Applied Computational Intelligence and Soft Computing

Imaging, Vision, and Pattern Recognition

Lead Guest Editor: Mourad Zaied
Guest Editors: Imed Bouchrika, Anil Kumar, Fouad Slimane, and Ridha Ejbali

Imaging, Vision, and Pattern Recognition

Imaging, Vision, and Pattern Recognition

Lead Guest Editor: Mourad Zaied
Guest Editors: Imed Bouchrika, Anil Kumar, Fouad Slimane, and Ridha Ejbali

Editorial Board

Contents

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2018, Article ID 1070183, 1 page
https://doi.org/10.1155/2018/1070183

Editorial

Imaging, Vision, and Pattern Recognition

Mourad Zaied,[1] **Imed Bouchrika**,[2] **Anil Kumar**,[3] **Fouad Slimane**,[4] **and Ridha Ejbali**[5]

[1] *University of Gabès, Gabès, Tunisia*
[2] *University of Souk Ahras Mohamed Chérif Messaadia, Souk Ahras, Algeria*
[3] *Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, India*
[4] *Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland*
[5] *Faculty of Sciences of Gabès, Gabès, Tunisia*

Correspondence should be addressed to Mourad Zaied; mourad.zaied@ieee.org

Received 15 April 2018; Accepted 15 April 2018; Published 3 May 2018

Pattern recognition, machine vision, and imaging are a set of techniques and methods belonging to machine learning. They focus on approach linked to recognition of patterns, regularities in data, computer vision, and image processing. Pattern recognition, machine vision, and imaging share other topics such as artificial intelligence and learning techniques.

Moreover, signal processing is a fundamental domain based on several techniques which encompasses the fundamental theory, applications, algorithms, and implementation of processing or transferring information. It uses mathematical, statistical, and computational representations and formalisms for representation, modeling, analysis, and synthesis of data.

This special issue is dedicated to latest developments in the area of machine learning methods. The target audiences were researchers in machine learning and computational intelligence applied to image processing, signal processing, biomedical system, and security of the environment. After a strict review, seven articles from researchers around the world were finally accepted. In one paper, A. Helwan and D. U. Ozsahin proposed an approach based on a sliding window based machine learning system for the left ventricle localisation in MR cardiac images. In another paper, L. Duan et al. proposed a deep hashing based fusing index method for large-scale image retrieval. In one of the papers, H. Pan et al. proposed a regular k-shrinkage thresholding operator for the removal of mixed Gaussian-impulse noise. In another paper, X. Sun et al. proposed a color image denoising based on guided filter and adaptive wavelet threshold. In one paper, N. Nematzadeh and D. M. W. Powers proposed local and global perception from multiple scales to multiscale. Y. Liang et al. proposed multifeature and neighbor based salient region detection for social images. In another one, W. Chinsatit and T. Saitoh proposed CNN-based pupil center detection for wearable gaze estimation system.

Acknowledgments

We thank all the authors and reviewers for their great contributions to this special issue. We would also like to thank Professor Hsien-Chung Wu, the Editor in Chief, for his full support.

Mourad Zaied
Imed Bouchrika
Anil Kumar
Fouad Slimane
Ridha Ejbali

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2018, Article ID 1014595, 16 pages
https://doi.org/10.1155/2018/1014595

Research Article

Saliency Aggregation: Multifeature and Neighbor Based Salient Region Detection for Social Images

Ye Liang,[1,2] **Congyan Lang**,[1] **Jian Yu**,[1] **Hongzhe Liu**,[2] **and Nan Ma**[2]

[1]*School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China*
[2]*Beijing Union University, Beijing 100101, China*

Correspondence should be addressed to Ye Liang; liangye@buu.edu.cn

Received 30 June 2017; Revised 20 October 2017; Accepted 14 November 2017; Published 1 January 2018

Academic Editor: Anil Kumar

The popularity of social networks has brought the rapid growth of social images which have become an increasingly important image type. One of the most obvious attributes of social images is the tag. However, the sate-of-the-art methods fail to fully exploit the tag information for saliency detection. Thus this paper focuses on salient region detection of social images using both image appearance features and image tag cues. First, a deep convolution neural network is built, which considers both appearance features and tag features. Second, tag neighbor and appearance neighbor based saliency aggregation terms are added to the saliency model to enhance salient regions. The aggregation method is dependent on individual images and considers the performance gaps appropriately. Finally, we also have constructed a new large dataset of challenging social images and pixel-wise saliency annotations to promote further researches and evaluations of visual saliency models. Extensive experiments show that the proposed method performs well on not only the new dataset but also several state-of-the-art saliency datasets.

1. Introduction

Images and videos are two of the main ways for social entertainments and communications. With the popularity of photo sharing websites, social images have become an important type. The most obvious feature of social images is that they typically have several tags to describe the contents. How to use the tags for multimedia tasks, such as image indexing and retrieval [1, 2], has attracted increasing attention these days [3]. However, tags are seldom considered in state-of-the-art salient region detection models. Therefore, in this paper, we focus on salient region detection of social images using both appearance features and tag features.

With the development of saliency detection, a large number of saliency detection algorithms have been developed [4–6]. It has been found that only relying on low-level features cannot achieve satisfactory results. The researches have proved that the hierarchical and deep architectures [7–12] for salient region detection are very effective. Thus, a salient region detection method based on deep learning is proposed in this paper. In addition, various priors are also very important in salient region detection [13], for example, face [14–16], car [17], color [14], center bias [13], and objectness [18–20]. Intuitively, the tags could potentially be important high-level semantic cues for salient region detection [16, 21]. Thus, tags are incorporated into our salient region detection models.

It is observed that different methods perform differently in saliency analysis [22]. The performance of saliency varies with individual images. The problem also exists in deep feature based methods and handcrafted feature based methods. So handcrafted feature based detection methods can be considered as complementarities to deep feature based detection methods. However, the fusion process is without ground truth. It is nontrivial to determine which saliency map is better. The good saliency aggregation model should work on each individual image and be able to consider the performance gaps appropriately. Therefore, how to fuse saliency maps of different detection methods is a key issue to be solved in the paper.

The framework of salient region detection is shown in Figure 1. It includes two parts: deep learning based salient region detection and handcrafted feature based salient

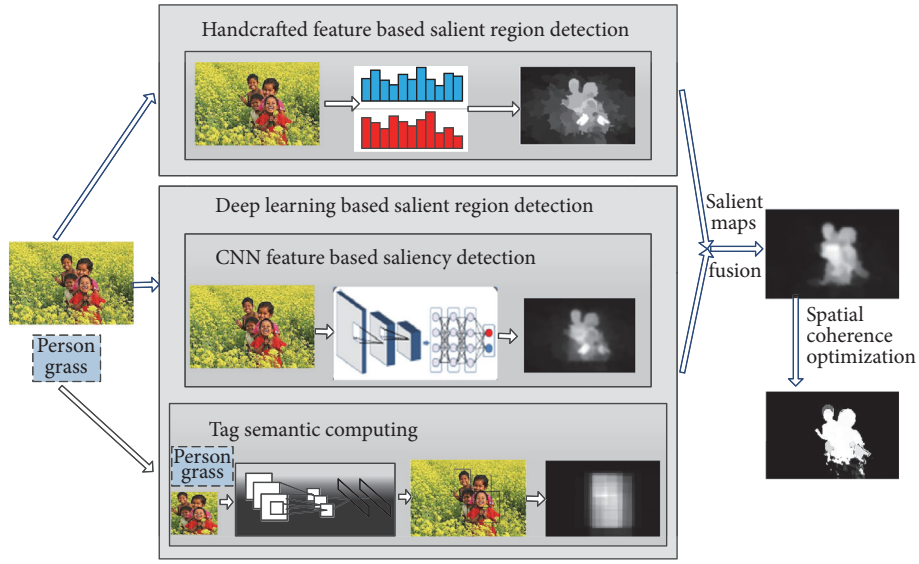


Figure 1: Framework.

region detection. Deep features include CNN (convolution neural network) features and tag features. Finally, the spatial coherence of saliency maps is optimized through the fully connected conditional random field model.

There are a variety of saliency detection benchmark datasets, either from saliency detection field [7, 8, 23–26] or from image segmentation field [27–29]. To promote further researches and evaluations on visual saliency detection for social images, it is necessary to construct a new dataset of social images.

The paper focuses on salient region detection of social images. The contributions of this paper are twofold. First, a deep learning based salient region detection method for social images is proposed, considering both appearance features and tag features. Second, tag neighbor and appearance neighbor based saliency aggregation method is proposed, which fuses state-of-the-art handcrafted feature based detection methods with our deep learning based detection method. The aggregation method is dependent on each specific individual image and considers the saliency performance gaps appropriately. So the detection model has fully taken advantage of image tags.

The rest of the paper is organized as follows. The deep learning based model is proposed in Section 2. Section 3 discusses the handcrafted feature based detection models. In Section 4, the saliency aggregation method is proposed. Spatial coherence optimization is discussed in Section 5. In Section 6, the new saliency dataset of social images is introduced. In Section 7, extensive experiments are performed and analyzed. Finally, conclusions are given in Section 8.

2. Deep Learning Based Salient Region Detection

Deep learning based salient region detection uses two types of features, appearance based CNN (convolution neural network) features and social image tag features. They are discussed in the following subsections.

2.1. CNN Based Salient Region Detection

2.1.1. Network Architecture. The deep network for appearance feature extraction has 8 layers [30] as shown in Figure 2. It includes 5 convolution layers, 2 fully connected layers, and 1 output layer. The bottom layer represents the input image and the adjacent upper layer represents the regions for deep feature extraction.

The convolution layers are responsible for the multiscale feature extraction. In order to achieve translation invariance, max pooling operation is performed after convolution operation. The learned feature is composed of 4096 elements. Fully connected layers are followed by ReLU (Rectified Linear Units) for nonlinear mapping. The dropout procedure is to avoid overfitting. ReLU performs the operation for each element in the following.

$$R\left(x^i\right) = \max\left(0, x^i\right), \tag{1}$$

where x is the feature of 4096 elements; if $x^i \geq 0$, then $\max(0, x^i) = x^i$; otherwise $\max(0, x^i) = 0$, $1 \leq i \leq 4096$.

The output layer uses softmax regression to calculate the probability of image patches being salient.

2.1.2. Multiscale CNN Feature Computation. In an image, salient regions have uniqueness, scarcity, and obvious difference with their neighborhoods. Inspired by literature [8], in order to effectively compute the saliency, three types of differences are computed, that is, the difference between the region and its neighborhoods, the difference between the region and the whole image, and the difference between the region and image boundaries. To compute these differences, four types of regions are extracted: (1) rectangle

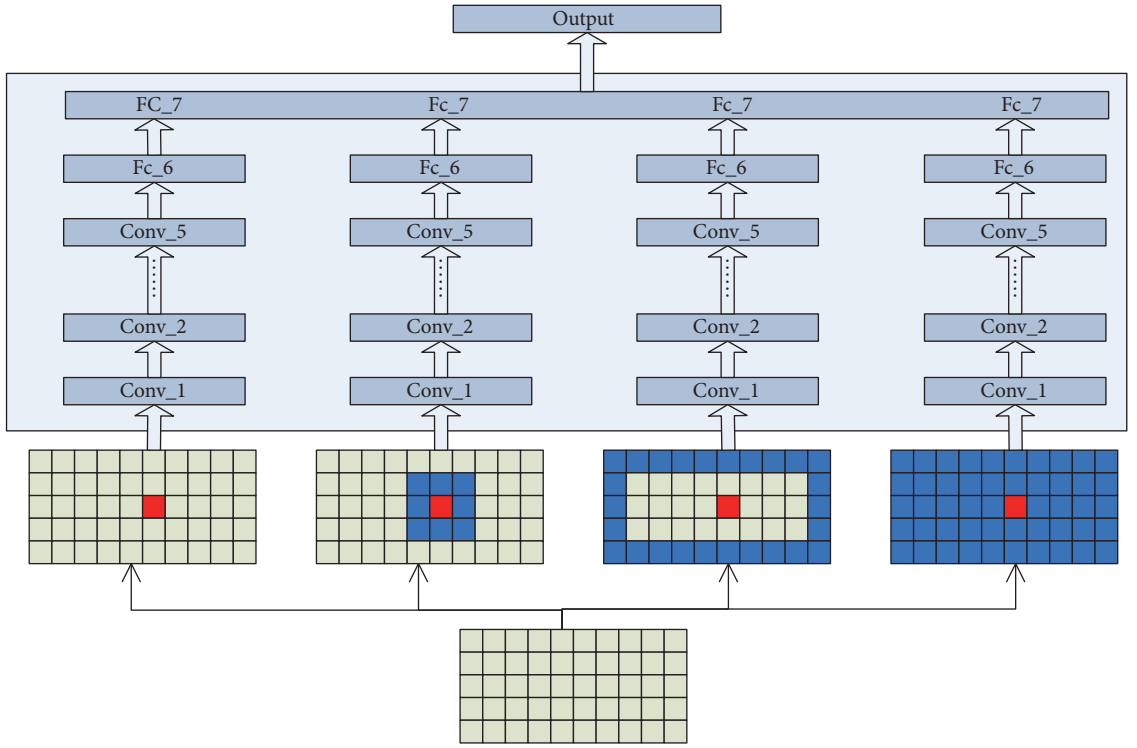


Figure 2: Architecture of network.

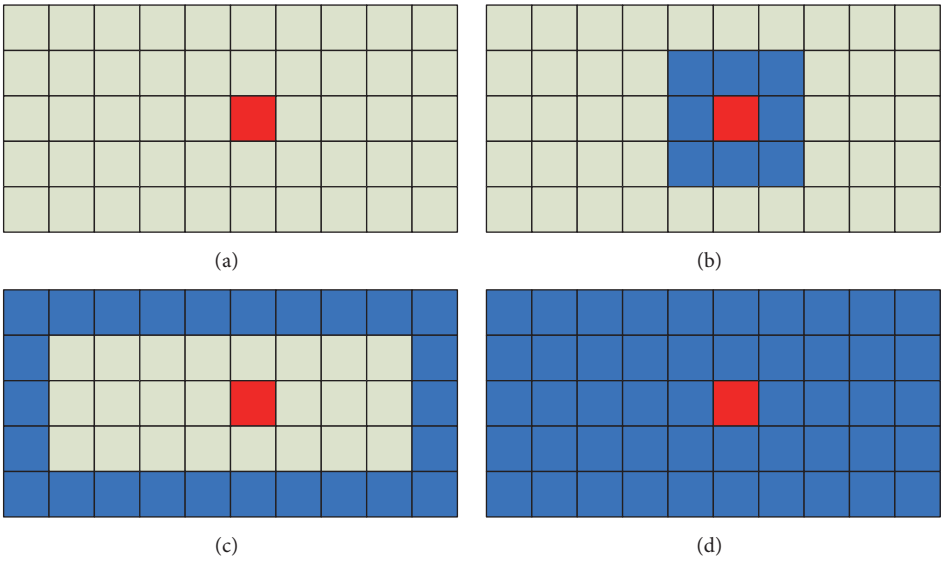


Figure 3: Four types of regions. (a) The red region denotes rectangle sample, (b) The blue regions denote neighborhoods of rectangle sample, (c) The blue regions denote boundaries of the image, (d) The blue regions denote image area except rectangle sample.

sample in a sliding window fashion; (2) neighborhoods of rectangle sample; (3) boundaries of the image; (4) image area except rectangle sample. Four types of regions are shown in Figure 3.

2.1.3. Training of CNN Network. Caffe [30], an open source framework, is used for CNN training and testing. The deep convolution neural network is originally trained on the ImageNet dataset. We extract multiscale features for each region and fine-tune the network parameters. For each image in the training set, we crop samples into 51×51 RGB patches in a sliding window fashion with a stride of 10 pixels. To label the sample patches, if more than 70% pixels in the example are salient, then this sample label is 1; otherwise it is 0. Using this annotation strategy, we obtain sample regions $\{B_i\}$ and corresponding labels $\{l_i\}$.

