{bi} + \eta\sigma_{bi}$ are concomitantly satisfied, where $i = l, a, b$ are the image channels, $p(x, y)$ is a given pixel value, $x \in [1, M]$, $y \in [1, N]$, $M \times N$ is the image dimension, and $\eta = 1.0$ standard deviation is used in this study. In addition, other values of $\eta \in [0.5, 2.0]$ can be used, but the value of $\eta = 1.0$ has been experimentally found to give good segmentation results in this study. Figure 2 shows the diagrammatic illustration of image patches used for the computation of mean values. The yellow ellipse represents a set of color pixels that is used for the computation of background mean. The red solid rectangle represents a set of pixels that is used for the computation of object mean. It is important to note the difference between Figures 2(b) and 2(d) from the rectangular shapes. The segmentation algorithm computes object mean for those pixels within the rectangular patch that differ from background pixels following the assumption of color prior [13, 40]. In Figure 2(b), not all pixels in the rectangular patch are object pixels, but in Figure 2(d) all pixels in the rectangular patch are object pixels, hence the principal reason for the observed difference in the rectangular shapes.

The color difference of background color feature with mean of this color feature, $\Delta b(x, y)$, and color difference of object color feature with mean of this color feature, $\Delta o(x, y)$, are computed for each pixel to preserve spatial information. These two measures are then aggregated to create a grayscale saliency map, $S = \{s(x, y)\}$, whose entry $s(x, y)$ can be determined as follows:

$$s(x, y) = \frac{255 \times \Delta b(x, y)}{\Delta b(x, y) + \Delta o(x, y) + 1}. \quad (7)$$

The resolution of a salient pixel is determined by the degree to which every value $s(x, y)$ of the salient pixel tends to the maximum grayscale value of 255. Salient pixels are those pixels of a dermoscopic image that contain useful information for diagnosis purpose. The binary saliency map, $B = \{b(x, y)\}$, is constructed to provide high resolution and good segmentation [38]. The value $b(x, y)$ tends to 255 for a salient pixel and 0 for a nonsalient pixel according to the following simple decision rule:

$$b(x,y) = \begin{cases} 255, & \text{if } \Delta o(x,y) < \Delta b(x,y) \\ 0, & \text{if } \Delta o(x,y) \geq \Delta b(x,y). \end{cases} \tag{8}$$

In fact, (7) and (8) can be combined into one equation such that nonsalient pixels are assigned the value of 0 to realize a high resolution grayscale saliency map as follows:

$$s(x,y) = \begin{cases} \dfrac{255 \times \Delta b(x,y)}{\Delta b(x,y) + \Delta o(x,y)}, & \text{if } \Delta o(x,y) < \Delta b(x,y) \\ 0, & \text{if } \Delta o(x,y) \geq \Delta b(x,y). \end{cases} \tag{9}$$

The saliency of a pixel as measured by (7)–(9) is controlled by the value of the color difference between the object color feature and mean of this feature. Large value of $\Delta o(x, y)$ corresponds to week saliency and low value of $\Delta o(x, y)$ corresponds to strong saliency. The parameters $\Delta b(x, y)$ and $\Delta o(x, y)$ can be computed using the accurate CIEDE2000 color difference formula which is symbolically denoted in this paper by $\Delta E_{2000}(p_1, p_2)$. The color difference between two given color values $p_1(L_1, a_1, b_1)$ (pixel color feature) and $p_2(L_2, a_2, b_2)$ (mean color feature) in the CIE $L^*a^*b^*$ color model is defined as [54, 74, 75]

$$\Delta E_{2000}(p_1, p_2) = \sqrt{\left(\frac{\Delta L_p}{K_L S_L}\right)^2 + \left(\frac{\Delta C_p}{K_C S_C}\right)^2 + \left(\frac{\Delta H_p}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C_p}{K_C S_C}\right)\left(\frac{\Delta H_p}{K_H S_H}\right)}. \tag{10}$$

The parametric weighting factors K_L, K_C, and K_H are correction terms for experimental conditions, where the differential color vector components that represent the differences in lightness, Chroma, and hue are

$$\begin{aligned} \Delta L_p &= L_2 - L_1, \\ \Delta C_p &= C_{2p} - C_{1p}, \\ \Delta H_p &= 2\sqrt{C_{1p} * C_{2p}} * \sin\left[\frac{\Delta h_p}{2}\right]^0, \end{aligned} \tag{11}$$

where

$$\Delta h_p = \begin{cases} h_{2p} - h_{1p} - 360, & \text{if } h_{2p} - h_{1p} > 180 \\ h_{2p} - h_{1p} + 360, & \text{if } h_{2p} - h_{1p} < -180 \\ h_{2p} - h_{1p}, & \text{else.} \end{cases} \tag{12}$$

The rotation function R_T that accounts for the interaction between Chroma and hue differences in the blue region is mathematically expressed as

$$R_T = -2\sin\left[60\exp\left\{-\left(\frac{h_{\text{pa}} - 275}{25}\right)^2\right\}\right]^r * \sqrt{\frac{C_{\text{pa}}^7}{C_{\text{pa}}^7 + 25^7}}. \tag{13}$$

The parametric weighting functions that adjust the total color difference for variation in the location of the color difference pair in the coordinates of the color model are

$$\begin{aligned} S_L &= 1 + \frac{0.015\left(L_{\text{pa}} - 50\right)^2}{\sqrt{20 + \left(L_{\text{pa}} - 50\right)^2}}, \\ S_C &= 1 + 0.045 * C_{\text{pa}}, \\ S_H &= 1 + 0.015 * C_{\text{pa}} * \Big(1 - 0.17 * \cos\left[h_{\text{pa}} - 30\right]^r \\ &\quad + 0.24 * \cos\left[2 * h_{\text{pa}}\right]^r + 0.32 * \cos\left[3 * h_{\text{pa}} + 6\right]^r \\ &\quad - 0.20 * \cos\left[4 * h_{\text{pa}} - 63\right]^r\Big). \end{aligned} \tag{14}$$

The symbols used in the rotation and parametric weighting functions are defined in terms of the hue angle for a pair of color samples as follows:

$$\begin{aligned} L_{\text{pa}} &= \frac{L_1 + L_2}{2}, \\ C_{\text{pa}} &= \frac{C_{1p} + C_{2p}}{2}, \\ h_{\text{pa}} &= \begin{cases} h_{1p} + h_{2p}, & \text{if } C_{1p} * C_{2p} = 0 \\ \dfrac{h_{1p} + h_{2p}}{2}, & \text{if } \left|h_{2p} - h_{1p}\right| \leq 180 \\ \dfrac{h_{1p} + h_{2p} + 360}{2}, & \text{if } h_{2p} + h_{1p} < 360 \\ \dfrac{h_{1p} + h_{2p} - 360}{2}, & \text{else,} \end{cases} \end{aligned} \tag{15}$$

where

$$h_{1p} = \begin{cases} 0, & \text{if } a_{1p} = b_1 = 0 \\ \tan^{-1}\left[\dfrac{b_1}{a_{1p}}\right]^\circ + 360^\circ, & \text{else} \end{cases}$$

$$h_{2p} = \begin{cases} 0, & \text{if } a_{2p} = b_2 = 0 \\ \tan^{-1}\left[\dfrac{b_2}{a_{2p}}\right]^{\circ} + 360^{\circ}, & \text{else.} \end{cases} \tag{16}$$

The expression $[x]^0$ in (11) and (16) means that "x" in radian is to be expressed in degree and the expression $[x]^r$ in (13) and (14) indicates that "x" in degree is to be expressed in radian. The other symbols appearing in the color difference equation are defined as follows:

$$\begin{aligned} C_{1p} &= \sqrt{a_{1p}^2 + b_1^2}, \\ C_{2p} &= \sqrt{a_{2p}^2 + b_2^2}, \\ a_{1p} &= (1 + G)\, a_1, \\ a_{2p} &= (1 + G)\, a_2, \\ G &= 0.5\left(1 - \sqrt{\frac{C_{\text{pa}}^7}{C_{\text{pa}}^7 + 25^7}}\right), \\ C_1 &= \sqrt{a_1^2 + b_1^2}, \\ C_2 &= \sqrt{a_2^2 + b_2^2}, \\ C_{\text{pa}} &= \frac{C_1 + C_2}{2}. \end{aligned} \tag{17}$$

3.2.4. Image Artifact Filtering. The computed binary saliency map is the input to the artifact filtering function, so any desirable algorithm can be used to filter the saliency map. The prime objective of the artifact filtering is to remove any extra element that might be remaining after segmentation and select a single connected region that is more likely to be the actual skin lesion. The two approaches for removing artifacts from images are preprocessing and postprocessing. This study implements the postprocessing approach to achieve computational efficiency because not all the three channels of the image are processed to remove artifacts.

This study applies the morphological analysis as the artifact filtering tool to remove undesired elements in the binary map while maintaining the structural properties of skin lesion. Morphological analysis is important in digital image processing because it can preserve structural properties of skin lesion and rigorously quantify many aspects of the geometrical structure of images in agreement with the human perception [16, 25]. The relationship between each part of an image can be identified when processing with morphological theory [25, 33]. The structural character of an image in a morphological approach is analyzed in terms of some predetermined geometric shapes such as disk, diamond, and squared shapes which are known as structuring elements [33]. The MATLAB median filter, clear border function, and morphological operations of opening and closing are used in this study. The median filter with structuring element of size 11×11 is first used to eliminate hairs and smooths against noise because of its capability to reduce bubble intensity and prevent fuzzy edges [16, 28]. It is widely used in digital image processing because it preserves edge information under certain conditions while removing oversegmentation. The filter considers each pixel in the input image in turn and looks at its nearby neighbors to decide whether or not it is a representative of its surroundings. It is usually evaluated by ordering all pixel values from the surrounding neighborhood and the pixel being considered is replaced with the middle pixel [76].

The opening operation smooths object contours, breaks thin connections, removes thin protrusions, and eliminates those objects smaller than the structuring element using morphological erosion followed by morphological dilation. The disk structural element is created to preserve the circular nature of lesion when performing morphological opening operation. The radius of the structural element is specified in this study to be 11 pixels so that large gaps can be filled adequately. The resulting binary image is then closed using the morphological closing operation by performing dilation followed by erosion. The same disk structural element that is created in the opening operation is used for the closing operation. The closing operation smooths object contours, joins narrow breaks, and fills long thin gulfs and holes smaller than the structuring element. The "clear border" function is finally used to remove vignette and disconnected objects touching the image borders. However, for nondisconnected objects touching the image borders, we recommend the use of a more effective border processing algorithm to avoid the inherent limitation of the MATLAB "imclearborder" function.

3.3. Algorithm Implementation. The algorithmic implementation of the method of perceptual color difference saliency (PCDS) is succinctly outlined based on mathematical equations (1)–(17). The asymptotic time complexity of the PCDS algorithm is $O(M \times N \times 3)$ for an input color image of dimensions $M \times N \times 3$. The PCDS algorithm is described step by step in Algorithm 1.

4. Discussion of Experimental Results

The experimental results obtained by the PCDS algorithm are discussed in this section. The PCDS algorithm is qualitatively and quantitatively compared to the spatially weighted dissimilarity (SWD) [77], principal component analysis (PCA) [78], Markov chain (MC) [79], and saliency based skin lesion segmentation (SSLS) [17] which are benchmark saliency segmentation algorithms. In addition, we establish comparison with the Otsu algorithm [71], K-means clustering [80], fuzzy C-means (FCM) clustering [81], and modified JSEG [12] which are benchmark nonsaliency segmentation algorithms. The source code for the SSLS algorithm with default parameter settings has been provided by the author whereas the source codes for the SWD, PCA, and MC algorithms are readily available at the following website: https://github.com/MingMingCheng/SalBenchmark/tree/master/Code/matlab.

Input: $M \times N \times 3$ *RGB* color image.
Output: $M \times N$ grayscale saliency map, $M \times N$ silhouette saliency map.
It is assumed that the standard color difference formula described by equations (10) to (17) has been implemented to be invoked in the computation of a saliency map in step (12) of this algorithm.
(1) **for all** $x = 0, 1, \ldots, M - 1$ **do**
(2) **for all** $y = 0, 1, \ldots, N - 1$ **do**
(3) transform the Adobe *RGB* image to CIE *XYZ* image using equations (1) and (2).
(4) transform the CIE *XYZ* image to CIE Lab image using equations (3) and (4).
(5) **end for**
(6) **end for**
(7) enhance the luminance channel of CIE Lab image using equations (5) and (6).
(8) compute mean of representative background pixels on an ellipsoidal patch.
(9) compute mean of representative object pixels within a rectangular patch.
(10) **for all** $x = 0, 1, \ldots, M - 1$ **do**
(11) **for all** $y = 0, 1, \ldots, N - 1$ **do**
(12) compute grayscale saliency map using equation (7) or equation (9).
(13) compute binary saliency map using equation (8).
(14) **end for**
(15) **end for**
(16) filter binary saliency map using morphological analysis or any desirable method.
(17) **stop**

Algorithm 1

4.1. Qualitative Evaluation of Segmentation Results. The purpose of the qualitative evaluation is to test the performance of the PCDS algorithm through qualitative comparison with existing saliency and nonsaliency based benchmark algorithms.

4.1.1. Comparison with Saliency Algorithms on ISBI 2016 Images. The segmentation results obtained by the PCDS algorithm is qualitatively compared with the results obtained by the existing benchmark saliency segmentation algorithms using test images acquired from the ISBI 2016 challenge corpus. Figure 3 shows a few examples of the original and ground truth images under varying conditions such as the presence of air bubbles (Im1 and Im2), presence of thick hair (Im3), low contrast (Im4, Im5, and Im6), and thin hair (Im7). In Figure 3, it can be seen that most of the skin lesions are correctly and consistently highlighted by the PCDS algorithm across all test images. However, when dermoscopic images possess air bubbles and illumination variation as in Im1, a situation whereby the skin lesion color distribution appears uneven, the other four benchmark saliency algorithms do not effectively and consistently highlight the skin lesion as our PCDS algorithm.

The PCDS algorithm generates an improved saliency map with more defined image boundaries when compared to the four other benchmark saliency algorithms. This is evidence in the case of Im2 that exhibits air bubbles and simultaneously presents similar color intensity between skin lesion and background skin. It is worth mentioning from an observation that virtually, for all images shown in Figure 3, the SWD algorithm has the poorest performance because it generates saliency maps with low resolution, blurry, and poorly defined borders. Moreover, it can be observed that all the saliency algorithms are able to achieve satisfactory results for dermoscopic images with high contrast as in Im3. However, for low contrast images such as Im4, Im5, and Im6, the PCA and SSLS algorithms do not uniformly highlight the salient objects. In fact, these algorithms could only highlight certain parts of the lesions while some parts share similar intensities with background color and salient lesions smaller in size when compared to the ground truth lesion. Although the MC algorithm has performed better than PCA and SSLS algorithms, it can be observed that saliency maps generated by the MC algorithm possess heterogeneous regions and fuzzy boundaries not uniformly highlighted. Contrarily, the PCDS algorithm outperforms the others in completely and uniformly highlighting the lesion objects with no varying colors. This indicates that the PCDS algorithm assigns uniform saliency values to the pixels within the salient objects.

In addition, another interesting observation from Figure 3 can be seen in Im7 that other benchmark algorithms highlight only the visible part of the skin lesion when a skin lesion possesses thin hair and low contrast. Interestingly, only the PCDS algorithm has detected the tail end of the lesion as seen in the ground truth lesion which can lead to diagnostic error. The impressive performance of the PCDS algorithm in segmenting all images considered can be attributed to the effective measurement of color difference between uneven lesion color distributions using the accurate CIEDE2000 formula.

4.1.2. Comparison with Saliency Algorithms on PH2 Images. The segmentation results obtained by the PCDS algorithm is qualitatively compared with the results of existing benchmark saliency algorithms using test images acquired from the PH2

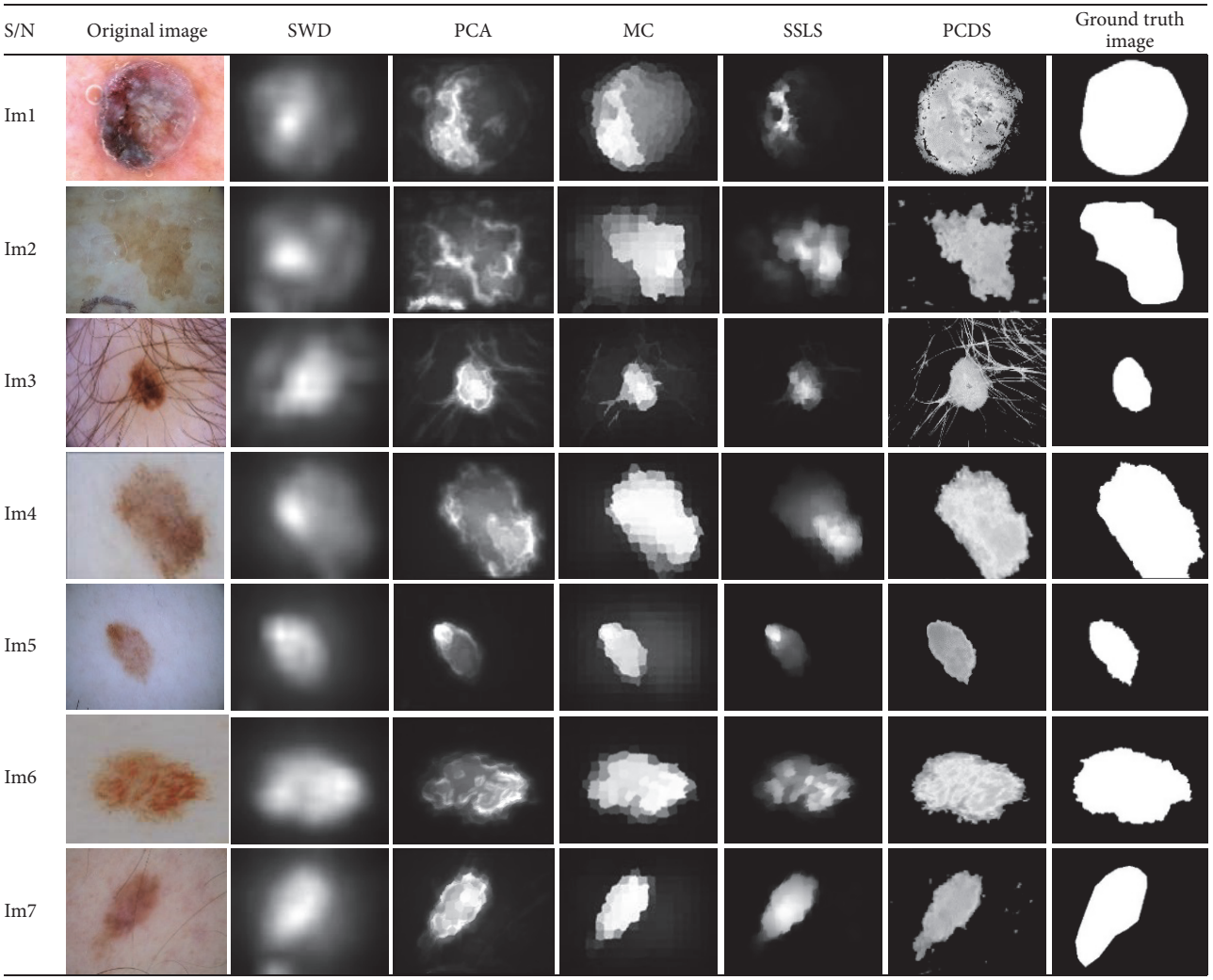


Figure 3: Qualitative illustration of saliency segmentation results obtained by benchmark saliency algorithms and PCDS algorithm on ISBI 2016 images.

corpus. Figure 4 shows some saliency maps produced by the PCDS algorithm along with those of other algorithms. Moreover, we have noted that the PCDS algorithm achieves good segmentation against the other algorithms. This is because the PCDS algorithm has the advantage of uniformly highlighting the whole salient object with high resolution as seen across the entire dermoscopic images.

The SWD algorithm has the least performance on PH2 images as seen in the segmentation results depicted in Figure 4. Moreover, it can be observed that saliency maps generated by the SWD algorithm are blurry and do not convey much useful information with respect to identifying the skin lesion. Although the PCA algorithm can correctly locate the skin lesion in the images, usually the algorithm highlights certain parts of the salient lesion boundaries as seen in Im6 and Im7 which can lead to diagnostic error. In addition, it can be observed further that the PCA algorithm fails at detecting the precise location of the skin lesion. It can be observed, for example, in Im1 and Im2, that the PCA algorithm detects skin lesion in such a way that it touches the image border which is oversegmentation.

The MC algorithm highlights skin lesion boundaries and detects skin lesion. However, it can be seen that boundaries of the saliency map are imprecise and fuzzy across the test images. This can result in the segmentation of healthy skin as skin lesion if fuzzy based thresholding algorithms such as the Huang and Wang [82] are applied for binary segmentation of the saliency map. Furthermore, it can be observed that the SSLS algorithm is able to highlight skin lesion boundaries, but still it cannot assign uniform salient pixel values in the inner part as in Im3, Im4, and Im6. In addition, it can be observed that the skin lesion produced by the SSLS algorithm in Im7 is smaller than ground truth skin lesion. In sharp contrast, it can be seen that the PCDS algorithm, to a greater extent, uniformly highlights the skin lesion, predicts precise location of the skin lesion, and produces well defined skin lesion borders. This clearly indicates that the PCDS algorithm shows a good performance and desirable

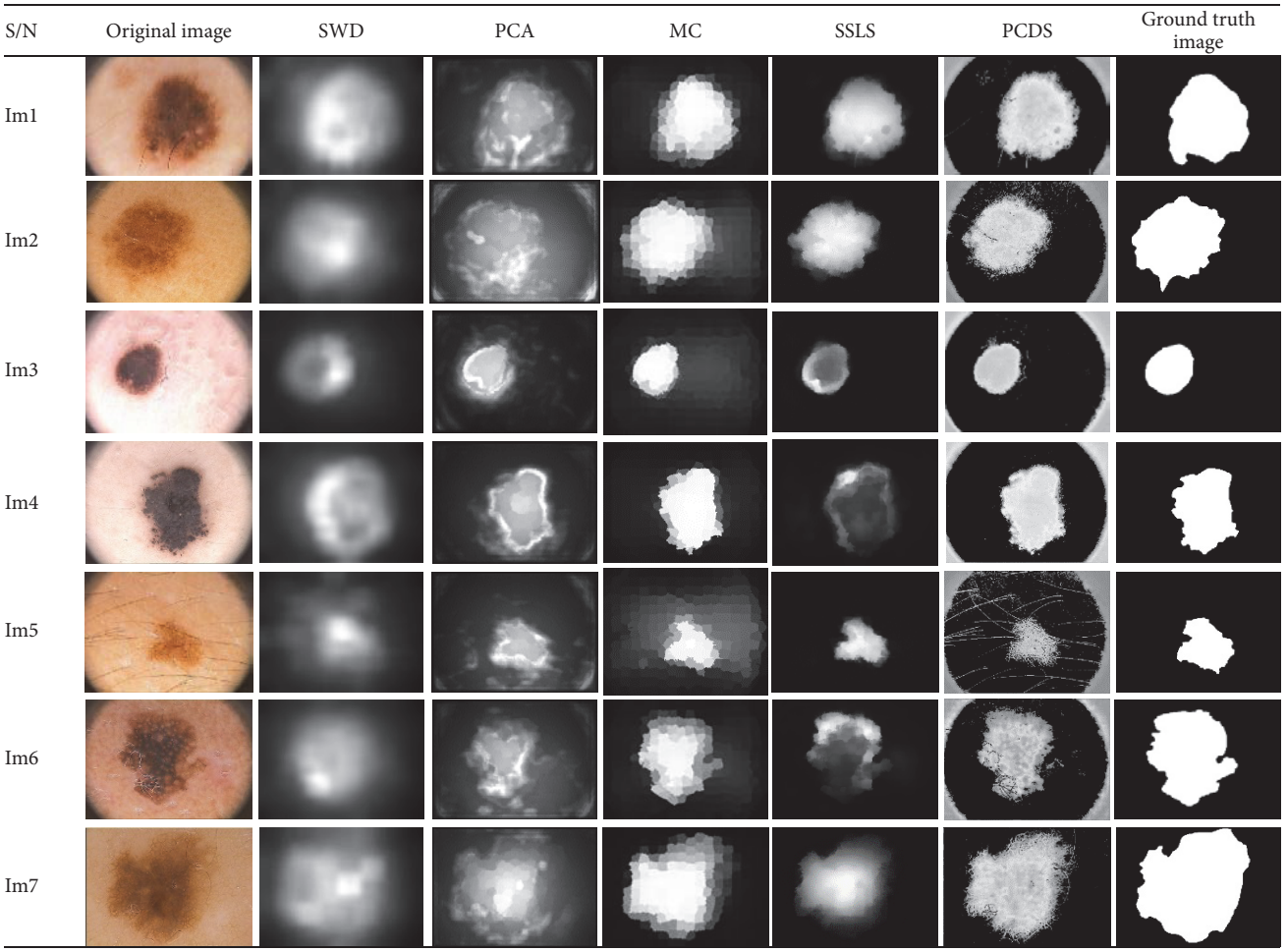


Figure 4: Qualitative illustration of saliency segmentation results obtained by benchmark saliency algorithms and PCDS algorithm on PH2 images.

saliency segmentation with reference to the ground truth dermoscopic images.

4.1.3. Comparison with Nonsaliency Algorithms on ISBI 2016 Images. The binary segmentation results obtained using the default thresholding method of the PCDS algorithm on ISBI 2016 images are presented in this section. The image artifact filtering method is not performed in this particular case in order to test the performance of the PCDS default thresholding without being aided. Figure 5 shows some examples of binary segmentation results produced by the PCDS default thresholding with other nonsaliency benchmark algorithms. The lesion images for the qualitative comparison are the same ISBI 2016 images presented in Figure 3, but in the absence of artifact filtering. However, it is worth mentioning that the implementation of the modified JSEG algorithm is inherently embedded with preprocessing and postprocessing methods to deal with artifacts which we do not have control over.

The results in Figure 5 show that, despite the absence of image artifact filtering, it is easy to note that binary segmentation results produced by the default PCDS thresholding show performance improvement. Specifically, one can see that PCDS algorithm gives a better segmentation result for Im1. It is observed that Otsu, K-means, and FCM algorithms produced incomplete binary segmented lesions smaller in size than ground truth lesions. This problem can be attributed to the illumination variation in the original dermoscopic image in Im1 that the algorithms cannot deal with intelligently. In addition, there is a considerable amount of border irregularities in the lesion borders of the binary segmented images produced by the modified JSEG algorithm. This is a conspicuous demerit as border irregularities caused by inaccurate segmentation can mislead the automatic diagnosis process.