In fine-tuning process, the cost function is the softmax loss with weight decay given by

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=0}^{1} l\{l_i = j\}\log P(l_i = j \mid \theta) + \lambda\sum_{k=1}^{8}\|W_k\|_F^2, \quad (2)$$

where θ is the learnable parameter of convolution neural network, including the bias and weights of all layers; $l\{\cdot\}$ is the indicator function; $P(l_i = j \mid \theta)$ is the probability of the ith sample being salient; λ is the parameter of weight decay; W_k is the weight of the kth layer. We use stochastic gradient descent to train the network with batch size $m = 256$, $\lambda = 0.0005$. The initial learning rate is 0.01. When the cost is stabilized, the learning rate is decreased by a factor of 0.1. 80 epochs are repeated for the training process. The dropout rate is set to 0.5 to avoid overfitting.

2.2. Tag Semantic Feature Computation. Due to the fact that objects are closely related to salient regions, we use object tags to compute semantic features. The probability that a region is a particular object reflects the possibility being a salient region to some extent. Therefore, the probabilities that regions are specific objects can be regarded as priors.

RCNN (Regions with CNN) [31] is based on deep learning and has been widely used because of its excellent object detection accuracy. In the paper, RCNN is used to detect objects; thus tag semantics are transformed into RCNN features.

Suppose there are X object detectors. For the kth detector, the detection process is as follows.

(1) Select N proposals which are more likely to contain the specific object.

(2) Compute the ith proposal probability ${p_k}^i$ of the ith proposal being the kth object, $1 \le k \le X$, $1 \le i \le N$. At the same time, each pixel in the ith proposal also has the same probability ${p_k}^i$.

(3) For N proposals, each pixel has the score $\sum_{i=1}^{N} {p_k}^i * {f_k}^i$ being the kth object. If the pixel is contained by ith proposal, then ${f_k}^i = 1$, else ${f_k}^i = 0$.

X dimension feature is obtained for each pixel after X objects detector detection. X dimension feature is normalized as f, $f \in R^X$. Each dimension of f indicates probability being a specific object.

2.3. Fusion of CNN Based Saliency and Tag Semantic Features. Assume that the saliency map is S_D and RCNN based semantic features is T; the fusion is

$$S = S_D \cdot \exp(T). \quad (3)$$

Tags are priors and play weights in fusion. S represents the fused saliency map.

3. Handcrafted Feature Based Salient Region Detection

It is observed that different methods perform differently in saliency analysis [22]. Although the overall detection effect based on deep features is better than that based on handcrafted features, the differences still exist on individual images. So handcrafted feature based salient maps can be considered as complementarities to deep feature based saliency maps. In Figure 4, the first column shows the original social images; the second shows the ground truth masks; the third shows the salient maps of DRFI method [25] which is based on handcrafted features; the last represents the salient maps of MDF method [8], which are based on deep features. We can see that the last column includes incomplete parts, unclear boundaries, and false detections. So in the paper, some state-of-the-art salient region detection methods based on handcrafted features are selected as complementarities to our proposed deep detection method.

4. Saliency Aggregation

4.1. Main Idea. It is observed that if a salient region detection method has good effects on a social image, this method has great possibility to get sound effect on similar images. The main idea of aggregation is based on this assumption.

In training process, sort lists of all detection methods on all images can be achieved. Sort lists can be seen as priors in testing.

In testing process, we search *KNN (K nearest neighbors)* images similar to the test image in the training set. Moreover, sort lists of KNN images are known in the training stage. KNN images can vote for detection methods through sort lists. Thus, the test image is able to obtain its sort list based on voting. Salient map of test image can be computed by aggregating its salient maps of different methods using sort lists.

Training process and testing process are shown in Figures 5 and 6.

4.2. Training Process. Given an image I in the training set, its ground truth is given by G; its salient maps using different detection methods is denoted as $S = \{S_1, S_2, S_3, \ldots, S_i, \ldots, S_M\}$. In this saliency map set, M is the number of detection methods, and S_i is the salient map of the ith method.

For every detection method, its salient maps can be compared with ground truth G and yield AUC (Area under ROC Curve) values. The greater the AUC value, the better the saliency detection performance. After AUC value computation, sort lists of all methods can be obtained.

For convenience, it is assumed that there are four detection methods. Sort lists are shown in Figure 7. The data structure is single linked list. Data domain of header node denotes image and pointer domain of header node points to data node. Nonheader node includes three domains: the first domain is the AUC value, the second domain is the method index, and the last domain is a pointer.

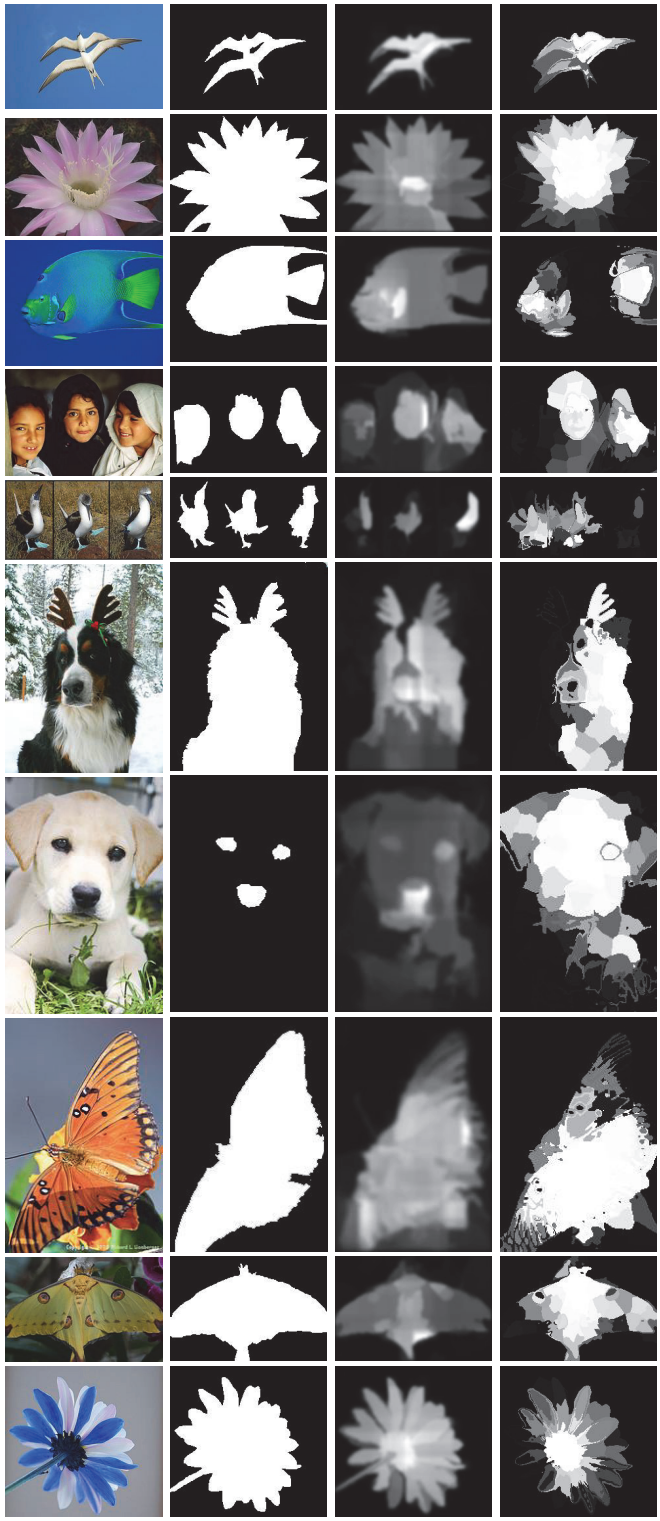


Figure 4: Examples of saliency detection results. Images in each column are original images, ground truth masks, salient maps of method DRFI [25], and salient maps of method MDF [8], respectively.

4.3. Testing Process. A social image has two parts: image and corresponding tags. In the testing set, image I and its tag set $T = \{t_1, t_2, \ldots, t_i, \ldots, t_N\}$ are given, where N is the number of tags. We search its neighbors through tag semantics and image appearance. Sort lists of neighbors can vote for salient maps of image I.

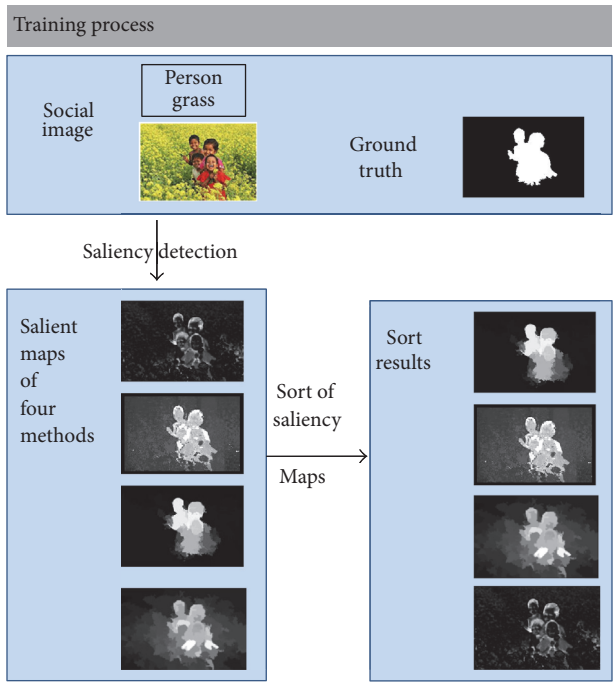


Figure 5: Training process.

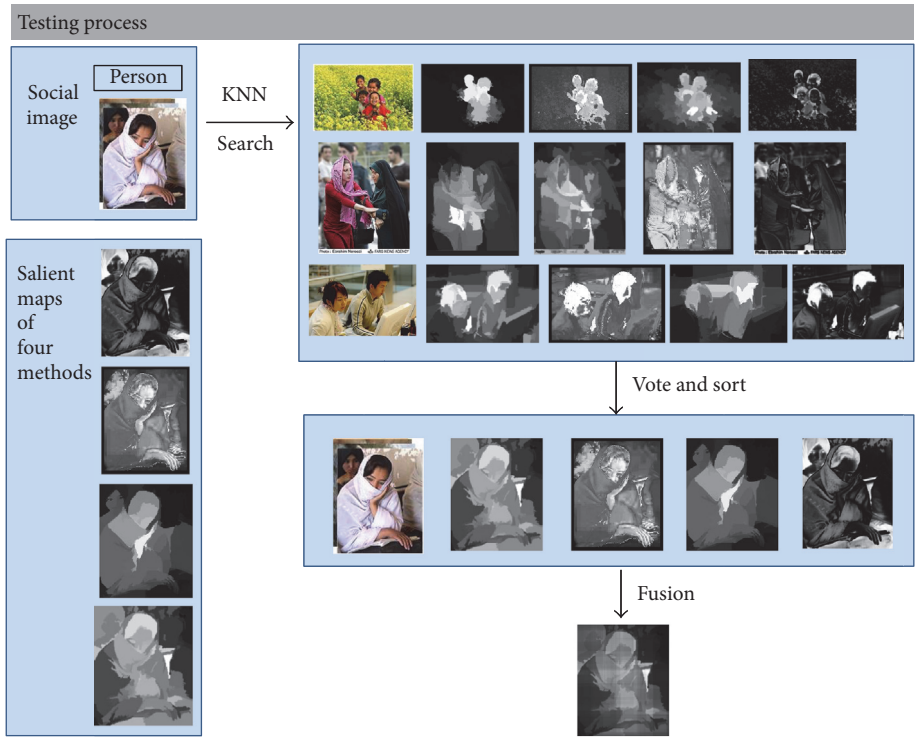


Figure 6: Testing process.

4.3.1. Tag Based Neighbor Search. There are two types of tags: object tags and scene tags. Because objects are closely related to salient regions, object tags are used in semantic search.

There are 37 object tags in the new dataset, including animal, bear, birds, cat, fox, zebra, horses, tiger, cow, dog, elk, fish, whale, vehicles, boats, cars, plane, train, person, police, military, tattoo, computer, coral, flowers, flags, tower, statue, sign, book, sun, leaf, sand, tree, food, rocks, and toy.

In these categories, animal has super class and subclass relationship with bear, birds, cat, fox, zebra, horses, tiger,

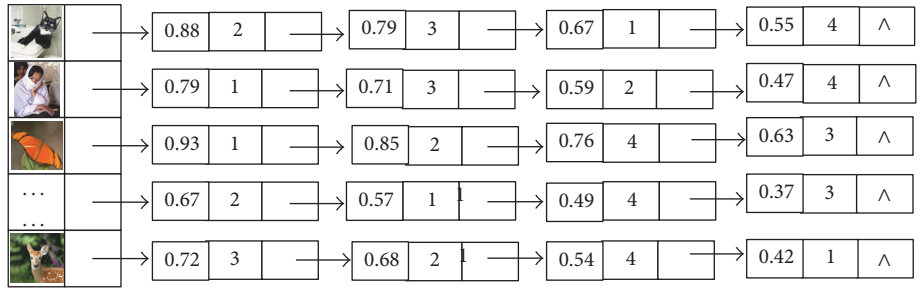


Figure 7: Images and their sort lists.

cow, dog, elk, fish, and whale; vehicles have super class and subclass relationship with boats, cars, plane, and train; person has super class and subclass relationship with police, military, and tattoo.

Although super class and subclass have great relevance in the class definition, many subclasses have a variety of differences in environment and appearance. So, for animal class, subclasses need exact matching to find neighbors; for vehicles class, subclasses need exact matching to find neighbors; because of particularity of class people, if there is no exact matching of subclass, matching can be performed at person level.

4.3.2. Appearance Based Neighbor Search. 256 dimensional histogram of RGB color space is used and χ^2 distance is computed.

4.4. Vote Based Saliency Maps Aggregation. Suppose the test image is I, the number of tag neighbors is k, and the number of appearance neighbors is k.

After tag based search in the training set, the detected neighbor number is y. If y is bigger than k, then k images are selected according to appearance similarities from y images. Finally, tag based neighbor set is given as

$$\operatorname{Im} g^T = \left\{\operatorname{Im} g_1^T, \operatorname{Im} g_2^T, \ldots, \operatorname{Im} g_i^T, \ldots, \operatorname{Im} g_x^T\right\}, \quad (4)$$

where x is the final number of neighbors; if $y \geq k$, then $x = k$; otherwise, $x = y$.

After appearance based similarity computation in the training set, k nearest neighbors are selected as

$$\operatorname{Im} g^A = \left\{\operatorname{Im} g_1^A, \operatorname{Im} g_2^A, \ldots, \operatorname{Im} g_i^A, \ldots, \operatorname{Im} g_k^A\right\}. \quad (5)$$

Merge sets (4) and (5) and get the set as

$$\operatorname{Im} g = \left\{\operatorname{Im} g_1, \operatorname{Im} g_2, \ldots, \operatorname{Im} g_x, \ldots, \operatorname{Im} g_{x+k}\right\}. \quad (6)$$

Each neighbor image has a sort list and contains the AUC values of all detection methods. The AUC values can vote for each detection method. Vote weights are summed as

$$\text{auc} = \left[\sum_{i=1}^{x+k} \text{auc}_i^1, \sum_{i=1}^{x+k} \text{auc}_i^2, \ldots, \sum_{i=1}^{x+k} \text{auc}_i^j, \ldots, \sum_{i=1}^{x+k} \text{auc}_i^M\right]. \quad (7)$$

In $\sum_{i=1}^{x+k} \text{auc}_i^j$, i is the ith neighbor and j is the jth detection method. M is the number of detection models.

The salient map set of image I is

$$S(I) = \left[S_1(I), S_2(I), \ldots, S_i(p), \ldots, S_M(I)\right], \quad (8)$$

where $S_j(I)$ is the saliency map of the jth detection method.

The fused saliency map can be computed as follows.

$$S_F(I) = S(I) \cdot \text{auc}^T. \quad (9)$$

5. Spatial Coherence Optimization

In saliency computations, the spatial relationship of adjacent regions is not considered, so it will result in noises on salient regions. In the field of image segmentation, the researchers use fully connected CRF (conditional random field) model [49] to achieve better segmentation results. Therefore, we use the fully connected CRF model to optimize the spatial coherence of saliency maps.

The objective function is defined as follows.

$$S(L) = -\sum_i \log P(l_i) + \sum_{i,j} \theta_{ij}(l_i, l_j), \quad (10)$$

where L is the binary variable being salient or not. $P(l_i)$ is the probability of pixel x_i being salient. Initially, $P(1) = S_i$, $P(0) = 1 - S_i$. S_i is the saliency of the pixel i.

$\theta_{i,j}$ is defined as follows.

$$\theta_{i,j} = u(l_i, l_j)\left[\omega_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_1^2} - \frac{\|I_i - I_j\|^2}{2\sigma_2^2}\right) + \omega_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_3^2}\right)\right]. \quad (11)$$

If $l_i \neq l_j$, then $u(l_i, l_j) = 1$, or else 0.

Both position information and color information are considered in $\theta_{i,j}$.

p_i is the position of pixel i and p_j is the position of pixel j.

I_i is the color of pixel i and I_j is the color of pixel j.

$\omega_1 \exp(-\|p_i - p_j\|^2/2\sigma_1^2 - \|I_i - I_j\|^2/2\sigma_2^2)$ suggests that adjacent pixels with similar colors should have similar saliency. σ_1 and σ_2 control color similarity and distance proximity.

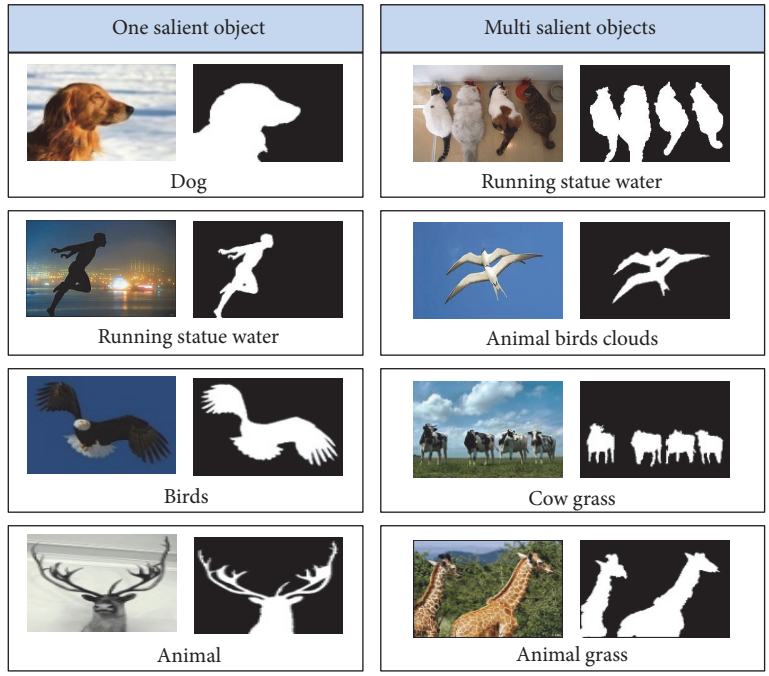


Figure 8: Images with one or multiple salient regions.

$\omega_2 \exp(-\|p_i - p_j\|^2/2\sigma_3^2)$ only considers position information. The purpose is to remove small areas.

6. Construction of Saliency Dataset of Social Images

The paper focuses on salient region detection of social images, so it is necessary to construct a new dataset of social images to promote further researches and evaluations of visual saliency models. The following will be discussed in detail.

6.1. Data Source. NUS-WIDE dataset [50] is a web image dataset constructed by NUS lab for media search. The images and the tags of this dataset are from Flickr which is a popular social web site. We randomly select 10000 images from NUS-WIDE dataset. The images come from thirty-eight folders of NUS-WIDE dataset, including carvings, castle, cat, cell phones, chairs, chrysanthemums, classroom, cliff, computers, cooling tower, coral, cordless cougar, courthouse, cow, coyote, dance dancing, deer, den, desert, detail, diver, dock, close-up, cloverleaf, cubs, doll, dog, dogs, fish, flag, eagle, elephant, elk, f-16, facade, and fawn.

6.2. Salient Region Annotation. Since the bounding boxes for salient regions are rough and can not reveal region boundaries, we adopt the pixel-wise annotation. In annotation process, nine subjects are asked to specify the attractive regions according to their first glance at the image.

To reduce label inconsistency of the annotation results, the pixel consistency score is computed. A pixel can be considered salient if 50% of subjects have selected it [23].

Finally, two subjects use Adobe Photoshop to segment salient regions.

6.3. Image Selection. First, 10000 images are randomly selected from NUS-wide dataset. Then, the images are further selected by the following criteria.

(1) The color contrast of any salient region and corresponding image is less than 0.7.

(2) Salient regions are rich in size. The proportion of salient regions to the corresponding image covers 10 grades, $[0, 0.1)$, $[0.1, 0.2)$, $[0.2, 0.3)$, $[0.3, 0.4)$, $[0.4, 0.5)$, $[0.5, 0.6)$, $[0.6, 0.7)$, $[0.7, 0.8)$, $[0.8, 0.9)$, $[0.9, 1]$.

(3) At least ten percent of the salient regions connected with the image boundaries.

After 5 rounds of selecting, the dataset contains 5429 images.

In the new dataset, the images have one or more salient regions; the positions of salient regions are not limited to image centers. The sizes of salient regions are varied. A great deal of images have complex/cluttered backgrounds. There are 78 tags which come from 81 tags of NUS-WIDE dataset. All these will bring challenges to salient region detection.

6.4. Typical Images of the New Dataset. In this section, typical examples of images, ground truth masks, and tags are listed below. Images can have one or multiple salient regions in Figure 8. The images may have cluttered and complex backgrounds in Figure 9. The sizes of salient regions are rich in Figure 10.

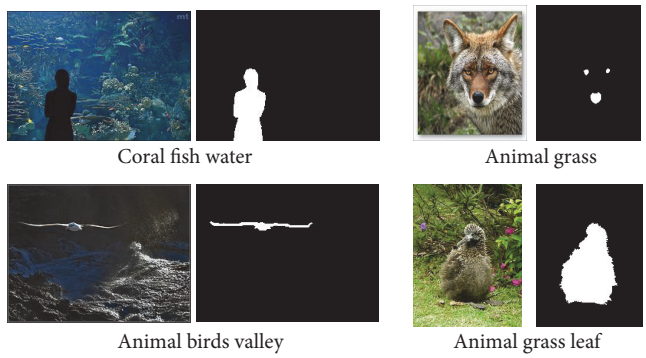


Figure 9: Images with cluttered and complex backgrounds.

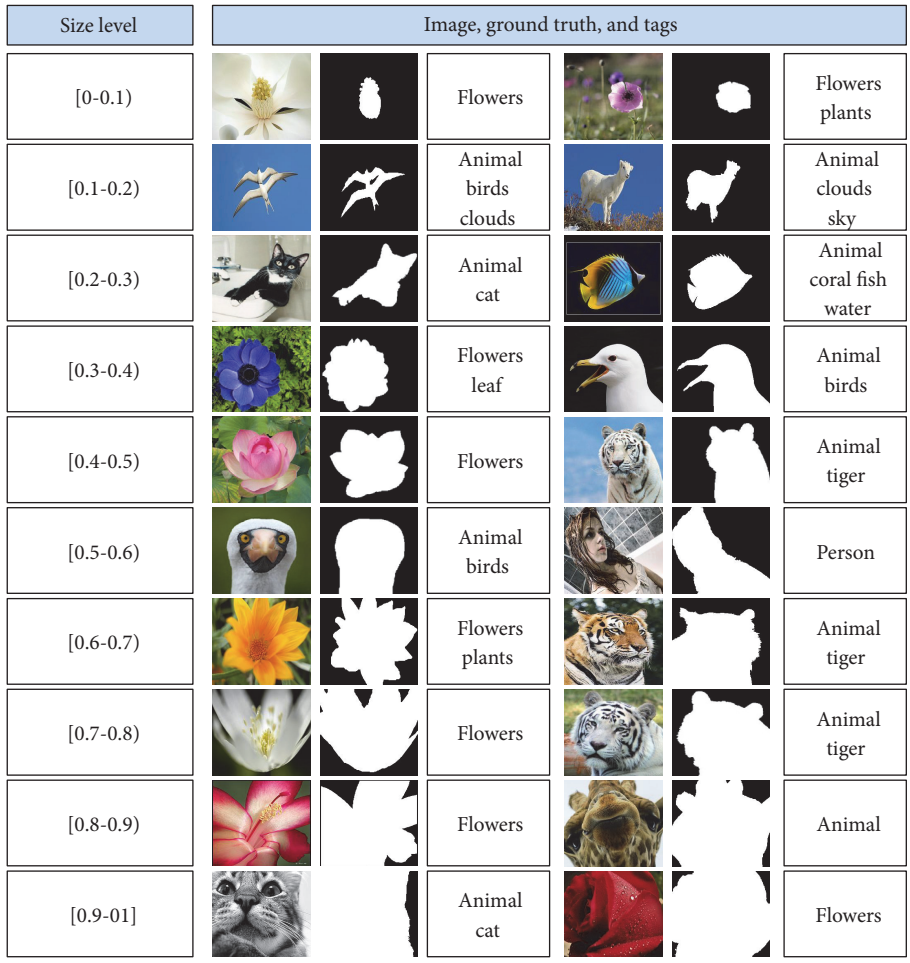


Figure 10: Images in various size levels.

7. Experiments

7.1. Experimental Setup

7.1.1. Experiments on the New Dataset. The aim of the paper is to solve salient region detection of social images. So the main experimental dataset is our new dataset, which is abbreviated as TBD (Tag Based Dataset).

We selected 20 object tags, including bear, birds, boats, buildings, cars, cat, computer, coral, cow, dog, elk, fish, flowers, fox, horses, person, plane, tiger, train, and zebra. Correspondingly, 20 RCNN object detectors were chosen to

extract RCNN features. Top 1000 proposals of each detector were used to compute RCNN features.

The proposed deep based detection method is abbreviated as DBS (Deep Based Saliency). DBS method was compared with 27 state-of-the-art methods in Section 7.2.1. 27 state-of-the-art methods are CB [34], FT [23], SEG [44], RC [14], SVO [17], LRR [39], SF [45], GS [37], CA [33], SS [47], HS [7], TD [48], MR [24], DRFI [25], PCA [41], HM [38], GC [36], MC [40], DSR [35], SBF [43], BD [42], SMD [46], BL [32], MCDL [9], MDF [8], LEGS [10], and RFCN [11]. These methods not only are very popular but also cover many types.

In addition, we also verify the performance of the aggregation method in Section 7.2.2.

7.1.2. Experiments on State-of-the-Art Datasets. We also carried out the experiments on six state-of-the-art datasets to validate our method. These datasets are MSRA1000 [23], DUT-OMRON [24], ECSSD [7], HKU-IS [8], PASCAL-S [51], and SOD [27]. In these datasets, SOD [27] is a dataset which is from segmentation field; others are from saliency field. Because these datasets have no image level tags, we extract objectness feature [19] of these datasets. Objectness is a kind of high-level semantic cues, so objectness cue is similar to tag feature. Compared with the method DBS, the method using objectness feature instead of tag feature is abbreviated as OBS (Objectness Based Saliency).

OBS method was compared with 11 state-of-the-art methods, including FT [23], RC [14], SF [45], HS [7], MR [24], DRFI [25], GC [36], MC [40], BD [42], MDF [8], and LEGS [10].

7.1.3. Evaluation Criteria. We adopted popular performance evaluations to quantitatively evaluate the results, including PR (Precision Recall) curves, ROC (Receiver Operating Characteristic) curves, F-measure value, AUC (Area under ROC Curve) value, and MAE (Mean Absolute Error) value, respectively.

7.2. Experiments on the New Dataset TBD

7.2.1. Experiments of Deep Learning Based Detection Method. DBS is compared with 27 state-of-the-art methods. The results are given in Table 1 and Figure 11.

Among the 28 methods in Table 1, the top four methods are all deep learning based methods, including MCDL [9], RFCN [11], MDF [8], and DBS. To some extent, deep learning based detection methods are better than handcrafted feature based methods, in terms of both completeness and accuracy of saliency maps. AUC value of DBS method is the highest. F-measure value of DBS method is slightly lower than RFCN [11]. MAE value of DBS is third low. The overall performance of DBS method is good.

Typical saliency maps are shown in Figure 11.

7.2.2. Experiments of Aggregation Method. The handcrafted feature based detection methods used as complementarities to DBS are DRFI [25], SMD [46], BL [32], and MC [40].

Table 1: F-measure, AUC, and MAE of DBS and 27 state-of-the-art methods.

	F-measure	AUC	MAE
CB	0.5472	0.7971	0.2662
SEG	0.4917	0.7588	0.3592
SVO	0.3498	0.8361	0.409
SF	0.3659	0.7541	0.2077
CA	0.5161	0.8287	0.2778
TD	0.5432	0.8081	0.2333
SS	0.2516	0.6714	0.2499
HS	0.5576	0.7883	0.2747
DRFI	0.5897	0.8623	0.2063
HM	0.4892	0.7945	0.2263
BD	0.5443	0.8185	0.1955
BL	0.5823	0.8562	0.266
MR	0.5084	0.7753	0.229
PCA	0.5392	0.8439	0.2778
FT	0.3559	0.6126	0.2808
RC	0.5307	0.8105	0.3128
LRR	0.5124	0.7956	0.3067
GS	0.5164	0.8136	0.2056
SMD	0.6033	0.8437	0.1976
GC	0.5063	0.7511	0.2596
DSR	0.5035	0.8139	0.2105
MC	0.574	0.8427	0.2287
SBF	0.493	0.848	0.2325
MCDL	0.6559	0.8813	0.1457
LEGS	0.6124	0.8193	0.1844
RFCN	0.6768	0.8803	0.1476
MDF	0.6574	0.8483	0.1556
DBS	0.6621	0.8917	0.1505

In neighbor searching, the number of tag neighbors is 4 and the number of appearance neighbors is 4.

In order to verify the effect of neighbors, appearance neighbor based method and tag neighbor based method are carried out, respectively. Appearance neighbor based aggregation method is abbreviated as ABS (Appearance Based Saliency). Tag neighbor based aggregation method is abbreviated as TBS (Tag Based Saliency). Tag neighbor and appearance neighbor based aggregation method is abbreviated as FBS (Fusion Based Saliency).

The detection performances of DBS, ABS, TBS, and FBS are compared in Table 2.