Moreover, Im2 reveals that Otsu, K-means, and FCM algorithms exhibit poor performances when the input image has low contrast between the skin lesion and healthy skin. It is also noticeable that, apart from the presence of image artifacts in the binary segmented images produced by Otsu thresholding, K-means, and fuzzy C-means, some parts of the healthy skin share similar color intensities as the lesion. This is an indication that Im2 contains heterogeneous regions with different visual properties. Most especially when the healthy skin intensity is similar to the lesion as it can be seen that the healthy skin in the segmented binary images

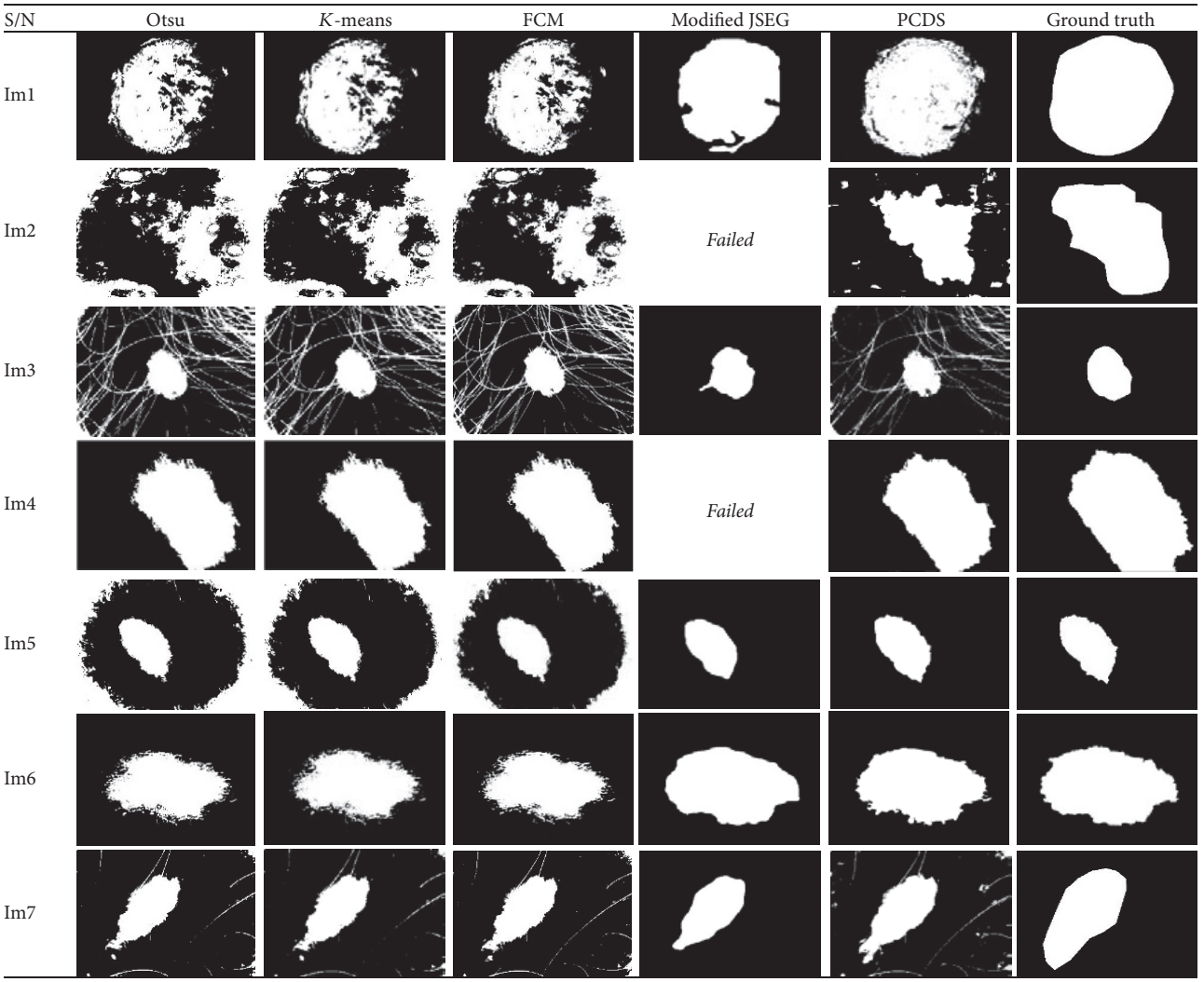


Figure 5: Qualitative illustration of the binary segmentation results obtained using four benchmark nonsaliency algorithms and default PCDS thresholding on ISBI 2016 images.

produced by these three algorithms share similar color intensities like skin lesion as seen in Im5. Still, on Im2, there is an indication that the modified JSEG algorithm failed to produce a binary segmented image because the algorithm is unsuccessful at segmenting Im2 as shown with white block written FAILED. The unsuccessful cases recorded by the modified JSEG algorithm as reported in this study is not the first of its kind. The original authors of the algorithm reported similar unsuccessful cases produced by the algorithm during experimentation [12]. Moreover, Norton et al. [83] reported an unsuccessful case of the modified JSEG for failing to segment fourteen test images in the most challenging situations. On the other hand, despite the absence of image artifact filtering, Im2 produced by the PCDS algorithm does not contain oil bubbles as seen in the original image. It is evident that the PCDS algorithm gives good binary segmentation results when compared to other benchmark nonsaliency algorithms.

In Im3, aside from the presence of thick hair, all the four nonsaliency algorithms produce binary segmented images similar to the ground truth images. This happens when there is a good contrast between the lesion and healthy skin; thus the lesion boundaries are well defined. However, it can be observed that the modified JSEG algorithm segmented hair trace to be part of the lesion which as stated earlier can result in diagnostic error. In Im6, we can see that the binary segmented image produced by the PCDS default thresholding is almost comparable to the result of the modified JSEG algorithm. It can be observed that the appearance of the PCDS default thresholding still produced well connected and precise lesion border than those of Otsu, K-means, and FCM algorithms as seen in Im6. Eventually, Im7 shows that the PCDS default thresholding produced a full representation of the skin lesion when compared to four other nonsaliency algorithms.

4.1.4. Comparison with Nonsaliency Algorithms on PH2 Images. The binary segmentation results obtained using the default thresholding method of the PCDS algorithm on PH2

images are presented in this section. Figure 6 shows some examples of the binary segmentation results in the absence of artifact filtering. There is no apparent differential between results produced by all the algorithms. However, the PCDS algorithm shows slight improvement when compared to the Otsu, K-means, and FCM for low contrast images as in Im5. Slight improvement in border irregularities can be seen in Im3, Im6, and Im7 produced by the modified JSEG algorithm when compared to the ground truth images. The less apparent differential is because the acquired PH2 images are not in varying imaging conditions as those of the ISBI 2016 images. However, many of the acquired PH2 test images exhibit vignette effect which is mainly due to the challenge of using round circular lens designed for a smaller sensor in dermatoscope [1].

In summary, the PCDS algorithm performs favorably against the benchmark algorithms as shown in Figures 3–6. The algorithm produces more stable discriminating saliency maps with high resolution and it uniformly highlights salient objects across the test images. Moreover, the algorithm extracts lesion borders in challenging conditions and it handles the problems of illumination variation and low contrast more effectively. These results validate the performance of the PCDS algorithm in handling challenging images and they demonstrate that implementation steps of the algorithm are relevant for its overall performance.

4.2. Quantitative Evaluation of Segmentation Results. The purpose of the quantitative evaluation is to test the performance of the PCDS algorithm through quantitative comparison of binary segmentation results with the existing benchmark saliency and nonsaliency algorithms on dermoscopic images acquired from the ISBI 2016 and PH2 corpora. The quantitative evaluation allows generalization to a large set of test images that cannot easily be achieved by qualitative evaluation because of few test samples. This study applies the precision (P), recall (R), accuracy (A), and dice (D) evaluation metrics to quantitatively score the binary segmentation results computed by the comparative algorithms. These evaluation metrics are widely used for judging the performance of binary segmentation algorithms [8, 13, 19, 20, 47, 48, 61, 83–85]. A binary segmentation algorithm with satisfactory performance has high precision, recall, accuracy, and dice values.

Precision is the ratio of the number of skin lesion pixels correctly identified to the total number of pixels in the saliency map. Recall is the ratio of the number of skin lesion pixels correctly identified to the total number of skin lesion pixels in the saliency map. Accuracy is the total number of pixels correctly identified to the total number of pixels in the saliency map. Dice coefficient measures agreement between the ground truth and result of automated segmentation method. The formal definitions of these evaluation metrics are based on the following parameters. True positive (T_p) is the count of skin lesion pixels correctly identified as skin lesion pixels. False negative (F_n) is the count of skin lesion pixels incorrectly identified as healthy skin pixels. False positive (F_p) is the count of healthy skin pixels incorrectly identified as skin lesion pixels. True negative (T_n) is the count of healthy skin pixels correctly identified as healthy skin. These measures are mathematically defined as follows [13, 47]:

$$
\begin{aligned}
P &= \frac{T_p}{T_p + F_p}, \\
R &= \frac{T_p}{T_p + F_n}, \\
A &= \frac{T_p + T_n}{T_p + T_n + F_p + F_n}, \\
D &= \frac{2T_p}{2T_p + F_n + F_p}.
\end{aligned}
\tag{18}
$$

The performance of binary segmentation using the PCDS algorithm is compared with the widely used Otsu thresholding algorithm [86] because thresholding algorithms are conventionally applied for binary segmentation of salient objects from grayscale maps [17, 87]. Table 1 shows ten comparative image segmentation algorithms compared in this study to establish the performance of binary segmentation using the PCDS algorithm.

4.2.1. Precision Scores. Table 2 lists the average precision (AVEP) scores and corresponding standard deviation (STDP) scores for each set of test images. It can be seen in Table 2 that the PCDS algorithm consistently recorded the highest AVEP score of 0.8911 and lowest STDP score of 0.1166 on ISBI 2016 test images. However, the SSLSOtsu algorithm recorded the lowest AVEP score of 0.6439 (0.2154) on ISBI 2016 images. Since the STDP score of 0.2154 for the SSLSOtsu algorithm is lower than that of the modified JSEG algorithm (0.2363) and Otsu algorithm (0.2445), the SSLSOtsu algorithm has better precision than modified JSEG and Otsu algorithms on some of the ISBI 2016 test images.

The PCDSOtsu algorithm consistently recorded the highest AVEP score of 0.9617 and lowest STDP score of 0.0503 on PH2 test images. However, the Otsu algorithm recorded the lowest AVEP score of 0.5557 and highest STDP score of 0.3697 on PH2 test images. The Otsu algorithm with the highest STDP score did not give better precision than any of the other algorithms on the PH2 test images. These results generally indicate that the PCDS algorithm consistently recorded good precision on ISBI 2016 test images, while the PCDSOtsu algorithm consistently recorded excellent precision on PH2 test images.

4.2.2. Recall Scores. Table 3 lists the average recall (AVER) scores and corresponding standard deviation (STDR) scores for each set of test images. It can be seen in Table 3 that the SSLSOtsu algorithm consistently recorded the highest AVER score of 0.9998 and lowest STDR score of 0.0012 on ISBI 2016 test images. The SWDOtsu algorithm recorded the lowest AVER score of 0.8014 (0.1893) on ISBI 2016 images. Since the STDR score of 0.1893 for the SWDOtsu algorithm is lower than that of the modified JSEG algorithm (0.2330),

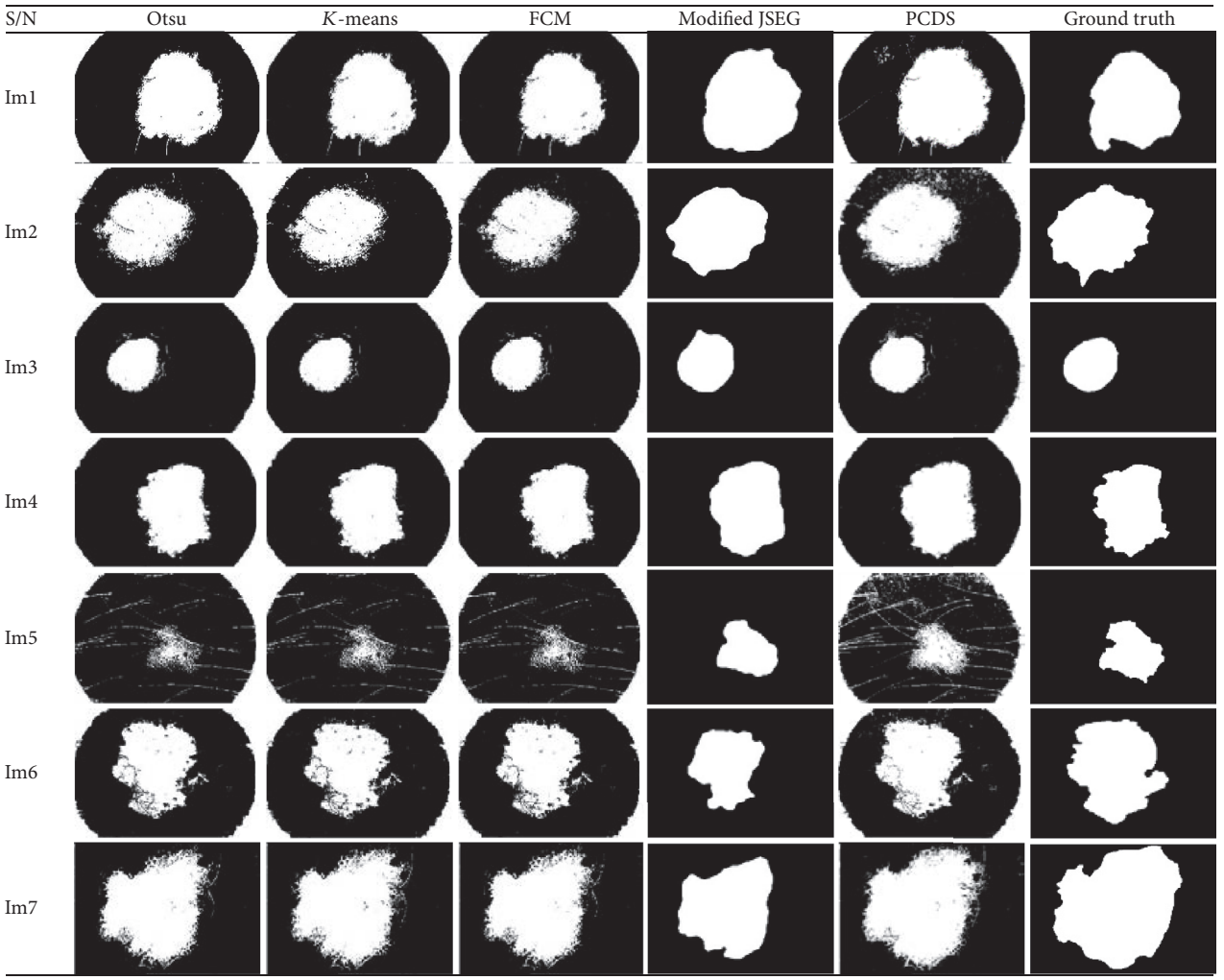


Figure 6: Qualitative illustration of binary segmentation results obtained using four benchmark nonsaliency algorithms and default PCDS thresholding on PH2 images.

Table 1: Comparative algorithms with descriptions.

Algorithm	Description
Otsu	Conventional Otsu thresholding algorithm applied to skin lesion segmentation.
K-means	Conventional K-means clustering algorithm applied to skin lesion segmentation.
FCM	Conventional fuzzy C-means clustering algorithm applied to skin lesion segmentation.
Modified JSEG	The modified JSEG image segmentation algorithm applied to skin lesion segmentation.
SWDOtsu	The conventional Otsu thresholding algorithm applied to threshold the saliency map computed by the spatially weighted dissimilarity (SWD) algorithm.
PCAOtsu	The conventional Otsu thresholding algorithm applied to threshold the saliency map computed by the principal component analysis (PCA) algorithm.
MCOtsu	The conventional Otsu thresholding algorithm applied to threshold the saliency map computed by the Markov chain (MC) algorithm.
SSLSOtsu	The conventional Otsu thresholding algorithm applied to threshold the saliency map computed by the saliency based skin lesion segmentation (SSLS) algorithm.
PCDSOtsu	The conventional Otsu thresholding algorithm applied to threshold the saliency map computed by the perceptual color difference saliency (PCDS) algorithm.
PCDS	The perceptual color difference saliency (PCDS) algorithm with a simple thresholding decision rule applied for binary segmentation of the saliency map.

Table 2: Precision scores of comparative algorithms on ISBI 2016 and PH2 images.

Algorithm	ISBI 2016		PH2	
	AVEP	STDP	AVEP	STDP
Otsu	0.8134	0.2445	0.5557	0.3697
K-means	0.8420	0.2000	0.7904	0.3031
FCM	0.8464	0.1979	0.7480	0.3261
Modified JSEG	0.8681	0.2363	0.8984	0.2438
SWDOtsu	0.8437	0.1883	0.9118	0.2388
PCAOtsu	0.8593	0.1675	0.9141	0.2005
MCOtsu	0.8494	0.1227	0.8969	0.1556
SSLSOtsu	0.6439	0.2154	0.8318	0.2095
PCDSOtsu	0.8823	0.2012	0.9617	0.0503
PCDS	0.8911	0.1166	0.9499	0.0554

Table 3: Recall scores of comparative algorithms on ISBI 2016 and PH2 images.

Algorithm	ISBI 2016		PH2	
	AVER	STDR	AVER	STDR
Otsu	0.9705	0.1677	0.8383	0.3701
K-means	0.9838	0.1196	0.9362	0.2392
FCM	0.9850	0.1195	0.9371	0.2393
Modified JSEG	0.9291	0.2330	0.8759	0.2414
SWDOtsu	0.8014	0.1893	0.6569	0.2144
PCAOtsu	0.9848	0.0392	0.8482	0.1911
MCOtsu	0.9971	0.0099	0.9620	0.1420
SSLSOtsu	0.9998	0.0012	0.9509	0.1975
PCDSOtsu	0.9817	0.1199	0.9554	0.0766
PCDS	0.9927	0.0228	0.9586	0.0467

Table 4: Accuracy scores of comparative algorithms on ISBI 2016 and PH2 images.

Algorithm	ISBI 2016		PH2	
	AVEA	STDA	AVEA	STDA
Otsu	0.9456	0.1204	0.9421	0.0638
K-means	0.9572	0.0580	0.9735	0.0548
FCM	0.9577	0.0570	0.9745	0.0392
Modified JSEG	0.9190	0.2297	0.9729	0.0433
SWDOtsu	0.8962	0.0732	0.9185	0.1172
PCAOtsu	0.9498	0.0626	0.9678	0.0353
MCOtsu	0.9564	0.0431	0.9861	0.0171
SSLSOtsu	0.8868	0.1086	0.9737	0.0382
PCDSOtsu	0.9622	0.1194	0.9888	0.0113
PCDS	0.9769	0.0303	0.9847	0.0114

Table 5: Dice scores of comparative algorithms on ISBI 2016 and PH2 images.

Algorithm	ISBI 2016		PH2	
	AVED	STDD	AVED	STDD
Otsu	0.8665	0.2312	0.6262	0.3815
K-means	0.8949	0.1770	0.8336	0.2929
FCM	0.8977	0.1819	0.7962	0.3237
Modified JSEG	0.8941	0.2301	0.8812	0.2330
SWDOtsu	0.8061	0.1584	0.7542	0.2100
PCAOtsu	0.9067	0.1244	0.8762	0.1886
MCOtsu	0.9120	0.0817	0.9291	0.1409
SSLSOtsu	0.7601	0.1820	0.8631	0.2316
PCDSOtsu	0.9166	0.1818	0.9360	0.1439
PCDS	0.9342	0.0709	0.9522	0.0287

the SWDOtsu algorithm recorded better recall than modified JSEG on some of the ISBI 2016 test images.

The MCOtsu algorithm recorded the highest AVER score of 0.9620 on PH2 test images. However, the STDR score of 0.1420 for the MCOtsu algorithm is higher than that of the PCDSOtsu algorithm (0.0766) and PCDS algorithm (0.0467). The PCDSOtsu and PCDS algorithms recorded better recall than MCOtsu algorithm on some PH2 test images. The SWDOtsu algorithm recorded the lowest AVER score of 0.6569 on PH2 test images. However, the STDR of 0.2144 for the SWDOtsu algorithm is lower than those of the nonsaliency based algorithms which implies that the SWDOtsu algorithm recorded better recall than nonsaliency based algorithms on some of the PH2 test images. These results generally indicate that the SSLSOtsu algorithm oversegment ISBI 2016 test images because it achieves imbalance precision (0.6439) and recall (0.9998) while the PCDS algorithm consistently gave excellent recall on PH2 test images because it achieves balance precision (0.9499) and recall (0.9586).

4.2.3. Accuracy Scores. Table 4 lists the average accuracy (AVEA) scores and corresponding standard deviation (STDA) scores for each set of test images. It can be seen in Table 4 that the PCDS algorithm consistently recorded the highest AVEA score of 0.9769 and lowest STDA score of 0.0303 on ISBI 2016 test images. The SSLSOtsu algorithm recorded the lowest AVEA score of 0.8868 (0.1086) on ISBI 2016 images. Since the STDA score of 0.1086 for the SSLSOtsu algorithm is lower than that of the PCDSOtsu algorithm (0.1194), modified JSEG algorithm (0.2297), and Otsu algorithm (0.1204), the SSLSOtsu algorithm recorded better accuracy than these algorithms on some of the ISBI 2016 test images.

The PCDSOtsu algorithm consistently recorded the highest AVEA score of 0.9888 and lowest STDA score of 0.0113 on PH2 test images. However, the SWDOtsu algorithm consistently recorded the lowest AVEA score of 0.9185 and highest STDA score of 0.1172. The SWDOtsu algorithm with the highest STDA score did not give better accuracy than any of the other algorithms on the PH2 test images. These results generally indicate that the PCDS algorithm consistently recorded excellent accuracy on ISBI 2016 test images, while the PCDSOtsu algorithm consistently recorded excellent accuracy on PH2 test images.

4.2.4. Dice Scores. Table 5 lists the average dice (AVED) scores and corresponding standard deviation (STDD) scores

for each set of test images. The PCDS algorithm can be seen in Table 5 to consistently record the highest AVED score of 0.9342 and lowest STDD score of 0.0709 on ISBI 2016 test images. The SSLSOtsu algorithm recorded the lowest AVED score of 0.7601 (0.1820) on ISBI 2016 test images. Since the STDD score of 0.1820 for the SSLSOtsu algorithm is lower than that of the modified JSEG algorithm (0.2301) and Otsu algorithm (0.2312), the SSLSOtsu algorithm performed better than modified JSEG and Otsu algorithms on some of the ISBI 2016 test images.

The PCDS algorithm gave the highest AVED score of 0.9522 and lowest STDD score of 0.0287 on PH2 test images. However, the Otsu algorithm recorded the lowest AVED score of 0.62627 and highest STDD score of 0.3697 on PH2 test images. The Otsu algorithm with the highest STDD score did not compute segmentation outputs with better agreement with the ground truth than any of the other algorithms on the PH2 test images. These results generally indicate that the PCDS algorithm consistently computed segmentation outputs that have excellent agreement with the ground truth images across the ISBI 2016 and PH2 test images.

4.2.5. Performance Scores. The coefficient of variation (CV) statistic is ultimately used in this study to determine the algorithm that gives best performance across different test images. The CV is a standardized dispersion measure of a probability distribution that represents the ratio of standard deviation to mean. The weighted mean of coefficient of variations (MCV) unifies the scores associated with a given evaluation criterion across different test images. The MCV value of 1 means low dispersion (excellent result) in the evaluation criterion and a value of 0 means high dispersion (inferior result) in the evaluation criterion. Given a set of distributions with mean values of $\mu_1, \mu_2, \ldots, \mu_k$ and standard deviation values of $\sigma_1, \sigma_2, \ldots, \sigma_k$, the MCV is determined with the largest weight given to the largest sample as follows:

$$\mathrm{MCV}\left(\mu, \sigma, w, n\right) = \sum_{i=1}^{n} \left(\frac{w_i \left(\mu_i - \sigma_i\right)}{\mu_i} \right), \tag{19}$$

where n is the total number of datasets and weight functions $w_1, w_2, \ldots, w_k$ sum up to unity:

$$\sum_{i=1}^{n} w_i = 1. \tag{20}$$

The main reason to use sample sizes as weight functions in MCV calculation is that an algorithm that performs well on a large set of test data is preferable to that which performs well on a small set of test data. In this study, the sizes of ISBI 2016 and PH2 test images are, respectively, 70 and 50. In fact, we deliberately selected more test images from the ISBI 2016 corpus because it has more challenging images than PH2 corpus. The PH2 contains 200 melanocytic lesions whereas the ISBI 2016 contains 900 dermoscopic images with ground truths of both sets of images available [13]. Consequently, $w_1 = 7/12$, $w_2 = 5/12$, and $n = 2$. In the special case of $n = 2$ (19) reduces to the following equation:

$$\begin{aligned} &\mathrm{MCVs}\left(\mu, \sigma, w, 2\right) \\ &\quad = \frac{w_1 \left(\mu_1 - \sigma_1\right) \mu_2 + w_2 \left(\mu_2 - \sigma_2\right) \mu_1}{\mu_1 \mu_2}. \end{aligned} \tag{21}$$

Table 6 shows the result of applying (21) to compute the MCV for precision (Precision_MCV), recall (Recall_MCV), accuracy (Accuracy_MCV), and dice (Dice_MCV). The overall performance score for each comparative algorithm is based on the utility function obtained by averaging the scores for all evaluation criteria. The result in Table 6 shows that, ranking in terms of the utility function, the PCDS algorithm recorded an excellent overall performance and is ranked in the first position while the Otsu algorithm is ranked in the tenth position. The ranking of each algorithm in terms of individual criterion is also given with the PCDS algorithm leading. Surprisingly, the PCDSOtsu algorithm did not rank second following the PCDS algorithm which means that the binary segmentation technique of the PCDS algorithm is effective. In the literature, the Otsu algorithm is acclaimed to be optimal for binary segmentation, but its performance is poor for segmentation of melanoma skin lesion in dermoscopic images as experienced in this study. Finally, it is important to note that the low performance scoring of the modified JSEG algorithm is mainly due to its inability to segment some of the test images.

5. Conclusion

This paper reports a new image segmentation algorithm based on perceptual color difference saliency (PCDS) that integrates both background and foreground information for segmentation of skin lesion in dermoscopic images. The PCDS algorithm has been tested on 120 challenging dermoscopic images acquired from the ISBI 2016 challenge and PH2 corpora. The algorithm has been quantitatively compared with a variety of saliency and nonsaliency benchmark algorithms using famous statistical evaluation metrics of precision, recall, accuracy, and dice. The experimental results of this study show that PCDS algorithm achieves excellent performance in segmenting skin lesion in dermoscopic images with different classes of challenges when compared to benchmark algorithms investigated in this study. Moreover, the PCDS algorithm tends to be more robust to the presence of air bubble, thick hair, and low contrast than other comparative algorithms investigated in this study.

Future work will focus on the extraction of distinctive skin lesion features for the classification of melanoma skin lesion in dermoscopic images using the PCDS algorithm for segmentation. In addition, we plan to extend the PCDS method to other existing color models and color difference formulae for comparative purpose. In addition, it will be prudent to look at other practical applications to test the performance of the PCDS algorithm on images with other challenges. The one important aspect of the PCDS algorithm that needs further investigation is the estimation of mean

Table 6: Overall performance scores of comparative algorithms.

Algorithm	Precision_MCV	Recall_MCV	Accuracy_MCV	Dice_MCV	Utility
Otsu	0.5475 (10)	0.7152 (10)	0.8975 (9)	0.5905 (10)	0.6877 (10)
K-means	0.7017 (7)	0.8226 (6)	0.9412 (5)	0.7382 (8)	0.8009 (6)
FCM	0.6820 (9)	0.8228 (7)	0.9485 (4)	0.7124 (9)	0.7914 (7)
Modified JSEG	0.7281 (6)	0.7389 (8)	0.8357 (10)	0.7397 (7)	0.7606 (9)
SWDOtsu	0.7607 (5)	0.7262 (9)	0.8992 (8)	0.7694 (5)	0.7889 (8)
PCAOtsu	0.7949 (4)	0.8829 (5)	0.9464 (3)	0.8303 (3)	0.8636 (4)
MCOtsu	0.8434 (3)	0.9327 (2)	0.9665 (2)	0.8846 (2)	0.9068 (2)
SSLSOtsu	0.6999 (8)	0.9128 (3)	0.9122 (7)	0.7485 (6)	0.8184 (5)
PCDSOtsu	0.8452 (2)	0.8953 (4)	0.9229 (6)	0.8202 (4)	0.8709 (3)
PCDS	0.8994 (1)	0.9663 (1)	0.9771 (1)	0.9432 (1)	0.9465 (1)

value of background color pixels and mean value of object color pixels because effectiveness of the algorithm heavily depends on accurate estimation of these statistics. It is also essential to combine color cue with other cues such as texture to further improve the performance of the PCDS algorithm.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This publication is supported by postgraduate research grants from the Durban University of Technology in South Africa. The authors wish to thank Professor M. Emre Celebi from the Department of Computer Science at the University of Central Arkansas, Conway, AR, USA, for providing the source codes for their previous work. The authors are also grateful to Mrs. Seena Joseph for running the OpenCV implementation of K-means algorithm on experimental data and the set.

References

[1] S. Khalid, U. Jamil, K. Saleem et al., "Segmentation of skin lesion using Cohen–Daubechies–Feauveau biorthogonal wavelet," *SpringerPlus*, vol. 5, no. 1, article no. 1603, 2016.

[2] D. A. Okuboyejo, O. O. Olugbara, and S. A. Odunaike, "Automating skin disease diagnosis using image classification," in *Proceedings of the 2013 World Congress on Engineering and Computer Science, WCECS 2013*, pp. 850–854, usa, October 2013.

[3] G. Schaefer, M. I. Rajab, M. E. Celebi, and H. Iyatomi, "Colour and contrast enhancement for improved skin lesion segmentation," *Computerized Medical Imaging and Graphics*, vol. 35, no. 2, pp. 99–104, 2011.

[4] F. Xie and A. C. Bovik, "Automatic segmentation of dermoscopy images using self-generating neural networks seeded by genetic algorithm," *Pattern Recognition*, vol. 46, no. 3, pp. 1012–1019, 2013.

[5] D. Meckbach, J. Bauer, A. Pflugfelder et al., "Survival according to BRAF-V600 tumor mutations - An analysis of 437 patients with primary melanoma," *PLoS ONE*, vol. 9, no. 1, Article ID e86194, 2014.

[6] E. Flores and J. Scharcanski, "Segmentation of melanocytic skin lesions using feature learning and dictionaries," *Expert Systems with Applications*, vol. 56, pp. 300–309, 2016.

[7] B. Bozorgtabar, M. Abedini, and R. Garnavi, "Sparse coding based skin lesion segmentation using dynamic rule-based refinement," in *Machine Learning in Medical Imaging*, pp. 254–261, 2016.

[8] F. Thompson and M. Jeyakumar, "Analytical research of segmentation methods on skin lesion," *International Journal of Applied Engineering Research*, vol. 11, pp. 7132–7138, 2016.