The performance of TBS is better than the performance of ABS. The reasons are as follows. ABS method is based on appearance feature based neighbor search. Appearance similar images cannot guarantee similar saliency maps. However, TBS method uses object information. The same or similar

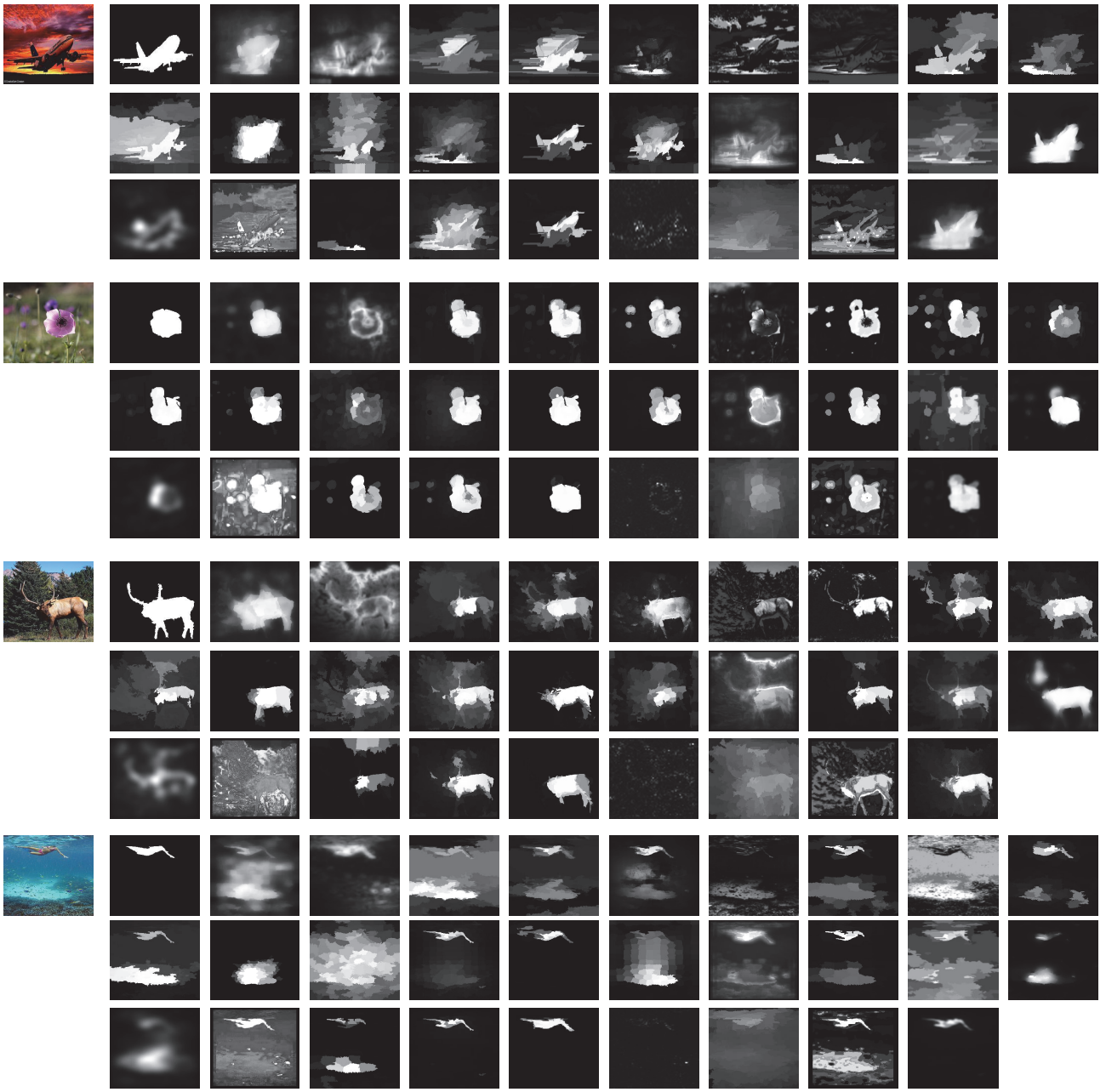


Figure 11: Visual comparisons of DBS with 27 state-of-the-art methods. The order of images are original image, ground truth mask, BL [32], CA [33], CB [34], DRFI [25], DSR [35], FT [23], GC [36], GS [37], HM [38], HS [7], LEGS [10], LRR [39], MC [40], MCDL [9], MR [24], PCA [41], BD [42], RC [14], RFCN [11], SBF [43], SEG [44], SF [45], SMD [46], MDF [8], SS [47], SVO [17], TD [48], and DBS.

objects can ensure similar salient regions to some extent. So the performance of TBS is better.

PR and ROC curves are shown in Figures 12 and 13. PR and ROC curves of FBS are higher than 27 state-of-the-art methods.

The examples of typical saliency maps of FBS method and DBS method are shown in Figure 14. It can be seen that the aggregation results are more complete and the details are better.

7.3. Experiments on State-of-the-Art Datasets. The experiment results are given in Table 3. We can see that AUC values of OBS are the highest on all datasets, F-measure values of OBS are the highest on all datasets, and MAE values are the lowest or the second lowest. The performance of OBS is the best. However, the improvements of OBS are not so obvious because objectness feature is not the accurate tag feature. Thus we believe that the results will be improved obviously if we use accurate tag annotation of images.

Table 2: F-measure, AUC, and MAE of DBS, ABS, TBS, and FBS.

	DBS	ABS	TBS	FBS
F-measure	0.6621	0.6652	0.6688	0.6712
AUC	0.8917	0.9061	0.9113	0.9166
MAE	0.1505	0.1497	0.1474	0.1452

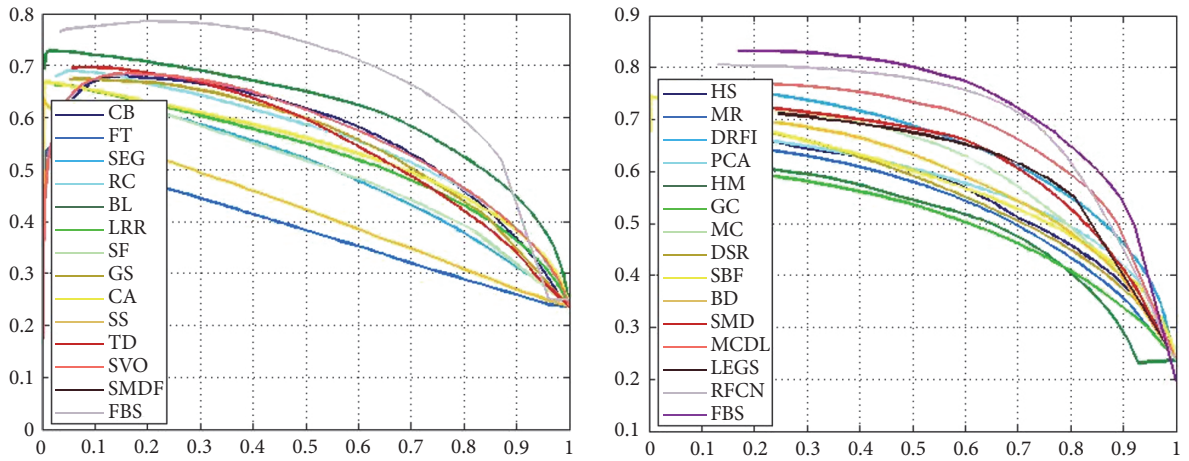


Figure 12: PR curves of FBS and 27 state-of-the-art methods.

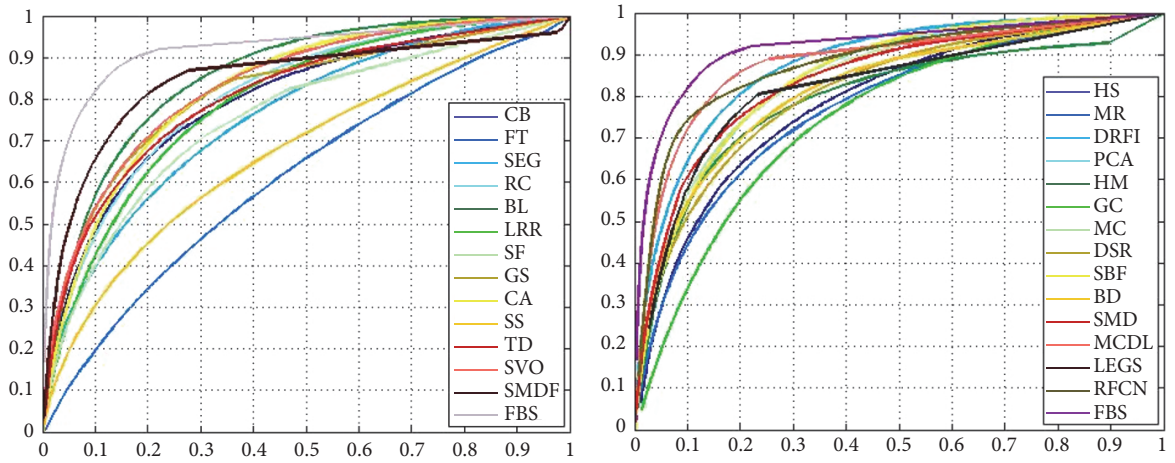


Figure 13: ROC curves of FBS and 27 state-of-the-art methods.

Experiments on state-of-the-art datasets validate the effectiveness of our proposed method DBS.

8. Conclusions

The paper focuses on salient region detection of social images. First, the proposed deep learning based salient region detection method considers both appearance features and tag features. Tag features are detected by RCNN models. Second, tag neighbor features and appearance neighbor features are added to the saliency aggregation model. Finally, a new database of challenging social images and pixel-wise saliency annotations is constructed, which can promote further researches and evaluations of visual saliency model.

Conflicts of Interest

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

Figure 14: Visual comparisons of FBS with DBS. The order of images is original image, ground truth mask, FBS, and DBS.

Table 3: F-measure, AUC, and MAE of OBS and 11 state-of-the-art methods on six state-of-the-art datasets.

Metric	AUC	F-measure	MAE
Dataset		MSRA1000	
FT	0.766	0.579	0.241
DRFI	0.966	0.845	0.112
RC	0.937	0.817	0.138
GC	0.863	0.719	0.159
HS	0.93	0.813	0.161
MC	0.975	0.894	0.054
MR	0.941	0.824	0.127
SF	0.886	0.7	0.166
BD	0.948	0.82	0.11
MDF	0.978	0.888	0.066
LEGS	0.958	0.87	0.081
OBS	0.984	0.893	0.061
Dataset		HKU-IS	
FT	0.71	0.477	0.244
DRFI	0.95	0.776	0.167
RC	0.903	0.726	0.165
GC	0.777	0.588	0.211
HS	0.884	0.71	0.213
MC	0.928	0.798	0.102
MR	0.87	0.714	0.174
SF	0.828	0.59	0.173
BD	0.91	0.726	0.14
MDF	0.971	0.869	0.072
LEGS	0.907	0.77	0.118
OBS	0.976	0.871	0.078
Dataset		PASCAL-S	
FT	0.627	0.413	0.309
DRFI	0.899	0.69	0.21
RC	0.84	0.644	0.227
GC	0.727	0.539	0.266
HS	0.838	0.641	0.264
MC	0.907	0.74	0.145
MR	0.852	0.661	0.223
SF	0.746	0.493	0.24
BD	0.866	0.655	0.201
MDF	0.921	0.771	0.146
LEGS	0.891	0.752	0.157
OBS	0.927	0.778	0.141
Dataset		ECSSD	
FT	0.663	0.43	0.289
DRFI	0.943	0.782	0.17
RC	0.893	0.738	0.186
GC	0.767	0.597	0.233
HS	0.885	0.727	0.228
MC	0.948	0.837	0.1
MR	0.888	0.736	0.189
SF	0.793	0.548	0.219
BD	0.896	0.716	0.171
MDF	0.957	0.847	0.106
LEGS	0.925	0.827	0.118
OBS	0.968	0.856	0.112
Dataset		DUT-OMRON	
FT	0.682	0.381	0.25
DRFI	0.931	0.664	0.15
RC	0.859	0.599	0.189
GC	0.757	0.495	0.218
HS	0.86	0.616	0.227
MC	0.929	0.703	0.088
MR	0.853	0.61	0.187
SF	0.81	0.495	0.147
BD	0.894	0.63	0.144
MDF	0.935	0.728	0.088
LEGS	0.885	0.669	0.133
OBS	0.943	0.731	0.091
Dataset		SOD	
FT	0.607	0.441	0.323
DRFI	0.89	0.699	0.223
RC	0.828	0.657	0.242
GC	0.692	0.526	0.284
HS	0.817	0.646	0.283
MC	0.868	0.727	0.179
MR	0.812	0.636	0.259
SF	0.714	0.516	0.267
BD	0.827	0.653	0.229
MDF	0.899	0.793	0.157
LEGS	0.836	0.732	0.195
OBS	0.907	0.801	0.163

Acknowledgments

This work was supported in part by the Program Project of Beijing Municipal Education Commission (KM201511417008), the National Natural Science Foundation of China (Grant no. 62372148), the National Natural Science Foundation of China (Grant no. 61272352), and Beijing Natural Science Foundation (4152016).

References

[1] S. Pare, A. Kumar, V. Bajaj, and G. Singh, "An efficient method for multilevel color image thresholding using cuckoo search algorithm based on minimum cross entropy," *Applied Soft Computing*, vol. 61, pp. 570–592, 2017.

[2] S. Pare, A. Bhandari, A. Kumar, and G. Singh, "An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix," *Expert Systems with Applications*, vol. 87, pp. 335–362, 2017.

[3] P. J. McParlane, Y. Moshfeghi, and J. M. Jose, "Collections for automatic image annotation and photo tag recommendation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8325, no. 1, pp. 133–145, 2014.

[4] W. Zhang, A. Borji, Z. Wang, P. Le Callet, and H. Liu, "The application of visual saliency models in objective image quality assessment: a statistical evaluation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1266–1278, 2016.

[5] 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.

[6] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[7] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 1155–1162, IEEE, Portland, Ore, USA, June 2013.

[8] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '15*, pp. 5455–5463, 2015.

[9] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 1265–1274, IEEE, Massachusetts, Mass, USA, June 2015.

[10] L. Wang, H. Lu, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '15*, pp. 3183–3192, IEEE, Massachusetts, Mass, USA, June 2015.

[11] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9908, pp. 825–841, 2016.

[12] H. Li, J. Chen, H. Lu, and Z. Chi, "CNN for saliency detection with low-level feature integration," *Neurocomputing*, vol. 226, pp. 212–220, 2017.

[13] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2106–2113, IEEE, Kyoto, Japan, October 2009.

[14] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[15] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 438–445, USA, June 2012.

[16] G. Zhu, Q. Wang, and Y. Yuan, "Tag-Saliency: combining bottom-up and top-down information for saliency detection," *Computer Vision and Image Understanding*, vol. 118, pp. 40–49, 2014.

[17] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 914–921, Barcelona, Spain, November 2011.

[18] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[19] Y. Jia and M. Han, "Category-independent object-level saliency detection," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV '13*, pp. 1761–1768, IEEE, Sydney, Australia, December 2013.

[20] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: uniqueness, focusness and objectness," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 1976–1983, IEEE, Sydney, Australia, December 2013.

[21] W. Wang, C. Lang, and S. Feng, "Contextualizing tag ranking and saliency detection for social images," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7733, no. 2, pp. 428–435, 2013.

[22] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: a data driven approach," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1131–1138, IEEE, Oregon, Ore, USA, June 2013.

[23] R. Achantay, S. Hemamiz, F. Estraday, and S. Süsstrunky, "Frequency-tuned salient region detection," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1597–1604, June 2009.

[24] C. Yang, L. H. Zhang, H. C. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3166–3173, 2013.

[25] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: a discriminative regional feature integration approach," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2013.

[26] T. Liu, Z. Yuan, J. Sun et al., "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[27] 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.

[28] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 315–327, 2012.

[29] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "iCoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '10*, pp. 3169–3176, IEEE, California, Calif, USA, June 2010.

[30] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, Orlando, Fla, USA, November 2014.

[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.

[32] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '15*, pp. 1884–1892, IEEE, Massachusetts, Mass, USA, June 2015.

[33] S. Goferman, L. Manor, and A. Tal, "Context-aware saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1915–1926, 2010.

[34] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior," in *Proceedings of the British Machine Vision Conference*, pp. 1–12, BMVA Press, 2011.

[35] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2976–2983, Sydney, Australia, December 2013.

[36] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 1529–1536, Sydney, Australia, December 2013.

[37] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proceedings of the 12th European conference on Computer Vision (ECCV '12)*, pp. 29–42, Florence, Italy, October 2012.

[38] X. Li, Y. Li, C. Shen, A. Dick, and A. V. D. Hengel, "Contextual hypergraph modeling for salient object detection," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV '13*, pp. 3328–3335, December 2013.

[39] X. H. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix Recovery," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 853–860, 2012.

[40] 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.

[41] 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, Oregon, Ore, USA, 2013.

[42] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 2814–2821, June 2014.

[43] 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.

[44] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Computer Vision—ECCV 2010*, vol. 6315 of *Lecture Notes in Computer Science*, pp. 366–379, Springer, Berlin, Germany, 2010.

[45] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: contrast based filtering for salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 733–740, June 2012.

[46] H. Peng, B. Li, R. Ji, W. Hu, W. Xiong, and C. Lang, "Salient object detection via Low-rank and Structured sparse Matrix Decomposition," in *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI '13*, pp. 796–802, July 2013.

[47] 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.

[48] C. Scharfenberger, A. Wong, K. Fergani, J. S. Zelek, and D. Clausi, "Statistical textural distinctiveness for salient region detection in natural images," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 979–986, June 2013.

[49] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with Gaussian edge potentials," *Advances in Neural Information Processing Systems*, pp. 109–117, 2011.

[50] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: a real-world web image database from national university of singapore," in *Proceedings of ACM International Conference on Image and Video Retrieval, CIVR '09*, 2009.

[51] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14*, pp. 280–287, IEEE, Massachusetts, Mass, USA, June 2014.

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 8179579, 22 pages
https://doi.org/10.1155/2017/8179579

Research Article

The Café Wall Illusion: Local and Global Perception from Multiple Scales to Multiscale

Nasim Nematzadeh[1,2] and David M. W. Powers[1]

[1]*College of Science and Engineering, Flinders University, Adelaide, SA, Australia*
[2]*Department of Science and Engineering, Faculty of Mechatronics, Karaj Branch, Islamic Azad University (KIAU), Karaj-Alborz, Iran*

Correspondence should be addressed to Nasim Nematzadeh; nasim.nematzadeh@flinders.edu.au

Received 29 June 2017; Revised 9 September 2017; Accepted 20 September 2017; Published 13 December 2017

Academic Editor: Mourad Zaied

Geometrical illusions are a subclass of optical illusions in which the geometrical characteristics of patterns in particular orientations and angles are distorted and misperceived as a result of low-to-high-level retinal/cortical processing. Modelling the detection of tilt in these illusions, and its strength, is a challenging task and leads to the development of techniques that explain important features of human perception. We present here a predictive and quantitative approach for modelling foveal and peripheral vision for the induced tilt in the Café Wall illusion, in which parallel mortar lines between shifted rows of black and white tiles appear to converge and diverge. Difference of Gaussians is used to define a bioderived filtering model for the responses of retinal simple cells to the stimulus, while an analytical processing pipeline is developed to quantify the angle of tilt in the model and develop confidence intervals around them. Several sampling sizes and aspect ratios are explored to model variant foveal views, and a variety of pattern configurations are tested to model variant Gestalt views. The analysis of our model across this range of test configurations presents a precisely quantified comparison contrasting local tilt detection in the foveal sample sets with pattern-wide Gestalt tilt.

1. Introduction

Visual processing starts within the retina from the photoreceptors passing the visual signal through bipolar cells to the Retinal Ganglion Cells (RGCs) whose axons carry the encoded signal to the cortex for further processing. The intervening layers incorporate several types of cell with large dendritic arbors, divided into horizontal cells that control for different illumination conditions and feedback to the receptor and bipolar cells and amacrine cells that feed into the center-surround organization of the Retinal Ganglion Cells. High-resolution receptors in the foveal area have a direct 1 : 1 pathways from photoreceptors, via bipolar cells to ganglion cells [1].

It is commonly believed that the center-surround organization in RGCs and their responses are the results of the lateral inhibitory effect in the outer and the inner retina [2] in which the activated cells inhibit the activation of nearby cells. At the first synaptic level, the lateral inhibition [2–4] enhances the synaptic signal of photoreceptors, which is specified as a retinal point spread function (PSF) seen as a biological convolution with the edge enhancement property [3]. At the second synaptic level, the lateral inhibition mediates the more complex properties such as the responses of directional selective receptive fields (RFs) [2].

The complexity of interneural circuitries and activation and responses of the retinal cells have been investigated [5, 6] in a search for the specific encoding role of each individual cell in the retinal processing leading to new insights. This includes the existence of a diverse range of Retinal Ganglion Cells (RGCs) in which the size of each individual type varies in relation to the eccentricity of neurons and the distance from the fovea [5] supporting our biological understanding of the retinal multiscale encoding [7], completed in the cortex [5, 6, 8]. ON and OFF cells of each specific type are noted to have a variant size [5, 9] as well. It is also reported that there are different channels for passing the encoded information of ON-center and OFF-surround (and vice versa) activation of retinal RFs [6] to the cortex. Moreover, the possibility of simultaneous activation of a group of RGCs (as a combined

activity) in the retina by the output of amacrine cells is noted in the literature [10–12]. Some retinal cells have been found with a directional selectivity property such as the cortical cells [5, 6]. It is noteworthy that, despite the complexity and variety of retinal cells circuitry and coding, there are a few constancy factors common to them, valid even for amacrine and horizontal cells. The constancy of integrated sensitivity is one of these factors mentioned in the literature [13–15] which is quite useful for quantitative models for visual system.

The perception of directional tilt in the Café Wall illusion might tend to direct explanations toward the cortical orientation detectors or complex cells [8, 16]. We have shown that the emergence of tilt in the Café Wall illusion specifically [17–21], and in tile illusions generally [17, 22], is a result of simple cells processing with circularly symmetric activation/inhibitions. Low-level filtering models [23, 24] commonly apply a filter similar to a Gaussian or Laplacian of a specific size on the Café Wall to show the appearance of slanted line segments referred to as Twisted Cord [25] elements in the convolved output. These local tilts are assumed then to be integrated into continuous contours of alternating converging and diverging mortar lines at a more global level [22–24]. A hybrid retinocortical explanation as a midlevel approach containing light spread, compressive nonlinearity, and center-surround transformation has been proposed by Westheimer [26]. Some other explanations rely on Irradiation Hypothesis [27] and Brightness Induction [28]. There are also high-level descriptive approaches such as "Border Locking" [29] and "Phenomenal Model" [30] for the illusion with little consideration to the underlying neurological mechanisms involved in the emergence of tilt in the Café Wall illusion.

Modelling the receptive field responses dates back to Kuffler's demonstration of roughly concentric excitatory center and inhibitory surround [31]. Then, Rodieck and Stone [32] and Enroth-Cugell and Robson [33] modelled the center and surround signals of the photoreceptors by two concentric Gaussians with different diameters. The computational modelling of early visual processing was followed by Marr and Ullman [34] who were inspired by Hubel and Wiesel's [8] discovery of cortical simple and complex cells. Laplacian of Gaussian (LoG) has been proposed by Marr and Hildreth [35] as an optimal operator for low-level retinal filtering and an approximation filter of Difference of Gaussians (DoG) instead of LoG, considering a ratio of ~1.6 for the Gaussians diameters.

The model here [17–22] is a most primitive implementation for the contrast sensitivity of RGCs based on classical circular center and surround organization of the retinal RFs [32, 33]. The output of the model is a simulated result for the responses of the retinal/cortical simple cells to stimuli/image. This image representation is referred to as an "edge map" utilizing Difference of Gaussians (DoG) at multiple scales to implement the center-surround activity as well as the multiscale property of the RGCs. Our explanation differs from the previous low-level models [23, 24, 27, 36] due to the concept of filtering at multiple scales in our model in which the scales are tuned to the resolutions of image features, not the resolutions of the individual retinal cells. We show also that our model is a quantitative approach capable of even predicting the strength of the Café Wall illusion based on different characteristics of the pattern [21].

This work is a complete collection of our findings on the underlying mechanism involved in our foveal and peripheral vision for modelling the perception of the induced tilt in the Café Wall illusion. It draws together and extends our previous studies on the foveal/local investigations of tilt on Café Wall illusion [18, 19] and extends our investigations for the peripheral/global analysis of the perceived tilt not in just one specific sample (to overcome the shortcomings of our previous studies [18, 19]), but for variations of different configurations modelling the Gestalt perception of tilt in the illusion.

In Section 2, we describe the characteristics of a simple classical model for simulating the responses of simple cells based on Difference of Gaussians (DoG) and utilize the model for explaining the Café Wall illusion qualitatively (Section 2.2.1) and quantitatively (Section 2.2.2). Afterwards, in Section 3, the experimental results on variations of foveal sample sets are provided (Section 3.1), followed by the report of quantitative tilt results for variations of different configurations of the Café Wall illusion with the same characteristics of mortar lines and tiles but with different arrangements of a whole pattern (Section 3.2), which had then been completed by a thorough comparison of the local and global mean tilts of the pattern found by our simulations (Section 3.3). We conclude by highlighting the advantages and disadvantages of the model in predicting the local and global tilt in the Café Wall pattern and proceed to outline a roadmap of our future work (Section 4).

2. Materials and Methods

2.1. Formal Description and Parameters. Applying a Gaussian filter on an image generates a blurred version of the image. In our DoG model, the difference of two Gaussian convolutions of an image generates one scale of the *edge map* representation. For a 2D signal such as image I, the DoG output, modelling the retinal ganglion cell responses with the center-surround organization, is given by

$$\begin{aligned}\mathrm{DoG}_{\sigma,s\sigma}(x,y) &= I \times \frac{1}{2\pi\sigma^2 \exp\left[-\left(x^2+y^2\right)/\left(2\sigma^2\right)\right]} - I \\ &\times \frac{1}{2\pi\left(s\sigma\right)^2 \exp\left[-\left(x^2+y^2\right)/\left(2s^2\sigma^2\right)\right]},\end{aligned} \tag{1}$$

where DoG is the convolved filter output, x and y are the horizontal and vertical distances from the origin, respectively, and σ is the scale of the center Gaussian ($\sigma = \sigma_c$). $s\sigma$ in (1) indicates the scale of the surround Gaussian ($\sigma_s = s\sigma_c$), and s is referred to as *Surround ratio* in our model as shown in

$$s = \frac{\sigma_{\text{surround}}}{\sigma_{\text{center}}} = \frac{\sigma_s}{\sigma_c}. \tag{2}$$

Increasing the value of s results in a wider suppression effect from the surround region, although the height of the

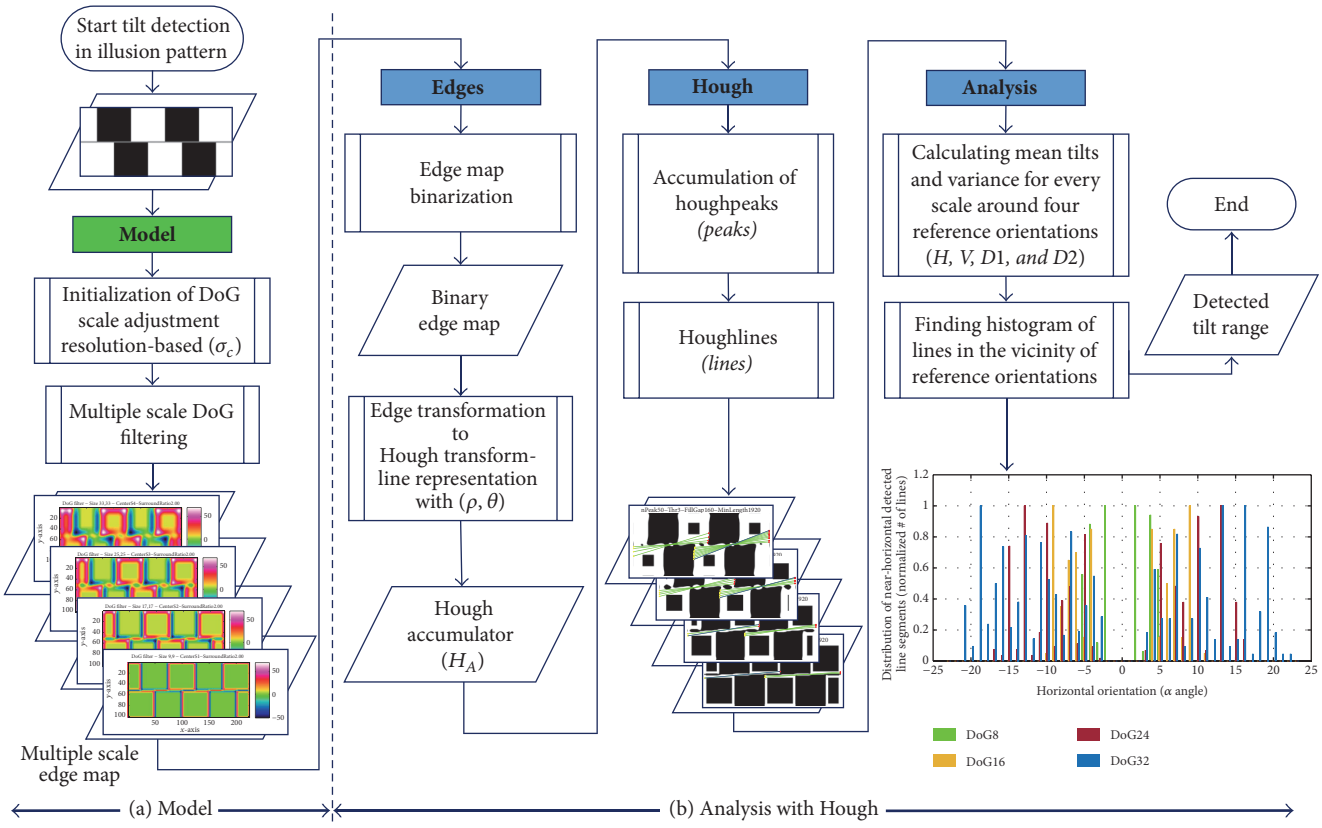


Figure 1: Flowchart of our model with Hough analytic processing pipeline (adapted from [19, 20]).

surround Gaussian declines (normalized Gaussians are used in our model). A broader range of *Surround ratios* from 1.4 to 8.0 have been tested with little difference to our results. We have considered another parameter in the model for the filter size referred to as *Window ratio* (h). To generate edge maps we have applied DoG filters within a window in which the values of both Gaussians are insignificant outside the window. The window size is determined based on the parameter h that determines how much of each Gaussian (center and surround) is included inside the DoG filter and the scale of the center Gaussian (σ_c) such that

$$\text{WindowSize} = h \times \sigma_c + 1. \quad (3)$$

+1 as given in (3) guaranties a symmetric DoG filter. In the experimental results the *Window ratio* (h) has been set to 8 to capture more than 95% of the surround Gaussians in the DoG convolved outputs.

2.2. Model and Image Processing Pipeline. An image processing pipeline has been used [18–21] here to extract edges and their angles of tilt (in the edge maps), as shown in Figure 1 for a crop section of a Café Wall pattern of size 2 × 4.5 tiles (the precise height is 2 tiles + mortar = $2T + M$). In this research, we concentrate on the analysis of the induced tilt in the Café Wall illusion, to include the details of the parameters used in the simulations in order to quantify the tilt angle in this stimulus by modelling our foveal and peripheral vision.

2.2.1. DoG Edge Map at Multiple Scales. The DoG representation at multiple scales is the output of the model, which is referred to as an *edge map* of an image. The DoG is highly sensitive to spots and moderately sensitive to lines that match the center diameter. We have used this representation for modelling the responses of visual simple cells especially on tile illusions in our investigations [18–20, 22]. An appropriate range for σ_c can be determined for any arbitrary pattern/image considering the pattern characteristics as well as the filter size matched with the image features (by applying (3)) in our model. The step sizes determine the accuracy of the multiple scale representation here and again are pattern specific for preserving the visual information with minimum redundancy but at multiple scales.