[9] A. Masood and A. A. Al-Jumaily, "Computer aided diagnostic support system for skin cancer: a review of techniques and algorithms," *International Journal of Biomedical Imaging*, vol. 2013, Article ID 323268, 22 pages, 2013.

[10] A. Pennisi, D. D. Bloisi, D. Nardi, A. R. Giampetruzzi, C. Mondino, and A. Facchiano, "Skin lesion image segmentation using Delaunay Triangulation for melanoma detection," *Computerized Medical Imaging and Graphics*, vol. 52, pp. 89–103, 2016.

[11] F. Dalila, A. Zohra, K. Reda, and C. Hocine, "Segmentation and classification of melanoma and benign skin lesions," *Optik - International Journal for Light and Electron Optics*, vol. 140, pp. 749–761, 2017.

[12] M. E. Celebi, Y. A. Aslandogan, W. V. Stoecker, H. Iyatomi, H. Oka, and X. Chen, "Unsupervised border detection in dermoscopy images," *Skin Research and Technology*, vol. 13, no. 4, pp. 454–462, 2007.

[13] H. Fan, F. Xie, Y. Li, Z. Jiang, and J. Liu, "Automatic segmentation of dermoscopy images using saliency combined with Otsu threshold," *Computers in Biology and Medicine*, vol. 85, pp. 75–85, 2017.

[14] D. D. Gómez, C. Butakoff, B. K. Ersbøll, and W. Stoecker, "Independent histogram pursuit for segmentation of skin lesions," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 1, pp. 157–161, 2008.

[15] D. A. Okuboyejo, O. O. Olugbara, and S. A. Odunaike, "Unsupervised restoration of hair-occluded lesion in dermoscopic Images," in *MIUA*, pp. 91–96, 2014.

[16] J. Premaladha and K. S. Ravichandran, "Novel Approaches for Diagnosing Melanoma Skin Lesions Through Supervised and Deep Learning Algorithms," *Journal of Medical Systems*, vol. 40, no. 4, article no. 96, pp. 1–12, 2016.

[17] E. Ahn, L. Bi, Y. H. Jung et al., "Automated saliency-based lesion segmentation in dermoscopic images," in *Proceedings of the 37th Annual International Conference of the IEEE Engineering in*

Medicine and Biology Society, EMBC 2015, pp. 3009–3012, Italy, August 2015.

[18] Y. Zhao, Y. Zheng, Y. Liu et al., "Intensity and Compactness Enabled Saliency Estimation for Leakage Detection in Diabetic and Malarial Retinopathy," *IEEE Transactions on Medical Imaging*, vol. 36, no. 1, pp. 51–63, 2017.

[19] A. Aksac, T. Ozyer, and R. Alhajj, "Complex networks driven salient region detection based on superpixel segmentation," *Pattern Recognition*, vol. 66, pp. 268–279, 2017.

[20] Q. Zhang, J. Lin, Y. Tao, W. Li, and Y. Shi, "Salient object detection via color and texture cues," *Neurocomputing*, vol. 243, pp. 35–48, 2017.

[21] C. Yang, L. Zhang, and H. Lu, "Graph-regularized saliency detection with convex-hull-based center prior," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 637–640, 2013.

[22] R. Dubey, A. Dave, and B. Ghanem, "Improving saliency models by predicting human fixation patches," in *Computer Vision*, vol. 9005, pp. 330–345, 2015.

[23] M. Celebi, Q. Wen, H. Iyatomi, K. Shimizu, H. Zhou, and G. Schaefer, "A State-of-the-Art Survey on Lesion Border Detection in Dermoscopy Images," in *Dermoscopy Image Analysis*, Digital Imaging and Computer Vision, pp. 97–129, CRC Press, 2015.

[24] D. A. Okuboyejo, O. O. Olugbara, and S. A. Odunaike, "CLAHE inspired segmentation of dermoscopic images using mixture of methods," *Transactions on Engineering Technologies*, pp. 355–365, 2014.

[25] H. Wang, R. H. Moss, X. Chen et al., "Modified watershed technique and post-processing for segmentation of skin lesions in dermoscopy images," *Computerized Medical Imaging and Graphics*, vol. 35, no. 2, pp. 116–120, 2011.

[26] M. E. Celebi, H. A. Kingravi, H. Iyatomi et al., "Border detection in dermoscopy images using statistical region merging," *Skin Research and Technology*, vol. 14, no. 3, pp. 347–353, 2008.

[27] P. Rubegni, A. Ferrari, G. Cevenini et al., "Differentiation between pigmented Spitz naevus and melanoma by digital dermoscopy and stepwise logistic discriminant analysis," *Melanoma Research*, vol. 11, no. 1, pp. 37–44, 2001.

[28] R. Kasmi, K. Mokrani, R. K. Rader, J. G. Cole, and W. V. Stoecker, "Biologically inspired skin lesion segmentation using a geodesic active contour technique," *Skin Research and Technology*, vol. 22, no. 2, pp. 208–222, 2016.

[29] M. Sadeghi, M. Razmara, T. K. Lee, and M. S. Atkins, "A novel method for detection of pigment network in dermoscopic images using graphs," *Computerized Medical Imaging and Graphics*, vol. 35, no. 2, pp. 137–143, 2011.

[30] O. O. Olugbara, E. Adetiba, and S. A. Oyewole, "Pixel intensity clustering algorithm for multilevel image segmentation," *Mathematical Problems in Engineering*, vol. 2015, Article ID 649802, 19 pages, 2015.

[31] M. Emre Celebi, Q. Wen, S. Hwang, H. Iyatomi, and G. Schaefer, "Lesion Border Detection in Dermoscopy Images Using Ensembles of Thresholding Methods," *Skin Research and Technology*, vol. 19, no. 1, pp. e252–e258, 2013.

[32] R. B. Oliveira, N. Marranghello, A. S. Pereira, and J. M. R. S. Tavares, "A computational approach for detecting pigmented skin lesions in macroscopic images," *Expert Systems with Applications*, vol. 61, pp. 53–63, 2016.

[33] M. Zortea, E. Flores, and J. Scharcanski, "A simple weighted thresholding method for the segmentation of pigmented skin lesions in macroscopic images," *Pattern Recognition*, vol. 64, pp. 92–104, 2017.

[34] P. Khuwuthyakorn, A. Robles-Kelly, and J. Zhou, "Object of interest detection by saliency learning," in *Proceedings of the European conference on Computer vision*, vol. 6312, pp. 636–649, 2010.

[35] W. Yang, D. Li, S. Wang, S. Lu, and J. Yang, "Saliency-based color image segmentation in foreign fiber detection," *Mathematical and Computer Modelling*, vol. 58, no. 3-4, pp. 846–852, 2013.

[36] C. A. Hussain, D. V. Rao, and S. A. Masthani, "Robust Preprocessing Technique Based on Saliency Detection for Content Based Image Retrieval Systems," *Procedia Computer Science*, vol. 85, pp. 571–580, 2016.

[37] E. Ahn, J. Kim, L. Bi et al., "Saliency-Based Lesion Segmentation Via Background Detection in Dermoscopic Images," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1685–1693, 2017.

[38] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: a benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[39] R. Garnavi, M. Aldeen, M. E. Celebi, A. Bhuiyan, C. Dolianitis, and G. Varigos, "Automatic segmentation of dermoscopy images using histogram thresholding on optimal color channels," *International Journal of Medicine and Medical Sciences*, vol. 1, no. 2, pp. 126–134, 2011.

[40] M. Zortea, S. O. Skrøvseth, T. R. Schopf, H. M. Kirchesch, and F. Godtliebsen, "Automatic segmentation of dermoscopic images by iterative classification," *International Journal of Biomedical Imaging*, vol. 2011, Article ID 972648, 19 pages, 2011.

[41] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[42] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proceedings of the 14th Annual ACM International Conference on Multimedia (MULTIMEDIA '06)*, pp. 815–824, October 2006.

[43] X. Hou, J. Harel, and C. Koch, "Image signature: highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, 2012.

[44] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, USA, June 2008.

[45] B. Schauerte and R. Stiefelhagen, "Predicting human gaze using quaternion DCT image signature saliency and face detection," in *Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision, WACV 2012*, pp. 137–144, USA, January 2012.

[46] M. J. Ogorzalek, G. Surowak, L. Nowak, C. Merkwirth, and M. J. Ogorzałek, "approaches for computer-assisted skin cancer diagnosis," *Optimization and Systems Biology*, pp. 20–22, 2009.

[47] A. R. Sadri, M. Zekri, S. Sadri, N. Gheissari, M. Mokhtari, and F. Kolahdouzan, "Segmentation of dermoscopy images using wavelet networks," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 1134–1141, 2013.

[48] J. Khan, A. S. Malik, N. Kamel, S. C. Dass, and A. M. Affandi, "Segmentation of acne lesion using fuzzy C-means technique with intelligent selection of the desired cluster," in *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2015*, pp. 3077–3080, Italy, August 2015.

[49] H. Nisar, Y. K. Ch'ng, T. Y. Chew, V. V. Yap, K. H. Yeap, and J. J. Tang, "A color space study for skin lesion segmentation," in

Proceedings of the 2013 IEEE International Conference on Circuits and Systems: "Advanced Circuits and Systems for Sustainability", ICCAS 2013, pp. 172–176, Malaysia, September 2013.

[50] M. Schikora and A. Schikora, "Image-based Analysis to Study Plant Infection with Human Pathogens," *Computational and Structural Biotechnology Journal*, vol. 12, no. 20-21, pp. 1–6, 2014.

[51] G. Khashayar, P. A. Bain, S. Salari, A. Dozic, C. J. Kleverlaan, and A. J. Feilzer, "Perceptibility and acceptability thresholds for colour differences in dentistry," *Journal of Dentistry*, vol. 42, no. 6, pp. 637–644, 2014.

[52] N. Alghazali, G. Burnside, M. Moallem, P. Smith, A. Preston, and F. D. Jarad, "Assessment of perceptibility and acceptability of color difference of denture teeth," *Journal of Dentistry*, vol. 40, no. 1, pp. e10–e17, 2012.

[53] F. Bayindir, S. Kuo, W. M. Johnston, and A. G. Wee, "Coverage error of three conceptually different shade guide systems to vital unrestored dentition," *Journal of Prosthetic Dentistry*, vol. 98, no. 3, pp. 175–185, 2007.

[54] O. E. Pecho, R. Ghinea, R. Alessandretti, M. M. Pérez, and A. Della Bona, "Visual and instrumental shade matching using CIELAB and CIEDE2000 color difference formulas," *Dental Materials*, vol. 32, no. 1, pp. 82–92, 2016.

[55] M. D. M. Pérez, R. Ghinea, M. J. Rivas et al., "Development of a customized whiteness index for dentistry based on CIELAB color space," *Dental Materials*, vol. 32, no. 3, pp. 461–467, 2015.

[56] C. Gómez-Polo, M. Portillo Muñoz, M. C. Lorenzo Luengo, P. Vicente, P. Galindo, and A. M. Martín Casado, "Comparison of two color-difference formulas using the Bland-Altman approach based on natural tooth color space," *Journal of Prosthetic Dentistry*, vol. 115, no. 4, pp. 482–488, 2016.

[57] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188–198, 2013.

[58] X. Shi, Y. Chen, Y. Y. Fu, and J. H. Luo, "Application of color difference meter in the quality inspection of food," *Science and Technology of Food Industry*, vol. 5, p. 117, 2009.

[59] R. Fernández-Vázquez, C. M. Stinco, D. Hernanz, F. J. Heredia, and I. M. Vicario, "Colour training and colour differences thresholds in orange juice," *Food Quality and Preference*, vol. 30, no. 2, pp. 320–327, 2013.

[60] M. Q. Shaw, J. P. Allebach, and E. J. Delp, "Color difference weighted adaptive residual preprocessing using perceptual modeling for video compression," *Signal Processing: Image Communication*, vol. 39, pp. 355–368, 2015.

[61] D. Gutman, N. C. Codella, E. Celebi et al., "Skin lesion analysis toward melanoma detection: a challenge at the international symposium on biomedical Imaging," https://arxiv.org/abs/1605.01397.

[62] T. Mendonca, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira, "PH2 - A dermoscopic image database for research and benchmarking," in *Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2013*, pp. 5437–5440, Japan, July 2013.

[63] K. Møllersen, H. Kirchesch, M. Zortea, T. R. Schopf, K. Hindberg, and F. Godtliebsen, "Computer-Aided Decision Support for Melanoma Detection Applied on Melanocytic and Nonmelanocytic Skin Lesions: A Comparison of Two Systems Based on Automatic Analysis of Dermoscopic Images," *BioMed Research International*, vol. 2015, Article ID 579282, 8 pages, 2015.

[64] M. Dowlati, S. S. Mohtasebi, M. Omid, S. H. Razavi, M. Jamzad, and M. De La Guardia, "Freshness assessment of gilthead sea bream (*Sparus aurata*) by machine vision based on gill and eye color changes," *Journal of Food Engineering*, vol. 119, no. 2, pp. 277–287, 2013.

[65] S. Hosseinpour, S. Rafiee, S. S. Mohtasebi, and M. Aghbashlo, "Application of computer vision technique for on-line monitoring of shrimp color changes during drying," *Journal of Food Engineering*, vol. 115, no. 1, pp. 99–114, 2013.

[66] N. A. Valous, F. Mendoza, D.-W. Sun, and P. Allen, "Colour calibration of a laboratory computer vision system for quality evaluation of pre-sliced hams," *Meat Science*, vol. 81, no. 1, pp. 132–141, 2009.

[67] D. Filko, R. Cupec, and E. K. Nyarko, "Evaluation of color and texture descriptors for matching of planar surfaces in global localization scheme," *Robotics and Autonomous Systems*, vol. 80, pp. 55–68, 2016.

[68] Y. Shi, "Adaptive illumination correction considering ordinal characteristics," in *Proceedings of the 2010 6th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2010*, China, September 2010.

[69] S. Rahman, M. M. Rahman, M. Abdullah-Al-Wadud, G. D. Al-Quaderi, and M. Shoyaib, "An adaptive gamma correction for image enhancement," *Eurasip Journal on Image and Video Processing*, vol. 2016, no. 1, article no. 35, 2016.

[70] J. Qi, S. Dong, F. Huang, and H. Lu, "Saliency detection via joint modeling global shape and local consistency," *Neurocomputing*, vol. 222, pp. 81–90, 2017.

[71] P. G. Cavalcanti and J. Scharcanski, "Macroscopic Pigmented Skin Lesion Segmentation and Its Influence on Lesion Classification and Diagnosis," in *Color Medical Image Analysis*, vol. 6 of *Lecture Notes in Computational Vision and Biomechanics*, pp. 15–39, Springer Netherlands, Dordrecht, 2013.

[72] J. R. Van Aken, "An Efficient Ellipse-Drawing Algorithm," *IEEE Computer Graphics and Applications*, vol. 4, no. 9, pp. 24–35, 1984.

[73] A. Agathos, T. Theoharis, and A. Boehm, "Efficient integer algorithms for the generation of conic sections," *Computers and Graphics*, vol. 22, no. 5, pp. 621–628, 1998.

[74] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations," *Color Research & Application*, vol. 30, no. 1, pp. 21–30, 2005.

[75] D. R. Pant and I. Farup, "Riemannian formulation of the CIEDE2000 color difference formula," in *Proceedings of the 18th Color and Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications, CIC18 2010*, pp. 103–108, usa, November 2010.

[76] X. Li, B. Aldridge, L. Ballerini, R. Fisher, and J. Rees, "Depth data improves skin lesion segmentation.," *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 12, no. 2, pp. 1100–1107, 2009.

[77] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pp. 473–480, USA, June 2011.

[78] R. Margolin, A. Tal, and L. Zelnik-Manor, "What makes a patch distinct?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 1139–1146, IEEE, Portland, Ore, USA, June 2013.

[79] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing Markov chain," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 1665–1672, IEEE, Sydney, Australia, December 2013.

[80] R. Melli, C. Grana, and R. Cucchiara, "Comparison of color clustering algorithms for segmentation of dermatological images," in *Proceedings of the Medical Imaging 2006: Image Processing*, vol. 6144 of *Proceedings of SPIE*, pp. 3S1–3S9, San Diego, Calif, USA, February 2006.

[81] H. Castillejos, V. Ponomaryov, L. Nino-De-Rivera, and V. Golikov, "Wavelet transform fuzzy algorithms for dermoscopic image segmentation," *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 578721, 11 pages, 2012.

[82] L.-K. Huang and M.-J. J. Wang, "Image thresholding by minimizing the measures of fuzziness," *Pattern Recognition*, vol. 28, no. 1, pp. 41–51, 1995.

[83] K.-A. Norton, H. Iyatomi, M. E. Celebi et al., "Three-phase general border detection method for dermoscopy images using non-uniform illumination correction," *Skin Research and Technology*, vol. 18, no. 3, pp. 290–300, 2012.

[84] Z. Wang, G. Xu, Z. Wang, and C. Zhu, "Saliency detection integrating both background and foreground information," *Neurocomputing*, vol. 216, pp. 468–477, 2016.

[85] L. Bi, J. Kim, E. Ahn, D. Feng, and M. Fulham, "Automated skin lesion segmentation via image-wise supervised learning and multi-scale superpixel based cellular automata," in *Proceedings of the 2016 IEEE 13th International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2016*, pp. 1059–1062, Czech Republic, April 2016.

[86] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[87] H.-L. Wang, M. Zhu, C.-B. Lin, and D.-B. Chen, "Ship detection in optical remote sensing image based on visual saliency and AdaBoost classifier," *Optoelectronics Letters*, vol. 13, no. 2, pp. 151–155, 2017.

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 6921742, 12 pages
https://doi.org/10.1155/2018/6921742

Research Article

Total Variation Image Restoration Method Based on Subspace Optimization

XiaoGuang Liu [1] **and XingBao Gao** [2]

[1] *School of Computer Science and Technology, Southwest University for Nationalities, Chengdu, Sichuan 610041, China*
[2] *College of Mathematics and Information Science, Shaanxi Normal University, Xian 710062, China*

Correspondence should be addressed to XiaoGuang Liu; dtcr-gg@163.com

Received 5 October 2017; Accepted 13 December 2017; Published 21 January 2018

Academic Editor: Daniel Zaldivar

The alternating direction method is widely applied in total variation image restoration. However, the search directions of the method are not accurate enough. In this paper, one method based on the subspace optimization is proposed to improve its optimization performance. This method corrects the search directions of primal alternating direction method by using the energy function and a linear combination of the previous search directions. In addition, the convergence of the primal alternating direction method is proven under some weaker conditions. Thus the convergence of the corrected method could be easily obtained since it has same convergence with the primal alternating direction method. Numerical examples are given to show the performance of proposed method finally.

1. Introduction

Digital image restoration has a wide application in various areas including Navigation, Aerospace, and Biomedicine (see [1–3] and the references therein). In general, the relationship between the original image $\widehat{f} \in R^p$ and the observed image $g \in R^q$ is

$$g = A\widehat{f} + b, \tag{1}$$

where $b \in R^q$ is the additive noise and the spatial-invariant matrix $A \in R^{q\times p}$ represents the degradation system caused by problems such as motion blur, distortion radiation, and distortion wavelets in seismic imaging, $p = p_1 p_2$ with p_1 and p_2 being the number of rows and columns, respectively, when images are expressed as a matrix. In particular, matrix A represents Block Toeplitz-plus-Hankel matrix (with Toeplitz-plus-Hankel blocks) or Block Toeplitz matrix (with Toeplitz blocks) when Neumann boundary condition and the zero boundary condition are used, respectively [4].

The objective of the image restoration is to estimate the original image $\widehat{f}$ according to some a priori knowledge about the degradation system A, the additive noise b, and the observed image g. However, it often tends to be very ill-conditioned when the reverse process of model (1) is only used to get an ideal estimation f^*. Thus one of the effective ways to solve these problems is to combine some a priori information of the original image and define the regularization solution; that is, f^* is a minimizer of the following cost (energy) function:

$$J(f) = \theta(Af - g) + \alpha\Phi(f), \tag{2}$$

where $\theta : R^q \to R$ is the measure of the difference between Af and g. The regularization term Φ embodies a priori information and a regularization parameter $\alpha > 0$ is used to control the tradeoff between the terms θ and Φ.

In general, $\theta(x) = \|x\|_2^2 = \sum_{i=1}^n x_i^2$ (the square of l_2-norm of vector $x \in R^n$) when the additive noise meets Gaussian distribution. $\theta(x) = \|x\|_1 = \sum_{i=1}^n |x_i|$ (l_1-norm of vector $x \in R^n$) when the additive noise meets non-Gaussian distributions such as uniform and speckle.

The regularization term $\Phi(f) = \sum_{i\in I} \phi(\|D_i f\|_2)$, where $I = \{1, 2, \ldots, p\}$, $\phi : R \to R_+ = \{t \in R : t \geq 0\}$ is the potential function, $D_i : R^p \to R^d$ is the difference operator which can be seen as a $d \times p$ matrix and used to create the difference

vector between ith pixel and its d neighboring pixels. Here $D_i = [D_i^1; D_i^2]$ with

$$\begin{aligned} &D_i^1[j] = 0 \quad \forall j \in I, \text{ if } i \in I_1 = \{p_2, 2p_2, \ldots, p_1p_2\}, \\ &D_i^1[i] = -1, \ D_i^1[i+1] = 1, \ D_i^1[j] = 0 \\ &\qquad \forall j \notin \{i, i+1\}, \text{ if } i \notin I_1, \end{aligned} \tag{3}$$

$$\begin{aligned} &D_i^2[j] = 0 \\ &\forall j \in I, \text{ if } i \in I_2 = \{(p_1-1)p_2 + 1, (p_1-1)p_2 + 2, \ldots, p_1p_2\}, \\ &D_i^2[i] = -1, \ D_i^2[i+p_2] = 1, \ D_i^2[j] = 0 \\ &\qquad \forall j \notin \{i, i+p_2\}, \text{ if } i \notin I_2. \end{aligned} \tag{4}$$

In image restoration, the potential function ϕ plays a key role so that it is intensively studied in recent decades [5–20]. Two classes of regularization terms are well known. One is the Tikhonov class [5–7], for example, $\Phi(f) = \sum_{i\in I} \|D_i f\|_2^2$. In these cases, the minimum point of (2) can be easily solved due to differentiability. However, these methods tend to make restored images overly smooth and could not protect the clear boundaries.

The other class is based on total variation (TV) regularization [8]:

$$\Phi(f) = \|f\|_{\mathrm{TV}} = \sum_{i\in I} \|D_i f\|_2 = \sum_{i\in I} \sqrt{(D_i^1 f)^2 + (D_i^2 f)^2}. \tag{5}$$

In comparison to the first class, an obvious benefit is that the clear boundaries can be recovered well. Thus many effective methods are proposed to solve the TV deblurring problem (see [8–23]). However, they suffer from some numerical difficulties, since the TV regularization is nondifferentiable. In order to overcome the drawback, in [16], the TV regularization is modified as

$$\Phi(f) = \|f\|_{\mathrm{TV},\varepsilon} = \sum_{i\in I} \sqrt{(D_i^1 f)^2 + (D_i^2 f)^2 + \varepsilon}, \tag{6}$$

where $0 < \varepsilon \ll 1$. Obviously, the regularization is differentiable and many classical optimization methods can be applied [24, 25]. Unfortunately, experimental results show that ε must be small enough to keep the restoration quality, but the efficiency of algorithms will be reduced with smaller ε [16, 20].

By using the following modified TV regularization,

$$\Phi(f) = \sum_{i\in I} \left|D_i^1 f\right| + \left|D_i^2 f\right|, \tag{7}$$

a primal-dual active set method is proposed in [17], but it has no rotational invariant [8].

Recently, the alternating direction method [26] is used to solve the l_2-TV and l_1-TV image restoration models, and better performance has been obtained [18–23, 27]. It divides original problem into some simple subproblems and solves them alternatively. The downside is that the search directions are not accurate enough which could reduce the performance of algorithms (see the analysis in Section 3).

Based on the considerations above, we will take the algorithms in [18, 20] as examples and present a method to improve the performance of the alternating direction method used in l_2-TV and l_1-TV models. By means of the subspace optimization [28], it corrects the search direction of the primal alternating direction method by using the energy function and a linear combination of the previous search directions. In addition, the convergence of the primal alternating direction method is proven under some weaker conditions. Thus the convergence of the corrected method could be easily obtained by the equivalence between them. Numerical examples are given to shown the performance of proposed method.

The outline of this paper is as follows. Some preliminaries are stated in Section 2. A l_1-TV image restoration algorithm based on subspace optimization is proposed in Section 3. The convergence analysis and numerical examples are given in Sections 4 and 5, respectively. Finally, some concluding remarks are given in Section 6.

2. Preliminaries

In this section, we first propose some basic definitions and properties; then the alternating direction method to solve l_1-TV model in [20] will be introduced. The alternating direction method in [18] (l_2-TV model) will be omitted here since it is more simpler. But the comparison experimental results of the method and its corrected method still will be listed in Section 5.2. In the following, let D_h and D_v be the one-side difference matrix on the horizontal direction and vertical direction, respectively, $D = (D_h, D_v)^\top$, $\mathrm{Null}(A)$ is the null space of matric A, x^i is the ith iteration, and $x_{(i)}$ is the ith element of vector x.

Definition 1 (see [20]). An operator $\mathscr{P}$ is called nonexpansive if for any $x_1, x_2 \in X \subset R^n$, we have

$$\|\mathscr{P}(x_1) - \mathscr{P}(x_2)\|_2 \le \|x_1 - x_2\|_2. \tag{8}$$

Specially, $\mathscr{P}$ is called β-averaged nonexpansive if there exists some nonexpansive operator $\mathscr{A}$ and $\beta \in (0, 1)$ such that $\mathscr{P} = (1-\beta)\mathscr{I} + \beta\mathscr{A}$, where $\mathscr{I}$ is the identity operator.

Lemma 2 (see [20]). *If $\rho > 0$, then the minimizer of $\psi(s) = |x-s|^2 + \rho|s|$ is given by the following:*

$$\nu(x) = \begin{cases} x - \dfrac{\rho}{2}, & x > \dfrac{\rho}{2}, \\ 0, & |x| \le \dfrac{\rho}{2}, \\ x + \dfrac{\rho}{2}, & x < -\dfrac{\rho}{2}. \end{cases} \tag{9}$$

We know from [20] that the operator $\nu(x)$ is nonexpansive.

Lemma 3 (see [29]). *Let φ be convex and semicontinuous, $\beta > 0$ and*

$$\hat{x} = \arg\min_x \|y - x\|_2^2 + \beta\varphi(x). \tag{10}$$

Define $\mathcal{S}$ such that $\hat{x} = \mathcal{S}(y)$ for each y; then $\mathcal{S}$ is (1/2)-averaged nonexpansive.

Definition 4 (see [25]). A function $\varphi : R^n \to R$ (i) is said to be proper over a set $X \subset R^n$ if $\varphi(x) < +\infty$ for at least one $x \in X$ and $\varphi(x) > -\infty$ for all $x \in X$ and (ii) is said to be coercive over a set $X \subset R^n$ if for every sequence $\{x_k\} \subset X$ such that $\|x_k\|_2 \to \infty$ we have

$$\lim_{k\to\infty} \varphi(x_k) = \infty. \tag{11}$$

When $X = R^n$, we say that φ is coercive on R^n.

Lemma 5 (see [30]). *Let $\varphi : R^n \to R$ be a closed, proper, and coercive function. Then the set of minima of φ over R^n is nonempty and compact.*

In [20], the l_1-TV image restoration model is

$$\arg\min_f \|Af - g\|_1 + \alpha \|f\|_{TV}. \tag{12}$$

To avoid the numerical difficulties caused by the nondifferentiability, two auxiliary variables $\omega \in R^q$ and $\eta \in R^p$ are used to cope with the nonsmooth terms $\|Af - g\|_1$ and $\|f\|_{TV}$. Then (12) has been transformed as

$$\begin{aligned}\arg\min_{f,\omega,\eta} F(f,\omega,\eta) = {} & \|Af - \omega\|_2^2 + \alpha \|\eta\|_{TV} \\ & + \alpha_1 \|\omega - g\|_1 + \alpha_2 \|\eta - f\|_2^2,\end{aligned} \tag{13}$$

where the parameters $\alpha_1 > 0$ and $\alpha_2 > 0$ are used to ensure the closeness of g and ω, f and η, respectively. When applied to (13), the alternating direction method is

$$\begin{aligned}\mathscr{H}_A(\omega^i, \eta^i) &:= f^{i+1} \\ &= \arg\min_f \|Af - \omega^i\|_2^2 + \alpha_2 \|\eta^i - f\|_2^2, \\ \mathscr{H}_{l_1}(f^{i+1}) &:= \omega^{i+1} \\ &= \arg\min_\omega \|Af^{i+1} - \omega\|_2^2 + \alpha_1 \|\omega - g\|_1, \\ \mathscr{H}_{TV}(f^{i+1}) &:= \eta^{i+1} \\ &= \arg\min_\eta \alpha_2 \|\eta - f^{i+1}\|_2^2 + \alpha \|\eta\|_{TV}.\end{aligned} \tag{14}$$

The first step in (14) is equivalent to solving the nonsingular linear system:

$$(A^\top A + \alpha_2 I) f = A^\top \omega^i + \alpha_2 \eta^i. \tag{15}$$

It can be solved directly; that is,

$$f = (A^\top A + \alpha_2 I)^{-1} (A^\top \omega^i + \alpha_2 \eta^i). \tag{16}$$

To let computational cost be low, many classical optimization and numerical methods such as CG method and preconditioned iteration methods can be used to solve it [24, 25, 31]. As in [4, 18], we suppose that the Neumann boundary conditions are used and the blurring function is symmetric in this paper; then A is a (block) Toeplitz-plus-Hankel matrix and it could be diagonalized by a discrete cosine transform matrix. Thus we could solve the nonsingular linear system utilizing lower cost, since the inverse of blurring matrix A can be computed by using fast cosine transforms; please see [4, 18, 27] for more details.

Since

$$\begin{aligned}&\|Af^{i+1} - \omega\|_2^2 + \alpha_1 \|\omega - g\|_1 \\ &\quad = \sum_{k=1}^{q} \left\{ \left((Af^{i+1})_{(k)} - \omega_{(k)} \right)^2 + \alpha_1 |\omega_{(k)} - g_{(k)}| \right\},\end{aligned} \tag{17}$$

the second step in (14) is equivalent to solving the minimizers of q functions in the form of $\nu(s) = |x - s|^2 + \alpha_1 |s|$. We can know from Lemma 2 that the kth element of its solution ω^{i+1} is

$$\omega_{(k)}^{i+1} = \begin{cases} (Af^{i+1})_{(k)} - \dfrac{\alpha_1}{2}, & (Af^{i+1} - g)_{(k)} > \dfrac{\alpha_1}{2}, \\ g_{(k)}, & \left|(Af^{i+1} - g)_{(k)}\right| \le \dfrac{\alpha_1}{2}, \\ (Af^{i+1})_{(k)} + \dfrac{\alpha_1}{2}, & (Af^{i+1} - g)_{(k)} < -\dfrac{\alpha_1}{2}. \end{cases} \tag{18}$$

The third step in (14) is a l_2-TV model with the blurring matrix A is the identity matrix, that is, image denoising problem. Many methods can solve it effectively [13–15], and Chambolle' projection algorithm [13] is used in [20].

3. The Method Based on Subspace Optimization

Based on the principle of alternating direction method above, it is easy to know that the solution $l^{i+1} = [f^{i+1}; \omega^{i+1}; \eta^{i+1}]$ of (14) is different from the solution of (13), so the performance of solving the minima of (13) may be lowered. Thus a method based on subspace optimization [28] will be proposed to improve the optimization performance of (14) in this section.

Firstly, we introduce subspace optimization method. Let $\bar{l}^{i+1} = [\bar{f}^{i+1}; \bar{\omega}^{i+1}; \bar{\eta}^{i+1}]$ denote the solution corrected by subspace optimization method, and

$$S^{i+1} = \left[t_0^{i+1}, t_1^{i+1}, \ldots, t_m^{i+1} \right], \tag{19}$$

where $t_0^{i+1} = l^{i+1} - \bar{l}^i$ and $t_k^{i+1} = \bar{l}^{i+1-k} - \bar{l}^{i-k}$ $(1 \le k \le m \le i)$, then

$$\bar{l}^{i+1} = \bar{l}^i + S^{i+1} \pi^{i+1}, \tag{20}$$

$$\pi^{i+1} = \arg\min_\pi F\left(\bar{l}^i + S^{i+1} \pi \right). \tag{21}$$

Supposing $\pi^{i+1} \in R^{m+1}$ is selected as $(1, 0, \ldots, 0)$ in (20), then the equation $\bar{l}^{i+1} = l^{i+1}$ is valid. In fact, we can know from (21) that π^{i+1} is a minimum point of the cost function: $F(\pi^{i+1}) = F(\bar{l}^i + S^{i+1}\pi)$; thus $F(\bar{l}^{i+1}) \le F(l^{i+1})$ must be true. It is helpful to improve the performance of original alternating direction method on every iteration, and the key problem is how to solve (21) now. To reduce the computational cost, let $S^{i+1} = [t_0^{i+1}]$ in this paper.

Next, the correction method of (14) will be proposed according to three cases.

Case 1 ($\omega_{(i)} \neq g_{(i)}$ and $D_i^1\eta \neq 0$ or $D_i^2\eta \neq 0$ for $\forall i \in I$). Obviously, the function $F(f,\omega,\eta)$ is smooth; we have

$$\nabla F = \begin{pmatrix} 2A^\top(Af-\omega) + 2\alpha_2(f-\eta) \\ 2(\omega - Af) + \alpha_1 \sum_{i\in I} \dfrac{\omega_{(i)} - g_{(i)}}{|\omega_{(i)} - g_{(i)}|} e_i \\ 2\alpha_2(\eta - f) + \alpha \sum_{i\in I} x_i^{-1/2} y_i z_i^\top \end{pmatrix},$$

$$\nabla^2 F = \begin{pmatrix} 2A^\top A + 2\alpha_2 I & -2A^\top & -2\alpha_2 I \\ -2A & 2I & 0 \\ -2\alpha_2 I & 0 & 2\alpha_2 I + \dfrac{\partial^2 \|\eta\|_{\mathrm{TV}}}{\partial \eta^2} \end{pmatrix}, \tag{22}$$

where e_i denotes the unit column vector with its ith element being 1 and the others being zero, $x_i = (D_i^1\eta)^2 + (D_i^2\eta)^2$, $y_i = D_i^1\eta + D_i^2\eta$, and $z_i = D_i^1 + D_i^2$. It is expensive to compute $\partial^2\|\eta\|_{\mathrm{TV}}/\partial\eta^2$, so we let $\partial^2\|\eta\|_{\mathrm{TV}}/\partial\eta^2 = 0$ to improve the efficiency of the algorithm because $\nabla^2 F$ is independent of variable l now.

On the other hand, since π^{i+1} is an optimum of the convex function F, we can know from (21) and Taylor series expansion that

$$\begin{aligned} \left(S^{i+1}\right)^\top \nabla F\left(\bar{l}^i + S^{i+1}\pi^{i+1}\right) &= 0, \\ \nabla F\left(\bar{l}^i + S^{i+1}\pi^{i+1}\right) &= \nabla F\left(\bar{l}^i\right) + \nabla^2 F S^{i+1}\pi^{i+1}. \end{aligned} \tag{23}$$

Equation (23) implies that

$$\left(S^{i+1}\right)^\top \left(\nabla F\left(\bar{l}^i\right) + \nabla^2 F S^{i+1}\pi^{i+1}\right) = 0. \tag{24}$$

Obviously, vector $S^{i+1} \neq 0$; then

$$\begin{aligned} \frac{1}{2}\left(S^{i+1}\right)^\top \nabla^2 F S^{i+1} &= \left\|A\left(f^{i+1} - \overline{f}^i\right)\right\|^2 + \left\|\omega^{i+1} - \overline{\omega}^i\right\|^2 \\ &\quad - 2\left(\omega^{i+1} - \overline{\omega}^i\right)^\top A\left(f^{i+1} - \overline{f}^i\right) + \alpha_2\left[\left\|f^{i+1} - \overline{f}^i\right\|^2\right. \\ &\quad \left. + \left\|\eta^{i+1} - \overline{\eta}^i\right\|^2 - 2\left(\eta^{i+1} - \overline{\eta}^i\right)^\top\left(f^{i+1} - \overline{f}^i\right)\right] \\ &= \left\|A\left(f^{i+1} - \overline{f}^i\right) - \omega^{i+1} + \overline{\omega}^i\right\|_2^2 + \alpha_2\left\|f^{i+1} - \overline{f}^i\right. \\ &\quad \left. - \eta^{i+1} + \overline{\eta}^i\right\|_2^2 \geq 0. \end{aligned} \tag{25}$$

Thus $(S^{i+1})^\top\nabla^2 F S^{i+1} = 0 \Leftrightarrow \|A(f^{i+1} - \overline{f}^i) - \omega^{i+1} + \overline{\omega}^i\|_2^2 + \alpha_2\|f^{i+1} - \overline{f}^i - \eta^{i+1} + \overline{\eta}^i\|_2^2 = 0 \Leftrightarrow A(f^{i+1} - \overline{f}^i) = \omega^{i+1} - \overline{\omega}^i$ and $f^{i+1} - \overline{f}^i = \eta^{i+1} - \overline{\eta}^i$. The two equalities do not hold in image restoration since the dimension of vector f is very huge. Therefore $(S^{i+1})^\top\nabla^2 F S^{i+1} > 0$, and

$$\pi^{i+1} = -\left(\left(S^{i+1}\right)^\top \nabla^2 F S^{i+1}\right)^{-1}\left(\left(S^{i+1}\right)^\top \nabla F\left(\bar{l}^i\right)\right) \tag{26}$$

from (24).

Case 2 ($I_3 = \{i \mid D_i^1\eta = D_i^2\eta = 0,\ i \in I\}$ is nonempty). Now, $\|\eta\|_{\mathrm{TV}}$ is nonsmooth. Let

$$\begin{aligned} \|\eta\|_{\mathrm{TV}+\varepsilon} &= \sum_{i\in I_3}\left(\left(D_i^1\eta\right)^2 + \left(D_i^2\eta\right)^2 + \varepsilon\right)^{1/2} \\ &\quad + \sum_{i\in I/I_3}\left(\left(D_i^1\eta\right)^2 + \left(D_i^2\eta\right)^2\right)^{1/2}, \end{aligned} \tag{27}$$

then $\lim_{\varepsilon\to 0}\|\eta\|_{\mathrm{TV}+\varepsilon} = \|\eta\|_{\mathrm{TV}}$ and $\partial\|\eta\|_{\mathrm{TV}+\varepsilon}/\partial\eta_{(i)} = (x_i + \varepsilon)^{-1/2} y_i z_i^\top = 0$ ($y_i = 0$) when $i \in I_3$. Thus $\partial\|\eta\|_{\mathrm{TV}}/\partial\eta_{(i)} = 0$ is set when $i \in I_3$.

Case 3 ($\exists i \in I$ such that $\omega_{(i)} = g_{(i)}$). The term $\alpha_1\|\omega_{(i)} - g_{(i)}\|_1$ is not differential at this case; the Huber function is often used to replace it [25]. For simplicity, $\partial\|\omega - g\|_1/\partial\omega_{(i)} = 0$ is selected when $\omega_{(i)} = g_{(i)}$.

In summary, the proposed algorithm is as follows.

Algorithm

Step 1. Initialize $\bar{l}^0$.

Step 2. Solve

(i) $f^{i+1} = \arg\min_f \|Af - \omega^i\|_2^2 + \alpha_2\|\eta^i - f\|_2^2$,

(ii) $\omega^{i+1} = \arg\min_\omega \|Af^{i+1} - \omega\|_2^2 + \alpha_1\|\omega - g\|_1$,

(iii) $\eta^{i+1} = \arg\min_\eta \alpha_2\|\eta - f^{i+1}\|_2^2 + \alpha\|\eta\|_{\mathrm{TV}}$.

Step 3. Compute π^{i+1} using (26), and let $\bar{l}^{i+1} = \bar{l}^i + S^{i+1}\pi^{i+1}$.

Step 4. If $\|f^{i+1} - f^i\|_2/\|f^{i+1}\|_2 < 10^{-4}$, then stop; otherwise go to Step 2.

4. Convergence Analysis

As known from (20) and (26), when $\lim_{i\to\infty}\|t_0^i\|_2 = 0$ and one of the sequences $\{l^i\}$ and $\{\bar{l}^i\}$ is convergent, then the other sequence is also convergent, where $t_0^i = l^i - \bar{l}^{i-1}$. Thus a simple proof of the convergence of algorithm (14) will be given below without the condition that $\mathscr{H}_{l_1}(\mathscr{H}_A(\cdot,\eta))$ and $\mathscr{H}_{\mathrm{TV}}(\mathscr{H}_A(\omega,\cdot))$ are asymptotically regular which is needed in [20], where

$$\begin{aligned} \omega^{i+1} &= \mathscr{H}_{l_1}\left(f^{i+1}\right) = \mathscr{H}_{l_1}\left(\mathscr{H}_A\left(\omega^i,\eta^i\right)\right) \\ &= \mathscr{T}_1\left(\omega^i,\eta^i\right), \end{aligned}$$

$$\begin{aligned}\eta^{i+1} &= \mathcal{H}_{\mathrm{TV}}\left(f^{i+1}\right) = \mathcal{H}_{\mathrm{TV}}\left(\mathcal{H}_A\left(\omega^i, \eta^i\right)\right)\\ &= \mathcal{T}_2\left(\omega^i, \eta^i\right),\end{aligned} \tag{28}$$

$$\begin{aligned}\mathcal{T}_1(\cdot) &= \mathcal{H}_{l_1}\left(\mathcal{H}_A(\cdot, \eta)\right),\\ \mathcal{T}_2(\cdot) &= \mathcal{H}_{\mathrm{TV}}\left(\mathcal{H}_A(\omega, \cdot)\right).\end{aligned} \tag{29}$$

Lemmas 6 and 7 below appeared in [18, 20], but the proofs given by us are simpler.

Lemma 6. *Suppose $Null(A) \cap Null(D) = \{0\}$; then the sets of fixed points of operators $\mathcal{T}_1$ and $\mathcal{T}_2$ are nonempty, respectively.*

Proof. It is easy to see that the objection function F in (13) satisfies the conditions of Lemma 5 [18, 20]. Then F has at least one minimizer $(\overline{f}, \overline{\omega}, \overline{\eta})$ that cannot be decreased by the alternating scheme (14). Thus

$$\begin{aligned}\overline{f} &= \mathcal{H}_A\left(\overline{\omega}, \overline{\eta}\right) = \arg\min F\left(\cdot, \overline{\omega}, \overline{\eta}\right),\\ \overline{\omega} &= \mathcal{H}_{l_1}\left(\overline{f}\right) = \arg\min F\left(\overline{f}, \cdot, \overline{\eta}\right),\\ \overline{\eta} &= \mathcal{H}_{\mathrm{TV}}\left(\overline{f}\right) = \arg\min F\left(\overline{f}, \overline{\omega}, \cdot\right),\end{aligned} \tag{30}$$

$$\begin{aligned}\overline{\omega} &= \mathcal{H}_{l_1}\left(\mathcal{H}_A\left(\overline{\omega}, \overline{\eta}\right)\right) = \mathcal{T}_1\left(\overline{\omega}\right),\\ \overline{\eta} &= \mathcal{H}_{\mathrm{TV}}\left(\mathcal{H}_A\left(\overline{\omega}, \overline{\eta}\right)\right) = \mathcal{T}_2\left(\overline{\eta}\right).\end{aligned} \tag{31}$$

Therefore $\overline{\omega}, \overline{\eta}$ are the fixed points of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. □

Lemma 7. *The operators $\mathcal{T}_1$ and $\mathcal{T}_2$ are nonexpansive.*

Proof. From Lemmas 2 and 3, we have

$$\begin{aligned}\left\|\mathcal{T}_1\omega_1 - \mathcal{T}_1\omega_2\right\|_2 &= \left\|\mathcal{H}_{l_1}\left(\mathcal{H}_A(\omega_1, \eta)\right)\right.\\ &\left. - \mathcal{H}_{l_1}\left(\mathcal{H}_A(\omega_2, \eta)\right)\right\|_2 = \left\|\psi\left(A\mathcal{H}_A(\omega_1, \eta)\right)\right.\\ &\left. - \psi\left(A\mathcal{H}_A(\omega_2, \eta)\right)\right\|_2 \le \left\|A\mathcal{H}_A(\omega_1, \eta)\right.\\ &\left. - A\mathcal{H}_A(\omega_2, \eta)\right\|_2\\ &= \left\|A\left(A^\top A + \alpha_2 I\right)^{-1}\left(A^\top\omega_1 + \alpha_2\eta\right)\right.\\ &\left. - A\left(A^\top A + \alpha_2 I\right)^{-1}\left(A^\top\omega_2 + \alpha_2\eta\right)\right\|_2\\ &= \left\|A\left(A^\top A + \alpha_2 I\right)^{-1}A^\top\left(\omega_1 - \omega_2\right)\right\|_2 \le \left\|\omega_1 - \omega_2\right\|_2,\end{aligned}$$

$$\begin{aligned}\left\|\mathcal{T}_2\eta_1 - \mathcal{T}_2\eta_2\right\|_2 &= \left\|\mathcal{H}_{\mathrm{TV}}\left(\mathcal{H}_A(\omega, \eta_1)\right)\right.\\ &\left. - \mathcal{H}_{\mathrm{TV}}\left(\mathcal{H}_A(\omega, \eta_2)\right)\right\|_2 \le \left\|\mathcal{H}_A(\omega, \eta_1)\right.\\ &\left. - \mathcal{H}_A(\omega, \eta_2)\right\|_2\\ &= \left\|A\left(A^\top A + \alpha_2 I\right)^{-1}\left(A^\top\omega + \alpha_2\eta_1\right)\right.\\ &\left. - A\left(A^\top A + \alpha_2 I\right)^{-1}\left(A^\top\omega + \alpha_2\eta\right)_2\right\|_2\\ &= \left\|A\left(A^\top A + \alpha_2 I\right)^{-1}A^\top\left(\eta_1 - \eta_2\right)\right\|_2 \le \left\|\eta_1 - \eta_2\right\|_2.\end{aligned} \tag{32}$$

This completes the proof. □

Lemma 8. *Let $\{\omega^i\}$ and $\{\eta^i\}$ be generated by (28); then $\sum_{k=1}^{\infty}\left\|\omega^{i_k+1} - \omega^{i_k}\right\|_2^2$ and $\sum_{k=1}^{\infty}\left\|\eta^{i_k+1} - \eta^{i_k}\right\|_2^2$ are bounded, where ω^{i_k} and η^{i_k} are the subsequence of $\{\omega^i\}$ and $\{\eta^i\}$, respectively.*

Proof. For $F(f, \omega^{i_k}, \eta^{i_k})$ in (13), we have

$$\begin{aligned}F\left(f^{i_k}, \omega^{i_k}, \eta^{i_k}\right) &= F\left(f^{i_k+1}, \omega^{i_k}, \eta^{i_k}\right)\\ &+ \left(f^{i_k} - f^{i_k+1}\right)^\top\frac{\partial F}{\partial f}\left(f^{i_k+1}, \omega^{i_k}, \eta^{i_k}\right)\\ &+ \frac{1}{2}\left(f^{i_k} - f^{i_k+1}\right)^\top\left(A^\top A + \alpha_2 I\right)\left(f^{i_k+1}, \omega^{i_k}, \eta^{i_k}\right)\\ &\cdot\left(f^{i_k} - f^{i_k+1}\right).\end{aligned} \tag{33}$$

Since f^{i_k+1} is the minimizer of $F(f, \omega^{i_k}, \eta^{i_k})$, so

$$\frac{\partial F}{\partial f}\left(f^{i_k+1}, \omega^{i_k}, \eta^{i_k}\right) = 0. \tag{34}$$

Then

$$\begin{aligned}F\left(f^{i_k}, \omega^{i_k}, \eta^{i_k}\right) - F\left(f^{i_k+1}, \omega^{i_k}, \eta^{i_k}\right)\\ \ge \alpha_2\left\|f^{i_k+1} - f^{i_k}\right\|_2^2.\end{aligned} \tag{35}$$

Notice that $F(f^{i_k+1}, \omega^{i_k+1}, \eta^{i_k+1}) \le F(f^{i_k+1}, \omega^{i_k}, \eta^{i_k})$, we get

$$\begin{aligned}F\left(f^{i_k}, \omega^{i_k}, \eta^{i_k}\right) - F\left(f^{i_k+1}, \omega^{i_k+1}, \eta^{i_k+1}\right)\\ \ge \alpha_2\left\|f^{i_k+1} - f^{i_k}\right\|_2^2.\end{aligned} \tag{36}$$

From Lemma 2, $\mathcal{H}_{l_1}$ is nonexpansive; that is,

$$\begin{aligned}\left\|f^{i_k+1} - f^{i_k}\right\|_2^2 &\ge \left\|\mathcal{H}_{l_1}\left(f^{i_k+1}\right) - \mathcal{H}_{l_1}\left(f^{i_k}\right)\right\|_2^2\\ &= \left\|\omega^{i_k+1} - \omega^{i_k}\right\|_2^2,\end{aligned} \tag{37}$$

(a) (b) (c) (d)

Figure 1: The restored Camera images by different methods with the uniform noise. The random numbers are in the interval (0, 0.2), the support being equal to 9×9. (a) Original image. (b) Observed image. (c) Image restored by algorithm in [20]. (d) Image restored by our method.

and thus

$$F\left(f^{i_k}, \omega^{i_k}, \eta^{i_k}\right) - F\left(f^{i_k+1}, \omega^{i_k+1}, \eta^{i_k+1}\right) \geq \alpha_2 \left\|\omega^{i_k+1} - \omega^{i_k}\right\|_2^2. \tag{38}$$

On the other hand,

$$F\left(f^{i_{k+1}}, \omega^{i_{k+1}}, \eta^{i_{k+1}}\right) \leq F\left(f^{i_k+1}, \omega^{i_k+1}, \eta^{i_k+1}\right), \tag{39}$$

and we have

$$\begin{aligned} &\sum_{k=1}^{\infty} \left[F\left(f^{i_k}, \omega^{i_k}, \eta^{i_k}\right) - F\left(f^{i_k+1}, \omega^{i_k+1}, \eta^{i_k+1}\right)\right] \\ &= F\left(f^{i_1}, \omega^{i_1}, \eta^{i_1}\right) + \sum_{k\to\infty} \left[F\left(f^{i_{k+1}}, \omega^{i_{k+1}}, \eta^{i_{k+1}}\right)\right. \\ &- F\left(f^{i_k+1}, \omega^{i_k+1}, \eta^{i_k+1}\right) \\ &\left. - F\left(f^{i_{k+1}+1}, \omega^{i_{k+1}+1}, \eta^{i_{k+1}+1}\right)\right] < F\left(f^{i_1}, \omega^{i_1}, \eta^{i_1}\right). \end{aligned} \tag{40}$$

Namely, $\sum_{k=1}^{\infty} \|\omega^{i_k+1} - \omega^{i_k}\|_2^2$ is bounded. Similarly, one can verify that $\sum_{k=1}^{\infty} \|\eta^{i_k+1} - \eta^{i_k}\|_2^2$ is also bounded.

Now we state and prove the convergence result of (14). □

Theorem 9. *Suppose $Null(A) \cap Null(D) = \{0\}$; the sequence $\{(\omega^i, \eta^i)\}$ generated by (14) with any initial point (ω^0, η^0) converges to a solution (ω^*, η^*) of (13).*

Proof. From Lemma 6, let $\overline{\omega}$ be any fixed point of $\mathscr{T}_1$; then

$$\begin{aligned} \left\|\omega^i - \overline{\omega}\right\|_2 &= \left\|\mathscr{T}_1\left(\omega^{i-1}\right) - \mathscr{T}_1\left(\overline{\omega}\right)\right\|_2 \leq \left\|\omega^{i-1} - \overline{\omega}\right\|_2 \\ &\leq \cdots \leq \left\|\omega^0 - \overline{\omega}\right\|_2 \end{aligned} \tag{41}$$

from Lemma 7. Thus the sequence $\{\omega^i\}$ is bounded, and there exists a subsequence $\{\omega^{i_k}\}$ with $\lim_{k\to\infty} \omega^{i_k} = \omega^*$. On the

(a) (b) (c) (d)

Figure 2: The restored Lena images by different methods with the speckle noise. The noise variance is 0.01, the support being equal to 7×7. (a) Original image. (b) Observed image. (c) Image restored by algorithm in [20]. (d) Image restored by our method.

other hand, $\sum_{k=1}^{\infty} \|\omega^{i_k+1} - \omega^{i_k}\|_2^2$ is bounded from Lemma 8; thus

$$\lim_{k\to\infty} \left\|\omega^{i_k+1} - \omega^{i_k}\right\|_2^2 = 0. \tag{42}$$

Let $k \to \infty$, then $\|\mathscr{T}_1(\omega^*) - \omega^*\|_2 = 0$; that is, ω^* is also a fixed point of $\mathscr{T}_1$. Similarly, to prove (41), one can also verify that the sequence $\{\|\omega^i - \omega^*\|_2\}$ is nonincreasing. Thus $\lim_{k\to\infty} \|\omega^i - \omega^*\|_2 = 0$ since $\lim_{k\to\infty} \omega^{i_k} = \omega^*$.

Following the argument above, one can prove that the sequence $\{\eta^i\}$ generated by (14) with any initial point η^0 converges to a solution η^* of (13). □

Remark 10. If $x \in \text{Null}(D)$, then $x_{(i)} = c$ for all $i \in I$, where c is a nonzero constant. Since matrix $A \geq 0$ in image restoration, so Ax is a nonzero vector. It follows that the assumption $\text{Null}(A) \cap \text{Null}(D) = \{0\}$ holds in general.

5. Experimental Results

In this section, some numerical results will be provided to show the performance of our method. As in [18, 20], the tested images are selected as Cameraman of 256×256 and Lena of 512×512. The algorithm in [20] and ours will be compared in Section 5.1 (l_1-TV model). The algorithm in [18] and its corrected algorithm will be compared in Section 5.2 (l_2-TV model). All the computational tasks are performed using MATLAB 2016a with Core(TM)2CPU with 2.83 GHz and 3.87 GB of RAM. The average value of ten tests would be selected.

The tested blurring function is chosen to be truncated 2D Gaussian function:

$$h(s,t) = \exp\left(\frac{-s^2 - t^2}{2\sigma^2}\right), \quad -3 \leq s,t \leq 3. \tag{43}$$

Here three sets of parameters are chosen: (1) the support being equal to 5×5 ($\sigma = 1$); (2) the support being equal to 7×7 ($\sigma = 1.5$); (3) the support being equal to 9×9 ($\sigma = 2$).

(a) (b) (c) (d)

Figure 3: The restored Camera images by different methods with Gaussian noise. The standard deviation of noise is 0.1; the support is 9×9. (a) Original image. (b) Observed image. (c) Image restored by algorithm in [18]. (d) Image restored by our method.

In all runs, CPU time is used to compare the efficiency, signal-to-noise ratio (SNR), and peak signal-to-noise ratio (PSNR):

$$\begin{aligned} \mathrm{SNR} &= 20 \log_{10} \left(\frac{\left\| \widehat{f} \right\|_2}{\left\| f^{i+1} - \widehat{f} \right\|_2} \right), \\ \mathrm{PSNR} &= -20 \log_{10} \left(\frac{\left\| f^{i+1} - \widehat{f} \right\|_2}{p_1 p_2} \right), \end{aligned} \tag{44}$$

which are used to measure the quality of the restored images. The stopping criterion of the algorithms should satisfies

$$\frac{\left\| f^{i+1} - f^{i} \right\|_2}{\left\| f^{i+1} \right\|_2} \leq 10^{-4}. \tag{45}$$

5.1. Comparison Experiment for the l_1-TV Model. In this subsection, a comparison of the algorithm in [20] with the proposed method is made under different non-Gaussian additive noises. Firstly, for the uniform noise, we let $\alpha = 0.008$, $\alpha_1 = 0.05$, and $\alpha_2 = 0.2$ as in [20]. The uniform random numbers appear in the intervals (0, 0.05), (0, 0.1), and (0, 0.2), respectively. Specially, [20] points out that the noise added to the blur image is independent and identically distributed, so the same set of parameters could be selected for different images when the same kind of noise is considered. This could reduce the computational cost of searching regularization parameters. For the speckle noise, as in [20, 23], let $\alpha \equiv 1$, $\alpha_1 = 2.5 \times 10^4$, the continuation scheme about α_2 will be used, easier subsequences with smaller α_2 can be solved quickly, and the later subproblems can also be solved relatively quickly with warm starts from previous solutions. The noise variance will be selected as 0, 0.01, and 0.05 respectively. To be fair, the same methods will be used to solve the primal alternating direction method in this paper.

Now, we select two cases that have significant improvement of the visual sense to illustrate our method's restoration quality. Under different conditions, more image restoration

(a) (b)

(c) (d)

Figure 4: The restored Lena images by different methods with Gaussian noise. The standard deviation of noise is 0.2; the support is 9×9. (a) Original image. (b) Observed image. (c) Image restored by algorithm in [18]. (d) Image restored by our method.

results with uniform noise and speckle noise will be summarized in Tables 1 and 2, respectively. Figure 1(a) shows the original Camera image. Figure 1(b) shows the observed Camera image, where the noise is the uniform noise, the random numbers are in the interval (0, 0.2), and the support is equal to 9×9. Figure 1(c) shows the restored Camera image by the algorithm in [20]. Figure 1(d) shows the restored Camera image by our method. Figure 2(a) shows the original Lena image. Figure 2(b) shows the observed Lena image, where the noise is the speckle noise, the noise variance is 0.01, the support is equal to 7×7. Figure 2(c) shows the restored Lena image by the algorithm in [20]. Figure 2(d) shows the restored Lena image by our method.