For Café Wall illusion, the DoG edge map indicates the emergence of divergence and convergence of the mortar lines in the pattern, similar to how it is perceived as shown in Figure 2. The edge map has been shown at six different scales in jetwhite color map [37] for a Café Wall of 3 × 8 tiles with 200 × 200 px tiles (T) and 8 px mortar (M). In order to extract the tilted line segments along the mortar lines referred to as Twisted Cord [25] elements, the DoG filter should be of the same size as the mortar size [18, 24, 36]. The edge map should contain both high frequency details as well as low frequency contents in the image. We start DoG filtering below the mortar size at scale 4 ($\sigma_c = 4$; as the finest scale) and extend the scales gradually until scale 28 for a large filter to capture the tiles fully, with incremental steps of 4 (in the figure we

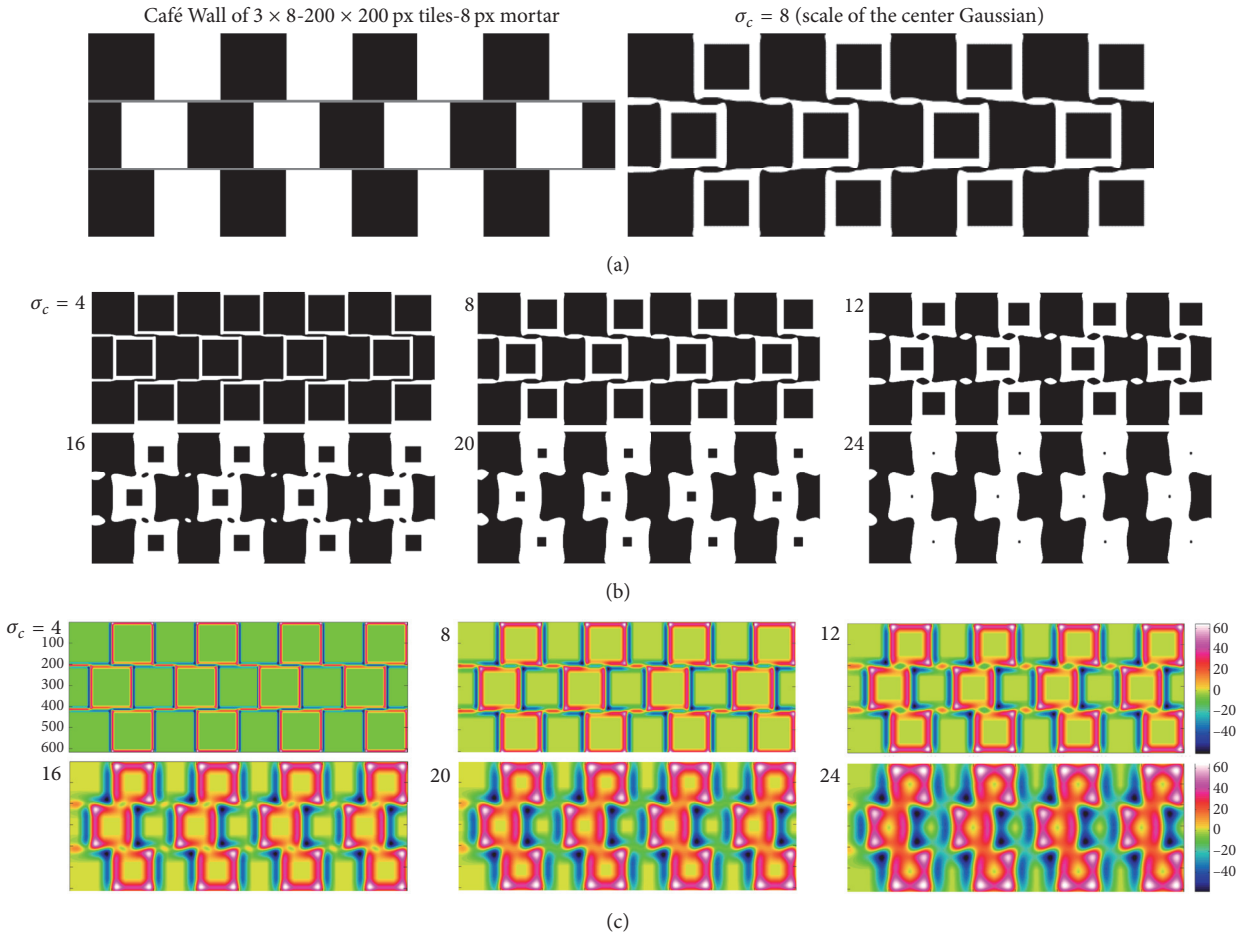


Figure 2: (a) Café Wall of 3 × 8 with 200 × 200 px tiles and 8 px mortar (left) and an enlarged DoG output at scale 8 (σ_c = 8) from the edge map of the pattern (right). (b) The binary edge map at six scales (σ_c = 4 to 24 with incremental steps of 4). (c) The same edge map but presented in the jetwhite color map [37]. The noncritical parameters of the DoG model are s = 2 and h = 8 (the *Surround* and *Window ratios,* resp.).

have shown this till σ_c = 24 due to shortage of space). Other noncritical parameters of the model are s = 2 and h = 8, representing the *Surround* and *Window ratios,* respectively.

The DoG outputs in Figure 2 show that the tilt cues appear at fine to medium scales and start to disappear as the scale of the center Gaussian increases in the model. At fine to medium scales, there are some corner effects that appear in the edge maps which highlight the emergence of tilted line segments and result in the appearance of square tiles that look similar to trapezoids. This may be referred to as wedges in the literature [29], inducing convergent and divergent mortar lines. So, at fine scales around the size of the mortar, we see the groupings of identically colored tiles with the Twisted Cord elements along the mortar lines. By increasing the scales gradually from the medium to coarse scales, when the mortar cues disappear completely in the edge map, other groupings of identically colored tiles are emerged in the edge map, connecting tiles in zigzag vertical orientation. What we see across multiple scales in the edge map of the pattern are two incompatible groupings of pattern elements: groupings of tiles in two consecutive rows by the mortar lines at fine scales with nearly horizontal orientation (as focal/local view) and then groupings of tiles in zigzag vertical direction at coarse scales (as peripheral/global view). These two incompatible groupings occur simultaneously across multiple scales and exhibit systematic differences according to the size of the Gaussian and predicts the change in illusion effects with distance from the focal point in the pattern.

We have shown that, in the edge map at multiple scales, not only do we extract the information of edges/textures with the shades and shadows around the edges, but we are also able to show the emergence of other cues related to tilt and perceptual grouping as features for mid-to-high-level processing [17, 22]. Also we have shown in another article that even the prediction of the strength of tilt effect in different variations of Café Wall illusion is possible from the persistence of mortar cues across multiple scales [21] in the edge map. Highly persistent mortar cue in the edge map is an indication for a stronger induced tilt in the stimulus.

2.2.2. Second-Stage Processing. The Hough analysis is used for quantitative measurement of tilt in our model and consists of three stages of Edges, Hough, and Analysis as shown in Figure 1, explained below.

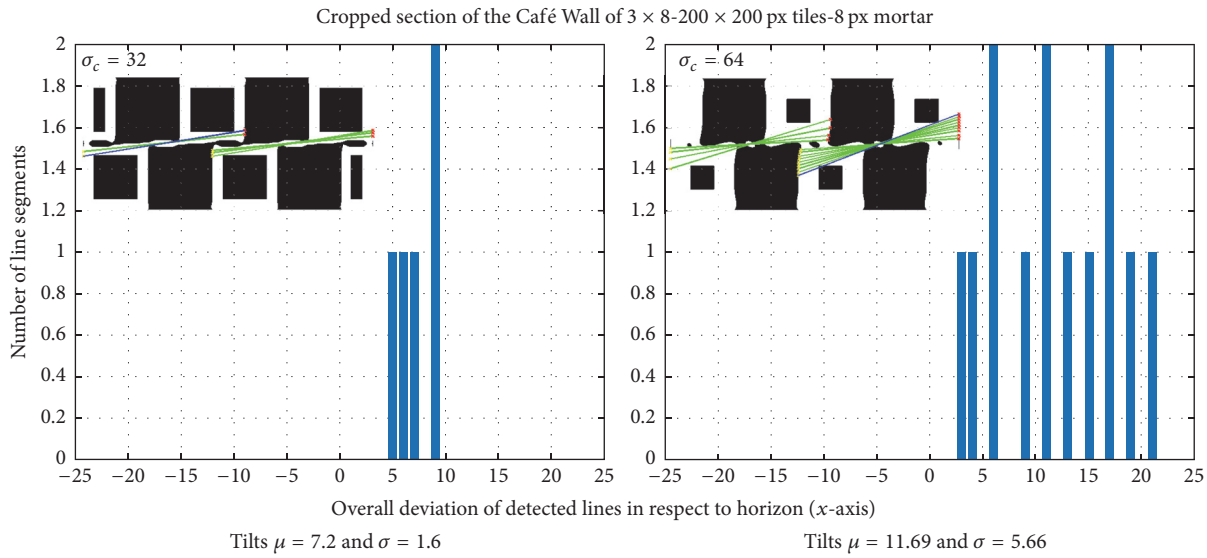


FIGURE 3: *Hough* stage output. Distribution of line segments detected near the horizontal orientation, presented for two scales of the DoG edge map at scales 32 and 64 (σ_c = 32, 64). Mean tilt and variance for each graph have been also provided.

Edges. We used here an analysis pipeline to characterize the tilted line segments presented in the edge map of the Café Wall pattern. First, the edge map is binarized and then the Standard Hough Transform (SHT) [38, 39] is applied to it to detect line segments inside the binary edge map at multiple scales. SHT uses a two-dimensional array called the accumulator (H_A) to store line information of edges based on the quantized values of ρ and θ in a pair (ρ, θ) using Hough function in MATLAB. ρ specifies the distance between the line passing through the edge point and θ is the counterclockwise angle between the normal vector (ρ) and the x-axis ranges from 0 to π, $[0, \pi)$. Therefore, every edge pixel (x, y) in the image space corresponds to a sinusoidal curve in Hough space such that $\rho = x \cdot \cos\theta + y \cdot \sin\theta$, with θ as free parameter corresponding to the angle of the lines passing through the point (x, y) in the image space. The output of *Edges* is the accumulator matrix (H_A) with all the edge pixel information.

Hough. The *Edges* stage provides all possible lines that could pass through every edge point of the edge map inside the H_A matrix. We are more interested in detecting the induced tilt lines inside the Café Wall image. Two MATLAB functions called *houghpeaks* and *houghlines* are employed for the further processing of the accumulator matrix (H_A). The *houghpeaks* function finds the peaks in the H_A matrix with three parameters of *NumPeaks* (maximum number of line segments to be detected), *Threshold* (threshold value for searching the peaks in the H_A), and *NHoodSize* (the size of the suppression neighbourhood that is set to zero after the peak is identified). The *houghlines* function extracts line segments associated with a particular bin in the accumulator matrix (H_A) with two parameters of *FillGap* (the distance between two line segments associated with the same Hough bin; line segments with shorter gaps are merged into a single line segment) and *MinLength* (specifies keeping or discarding the merged lines; lines shorter than this value are discarded).

A sample output of *Hough* processing stage is given in Figure 1 with the detected houghlines displayed in green on the binary edge map at four different scales (the cropped section is selected from a Café Wall with 50 × 50 px tiles and 2 px mortar, with the DoG scales from 0.5M to 2M in the figure around the mortar size with the incremental steps of 0.5M). The results of *Hough* analysis stage for a different cropped section of a Café Wall pattern with higher resolution (cropped from a Café Wall with 800 × 800 px tiles and 32 px mortar) are shown in Figure 3 for two scales of the DoG edge map (σ_c = 32, 64-M and 2M; Blue lines indicate the longest line segments detected). The histograms of detected houghlines near the horizontal orientation have been provided for these scales. The absolute mean tilts and the standard deviation of tilts are calculated and presented in the figure below the graphs.

Analysis. The detected line segments and their angular positions are saved inside four orientation matrices considering the closest to any of the reference orientations of horizontal (H), vertical (V), positive diagonal (+45°, $D1$), and negative diagonal (−45°, $D2$) orientations. We consider an interval of [−22.5°, 22.5°) around each reference orientation to cover the whole space. The statistical analysis of tilt angles of the detected lines around each reference orientation is the output of this stage and includes the mean tilts and the standard errors around the means for each scale of the DoG edge map.

Hough Parameters for Tilt Investigations of Café Wall Stimulus. Recall that *NumPeaks* indicates the maximum number of line segments to be detected and *FillGap* shows the distance between two line segments associated with the same Hough bin in which line segments with shorter gaps are merged

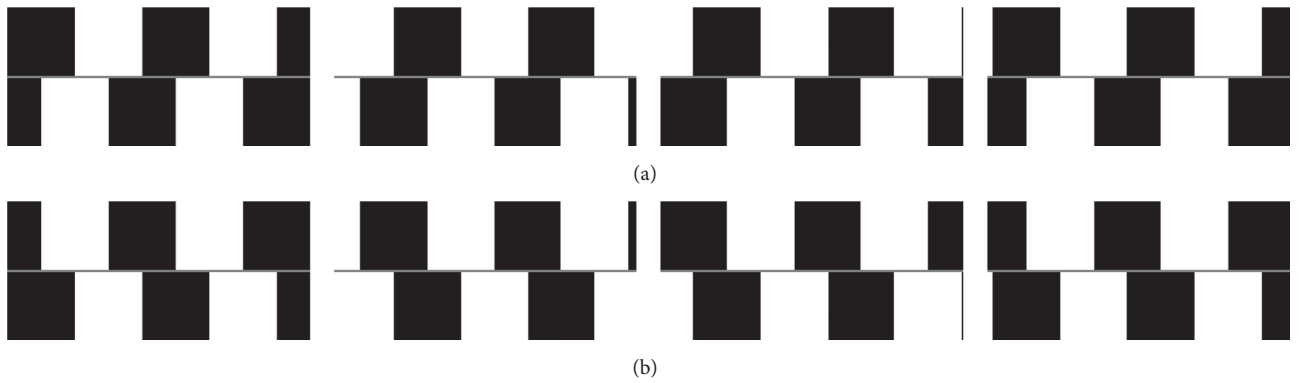


Figure 4: Examples of systematically cropped samples along Falling (a) and Rising mortar lines (b), selected from the Café Wall of 3 × 9 tiles with 400 × 400 px tiles (T) and 16 px mortar (M), Café Wall 3 × 9 T400-M16. In total, 50 samples are taken along each mortar line with an offset of 32 px between samples in each step. Cropped samples have a size of $(2T + M) \times 4.5T$ with the mortar line positioned in the middle.

into a single line segment. The other Hough parameter is *MinLength*, which specifies keeping or discarding of the line segments considering this minimum length and discarding the lines shorter than this value. To select an appropriate value for these parameters we should consider pattern features and the scales of the DoG edge map. In the Café Wall pattern, in order to detect the Twisted Cord elements at fine scales, the *MinLength* value should be in a reliable range. The Twisted Cord elements have a minimum length of $2.5T$ (*MinLength* $\approx 2.5T$), and therefore, for a Café Wall with 200 × 200 px tiles, *MinLength* = 500 px. We set this parameter a bit smaller than this value equal to *MinLength* $\approx 2.25T = 450$ px for our experiments in Section 3. The *FillGap* parameter is chosen equal to 1/5th of a tile size ($1/5T$) in our experiments (to merge the disconnected mortar cues of each Twisted Cord elements at fine to medium scales in the edge maps). *NumPeaks* is selected appropriately based on the size of the pattern, and, for small foveal sets (Section 3.1), this is set to 100 but has a higher range, 520 and 1000 for larger Café Wall stimulus for global investigation of tilt (Section 3.2).

3. Results and Discussions

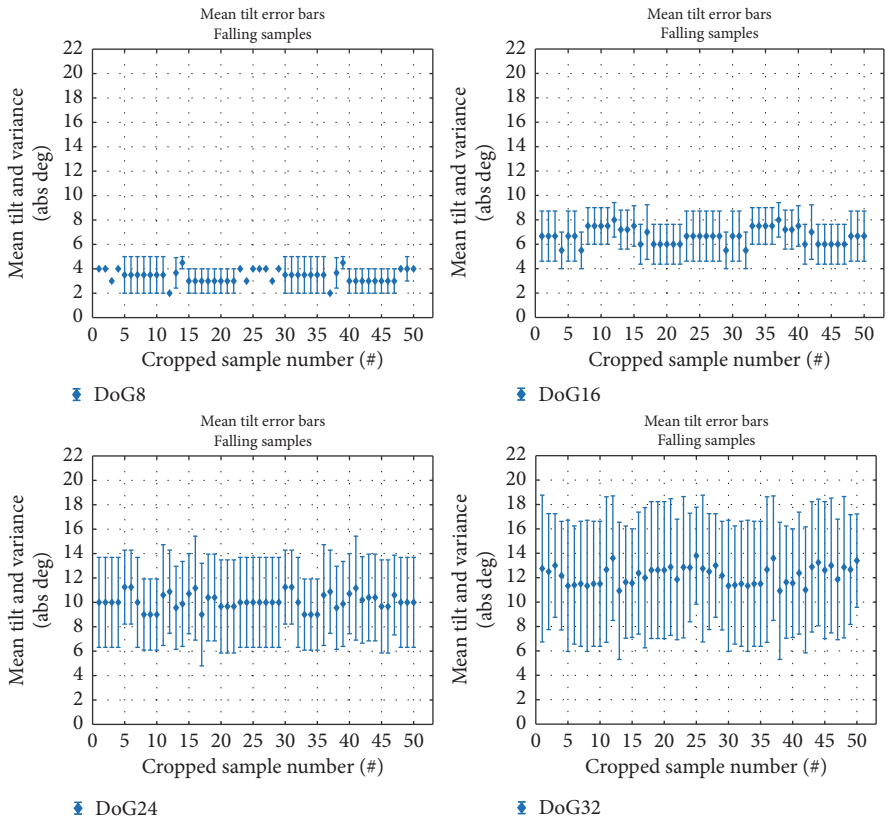
3.1. Local Tilt Investigation

3.1.1. Falling and Rising Mortar Investigation. This work draws together and extends our previous studies on the foveal/local investigations of tilt on Café Wall illusion [18, 19], and the extension of our investigations for the peripheral/global analysis of the perceived tilt not in just one specific sample but for variations of different configurations. The quantitative mean tilts of similar shape samples but with variant resolutions have been investigated in our previous work [20]. We have shown that, for variations with different resolutions, the tilt prediction of the model stays nearly the same when the dependent parameters of the model to the spatial content of the pattern have been updated accordingly in each resolution (σ_c and Hough parameters).

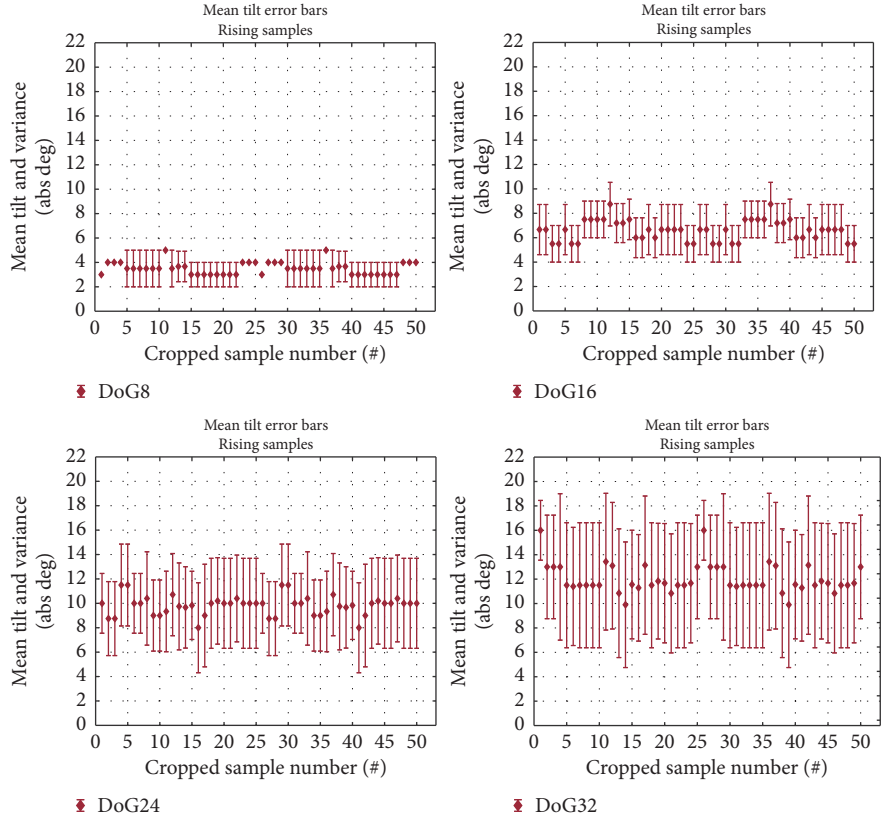
We report here the evaluation results of our model's predictions for the *direction of detected tilts* for two types of mortar lines in the Café Wall illusion [20]. Instead of referring to the mortar lines as either convergent or divergent, we rather talk about Falling or Rising mortar lines, in which, in the Falling mortar, the direction of induced tilt is downwards on its right side compared to the horizontal direction and for the Rising mortar the vice versa. For instance, in Figure 2(a)-(left) the top mortar line is Falling while the bottom one is Rising. In this experiment the cropped samples specifically selected in such a way to contain only one mortar line indicate the emergence of tilt in only one direction of either positive or negative in the DoG edge maps (Falling or Rising). The samples have a height of two tiles and the mortar line in between ($2T + M$) and the width of 4.5 tiles ($4.5T$; T: tile size, M: mortar size), with the same height above and below the mortar line. In Section 3.1.2, we show the results for samples of variant sizes and different cropping technique for a more general investigation of the foveal/local perception of the induced tilt in the pattern.

To fix parameters not being investigated, we restrict consideration initially to the Café Wall of 3 × 9 tiles with 400 × 400 px tiles and 16 px mortar. Here, in a systematic approach, 50 samples were selected from the Falling mortar and 50 samples from the Rising mortar with the dimensions described above from the Café Wall of 3 × 9 tiles. The sampling process starts from the leftmost side of the pattern and with a horizontal shift size/offset of 32 pixels between the samples for the cropping window. A few examples of the Falling and Rising samples have been provided in Figure 4. The cropped samples at the bottom of the figure are symmetrical crops from the Rising mortar lines selected from the stimulus (Café Wall of 3 × 8). In the DoG edge maps of these samples, the scale of the center Gaussian is in the range of $1/2M$ to $2M$ with the incremental steps of $1/2M = 8$ px (σ_c = 8, 16, 24, and 32) to detect both mortar lines and the outlines of the tiles for detecting near-horizontal tilts in the edge maps.

For individual samples of the Falling and Rising mortar, the near-horizontal mean tilts and variance of the detected houghlines have been shown in Figure 5. As the scale of the center Gaussian (σ_c) in our model increases, the variance of tilt also increases. The mean tilt results of the Falling and



(a)



(b)

FIGURE 5: Mean tilts and variance error bars for individual samples of Falling (a) and Rising mortar lines (b) specified along horizontal axis (100 samples in total; DoG8 means $\sigma_c = 8$). As explained in Figure 4, the cropped samples are from the Café Wall of 3×9 tiles with 400×400 px tiles and 16 px mortar (DoG4: $\sigma_c = 4$).

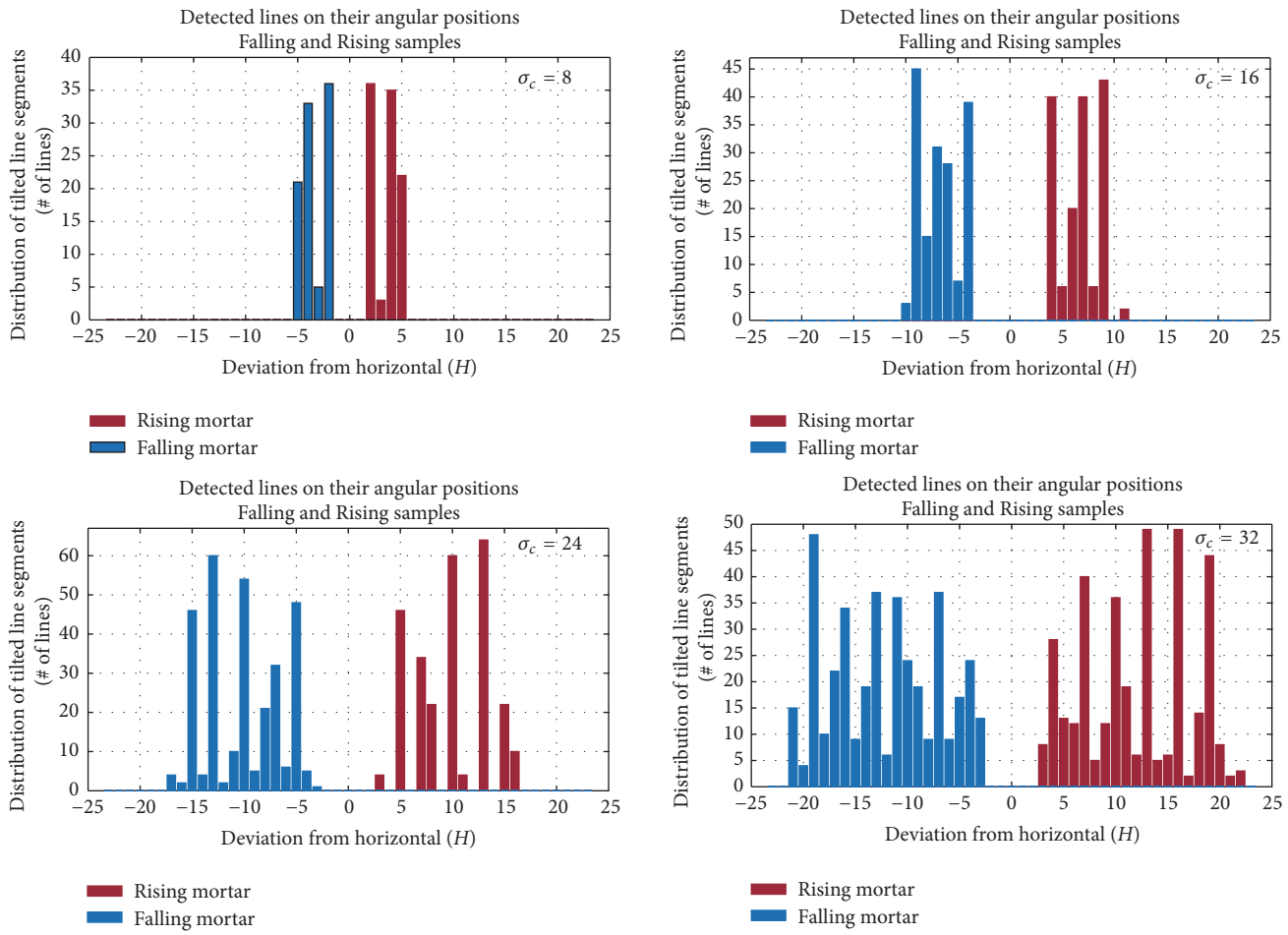


FIGURE 6: Distribution of line segments detected by deviation in degrees from the horizontal (x-axis), for the total 100 samples of Falling (Blue bars) and Rising mortar lines (Red bars), as explained in Figure 4. The scales of the edge maps (σ_c) range from 8 to 32 with step sizes of 8 ($1/2M$ to $2M$ with incremental steps of $1/2M = 8$ px). Other parameters of the model and Hough analysis are as follows: $s = 2$ and $h = 8$ (the *Surround* and *Window ratios*), with *NumPeaks* = 50, *Threshold* = 3, *FillGap* = 80, and *MinLength* = 960 as the Hough parameters (see [20] for the resolution analysis and its effect on the Hough parameters).

Rising mortar samples in Figure 5 indicate that both types of mortar lines follow nearly the same pattern.

The near-horizontal line segments detected in the DoG edge maps (houghlines) at four scales (σ_c = 8, 16, 24, and 32) are shown in Figure 6 for the Falling (Blue bars) and the Rising (Red bars) mortar samples. These graphs are summarized in Figure 7 in a single graph, indicating normalized distribution of line segments detected with their deviations in degrees from the horizontal (x-axis) orientation for the 100 samples. When the DoG scale increases, the detected tilt range covers a wider neighbourhood area around the horizontal axis as their details given in Figure 6. The deviations of the detected lines from the horizontal orientation in Figures 6 and 7 are very small at the finest scale ($\sigma_c = 8$). The range of tilt angles increases along the following scales of the edge maps reaching a wide range of variations at scale 32 ($\sigma_c = 32$ or DoG32) that is not reflected in our subjective experience of tilt in the pattern (it is overestimated at this scale). In the literature [18, 19, 22, 24, 36] it is noted that the size of DoG filter should be close to the size of the mortar for the Twisted Cord elements to appear along the mortar lines, here DoG16 ($\sigma_c = 16$). We have demonstrated that this is not applicable for Café Wall patterns with very thick mortar lines [21]. In this case, the mortar cues are lost completely in the edge map even by applying DoG filters smaller than the mortar size. The strength of the illusion is highly dependent on the characteristics of the pattern such as the luminance of the mortar, the mortar size, the contrast of the tiles, the aspect ratio of the tile size to the mortar size, and other parameters of the stimulus. We noted that there exists a correlation between the strength of the illusion with the persistence of the mortar cues in the edge maps across multiple scales [21].

3.1.2. Variant Sampling Sizes: Two Methods of Sampling. As explained in Section 2.2.2, an analytical processing pipeline is used to quantify the tilt angles in the DoG edge maps. For modelling variant foveal views, several sampling sizes and aspect ratios have been investigated across multiple scales in order to find the confidence intervals around the predicted tilts reported in our previous work [18, 19]. These variations

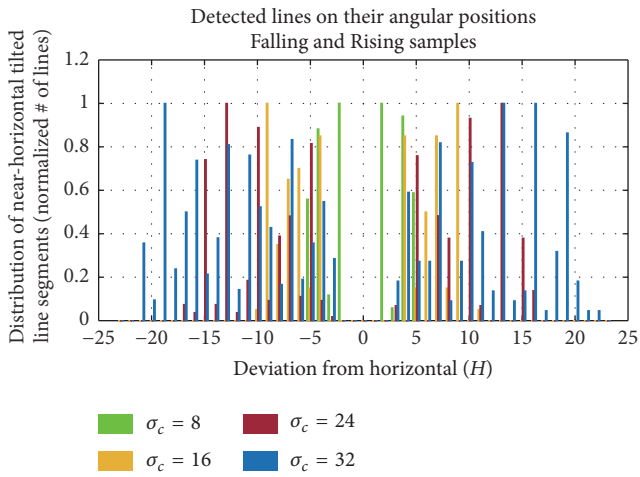


FIGURE 7: The normalized graph of lines detected from the DoG edge maps of the 100 cropped samples by deviation in degrees from the horizontal (H) orientation. Samples are from the Falling and Rising mortar lines from the Café Wall 3 × 9-T400-M16. The characteristics of the samples are provided in Figure 4, and the edge maps have four different scales (σ_c = 8, 16, 24, and 32, based on [20]).

are verified and quantified in simulations using two different sampling approaches. The contrast of local tilt detection with a global average across the whole Café Wall pattern will be discussed in Section 3.2.

The eyes process the visual scene at different scales at different times, due to the intentional and unintentional eye movements while we look at the scene (pattern). Notably overt saccades and gaze shifts result in a rapid scan of the field of view by the fovea for the pertinent high-resolution information. Our visual perception of tilt in the Café Wall is affected by our *fixation* on the pattern. The induced tilts weakened in a region around fixation point, but the peripheral tilts stay unaffected with stronger tilts. It seems that, in the Café Wall illusion, the final induced tilts get greater effect from the peripheral tilt recognition compared to the foveal/local tilt perception. The possible correlations that might exist between the tilt effect to our foveal/peripheral view of the pattern due to gaze shifts and saccades are further investigated here. The local "cropped" samples simulate foveal-sized locus only, but different scales of the DoG edge maps represent different degrees of eccentricity (the distance from the fovea) in the periphery.

In this section we are reporting the experimental results from [18, 19] and we restrict consideration initially to Café Wall of 9 × 14 tiles with 200 × 200 px tiles and 8 px mortar (Figure 8(a)), with three "foveal" crop sizes to be explored, Crop4 × 5 (Cropped section of a 4 × 5 tiles), Crop5 × 5, and Crop5 × 6 (Figure 8(b)). Although the size of foveal image can be estimated by factors such as specific image size, viewing distance, or human subject in mind (which usually are considered in psychophysical experiments), the sample sizes explored in our experiments for the simulation results are selected for convenience without considering those restrictive factors.

Two sampling methods have been applied: *Systematic* and *Random Cropping*. In the *"Systematic Cropping"* [18] for each specified crop window size, 50 samples are taken from the Café Wall of 9 × 14 tiles, in which the top left corner for the first sample is selected randomly from the pattern and, for the rest of the samples, the cropping window shifts horizontally to the right with an offset of 4 pixels in each step. The total shift is equal to a tile size (200 px) at the end, so there is no repeated versions of any samples. In the *"Random Cropping"* [19] approach, for each specified crop window size, 50 samples are taken from randomly selected locations with the only consideration of the crop borders to stay inside the pattern.

The range of DoG scales of these samples is from 0.5M to 3.5M with the incremental steps of 0.5M, and coarser scales exceeding the tile size (T; using (3)), resulting in a very distorted edge pattern. We calculated [19] not only the near horizontal mean tilts but also the vertical and diagonal tilts at medium to coarse scales in this experiment. Unlike the previous experiment in Section 3.1.1, detecting only horizontal tilts, we extended the range of scales in the model from 2M in the previous experiment to 3.5M for these experiments. With DoG filters larger than the mortar size, ultimately reaching the coarsest size selected (3.5M, using (3)), then the tiles are fully captured at the coarsest scales of the edge maps. These are being used for the vertical and diagonal deviations and the groupings of tiles in zigzag vertical orientation in our investigations. The Hough parameters (of both houghpeaks and houghlines functions) should have a proper range to detect the near-horizontal slanted line segments in the edge maps at fine scales (refer to Section 2.2.2). For example, *FillGap* should assign a value to fill small gaps between the line segments that appeared in the edge map at fine scales to detect near-horizontal tilted lines, and *MinLength* should be larger than an individual tile size (T) to avoid the detection of the outlines of the tiles in the calculations. The values of Hough parameters depend on the pattern's attributes (features) and they are selected empirically for the tilt investigation in the experiments. To attain reliable and comparable tilt results, a constant set of these parameters have been used in this experiment and in Section 3.2 for the global tilt investigation, which is as follows: *NumPeaks* = 100, *Threshold* = 3, *FillGap* = 40, and *MinLength* = 450. Other values for *NumPeaks* have been also tested for the global tilt investigation in Section 3.2 (520 and 1000).