From Figures 1 and 2, we see that the quality of images restored by our method is improved obviously compared to those restored by the method of [20] in the two cases. For the uniform noise and speckle noise, Tables 1 and 2 show that the SNR and PSNR of the images restored by our method both are higher than the results of [20] in most cases. For the same requirement of relative error, our proposed algorithm is clearly faster than the competing algorithm in [20]. Referring to Sections 2 and 3, the proposed algorithm uses subspace optimization method to correct the search direction, which could ensure $F(\bar{l}^{i+1}) \leq F(l^{i+1})$ is valid on the ith iteration. So it is clear that the proposed method is more efficient. The analysis is verified by the experiments results above.

5.2. Comparison Experiment for the l_2-TV Model. In the subsection, the algorithm in [18] and ours will be compared. Being different from [19, 20], [18] involves a fitting of the auxiliary variable to Df which has superior performance compared to fitting f [18, 23, 32]. The additive noise is selected as Gaussian noise. Here four sets of the standard deviation of noise are chosen: 0.01, 0.05, 0.1, and 0.2. Same methods appearing in [18] will be used to solve the primal alternating direction method.

Similarly, we will first select two cases that have significant improvement of the visual sense to illustrate our method's restoration quality. Figures 3(a) and 4(a) show the original

Table 1: Summary results for the uniform noise.

Image	Noise	Blur	SNR		PSNR		CPU	
			[20]	Ours	[20]	Ours	[20]	Ours
Camera	(0, 0.05)	5 × 5	21.25	21.46	26.83	26.84	5.49	4.84
		7 × 7	19.35	19.37	24.93	24.92	6.82	5.99
		9 × 9	18.28	18.30	23.86	23.88	8.47	7.99
	(0, 0.1)	5 × 5	18.12	18.13	23.70	23.73	6.12	5.93
		7 × 7	17.20	17.21	22.78	22.77	8.14	7.09
		9 × 9	16.52	16.52	22.11	22.16	8.95	8.17
	(0, 0.2)	5 × 5	12.86	12.86	18.45	18.57	7.57	7.07
		7 × 7	12.90	12.91	18.48	18.48	7.82	7.00
		9 × 9	12.81	12.88	18.39	18.46	9.14	8.03
Lena	(0, 0.05)	5 × 5	24.29	24.31	29.94	29.93	29.84	27.60
		7 × 7	23.42	23.46	29.07	29.11	39.13	35.62
		9 × 9	22.60	22.67	28.26	28.26	44.05	37.16
	(0, 0.1)	5 × 5	19.37	19.37	25.03	25.07	30.50	24.35
		7 × 7	19.18	19.15	24.84	24.84	38.24	33.65
		9 × 9	18.90	18.92	24.55	24.57	45.47	40.36
	(0, 0.2)	5 × 5	13.65	13.74	19.31	19.36	44.06	39.56
		7 × 7	13.58	13.63	19.24	19.33	41.28	36.33
		9 × 9	13.25	13.25	18.99	18.97	45.49	40.68

Table 2: Summary results for the speckle noise.

Image	Noise	Blur	SNR		PSNR		CPU	
			[20]	Ours	[20]	Ours	[20]	Ours
Camera	0	5 × 5	22.06	22.12	27.64	27.64	5.09	4.36
		7 × 7	19.94	19.93	25.52	25.57	7.55	6.05
		9 × 9	18.72	18.74	23.89	23.90	8.13	6.88
	0.01	5 × 5	20.32	20.33	25.90	25.93	5.88	5.02
		7 × 7	19.22	19.27	24.83	24.84	8.30	6.91
		9 × 9	18.28	18.28	24.30	24.35	8.13	7.04
	0.05	5 × 5	15.77	15.79	21.35	21.35	9.48	8.25
		7 × 7	14.95	14.96	20.54	20.55	9.16	7.25
		9 × 9	12.63	12.62	18.21	18.24	9.87	8.23
Lena	0	5 × 5	28.33	28.34	33.99	34.01	29.94	27.61
		7 × 7	26.07	26.10	31.72	31.72	37.44	32.56
		9 × 9	24.59	24.62	30.25	30.61	45.29	38.65
	0.01	5 × 5	23.64	23.69	29.30	29.32	36.86	32.58
		7 × 7	23.86	23.94	28.92	29.00	42.32	36.32
		9 × 9	23.24	23.24	28.89	28.92	50.97	43.65
	0.05	5 × 5	18.23	18.25	23.89	23.90	61.20	55.63
		7 × 7	16.58	16.57	22.24	22.31	61.31	56.86
		9 × 9	16.48	16.52	20.14	20.14	71.48	63.53

images. Figures 3(b) and 4(b) show the observed images. Figure 3(c) shows the restored Camera image by algorithm in [18]; Figure 3(d) shows the restored Camera image by our method, where the standard deviation of noise is 0.1 and the support is equal to 9 × 9. Figure 4(c) shows the restored Lena image by algorithm in [18]; Figure 4(d) shows the restored Lena image by our method, where the standard deviation of noise is 0.2 and the support is equal to 9 × 9. Table 3 shows more image restoration results with the different support and standard deviation of noise.

As in Section 5.1, Figures 3 and 4 show that the quality of restoration images by using our method is better than those restored by the algorithm of [18] in the two cases. We also can know from Table 3 that the SNR and PSNR of the images restored by our method both are higher than the results of [18] in most cases. With the same reasons mentioned in l_1-TV model, all computational time required by our method is significantly less than that required by the algorithm in [18]. So the performance of the algorithm in [18] also has been improved through the correction.

Table 3: Summary results for Gaussian noise.

Image	Noise	Blur	SNR		PSNR		CPU	
			[18]	Ours	[18]	Ours	[18]	Ours
Camera	0.01	5 × 5	21.99	21.98	27.57	27.61	2.45	2.31
		7 × 7	19.90	19.93	25.49	25.49	2.67	2.32
		9 × 9	18.69	18.69	24.27	24.28	2.73	2.63
	0.05	5 × 5	20.64	20.69	26.22	26.24	1.83	1.62
		7 × 7	19.04	19.12	24.62	24.66	7.60	7.02
		9 × 9	18.01	18.07	23.59	23.59	2.57	2.26
	0.1	5 × 5	17.97	17.99	23.55	23.58	1.84	1.56
		7 × 7	17.06	17.11	22.64	22.63	2.20	2.12
		9 × 9	16.41	16.40	22.00	22.61	2.22	1.98
	0.2	5 × 5	13.37	17.38	18.95	18.94	1.78	1.51
		7 × 7	13.11	13.10	18.70	18.73	1.68	1.53
		9 × 9	12.91	12.93	18.49	18.61	1.86	1.68
Lena	0.01	5 × 5	28.05	28.10	33.71	33.73	11.89	10.32
		7 × 7	25.90	25.91	31.56	31.62	12.93	10.33
		9 × 9	24.47	24.49	30.13	30.15	13.32	10.92
	0.05	5 × 5	24.09	24.06	29.74	29.77	10.39	9.66
		7 × 7	23.12	23.12	28.78	28.79	11.26	9.65
		9 × 9	22.33	22.42	27.99	28.12	13.18	11.56
	0.1	5 × 5	19.50	19.55	25.16	25.17	9.14	8.68
		7 × 7	19.19	19.23	24.85	24.84	10.26	9.33
		9 × 9	18.86	18.88	24.52	24.51	11.79	9.91
	0.2	5 × 5	13.81	13.81	19.47	19.44	7.14	6.56
		7 × 7	13.83	13.86	19.49	19.83	8.47	7.82
		9 × 9	13.78	14.01	19.44	19.87	9.78	7.97

6. Concluding Remarks

Taking the l_1-TV model in image restoration as an example, a modified method is proposed to overcome drawbacks of the primal alternating direction method in this paper. We have illustrated that our method about how to correct the search direction improves the optimization performance. In addition, the convergence of the primal alternating direction method has been proven under some weaker conditions, and thus the convergence of proposed method is easily obtained by the equivalence between them. The experimental results based on two models in [18, 20] show that the proposed method could enhance the quality of restored images in most cases and the efficiency of algorithms has been significantly improved. In fact, our method can be applied to many other cases optimized alternatively.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities of Southwest University for Nationalities (no. 2015NZYQN30), the Key Fund Project of Sichuan Provincial Department of Education (no. 17ZA0414), and the National Science Foundation of China (no. 61273311). The authors are grateful to the author, Xiaoxia Guo, of [20] for providing programmes.

References

[1] A. K. Katsaggelos, *Digital Image Restoration*, Springer-Verlag, Berlin, Germany, 1991.

[2] A. S. Carasso, "Linear and nonlinear image deblurring: a documented study," *SIAM Journal on Numerical Analysis*, vol. 36, no. 6, pp. 1659–1689, 1999.

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[4] M. K. Ng, R. H. Chan, and W.-C. Tang, "A fast algorithm for deblurring models with Neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 851–866, 1999.

[5] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[6] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[7] A. N. Tikhonov and V. Y. Arsenin, "Solutions of Ill-Posed Problem," *Mathematics of Computation*, vol. 23, 491 pages, 1977.

[8] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen, "Image restoration using total variation with overlapping group sparsity," *Information Sciences*, vol. 295, pp. 232–246, 2015.

[9] L. I. Rudin and S. Osher, "Total variation based image restoration with free local constraints," in *Proceedings of the 1st IEEE International Conference on Image Processing*, vol. 1, pp. 31–35, IEEE, Austin, Tex, USA, November 1994.

[10] A. Langer, "Automated parameter selection for total variation minimization in image restoration," *Journal of Mathematical Imaging and Vision*, vol. 57, no. 2, pp. 239–268, 2017.

[11] X. Liu, "Augmented Lagrangian method for total generalized variation based Poissonian image restoration," *Computers & Mathematics with Applications. An International Journal*, vol. 71, no. 8, pp. 1694–1705, 2016.

[12] J. Liu, T.-Z. Huang, X.-G. Lv, and S. Wang, "High-order total variation-based Poissonian image deconvolution with spatially adapted regularization parameter," *Applied Mathematical Modelling*, vol. 45, pp. 516–529, 2017.

[13] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[14] T. F. Chan and K. Chen, "An optimization-based multilevel algorithm for total variation image denoising," *Multiscale Modeling & Simulation. A SIAM Interdisciplinary Journal*, vol. 5, no. 2, pp. 615–645, 2006.

[15] M. K. Ng, L. Qi, Y.-F. Yang, and Y.-M. Huang, "On semismooth Newton's methods for total variation minimization," *Journal of Mathematical Imaging and Vision*, vol. 27, no. 3, pp. 265–276, 2007.

[16] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 1964–1977, 1999.

[17] M. Hintermuller and K. Kunisch, "Total bounded variation regularization as a bilaterally constrained optimization problem," *SIAM Journal on Applied Mathematics*, vol. 64, no. 4, pp. 1311–1333, 2004.

[18] Y. Huang, M. K. Ng, and Y.-W. Wen, "A fast total variation minimization method for image restoration," *Multiscale Modeling & Simulation. A SIAM Interdisciplinary Journal*, vol. 7, no. 2, pp. 774–795, 2008.

[19] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008.

[20] X. Guo, F. Li, and M. K. Ng, "A fast l1-TV algorithm for image restoration," *SIAM Journal on Scientific Computing*, vol. 31, no. 3, pp. 2322–2341, 2009.

[21] Z. Zhi, Y. Sun, and Z.-F. Pang, "Two-stage image segmentation scheme based on inexact alternating direction method," *Numerical Mathematics: Theory, Methods and Applications*, vol. 9, no. 3, pp. 451–469, 2016.

[22] F. Wang, X.-L. Zhao, and M. K. Ng, "Multiplicative noise and blur removal by framelet decomposition and l1-based L-curve method," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4222–4232, 2016.

[23] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 569–592, 2009.

[24] W. Y. Sun and Y. X. Yuan, *Optimization Theory and Method: Nonlinear Programming*, Springer-Verlag, Berlin, Germany, 2006.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[27] Z.-J. Bai, D. Cassani, M. Donatelli, and S. Serra-Capizzano, "A fast alternating minimization algorithm for total variation deblurring without boundary artifacts," *Journal of Mathematical Analysis and Applications*, vol. 415, no. 1, pp. 373–393, 2014.

[28] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis* , vol. 23, no. 3, pp. 346–367, 2007.

[29] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation. A SIAM Interdisciplinary Journal*, vol. 4, no. 4, pp. 1168–1200, 2005.

[30] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, Belmont, Mass, USA, 2003.

[31] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, NY, USA, 1971.

[32] M. Nikolova, M. K. Ng, and C.-P. Tam, "Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3073–3088, 2010.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6204742, 18 pages
https://doi.org/10.1155/2017/6204742

Research Article

Indian Classical Dance Classification with Adaboost Multiclass Classifier on Multifeature Fusion

K. V. V. Kumar, P. V. V. Kishore, and D. Anil Kumar

Department of Electronics and Communications Engineering, KL University, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh, India

Correspondence should be addressed to P. V. V. Kishore; pvvkishore@kluniversity.in

Received 1 June 2017; Revised 27 July 2017; Accepted 17 August 2017; Published 26 September 2017

Academic Editor: Daniel Zaldivar

Extracting and recognizing complex human movements from unconstraint online video sequence is an interesting task. In this paper the complicated problem from the class is approached using unconstraint video sequences belonging to Indian classical dance forms. A new segmentation model is developed using discrete wavelet transform and local binary pattern (LBP) features for segmentation. A 2D point cloud is created from the local human shape changes in subsequent video frames. The classifier is fed with 5 types of features calculated from Zernike moments, Hu moments, shape signature, LBP features, and Haar features. We also explore multiple feature fusion models with early fusion during segmentation stage and late fusion after segmentation for improving the classification process. The extracted features input the Adaboost multiclass classifier with labels from the corresponding song (tala). We test the classifier on online dance videos and on an Indian classical dance dataset prepared in our lab. The algorithms were tested for accuracy and correctness in identifying the dance postures.

1. Introduction

Automatic human action recognition is a complicated problem for computer vision scientists, which involves mining and categorizing spatial patterns of human poses in videos. Human action is defined as a temporal variation of human body in a video sequence, which can be any action such as dancing, running, jumping, or simply walking. Automation encompasses mining the video sequences with computer algorithms for identifying similarities between actions in the unknown query dataset with that of the known dataset. Last decade has seen a jump in online video creation and the need for algorithms that can search within the video sequence for a specific human pose or object of interest. The problem is to extract, identify a human pose, and classify into labels based on trained human signature action models [1]. The objective of this work is to extract the signature of Indian classical dance poses from both online and offline videos given a specific dance pose sequence as input.

However, the constraints are video resolution, frame rate, background lighting, scene change rate, and blurring to name a few. The analysis on online content is a complicated process as the most of the users end up uploading the videos with poor quality, which shows all the constraints as a hindrance in automation of video object segmentation and classification. Dance video sequences online are having a far many constraints for smooth extraction of human dance signatures. Automatic dance motion extraction is complicated due to complex poses and actions performed at different speeds in sink to music or vocal sounds. Figure 1 shows a set of online and offline (lab captured) Indian classical dance videos for testing the proposed algorithm.

Indian classical dance forms are a set of complex body signatures produced from rotation, bending, and twisting of fingers, hands, and body along with their motion trajectory and spatial location. There are 8 different classical Indian dance forms; Bharatanatyam, Kathakali, Kathak, Kuchipudi, Odissi, Sattriya, Manipuri, and Mohiniyattam [2–4]. Extracting these complex movements from online videos and classification requires a complex set of algorithms working in sequence. We propose to use silhouette detection and background elimination, human object extraction, local texture with shape reference model, and 2D point cloud to represent the dancer pose. Five features are calculated that represent the exact

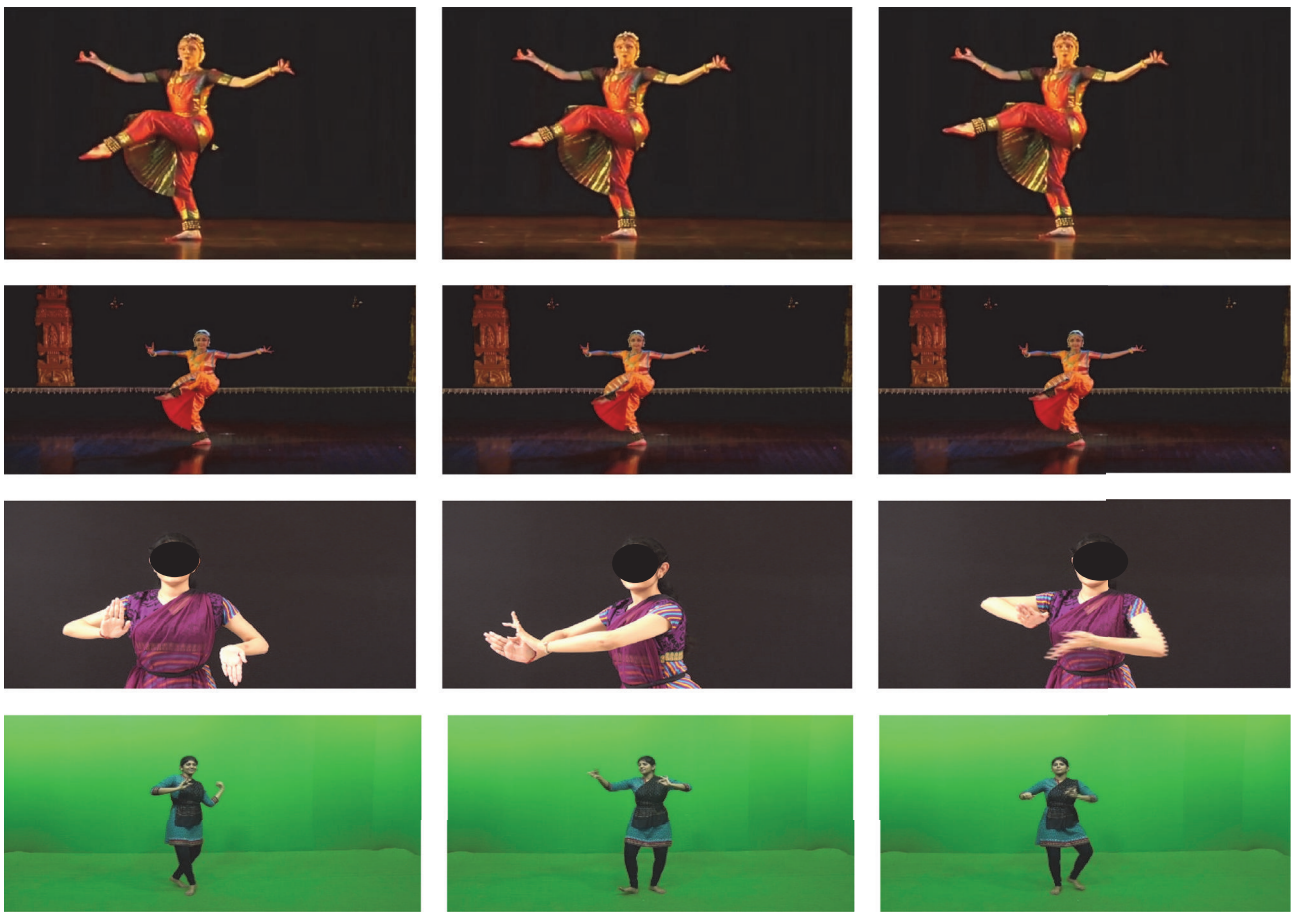


Figure 1: Online and offline dance datasets used in this work and the video constraints.

shape of the dancer in the video sequence. For recognition, a multiclass multilabel Adaboost algorithm is proposed to classify query dance video based on the dance dataset.

The rest of the paper is organized into literature survey on the proposed techniques, theoretical background on the proposed models, and experimental results. The proposed model is compared with SVM and Graph Matching (GM) classifier already proposed by us in our previous work.

2. Literature Survey

Local information of the human in the video is the popular features for action segmentation and classification in recent times. This section focuses on giving a current trend in human action recognition and how it is used in recent works for classifying dance performances. The human action recognition is subdivided into video object extraction, feature representation, and pattern classification [5, 6]. Based on these models, numerous visual illustrations have been proposed for discriminating human action based on shape templates in space-time [5], shape matching [6], interest points in 2D space time models [7], and representations using motion trajectories [8]. Impressively, dense trajectory based methods [9] have shown good results for action recognition by tracking sampled points through optical flow fields. Optical flow fields are based on preconditioned on brightness and object motion in a video [10]. The algorithms assume uniform brightness variations and object motions in consecutive frames which produce excellent results under minimum constrained video recording. Minimum constrained video recordings are having uniform brightness, less blurring, fixed camera angle, and high contrast between object and background. Finding such a video happens in a movie or in a lab setup. Hence, these approaches need to be robust in estimating human actions, which is still an open-ended problem on real time videos. Data driven methods with multiple feature fusion [11] with artificial intelligence models [12] are currently being explored with the increase in computing power.

In this work, human action recognition on Indian Classical Dance [13] videos is performed on recordings from both offline (controlled recording) and online (Live Performances, YouTube) data. Indian classical dance forms are practised from 5000 years worldwide. However, it is difficult for a dance lover to fully hold the content of the performance as it is made up of hand poses, body poses, leg movements, hands with respect to face and torso, and finally facial expressions. All these movements should synchronize in precision with both vocal song and the corresponding music for various instruments. Apart from these complications, the dancer wears complicated dresses with nice makeup and at times during performance the backgrounds are changing

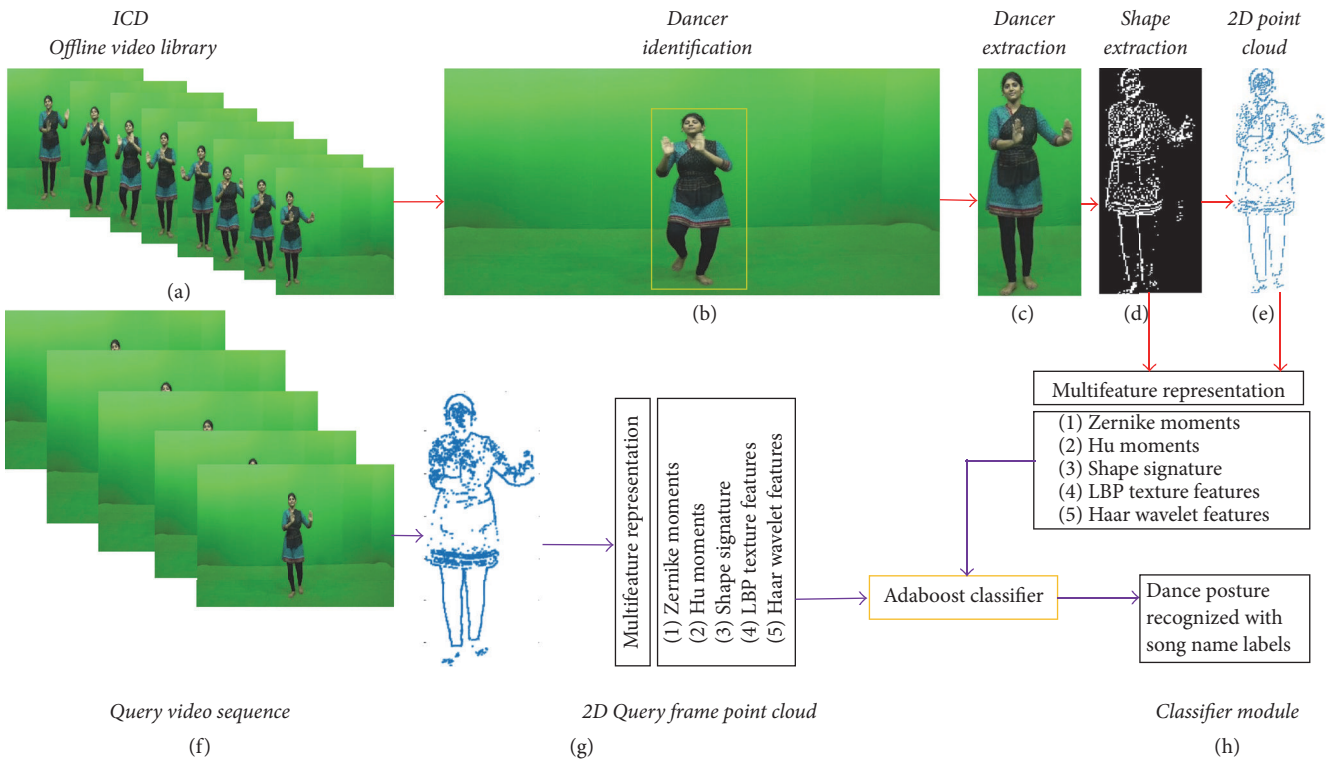


FIGURE 2: Flow diagram of the proposed process for Indian classical dance recognition. (a) Training datasets, (b) detected dance object, (c) extracted, (d) DWT and LBP features, (e) feature points, (f) query dance video, (g) query feature points, and (h) multifeature extraction and classification.

depending on the story which truly makes this an open-ended problem. Mohanty et al. [14] highlight the difficulties in using state-of-the-art pose estimation algorithms such as skeleton estimation [15] and pose estimation [16] which fail to track the dancers moves in both offline and online videos. The author in [14] proposes using deep learning based convolutional neural networks (CNNs) and shows they perform well in estimating the correct pose of dancers on both 3D Kinect dance poses and online videos. CNNs require large training data for a specific class of inputs which makes them computationally slow for a video datasets that change for every 2 frames. In real time, there is no Kinect [17] like effects and hence 2D video analysis must be refined in accuracy for identifying poses of Indian dance forms. Samanta et al. [18] used histogram of oriented optical flow (HOOF) features with sparse representations. Support vector machine classifier (SVM) classifies the Indian classical dance poses from KTH dataset with an accuracy of 86.67%. In our previous work [19], we approached the same problem with SVM classifier on dance videos and found that only multiclass SVMs should be considered. Moreover, optical flow on online videos suffers at lot due to inconsistencies during capture and sharing process.

In [20], Samanta and Chanda proposed a video descriptor manifold on ICD YouTube videos and KTH dataset with nonlinear SVM classifier and recorded recognition accuracies in the range 70 to 95%. The other works proposed also used SVM classifier with simple image processing models on images of dancers for pose estimation which can be found in [21–23]. In [24], authors used Kinect sensor to capture leg poses in Indian classical dance forms and classification is initiated with SVM classifier for a set of 40 poses. Kinect sensor produces skeleton data of the human body pose and fails to reproduce data related to fingers which are important in classifying a dance pose.

The objective is to select features that represent a sign and are easily distinguishable in closely related sign words and are computationally efficient. The attributes for a selfie sign language recognizer chosen are shape signature [25] for hand and head shapes, Hu moments [26] for hand orientations, hand–head distance, and hand position vectors for tracking. The chosen attributes perfectly characterize a sign in Indian sign language.

Classifying at faster rate on a huge dataset is a complicated problem. Adaboost [27] classifier is fast and efficient algorithm for large datasets [28]. Inspired by [29, 30], the feature matrix is labelled and inputted to Adaboost classifier for training and testing. The performance indicators are recall-precision curves and execution time on mobile and are recorded to check the robustness of the algorithm and feasibility of implementing more efficiently.

In this paper, we propose an multiclass multilabel Adaboost (MCMLA) based classification problem on multidimensional feature vector. We show that this can be used to match large unconstrained dance features which are automatically extracted from video datasets. The feature representation of video objects depends on the efficiency of video segmentation algorithms. As illustrated in Figure 2, the proposed Adaboost can effectively recover the query video

frames from the dance dataset, by shape–texture observation model defined by discrete wavelet transform (DWT) and local binary patterns (LBP).

In summary, our MCMLA algorithm on online and offline Indian classical dance videos combines the representational flexibility and trivial computations. We perform experiments on two different datasets of Indian classical dance Bharatanatyam and Kuchipudi created from online downloads and offline controlled lab capture. The proposed method is compared with other GM models which are outperformed by a considerable margin in speed.

3. Proposed Methodology

The proposed algorithm framework is shown in Figure 2. An Indian Classical Dance (ICD) video library is created combining online and offline videos. Dancer identification, dancer extraction, local shape feature extraction, and classifier are the modules of the system. Further feature fusion concept from [31] is also explored in this wok using 5 feature types, Zernike moments, Hu moments, shape signature, LBP features, and Haar features. Adaboost algorithm explores the relativity between the query dance sequence and known dataset.

3.1. Dancer Identification. Most of the dance videos are poorly illuminated or fully brightened with too much background information during capture. Commercial video cameras have a frame rate of 30 fps and dance movements are sometimes faster and at time slower which makes the object blurry. The objective is to extract moving dancer and segment it for further processing. This helps to prevent the algorithm from constantly upgrading the background information and modelling the object characteristics in real time. The dancer identification module is based on one of the silhouette extraction methods proposed in [32]. A significant indication in determining dancers motion for extraction lies in the temporal changes in the dancer's silhouette during performance. To avoid background modelling and foreground extraction models, we propose using the following procedure.

The dance video sequence $\mathbf{V}(\mathbf{x}, \mathbf{y}, \mathbf{t}) \subset \mathbb{R}^+$, with $(\mathbf{x}, \mathbf{y}) \subset \mathbb{Z}^+$, gives pixel location and $\mathbf{t} \subset \mathbb{Z}^+$ is the frame number. Each frame in $\mathbf{V}$ is having RGB planes and is of size $\mathbf{N} \times \mathbf{M} \times 3$. This part of the module is only for motion segmentation and object extraction; color can be discarded. RGB is converted to gray scale and contrast enhanced to improve the frame quality. The frame $\mathbf{V^t}$ at $\mathbf{t}$ is mean filtered with mask defined by $\mathbf{m}(\mathbf{x}, \mathbf{y})$ with

$$\mathbf{V_m^t}(\mathbf{x}, \mathbf{y}) = \mathbf{V^t}(\mathbf{x}, \mathbf{y}) \otimes \mathbf{m}(\mathbf{x}, \mathbf{y}). \tag{1}$$

The size of $\mathbf{m}$ is updated based on the frame size $\mathbf{N} \times \mathbf{M}$ for faster computations, where the object area is small compared to the background area. The $\otimes$ operator is linear convolution and the averaged frame is of the same size as the input frame. The next step applies a Gaussian filter of μ mean and σ variance on the input frame $\mathbf{V^t}$:

$$\mathbf{V_g^t}(\mathbf{x}, \mathbf{y}) = \mathbf{V^t}(\mathbf{x}, \mathbf{y}) \otimes \mathbf{g}(\mu, \sigma). \tag{2}$$

The size of the Gaussian mask is determined by the input video frame. Euclidian distance metric $\mathbf{S^t}(\mathbf{x}, \mathbf{y})$ between $\mathbf{V_m^t}$ and $\mathbf{V_g^t}$ gives the saliency map of the moving pixels in the frame

$$\mathbf{S^t}(\mathbf{x}, \mathbf{y}) = \left\| \mathbf{V_g^t}(\mathbf{x}, \mathbf{y}) - \mathbf{V_m^t}(\mathbf{x}, \mathbf{y}) \right\|_2. \tag{3}$$

The second order normed distance map is shown in Figure 3 which identifies the dancer's silhouette. However, to extract the dancer, a mask of this silhouette is used to determine the connected components in the object. Figure 3(d) shows the silhouette mask and connected component output is in Figure 3(e).

The centroid of the mask is mapped on the frame to crop out the moving dancer in the frame. The method is effective in all lighting conditions putting constraints on the input video frame size in selecting the masks used for mean and Gaussian filters. The boxed and extracted dancer from the video sequence is shown in Figures 3(g) and 3(h), respectively. The extracted dancer is free from background variations in the video sequence. If a portion of background still appears at this stage it can be nullified during the matching phase. Applying feature extraction on the extracted dancer allows for lesser computations as the background is almost eliminated and leads to good matching accuracy.

3.2. Feature Extraction. Which features can help recognize and classify dance correctly is the question. From a dancer's perspective, to identify a dance type, body posture, hand shapes, and their movements in space are the vital features. Feature extraction phase explores the methodology in extracting these features. There are many shape descriptors available in literature for characterizing shape features [33]. Lighting, frame inconsistency, contrast, blurring, and frame size are some of the critical factors that affect feature extraction algorithms. In addition, the dancer velocity during performance has instincts for a faster shape extractor.

3.2.1. Haar Wavelet Features: Global Shape Descriptor. For removing video frame noise during capture and to extract local shape information, we propose a hybrid algorithm with Discrete wavelet transform (DWT) [34] and Local Binary Patterns (LBP) [35]. The objective at this stage is to represent moving dancers shape with a set of wavelet coefficients. Here we propose using Haar wavelet at level 1. At level 1, Haar wavelet decomposes the video frame $\mathbf{V^t}$ into 4 subbands. Figure 4 shows the 4 subbands at 2 levels. At 1st level we have 4 subbands and at 2nd level have 8 subbands. In the 1st level, the three subbands represent the shape information at three different orientations: Vertical $\mathbf{v}$, Horizontal $\mathbf{h}$, and Diagonal $\mathbf{d}$. Combining the three subbands and averaging the wavelet coefficients normalizes the large values.

$$\mathbf{W_S^t} = \frac{\mathbf{h} + \mathbf{v} + \mathbf{d}}{3}. \tag{4}$$

The averaged shape harr wavelet coefficients $\mathbf{W_S^t}$, along with $\{\mathbf{h}, \mathbf{v}, \mathbf{d}\}$ subband coefficients are reconstructed to spatial domain. Figure 4 shows the reconstructed spatial domain

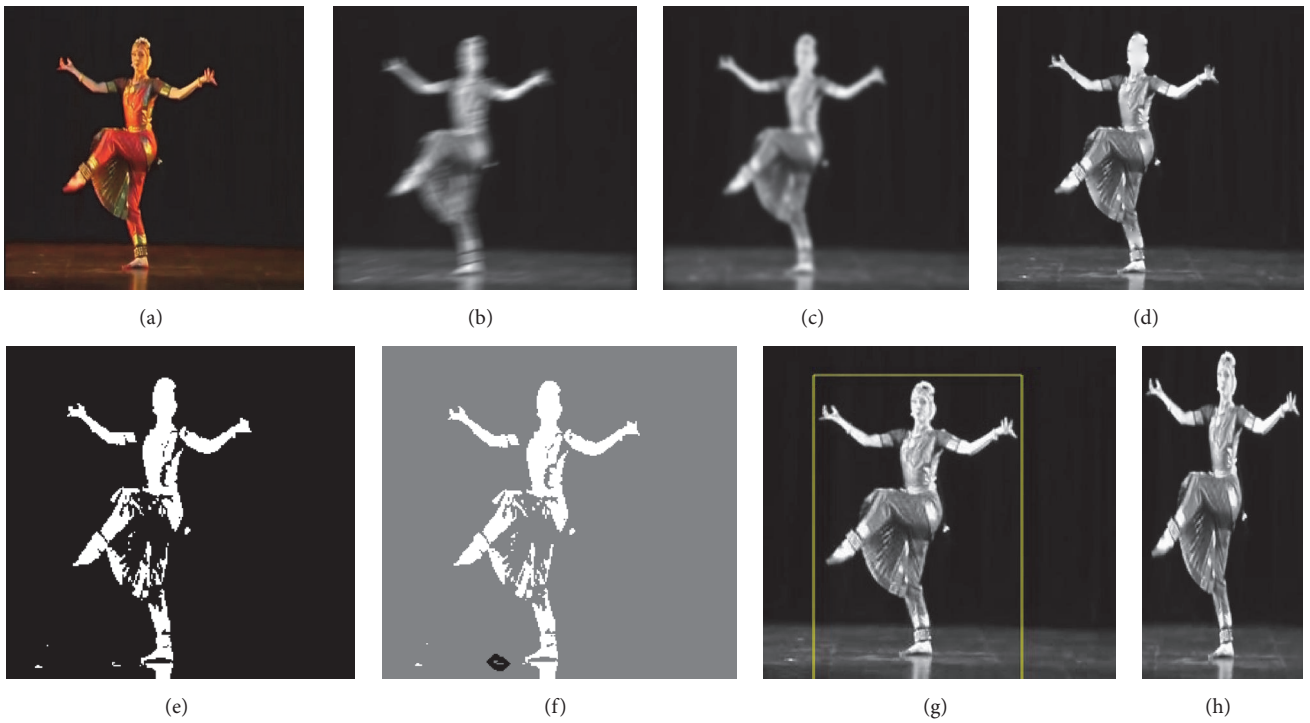


Figure 3: Dancer extraction. (a) Original frame, (b) mean filtered, (c) Gaussian filtered, (d) distance saliency map, (e) silhouette mask, (f) connected components labelling, (g) identified dancer, and (h) dancer extracted.

Figure 4: Harr wavelet subbands representing shape in three different orientations.

frame producing the exact hand shapes. These shape features can be used as nodes and a graph can be constructed for recognition. However, background noise is still a major concern at this stage. Local pixel information becomes vital in selection of nodes that exactly represent the graph.

3.2.2. Thresholding. Apply threshold on the reconstructed ICD video frame $\mathbf{V}_{\mathbf{r}}^{\mathbf{t}}$ as

$$\mathbf{T}^{\mathbf{t}} = \sqrt{\frac{1}{\mathbf{NM}} \sum_{\mathbf{j}=1}^{\mathbf{M}} \sum_{\mathbf{i}=1}^{\mathbf{N}} \left(\mathbf{V}^{\mathbf{t}}(\mathbf{j}, \mathbf{i})\right)^2}. \tag{5}$$

The binarized video frame $\mathbf{B}^{\mathbf{t}}$ is

$$\mathbf{B}^{\mathbf{t}} = \mathbf{V}_{\mathbf{r}}^{\mathbf{t}} > \mathbf{T}^{\mathbf{t}}. \tag{6}$$

To extract the nodes for the graph, local pixel patterns provide exact shape representation.

3.2.3. Local Binary Patterns and Local Shape Models. LBP compares each pixel in a predefined neighbourhood to summarize the local structure of the image. For an image pixel $\mathbf{B}^{\mathbf{t}}(\mathbf{x}, \mathbf{y}) \in \mathfrak{R}^{+}$, $(\mathbf{x}, \mathbf{y})$ gives the pixel position in the intensity image. The neighbourhood of a pixel can vary from

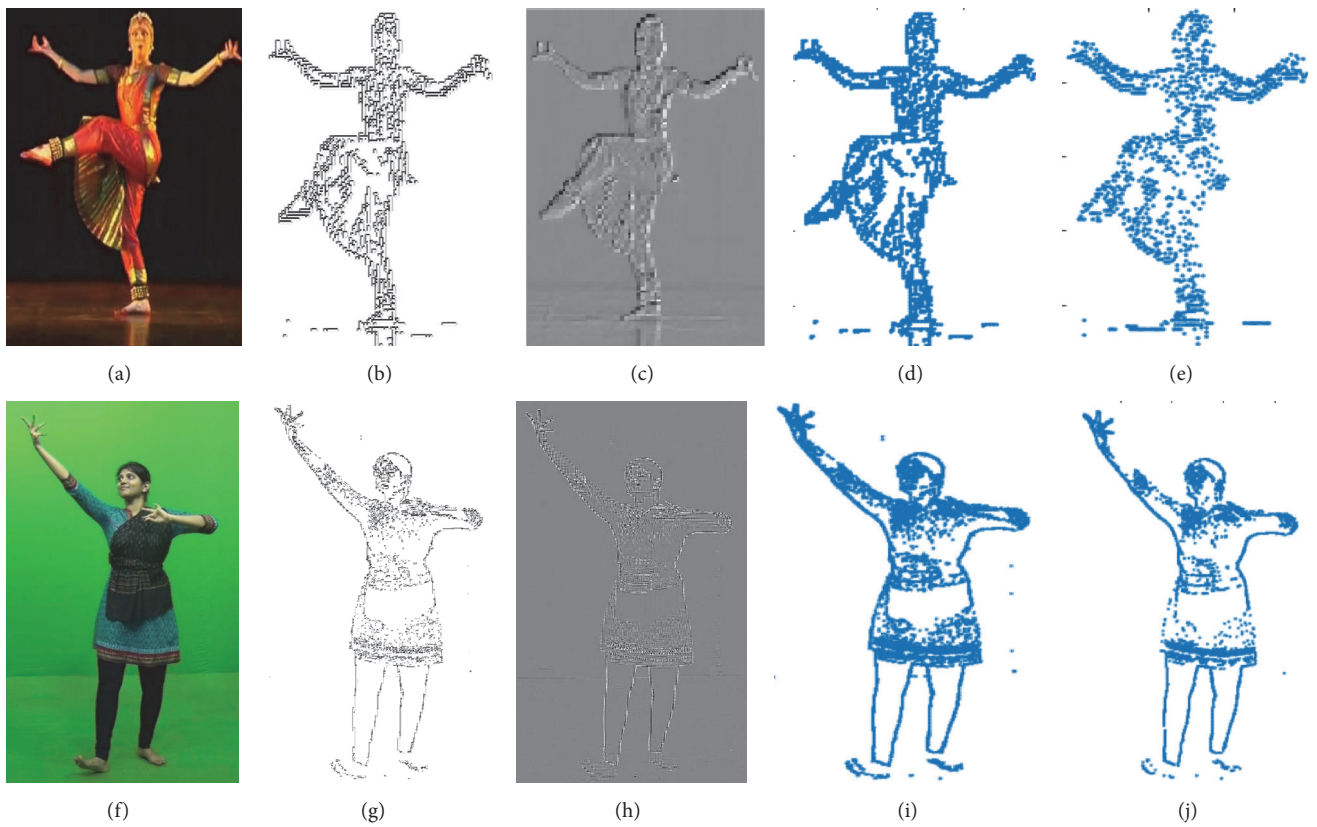


Figure 5: (a) Original dancer in online video frame, (b) LBP features from reconstructed and thresholded wavelet coefficients, (c) Haar wavelet features, (d) sparse coded wavelet reconstructed LBP features, and (e) sparse coded wavelet features. (f)–(j) Same as (a)–(e) for offline lab captured video frame.

3 pixels with radius $\mathbf{r} = 1$ or a neighbourhood of 12 pixels with $\mathbf{r} = 2.5$. The value of pixels using LBP code for a centre pixel $(\mathbf{x_c}, \mathbf{y_c})$ is given by

$$\mathbf{L_S^t} = \mathbf{LBP}\left(\mathbf{x_c},\mathbf{y_c}\right) = \sum_{\mathbf{j}=1}^{\mathbf{P}} \mathbf{B^t}\left(\mathbf{g_p} - \mathbf{g_c}\right) 2^{\mathbf{P}}$$
$$\mathbf{B^t}\left(\mathbf{x}\right) = \begin{cases} 1 & \forall \mathbf{x} \geq 0 \\ 0 & \textit{Otherwse}, \end{cases} \tag{7}$$

where $\mathbf{g_c}$ is binary value of centre pixel at $(\mathbf{x_c},\mathbf{y_c})$ and $\mathbf{g_p}$ is binary value around the neighbourhood of $\mathbf{g_c}$. The value of $\mathbf{P}$ gives the number pixels in the neighbourhood of $\mathbf{g_c}$. The local shape descriptor $\mathbf{L_S^t}$ of the human dancers pose projects maximum number of points on to graph.

3.3. Multifeatures: Zernike, Hu Moments, Shape Signature, LBP, and Haar. Figure 5 shows the extracted dancer represented with LBP features and Haar wavelet features. Local shape features in Figure 5(a) are used to construct a graph. Given a motion frame in a ICD video sequence $\mathbf{V^t}$ successfully extracted local shape features $\mathbf{L_S^t}$ and transformed into a binary shape matrix $\mathbf{B_S^t}$ of ones and zeros using (5). A sparse representation of $\mathbf{B_S^t}$ eliminates all zeros and retains only ones and their locations in $\mathbf{M_S^t(x, y, w)}$, where $\mathbf{x}, \mathbf{y}$ are shape point locations and $\mathbf{w}$ is shape feature weight vector. Figures 5(d) and 5(e) show a sparse representation for both wavelet reconstructed LBP (WR_LBP) and only Harr wavelet features (HWF), respectively. The points on the motion object are formed by extracting the location of the pixel and its feature value determines the shape of the dance pose. From these feature point locations and values a graph is constructed in this work.

Haar and LBP features are enough to label the dancers in the frames and put them under the classifier. This is early fusion of features at the segmentation stage. The fusion operator is principle component analysis. PCA of wavelet features and LBP features is concatenated using the following expression:

$$\mathbf{E_{fu}} = \mathbf{PCA}\left(\mathbf{W_S^t}\right) \cup \mathbf{PCA}\left(\mathbf{L_S^t}\right). \tag{8}$$

However, 3 more features are proposed in this work that can effectively represent shape features of the dancer. Dancers in Indian classical dance videos have a large motion vector field and Haar and LBP depend on variations in lighting, camera movement, and background. To counterbalance camera movements, we propose using Zernike Moments (ZM)

to represent dancer in each frame on the 2D point cloud extracted from WR_LBP vectors.

Dancer body orientations provide rotation invariant feature of a dance movement in a dancing space. However, these movements are incorrectly classified if there are unavoidable sudden camera vibrations. Moments $\mathbf{M_{pq}}$ project 2D points in dance segments $\mathbf{B_s^t(x,y)}$ on to basis $\mathbf{x^p y^q}$ which results in a piecewise continuous linear function in the spatial plane. Moments and moment functions are used as pattern shape features in number of applications [36–38]. Geometric moments are usually defined as

$$\mathbf{M_{pq}} = \iint_{\mathbb{R}} \mathbf{x^p y^q B_s^t}(\mathbf{x},\mathbf{y})\,\mathbf{dx}\,\mathbf{dy}, \tag{9}$$

where $\mathbf{M_{pq}}$ is the $(\mathbf{p}+\mathbf{q})$th order moment of $\mathbf{B_s^t} > 0$. However, geometric moments do not exhibit any invariance properties such as translation, rotation, and scaling. Teague [39] proposed ZM to recover image from moments representing the image in terms of orthogonal polynomials. ZM are used in pattern recognition applications [40, 41].

ZM project the 2D point cloud $\mathbf{B_s^t(x,y)}$ to a set of orthogonal polynomials, called Zernike polynomials defined as a complete set

$$\mathbf{Z_{nm}}(\rho,\theta) = \mathbf{R_{nm}}(\rho)\,\mathbf{e^{jm\theta}}, \tag{10}$$

where $\mathbf{R_{nm}}(\rho)$ are real valued radial polynomials defined in [39] and ρ is moment magnitude and θ is angle.

The orthogonality of ZM is modelled as

$$\begin{aligned}&\iint_{\substack{0\le\rho\le1\\0\le\theta\le2\pi}} \mathbf{Z^*_{nm}}(\rho,\theta)\,\mathbf{Z_{n'm'}}(\rho,\theta)\,\rho\mathbf{d}\rho\,\mathbf{d}\theta\\ &\quad= \frac{\pi}{\mathbf{n}+1}\delta_{\mathbf{nn'}}\delta_{\mathbf{mm'}},\end{aligned} \tag{11}$$

where "$*$" indicates complex conjugate and $\delta_{\mathbf{nn'}}$ should satisfy

$$\delta_{\mathbf{nn'}} = \begin{cases} 1, & \mathbf{n} = \mathbf{n'} \\ 0, & otherwise. \end{cases} \tag{12}$$

ZM of order $\mathbf{n}$ with repetition $\mathbf{m}$ for continuous feature set $\mathbf{B_s^t(x,y)}$ per frame $\mathbf{t}$ over a unit disk is defined as

$$\mathbf{A_{nm}} = \frac{\mathbf{n}+1}{\pi} \iint_{Unit\ Disk} \mathbf{Z^*_{nm}}(\mathbf{x},\mathbf{y})\,\mathbf{B_s^t}(\mathbf{x},\mathbf{y})\,\mathbf{dx}\,\mathbf{dy}. \tag{13}$$

If θ_0 is the rotational angle, with original ZM and rotated ZM as $\mathbf{A_{nm}}$ and $\mathbf{A^R_{nm}}$, respectively, we have

$$\begin{aligned}\left|\mathbf{A^R_{nm}}\right| &= \left|\mathbf{A_{nm}}\mathbf{e}^{-\mathbf{jm}\theta_0}\right| = \mathbf{A_{nm}};\\ \Theta^{\mathbf{R}}_{\mathbf{nm}} &= \Theta_{\mathbf{nm}} - \mathbf{m}\theta_0,\end{aligned} \tag{14}$$

where $|\mathbf{A_{nm}}|$ and $\Theta_{\mathbf{nm}}$ represent magnitude and phase, respectively. In (14) the magnitude remains constant while the image rotates, whereas the phase changes with image rotation. Hence, most of the applications use ZM magnitude as a feature vector for pattern classification. In this work, we propose ZM magnitude on 2D shape point cloud as invariant feature representing the small variations in camera movement that occur in online video capture of the dance performance. To represent changes in the dancer dimensions which happen nonlinearly, we propose using a nonlinear function defined over geometric moments.

Hu moments in [42] are nonorthogonal centralized moments that are scale, translation, and rotation invariant. Human dancers come in all shapes and sizes and the features describing them may change with dancer. Modelling invariance in dancer shape features from the 2D point segments of Figure 5(e) or Figure 5(j) is done with Hu moments. Hu moments are derived for 2D normalized central moments M^H_{pq} using algebraic invariants:

$$\begin{aligned}\hbar_1 &= M_{2,0} + M_{0,2},\\ \hbar_2 &= M_{2,0} - M_{0,2} - 4M^2_{1,1},\\ \hbar_3 &= \left(M_{3,0} - 3M_{1,2}\right)^2 + \left(3M_{2,1} - M_{3,0}\right)^2,\\ \hbar_4 &= \left(M_{3,0} + M_{1,2}\right)^2 + \left(M_{2,1} + M_{0,3}\right),\\ \hbar_5 &= \left(M_{3,0} - 3M_{1,2}\right)\left(M_{3,0} + M_{1,2}\right)\\ &\quad\cdot\left[\left(M_{1,2} + M_{3,0}\right)^2 - \left(M_{2,1} + M_{0,3}\right)^2\right]\\ &\quad+ \left(3M_{2,1} - M_{3,0}\right)\left(M_{2,1} + M_{0,3}\right)\\ &\quad\cdot\left[3\left(\left(M_{3,0} + M_{1,2}\right)^2 - \left(M_{3,0} + M^2{}_{1,2}\right)\right)\right],\\ \hbar_6 &= \left(M_{2,0} - M_{0,2}\right)\left[\left(M_{3,0} + M_{1,2}\right)^2 - \left(M_{2,1} + M^2_{0,3}\right)\right]\\ &\quad+ 4M_{11} + \left(M_{3,0} + M_{1,2}\right)\left(M_{0,3} + M_{2,1}\right),\\ \hbar_7 &= \left(3M_{2,1} - M_{,3}\right)\left(M_{3,0} + M_{1,2}\right)\\ &\quad\cdot\left[\left(M_{1,2} + M_{3,0}\right)^2 - \left(M_{2,1} + M_{0,3}\right)^2\right]\\ &\quad- \left(M_{3,0} - 3M_{1,2}\right)\left(M_{2,1} + M_{0,3}\right)\\ &\quad\cdot\left[3\left(\left(M_{3,0} + M^2{}_{1,2}\right) - \left(M_{2,1} + M_{0,3}\right)^2\right)\right].\end{aligned} \tag{15}$$

These 7 moments are calculated for every extracted dance shape in each frame.

Every pixel is represented with a shape feature [43] or a descriptor [44] in each frame to model dancer shape for classification. However, to reduce the computations on the feature vector during recognition, we propose using shape signatures in [25] to represent dancer shapes. In [25], shapes are represented on a multiscale model based on integral kernels. A shape descriptor or signature is an integral

invariant at various scales forming a shape signature function on dancer shapes $B_s^t(x, y)$

$$S_s(\Gamma) = \frac{\int_x \int_y B_s^t(x,y) \cdot \left(1 - G_\rho * B_s^t(x,y)\right) dy\, dx}{\int_x \int_y B_s^t(x,y)\, dx\, dy}, \quad (16)$$

where G_ρ is a Gaussian kernel in 2D with $\gamma > 0$ and zero mean. Also,

$$\gamma = \Gamma\left(\int_x \int_y F_s(x,y)\right)^{1/2}. \quad (17)$$

Integrating shape feature values over shape 2D spatial domain results in a shape signature value normalized by area of the shape to achieve scale invariance. The range of shape signature is [0, 1] based on value of scale γ. For low values of scale, it is 0 and for large scales it approaches 1.

3.4. Feature Matrix Construction. From the 5 features, a complete feature matrix is constructed per dance frame. Early fusion is performed with Haar and LBP features with PCA at the end of dancer segmentation and are labelled with dance vocal words. This feature matrix is named as EFDF (early fused dance features). It is a 2D feature matrix of size $256 \times 256 \times 2$. Max pooling is performed to reduce the feature set to $1 \times 256 \times 2$ per frame. Further, late fusion model is introduced with more robust set of features in the Zernike moments, Hu moments, and shape signatures (SS) on 2D shape point cloud of the dancer in the video frame. These late features are mixed with Haar and LBP features to create a multifeature matrix per frame. We calculated 5 ZM, 7 Hu Moments, 1 SS, max pooled, and thresholded LBP and Haar features limited to 30 features per frame. The final feature matrix in late fusion strategy is $5 \times 7 \times 1 \times 30 \times 30$ per frame. These features are carefully labelled with vocal words representing the dance form in the video frame. For a 25-frame dance word *"swami ra ra,"* we have 73×25 feature matrix. This feature matrix or a set of matrices are inputted to MCMLAB classifier.

3.5. Dance Classifier: Adaboost Multiclass Multilabel. Boosting based classifications [29, 30, 45] find very precise hypothesis from a set of weak hypotheses. Here hypothesis is a classification rule. Set of weak hypotheses are simple rules that generate a predictable classification. Let $T = [(f_1, L_1), (f_2, L_2), (f_3, L_3), \ldots, (f_v, L_v)]$ be a set of training examples at an instance f_i on ith frame in feature space f with labels L_i on label space L. The algorithm accepts the training samples T along with some class distribution $D = \{1, \ldots, m\} \in \mathbb{R}$ represented as weak learners. On the input, the weak learner computes a weak hypothesis H. Generally, $H : f \to \mathbb{R}$. The interpretation for classification is based on $\text{sign}\{H(f)\} = \{+1, -1\} \to \{f_i\}$ for a binary classifier. $|H(f_i)|$ gives prediction confidence.

The key to boosting is to use the weak learner to produce a very precise prediction rule by repeatedly addressing the weak learner on different distribution of training examples. In this work, a multiclass version of Adaboost is used having a set of strings as class labels. The problem is modelled as given T and size of final strong classifier C. The Adaboost initializes the distribution function as

$$D_1(i,l) = \frac{1}{vm} \quad \forall i = 1, \ldots, v,\ l = 1, \ldots, m, \quad (18)$$

where $l = |L|$. For $c = 1, \ldots, C$, we select a weak classifier $H_c : T \times L \to [-1, 1]$ with distribution D_c, to maximize the absolute value of

$$a_c = \sum_{i,c} D_c(i,l)\, L_i(l)\, H_c(f_i, l). \quad (19)$$

We choose the biasing value α_c as

$$\alpha_c = \frac{1}{2} \ln\left(\frac{1 + a_c}{1 - a_c}\right) \quad (20)$$

and update the distribution function as

$$D_{c+1}(i,l) = \frac{D_c(i,l)\, e^{-\alpha_c L_i[l] H_c(f_i, l)}}{N_c}, \quad (21)$$

where N_c is normalization factor to keep the distribution as probability density function. The final output strong classifier is

$$H(f,l) = \text{sign}\left(\text{sum}_c\left(\alpha_c H_c(f,l)\right)\right). \quad (22)$$

For the multiclass problem $c_1, \ldots, c_k$, we use the real valued 2D Look-Up Table model in [41], which is defined as

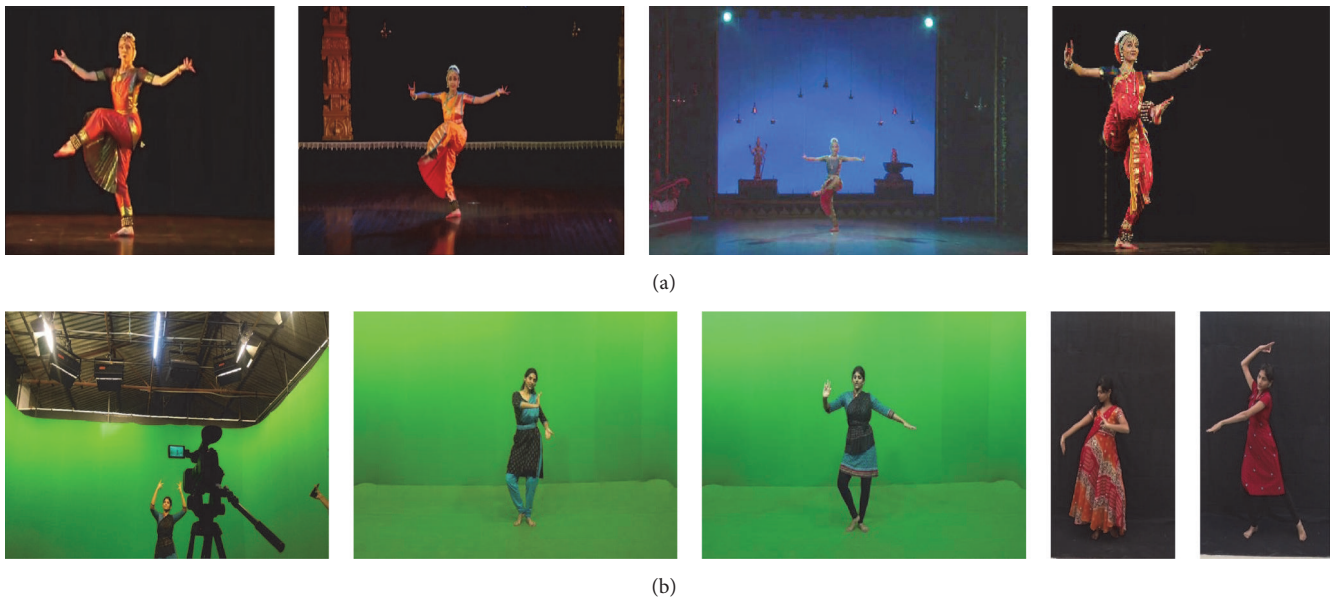
$$H_{\text{LUT}}(f, L) = \sum_{i=1}^{n} \sum_{j=1}^{k} \left(2P_l^{(j)} - 1\right) B_n^{j,l}(f,l), \quad (23)$$

where $P_l^{(j)} = P(f \in C_l \mid \text{frame} \in \text{sign})$ and $B_n^{i,l}(f,l) = \{1, f \in l; 0, \text{otherwise}\}$. From this weak hypothesis, through training a strong hypothesis is generated to recognize sign labels $Z_i = H(f_i)$.

4. Experimentation and Results

A set of 4 experiments are initiated to test the robustness of the proposed multifeature fusion with Adaboost classifier. Our Indian classical dance datasets consist of performances on "Bharatanatyam" and "Kuchipudi" from online YouTube videos and offline dance videos in controlled environment at KL University, cams department studio. We have created 4 dance videos from 5 dancers for 2 songs in two different dance styles. Similar online YouTube downloaded dance performances are also collected. Matching frames in each dataset are shown in Figure 6.

In exp-1 and exp-3, we use offline and online dataset of same dancer video for training and testing with early fusion and late fusion of multiple features 28 words in the dance sequence. Each mudra pose is coordinated with vocal manually by labelling each set. Variations in number of frames per label are nullified and normalized to 15 Key frames per dance pose across all video data. Exp-2 and Exp-4 are conducted based on different training and test



(a)

(b)

Figure 6: (a) Online Indian classical dance datasets from YouTube. (b) Offline dance video dataset created at KL University in a controlled lab environment.

video set from the online and offline dataset individually. These experiments test features in their early and late fusion models with Adaboost classifier. In the next phase, we test the classifier based on accuracy and efficiency in classifying dance gestures.

We use three performance evaluators for validating the results. They are precision–recall curves, percentage recognition rate, and the computation time per sign. For a strong hypothesis $\mathbf{H}$ resulted from Adaboost training and testing for an input feature $\mathbf{f_i}$ on a trained distribution $\mathbf{D}$ with $\mathbf{Z_i} = \mathbf{H}(\mathbf{f_i})$ predicted labels. The following metrics in [46] are

$$\text{Precision}\,(\mathbf{H},\mathbf{D}) = \frac{1}{|\mathbf{D}|}\sum_{\mathbf{i}=1}^{|\mathbf{D}|}\frac{|\mathbf{L_i}\cap\mathbf{Z_i}|}{|\mathbf{Z_i}|}, \tag{24}$$

$$\text{Recall}\,(\mathbf{H},\mathbf{D}) = \frac{1}{|\mathbf{D}|}\sum_{\mathbf{i}=1}^{|\mathbf{D}|}\frac{|\mathbf{L_i}\cap\mathbf{Z_i}|}{|\mathbf{L_i}|}, \tag{25}$$

$$\text{Recognition} = \frac{1}{|\mathbf{D}|}\sum_{\mathbf{i}=1}^{|\mathbf{D}|}\frac{|\mathbf{L_i}\cap\mathbf{Z_i}|}{|\mathbf{L_i}\cup\mathbf{Z_i}|}. \tag{26}$$

Exp-I uses input videos from dance data set captured in the controlled environment. The dancer identification, feature extraction, and graph representation for the dancer are shown in Figure 7. Saliency maps from average and Gaussian distance metric create Silhouette, which identifies the dancer in the video frame. The dimensions of the bounding box extract the dancer. Then the dancers features are extracted with segmentation as early features. Haar wavelet at level 1 is averaged in high frequency components to remove background and IDWT is performed to recover the global shapes on the dancer. Applying LBP on the resulting IDWT dance frame captures local shape information. At this stage, we perform Adaboost multiclass multilabel classification with the PCA based Haar and LBP feature fusion.

Early fused Haar-LBP features of offline dance video of a dancer are used to train the Adaboost classifier. For the same dance video shot with slight variations is provided as query dance video for same set of labels. The resulting confusion matrix from early fusion of features on same training and testing set is shown in Figure 8.

The normalized values in the confusion matrix are calculated using (26). We did plot the confusion matrix for the song sequence "Siva Shamboo" with first 28 word labels. The average recognition on total 116 labels from two dances in "kuchipudi" from a dancer is 0.96. From other dancer's videos, the Adaboost classifier averaged around 0.955. Exp-1 is repeated with all the same parameters with late fusion with Zernike moments, Hu Moments, and liner shape signature along with Haar and LBP features. Late feature matrix is a 73×25 matrix per label. For our 116-label dance sequence we have a $73 \times 25 \times 116$ dimension feature vector. Figure 9 shows the results of the classifier in the form of confusion matrix (showing 28 labels) from (26).

The results show an average of 0.99 for all dance videos in the dataset. Clearly, multiple features and late fusion have increased the ability of the classifier to recognize dance poses correctly. False matching is much less in this experiment as the datasets used for training and testing are similar, that is, same dancer and same performance.

In Exp-2, the performance is the same, meaning same labels for training but the dancer performing the dance will be different. However, testing with a different query video having different dancer results in lower recognition rates compared to the previous experiment. Here also, the simulations are performed for early feature fusion and late fusion. Confusion matrices from the two simulations are shown in Figures 10 and 11, respectively. The average recognition dropped due to noncoherence between dance moves with respect to body

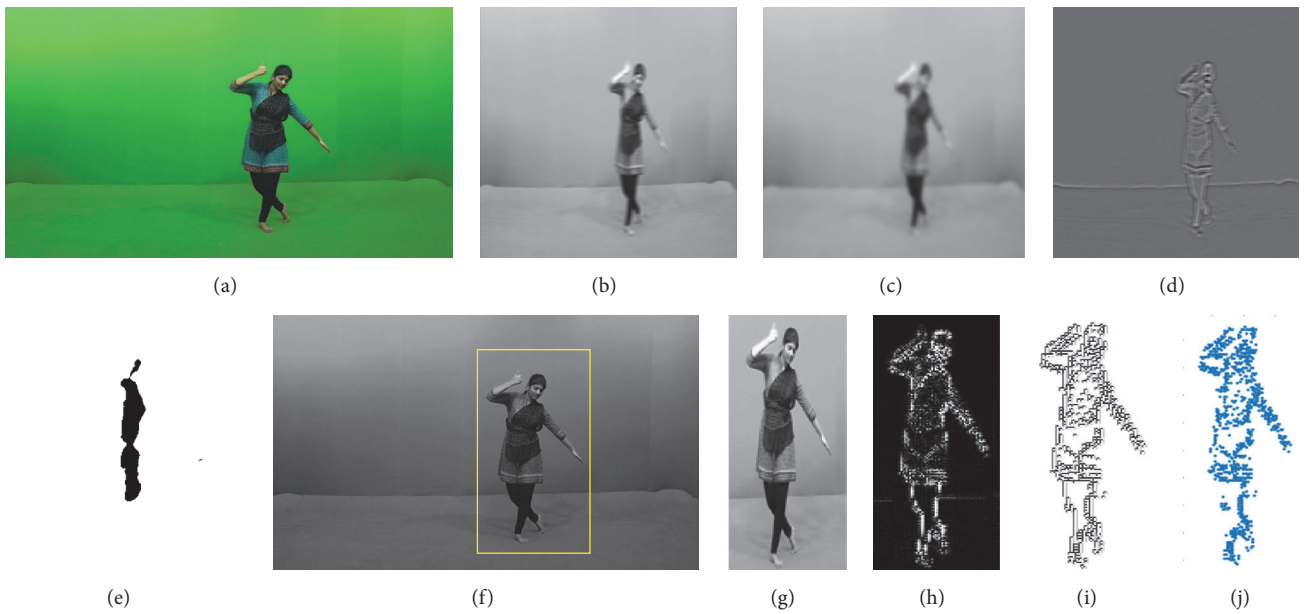


FIGURE 7: (a) Offline frame shot from ICD video, (b) Gaussian smoothing, (c) averaging, (d) saliency map, (e) silhouette creation, (f) dancer identification, (g) extracted dancer, (h) wavelet reconstructed features, (i) LBP features from wavelet, and (j) constructed 2D point shape cloud.

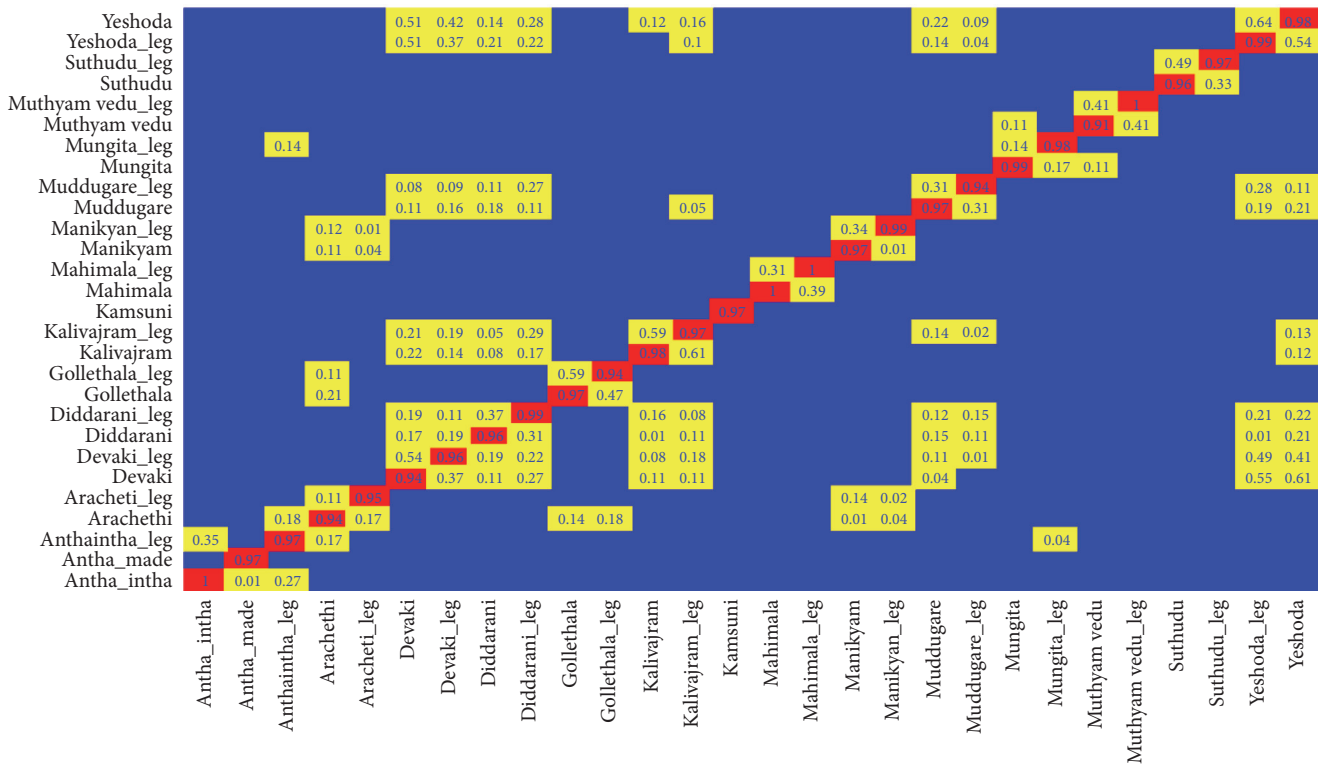


FIGURE 8: Exp-1 early fusion confusion matrix with same offline dancer in training and test video.

shapes, pose shapes, and movement speeds between the two dances.

Early feature fusion in exp-2 provided a recognition 0.81 averaged over a set of 4 video samples of the same song. However, late feature fusion with more versatile feature representing a dance frame produced a recognition rate of 0.91. Exp-3 trains the Adaboost classifier with online dance video content and tests with the same set for early fusion and late feature fusion. The initial shape extraction from online dance video content is shown in Figure 12.

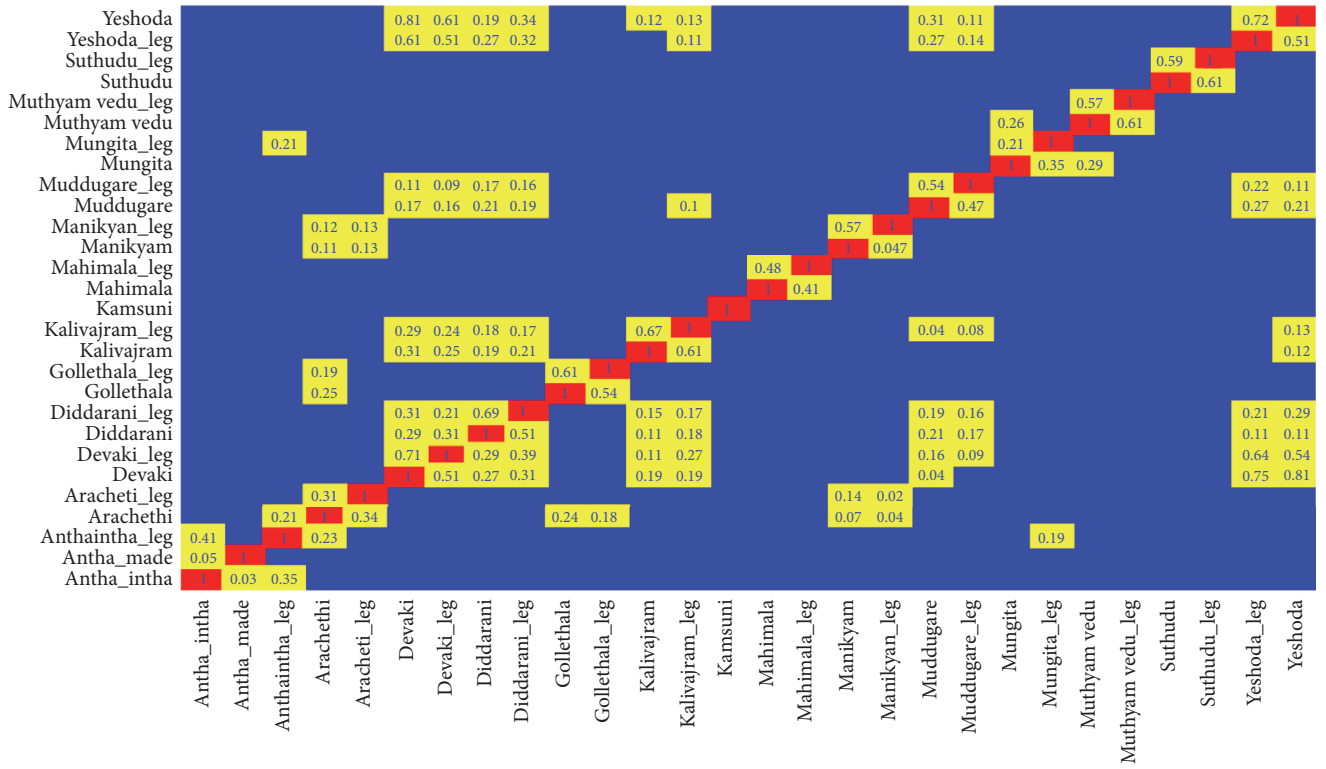


Figure 9: Exp-1 late fusion confusion matrix with same offline dancer in training and test video.

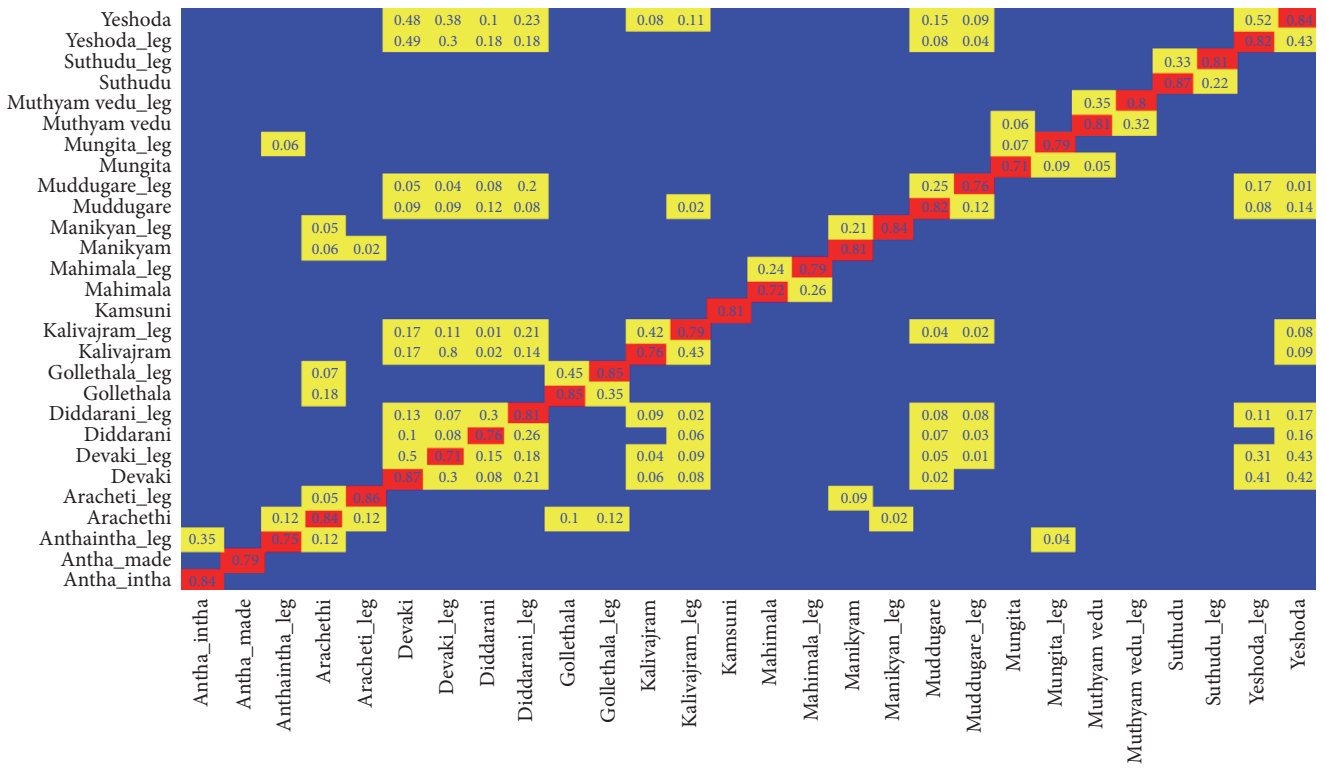


Figure 10: Exp-2 early fusion confusion matrix with different offline dancer in training and test video.

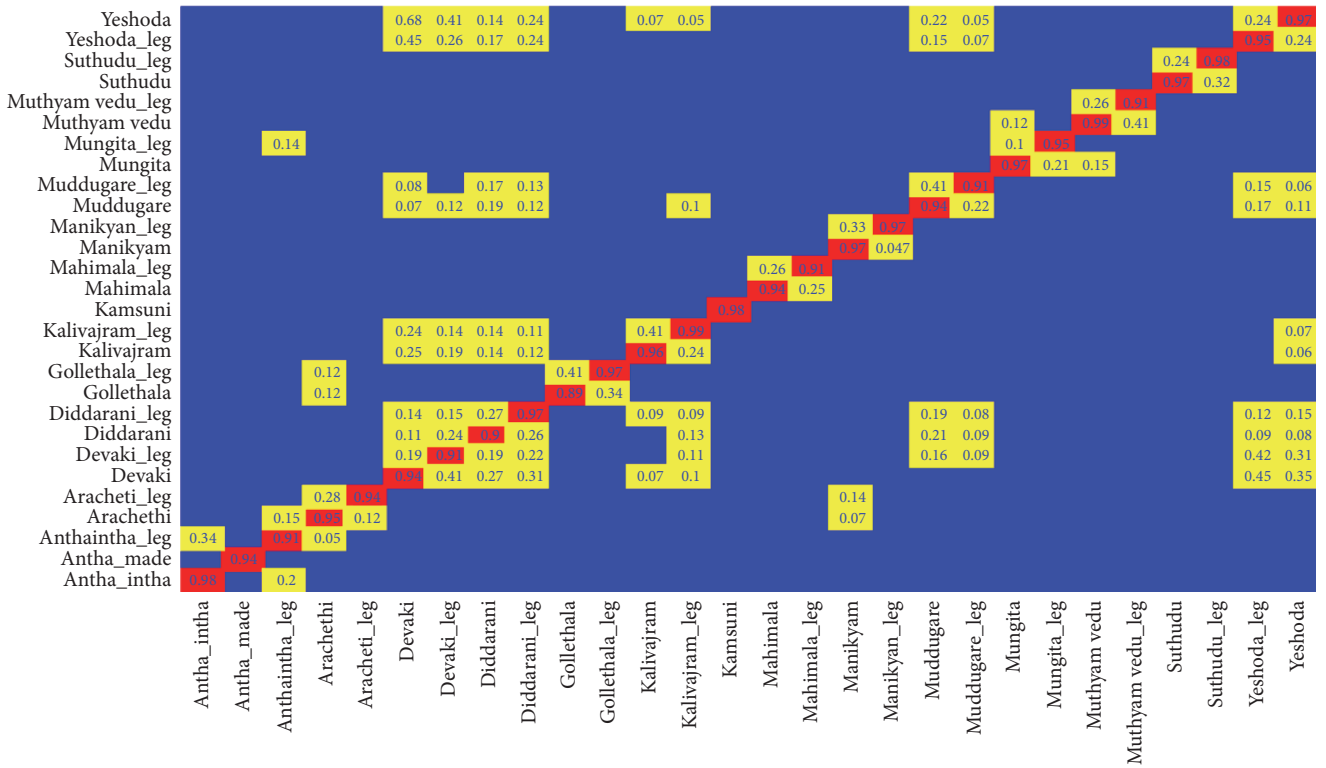


Figure 11: Exp-2 late fusion confusion matrix with different offline dancer in training and test video.

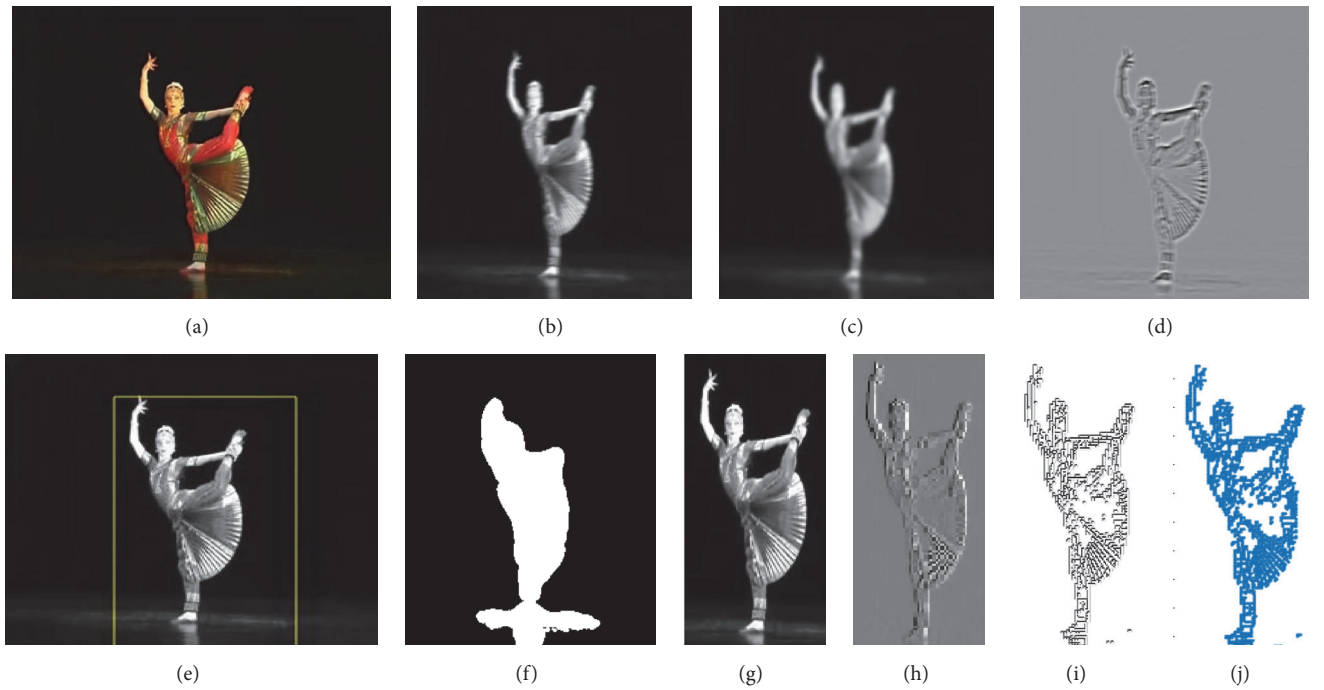


Figure 12: (a) Gray frame shot from ICD video, (b) Gaussian smoothing, (c) averaging, (d) saliency map, (e) silhouette creation, (f) dancer Identification, (g) extracted Dancer, (h) wavelet reconstructed features, (i) LBP Features from Wavelet, and (j) constructed 2D Point Shape Cloud.

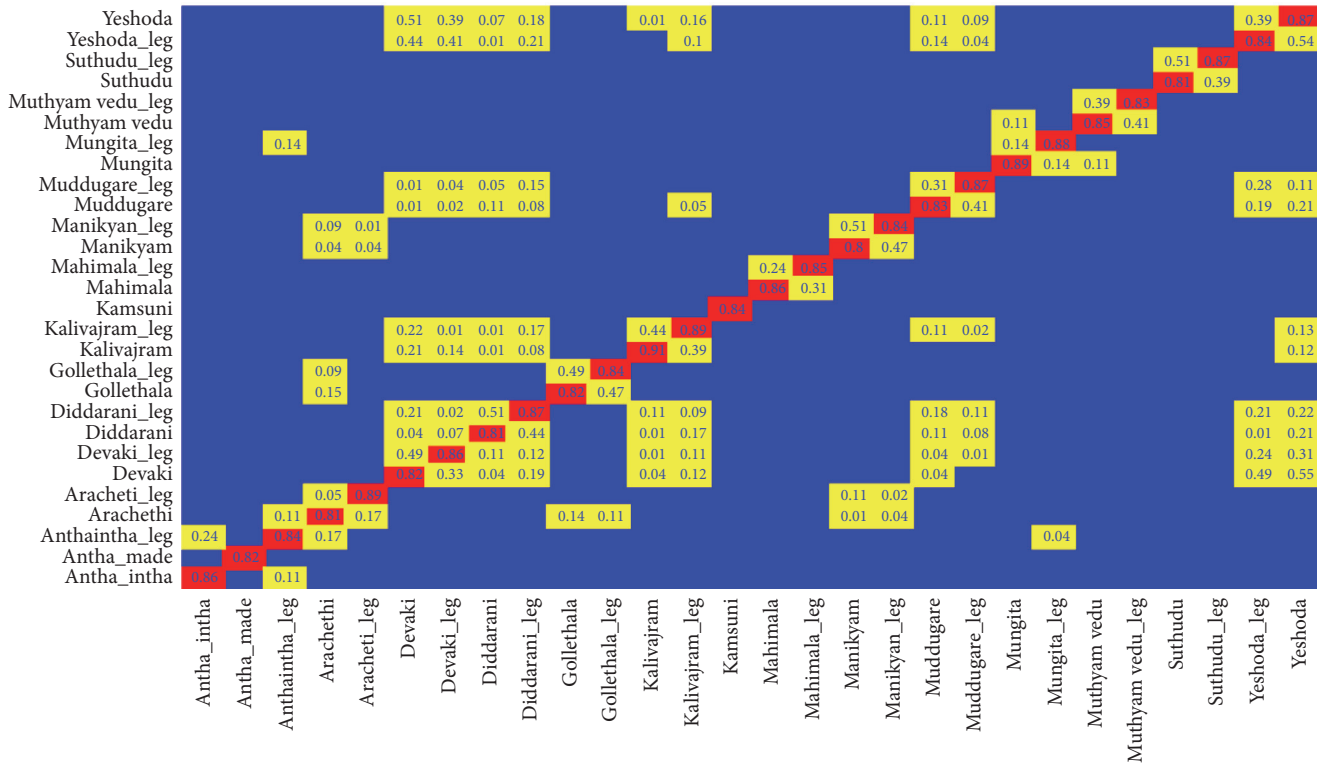


FIGURE 13: Exp-3 early fusion confusion matrix with same online dancer in training and test video.

The confusion matrices for early fusion and late fusion are presented in Figures 13 and 14 for exp-3, online dance videos, respectively.

The recognition rate decreased for online dance videos as these are unconstrained in real time capture. Problems such as vibrant video backgrounding, poor or over lighting, and motion blurring in online videos made the feature extraction complicated. However, the proposed algorithm with early feature fusion has resulted in an average recognition rate of 0.84 for one test video under these circumstances. The average recognition rate for 4 test samples in exp-3 is 0.83. The problems in online videos are effectively handled by increasing the number of features to represent a dance pose. Figure 14 gives the confusion matrix with late fusion features. The average recognition rate is 0.93 for 4 test videos.

Similarly, we obtained average recognition of 0.68 with early feature fusion and 0.82 with late fusion, respectively. The confusion matrices are shown in Figures 15 and 16.

To summarize, the results obtained with the proposed early fusion and late feature fusion on offline and online dance videos are presented in Table 1. The dataset consists of 4 online and offline videos. For each video, we computed the average recognition rate obtained by taking mean of all recognition rates per dance pose.

From Table 1, the performance of the Adaboost classifier on different online and offline dance videos can be recorded. This data can be interpreted to understand the ability of features to uniquely model dancers pose in a real time dance video to measure the dancers' performance. Harr and LBP together model global and local dance shapes, respectively. However, these features have constraints on motion blurring, scale variations, brightness, and contrast variation in the video frames. Apart from these image variations, we have camera vibrations, dancer shape variations, and video backgrounds to restrict these feature vectors from uniquely modelling a dancer in the video frame. To compensate for these variations, additional features in the form of shape signatures, Zernike moments, and Hu moments are proposed to represent a dance pose. Shape signature models the dancer pose uniquely with an integral kernel having Gaussian characteristics. The shape signature is calculated on the shape curve on a set of frames which matches similar dance pose at a different location in a video sequence. Similarly, ZM are linear orthogonal representation of video data that handle dancer's movements in spatial domain. Hu moments handle camera vibrations and other nonlinear movements of the dancer in the video frames.

Early fusion and late fusion concepts are introduced to understand which set of features can correctly and uniquely model a dance pose irrespective of the constraints during video capture. Simulations show that late fusion and more number of features are necessary for training Adaboost classifier. Same dancer videos resulted in better recognition rates compared to different dancer in both offline and online dance videos. We also tested the classifier with HOG (Histogram of oriented Gradients), SIFT (Scale Invariant Feature Transform), and SURF (Speeded Up Robust Features) with MCMSAB classifier. Training vector is made of 50 best

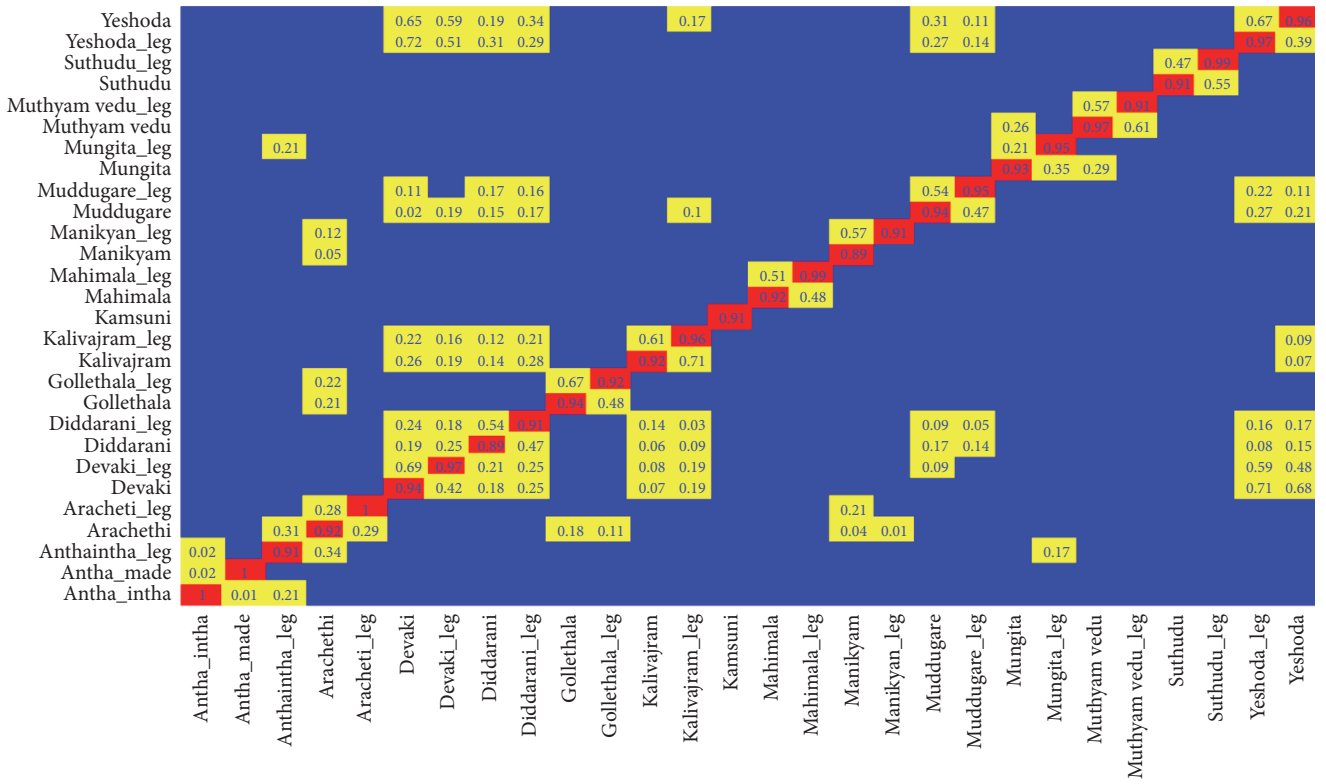


FIGURE 14: Exp-3 late fusion confusion matrix with same online dancer in training and test video.

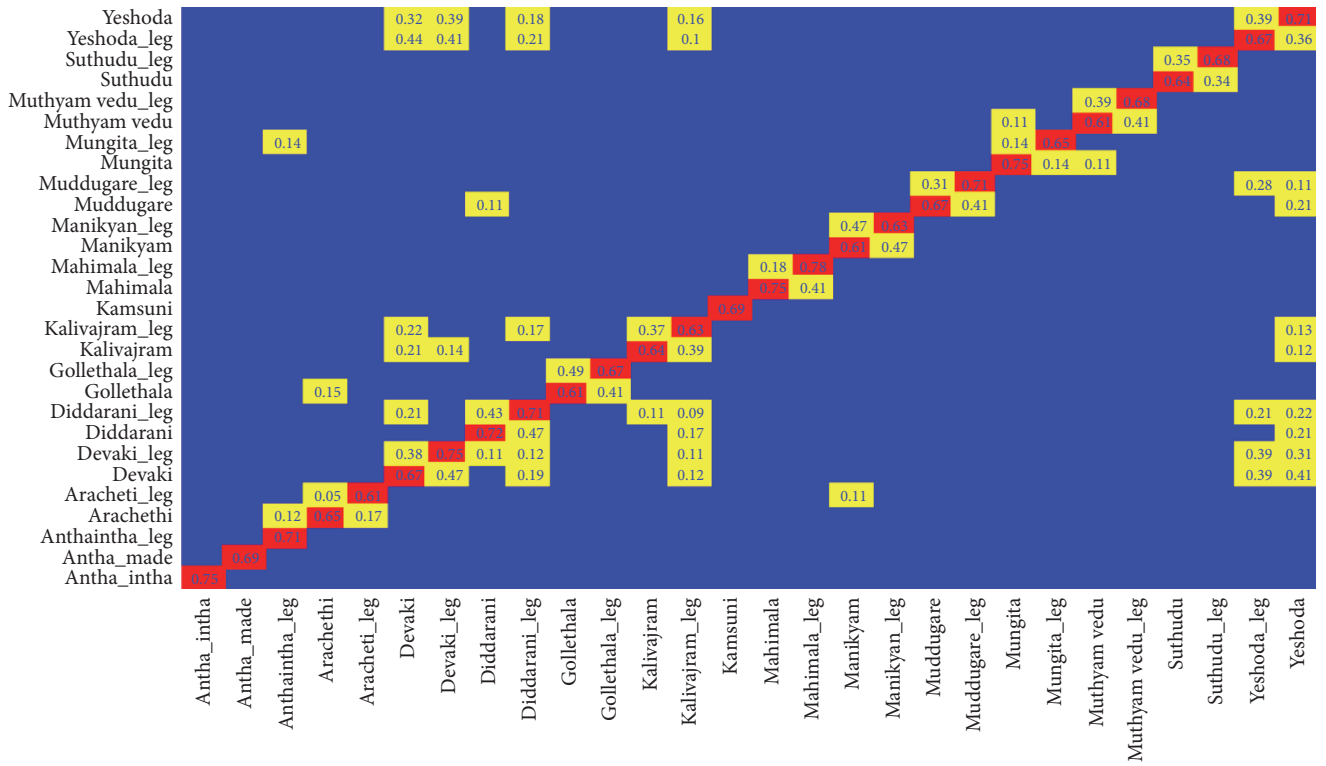


FIGURE 15: Exp-4 early fusion confusion matrix with different online dancer in training and test video.

Table 1: Summary of mean recognition rates for 4 offline and online dance performances for the song "Siva Shamboo" in kuchipudi dance.

Training dance video	Early PCA fusion on with Haar and LBP				Late fusion with Haar, LBP, ZM, HuM and SS			
	Exp-1	Exp-2	Exp-3	Exp-4	Exp-1	Exp-2	Exp-3	Exp-4
Offline video-1	0.95	0.88			0.99	0.92		
Offline video-2	0.93	0.82			0.99	0.91		
Offline video-3	0.94	0.78			0.98	0.85		
Offline video-4	0.87	0.74			0.92	0.81		
Online video-1			0.92	0.80			0.98	0.85
Online video-2			0.93	0.78			0.97	0.82
Online video-3			0.92	0.69			0.99	0.77
Online video-4			0.81	0.66			0.88	0.71

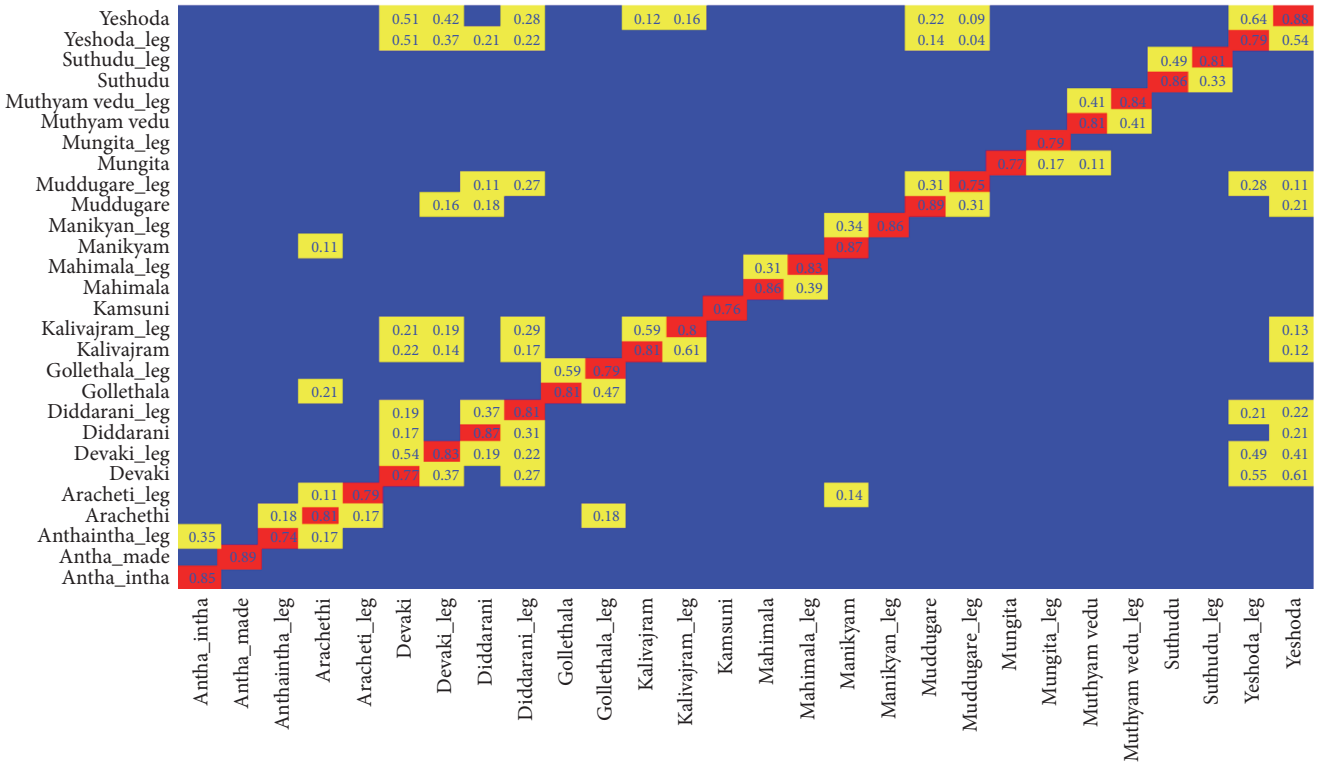


Figure 16: Exp-4 late fusion confusion matrix with different online dancer in training and test video.

features and the same number is used for testing. The average recognition rates were 0.84 for HOG, 0.82 for SIFT, and 0.8 for SURF in Exp-1 where same training and query dance video is used. But for different dancer videos the recognition dropped to 0.67, 0.65, and 0.59 in case of exp-2 for HOG, SIFT, and SURF, respectively. Similar results were seen in Exp-3 and Exp-4 for online dance videos. However, in Exp-4, for different dance video sets for training and testing have drastically reduced the recognition rate of the classifier by 50%. The drop-in classifier performance can be attributed to the poor feature extraction due a large variation in the video frames even though they have same dance pose.

HOG, SIFT, and SURF features are extracted from the original gray scale video frame. To improve their performance the algorithms are applied on the extracted dancer from our Haar-LBP sparse segmentation module. On the segmented and extracted dancer, the average recognition rate from HOG is 0.89, SOFT is 0.91, and SURF is 0.82 for exp-1. In exp-2 these values were again reduced by a factor of 18%. Exp-3 and Exp-4 showed an improvement of 30% increase in average recognition rate for all three feature extraction models. However, after consecutive testing and measurements these state-of-the art features reported less overall average recognition rates compared to the proposed late fusion features that form a complete set to model a dance video sequence.

The set of 5 late fused features with 73 attributes per frame performed well with MCMLAB classifier compared to early fusion and other feature extraction models; it is time to test for different classifiers. We applied the same

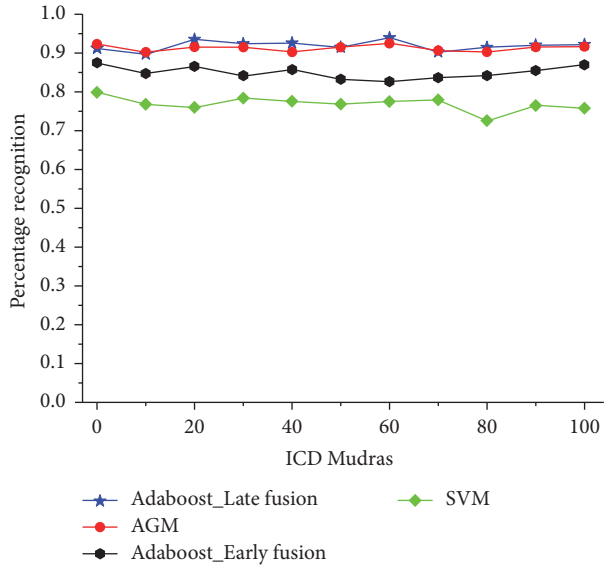


Figure 17: Comparison of classifiers for dance pose recognition.

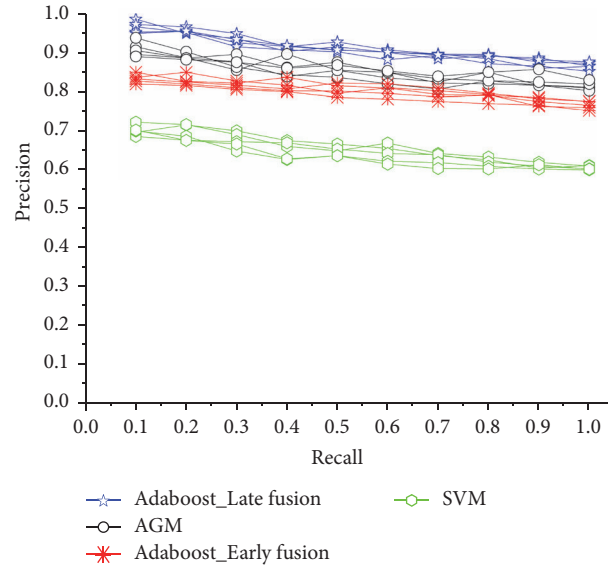


Figure 18: Classifier performance measurements.

late fusion features to adaptive graph matching (AGM) and Support Vector machine classifier. In AGM, each node is modelled with 73 features and the distance between the features is models as edge. For each dance pose video, a one-to-one matching on both node and edge feature is calculated. Recognition rate is calculated using (24). Multiclass SVM is the most trusted classifier for character recognition. Hence we apply the late fused features to a SVM classifier and each dance pose is measured.

Three classifier are compared here with late fusion features and the other Adaboost with early fusion features taking it to 4 classifiers. All these classifiers are compared for recognition rate and efficiency. We plot average recognition for the 4 classification models averaged across 4 offline and online videos in Figure 17.

From Figure 17, the Adaboost with late feature fusion with 5 features outperforms the AGM and SVM. However, AGM comes close to Adaboost classifier and sometimes is better than MCMLAB on late fusion features for dance pose recognition. Nevertheless, AGM is far slower than MCMLAB algorithm. SVM came last in the comparison due to inefficiency in defining initial support vectors for classification from the fused feature vectors. MCMLAB with early fusion features is better than SVM classifier. MCMLAB is better, if the example set is uniquely defined by the feature set and the proposed features are the best choice for Indian classical dance pose recognition.

Equations (25) and (26) are used to calculate precision and recall with late and early fusion on Adaboost for 2 offline and 2 online video sets. Similarly, the same datasets are used for AGM and SVM classifiers. One video will be used for same feature train–test model and other is a different train–test model. The average precision and recall values are plotted as a graph in Figure 18.

The ability to recall precisely the same label as the query video as a performance measure is plotted in Figure 18. It shows the same result as above, making MCMLAB as the

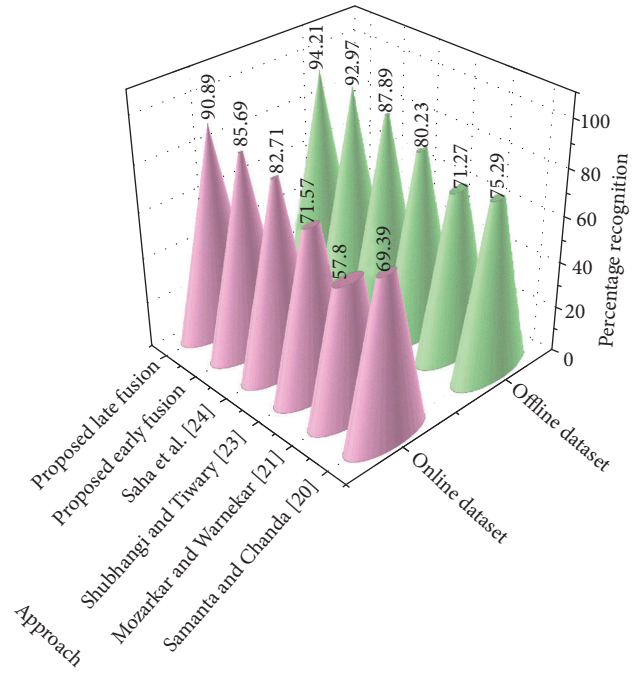


Figure 19: Compassions with other models of ICD with the proposed early and late fusion models with Adaboost classifier.

classifier for dance pose recognition with Haar, LBP, ZM, Hu moments, and SS giving satisfactory outcomes in robust comparisons.

Figure 19 shows the percentage recognition obtained from methods in [20, 21, 23, 24] along with the proposed early and late feature fusion models. The recognition percentage is averaged over the entire dataset. The plots in Figure 19 highlight the use of multiple features for various representations of moving objects in a dance video for accurate classification and recall.

5. Conclusion

Indian classical dance classification is a complex problem for machine vision research. The features representing the dancer should focus on the entire human body shapes. Hand and leg shape segmentation is a critical part of a ICD. In this work, we proposed a fully automated ICD consisting of dancer identification, extraction, segmentation, and feature representation and classification. Saliency based dancer identification and extraction helps in reducing the image space. Wavelet reconstructed local binary patterns are used for feature representation preserving local shape content of hands and legs. Two fusion models are proposed for feature fusion. Early fusion at the segmentation stage with PCA based Haar wavelet and LBP is used and late fusion using the Zernike moments, Hu moments, and shape signatures is used with Haar and LBP is proposed. Multiclass multilabel Adaboost on features of early fusion and late fusion between two sets of dance video data is the classifier. Multiple experiments on online and offline ICD video data are tested. Dance video data is labelled as per the vocal song sequence. The early and late features and classifiers performance tests show that the proposed late fusion features and multiclass multilabel Adaboost classifier give better classification accuracy and seed compared to AGM and SVM. More action features can be added for representing dancer more realistically by elimination backgrounds and blurring artefacts to improve the efficiency of the classifier.

Conflicts of Interest

The authors declare that they have no conflicts of interest related to this research in any form.

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[2] J. G. Lochtefeld, *The Illustrated Encyclopaedia of Hinduism*, vol. 1, The Rosen Publishing Group, 2002.

[3] P. Chakravorty, "Hegemony, dance and nation: the construction of the classical dance in India," *South Asia: Journal of South Asia Studies*, vol. 21, no. 2, pp. 107–120, 1998.

[4] A. Sinha, *Let's Know Dances of India*, Star Publications, 2006.

[5] H. Rahmani, A. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1.

[6] E. Rodolà, S. R. Bulò, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 4177–4184, Columbus, OH, USA, June 2014.

[7] D. Das Dawn and S. H. Shaikh, "A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector," *Visual Computer*, vol. 32, no. 3, pp. 289–306, 2016.

[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 3551–3558, Sydney, Australia, December 2013.

[9] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3169–3176, June 2011.

[10] P. V. V. Kishore, M. V. D. Prasad, D. A. Kumar, and A. S. C. S. Sastry, "Optical Flow Hand Tracking and Active Contour Hand Shape Features for Continuous Sign Language Recognition with Artificial Neural Networks," in *Proceedings of the 6th International Advanced Computing Conference, IACC 2016*, pp. 346–351, Bhimavaram, India, February 2016.

[11] A. Jalal, Y. Kim, Y. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognition*, vol. 61, pp. 295–308, 2017.

[12] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[13] K. Vatsyayan, Indian classical dance. Ministry of Information and Broadcasting, Government of India, 1992.

[14] A. Mohanty, P. Vaishnavi, P. Jana et al., "Nrityabodha: towards understanding indian classical dance using a deep learning approach," *Signal Processing: Image Communication*, vol. 47, pp. 529–548, 2016.

[15] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pp. 1385–1392, Colorado Springs, Colo, USA, June 2011.

[16] F. Wang and Y. Li, "Beyond physical connections: tree models in human pose estimation," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 596–603, Portland, Ore, USA, June 2013.

[17] S. Saha, S. Ghosh, A. Konar, and A. K. Nagar, "Gesture recognition from Indian classical dance using kinect sensor," in *Proceedings of the 5th International Conference on Computational Intelligence, Communication Systems, and Networks, CICSyN 2013*, pp. 3–8, Madrid, Spain, June 2013.

[18] S. Samanta, P. Purkait, and B. Chanda, "Indian Classical Dance classification by learning dance pose bases," in *Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision, WACV 2012*, pp. 265–270, Breckenridge, Colo, USA, January 2012.

[19] K. V. V. Kumar and P. V. V. Kishore, "Indian Classical Dance Mudra Classification Using HOG Features and SVM Classifier," in *Proceedings of the In Proceedings of International Conference on smart computing and information systems*, India, 2017.

[20] S. Samanta and B. Chanda, "Indian classical dance classification on manifold using jensen-bregman logdet divergence," in *Proceedings of the 22nd International Conference on Pattern Recognition, ICPR 2014*, pp. 4507–4512, Stockholm, Sweden, August 2014.

[21] S. Mozarkar and C. S. Warnekar, "Recognizing bharatnatyam mudra using principles of gesture recognition gesture recognition," *International Journal of Computer Science and Network*, vol. 2, no. 2, pp. 46–52, 2013.

[22] M. Devi and S. Saharia, "A two-level classification scheme for single-hand gestures of Sattriya dance," in *Proceedings of the 2016 International Conference on Accessibility to Digital World (ICADW)*, pp. 193–196, Guwahati, India, December 2016.

[23] Shubhangi and U. S. Tiwary, "Classification of Indian classical dance forms," *Lecture Notes in Computer Science (including*

subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10127, pp. 67–80, 2017.

[24] S. Saha, S. Ghosh, A. Konar, and R. Janarthanan, "A study on leg posture recognition from Indian classical dance using Kinect sensor," in *Proceedings of the 2013 International Conference on Human Computer Interactions, ICHCI 2013*, Chennai, India, August 2013.

[25] B.-W. Hong and S. Soatto, "Shape matching using multiscale integral invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 151–160, 2015.

[26] D. Dahmani and S. Larabi, "User-independent system for sign language finger spelling recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1240–1250, 2014.

[27] G. Rätsch, T. Onoda, and K. R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[28] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 79–84, May 2004.

[29] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Statistics and Its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[30] C. Qi, Z. Zhou, Y. Sun, H. Song, L. Hu, and Q. Wang, "Feature selection and multiple kernel boosting framework based on PSO with mutation mechanism for hyperspectral classification," *Neurocomputing*, vol. 220, pp. 181–190, 2017.

[31] C. I. Patel, S. Garg, T. Zaveri, A. Banerjee, and R. Patel, "Human action recognition using fusion of features for unconstrained video sequences," *Computers and Electrical Engineering*, 2015.

[32] J. Wang, M. She, S. Nahavandi, and A. Kouzani, "A review of vision-based gait recognition methods for human identification," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, DICTA 2010*, pp. 320–327, Sydney, Australia, December 2010.

[33] D. P. Tian, "A review on image feature extraction and representation techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 4, pp. 385–395, 2013.

[34] M. Yang, K. Kpalma, and R. Joseph, A survey of shape feature extraction techniques. (2008): 43–90.

[35] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[36] E. G. Karakasis, A. Amanatiadis, A. Gasteratos, and S. A. Chatzichristofis, "Image moment invariants as local features for content based image retrieval using the Bag-of-Visual-Words model," *Pattern Recognition Letters*, vol. 55, pp. 22–27, 2015.

[37] G. A. Papakostas, D. E. Koulouriotis, E. G. Karakasis, and V. D. Tourassis, "Moment-based local binary patterns: a novel descriptor for invariant pattern recognition applications," *Neurocomputing*, vol. 99, pp. 358–371, 2013.

[38] S. Nigam and A. Khare, "Integration of moment invariants and uniform local binary patterns for human activity recognition in video sequences," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 17303–17332, 2016.

[39] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[40] D. Zhang and G. Lu, "Content-based shape retrieval using different shape descriptors: A comparative study," in *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo, ICME 2001*, pp. 1139–1142, jpn, August 2001.

[41] M. Khare, R. K. Srivastava, and A. Khare, "Object tracking using combination of daubechies complex wavelet transform and Zernike moment," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 1247–1290, 2017.

[42] M. K. Hu, "Visual pattern recognition by moment invariant," *IRE Transactions of Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[43] P. V. V. Kishore, P. E. Rajesh Kumar, K. Kumar, and S. R. C. Kishore, "Video audio interface for recognizing gestures of indian sign," *International Journal of Image Processing (IJIP)*, vol. 5, no. 4, 479 pages, 2011.

[44] P. V. V. Kishore, M. V. D. Prasad, C. R. Prasad, and R. Rahul, "4-Camera model for sign language recognition using elliptical fourier descriptors and ANN," in *Proceedings of the 4th International Conference on Signal Processing and Communication Engineering Systems, SPACES 2015 - In Association with IEEE*, pp. 34–38, ind, January 2015.

[45] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 79–84, May 2004.

[46] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Advances in Knowledge Discovery and Data Mining*, vol. 3056 of *Lecture Notes in Computer Science*, pp. 22–30, Springer, Berlin, Germany, 2004.