Figure 9 shows a binary DoG edge map at seven different scales (a), the edge map presented in the jetwhite color map (b), and the detected houghlines displayed in green on the edge map (c) for a sample of Crop4 × 5 tiles, selected from the Café Wall of 9 × 14 tiles (Figure 8).

The absolute mean tilts in box plot have been graphed for the detected lines in the DoG edge maps at seven different scales for each sample set and the two sampling methods and around the four reference orientations of horizontal (H), vertical (V), and diagonals ($D1$, $D2$) orientations in Figures 10(a) and 10(b). As the figure indicates, the *"Random Cropping"* approach produces more stable tilt results across variant foveal sample sizes compared to the *"Systematic"* sampling method. We noted [19] that the *Systematic* sampling

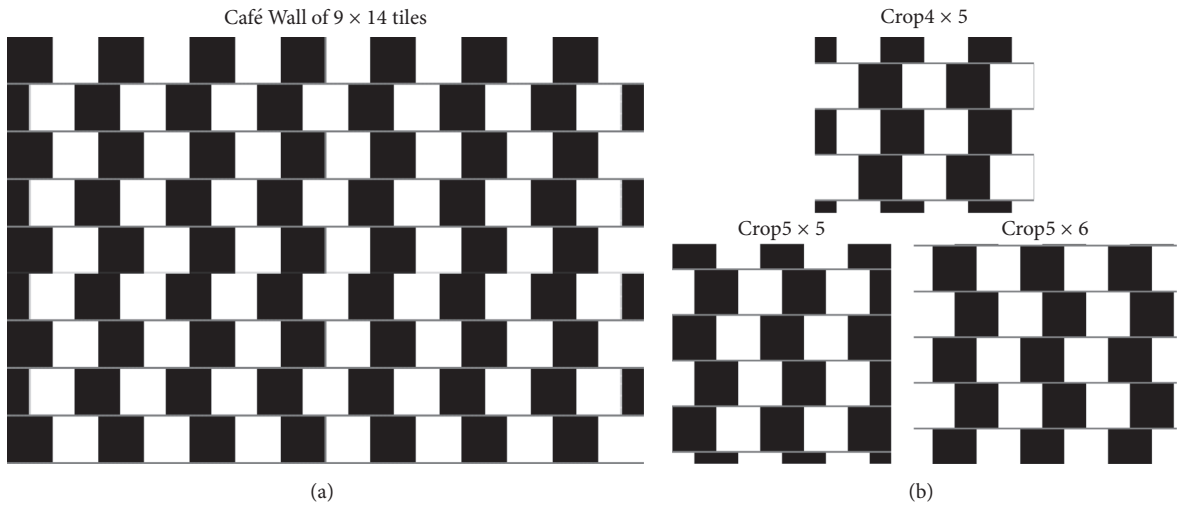


FIGURE 8: Café Wall of 9 × 14 tiles with 200 × 200 px tiles and 8 px mortar (a) and three "foveal" sample sizes explored (based on [18, 19]) (Crop$H \times W$ is $H \times W$ tiles).

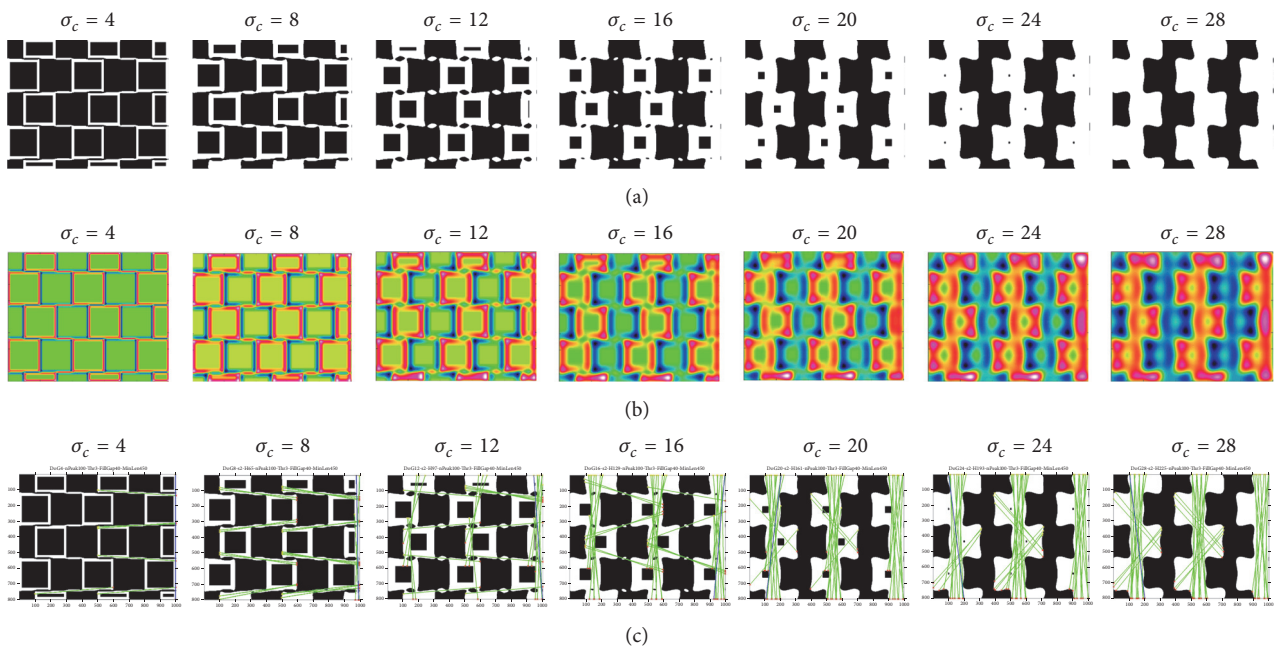
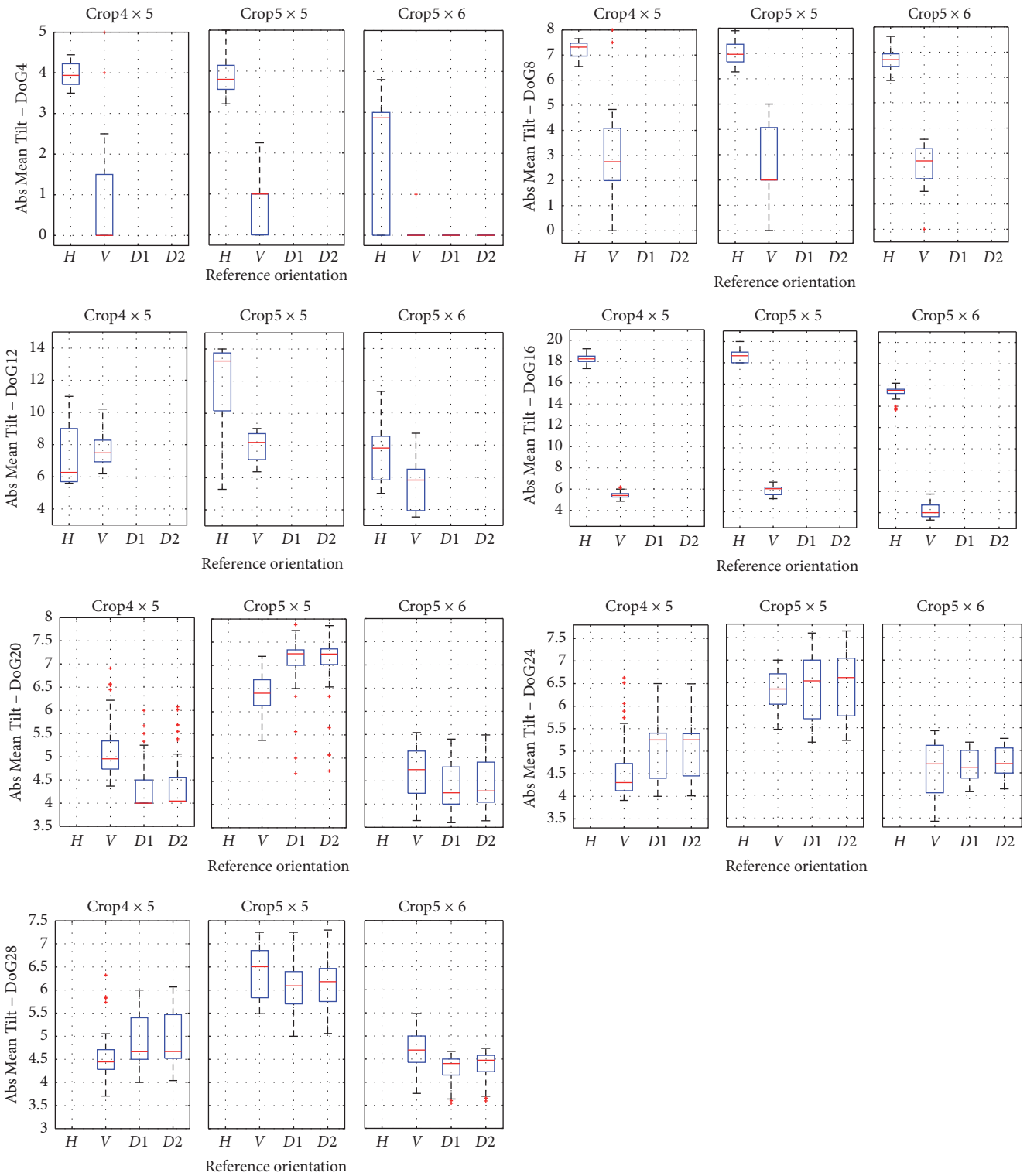


FIGURE 9: (a) A binary DoG edge map at seven scales (σ_c = 4, 8, 12, 16, 20, 24, and 28) of a cropped section of size 4 × 5 tiles from a Café Wall with 200 × 200 px tiles and 8 px mortar. (b) The DoG edge map displayed in the jetwhite color map [37]. (c) Detected houghlines from the edge map displayed in green, overlaid on the binary edge map with Hough parameters as: *NumPeaks* = 100, *Threshold* = 3, *FillGap* = 40, and *MinLength* = 450 (based on [19]).

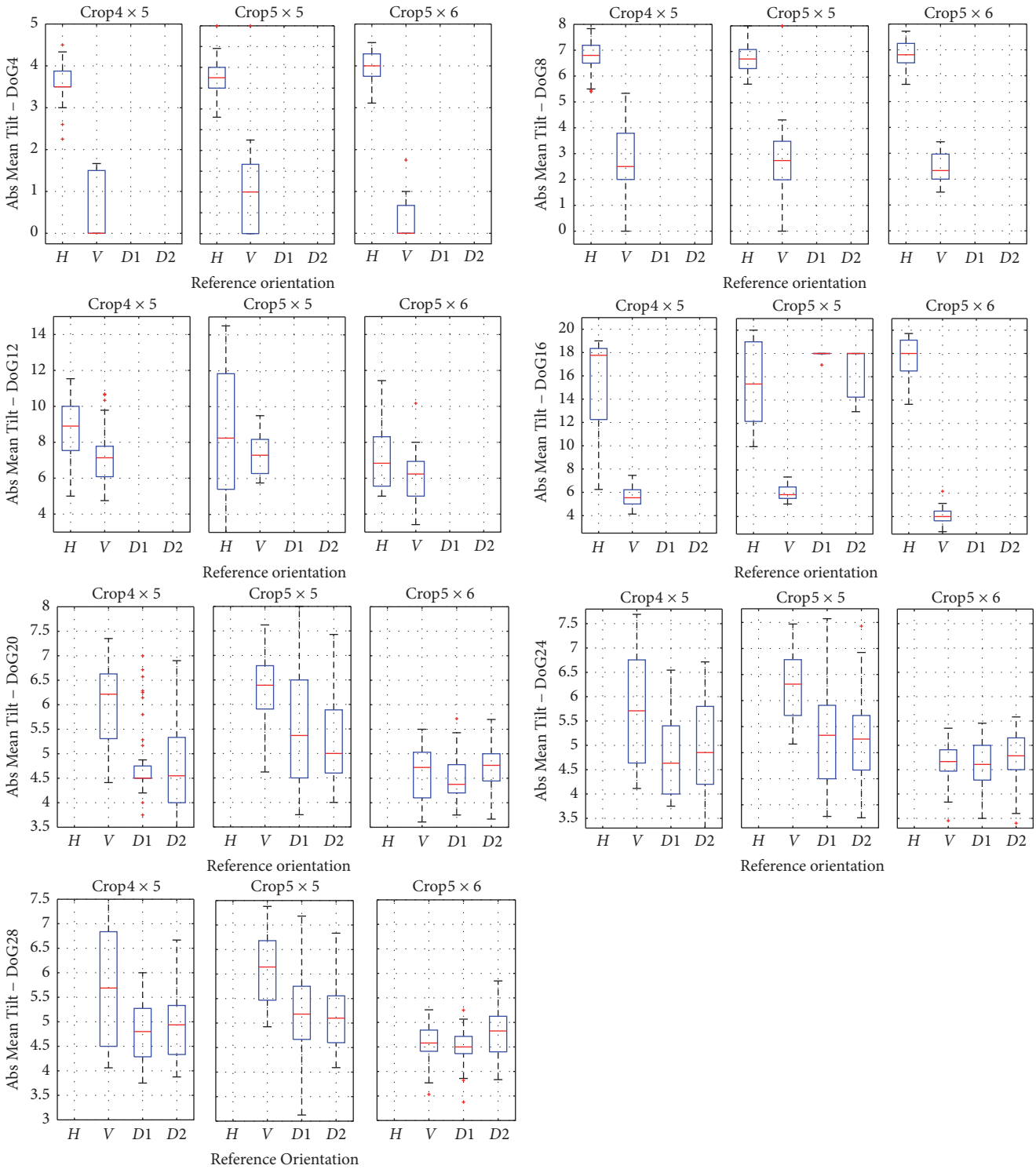
approach is closer to the bias of our saccades and gaze shifts toward interest points, but the *Random* sampling is a more standard statistical approach. At fine to medium scales of both sampling methods, there are only horizontal and vertical lines detected. A few samples of Crop5 × 5 have $D2$ components at scale 16 due to the border effects (only 4 out of 50 samples). The results for near-horizontal mean tilts at scale 8 (σ_c = 8) show a nearly stable range around 7° in all samples (the DoG filter size apparently correlates with the mortar size). As the scale increases from 20 onwards, there are no near-horizontal lines detected, but more vertical and diagonal lines are extracted from the edge maps. This is because the mortar cues in the edge maps at these scales start to fade, and also the enlargement of the outlines of the tiles results in more lines detected around the vertical and diagonal orientations at coarse scales of the DoGs. By increasing the scale, the horizontal mean tilt also increases and at scales 8 and 12 it is nearly 8°; however, at the finest scale (σ_c = 4) the horizontal mean tilt is quite small (~3.5°). When we fixate on the pattern, we encounter a weaker tilt



Reference orientations:
(*H*) Horizontal, (*V*) vertical
(*D*1) Positive diagonal (+45°)
(*D*2) Negative diagonal (−45°)
Crop4 × 5: cropped window of a 4 × 5 Tiles
DoG4: $\sigma_c = 4$

(a) Systematic Cropping

Figure 10: Continued.



Reference orientations:
(*H*) Horizontal, (*V*) vertical
(*D*1) Positive diagonal (+45°)
(*D*2) Negative diagonal (−45°)
Crop4 × 5: cropped window of a 4 × 5 Tiles
DoG4: $\sigma_c = 4$

(b) Random Cropping

FIGURE 10: Mean tilts and standard errors around the four reference orientations (*H*, *V*, *D*1, and *D*2), for the three "foveal" sample sets (Figure 8) with the two sampling approaches of (a) Systematic and (b) Random methods. The parameters of the model and Hough processing are as follows: edge maps at seven DoG scales (σ_c = 4, 8, 12, 16, 20, 24, and 28), s = 2, and h = 8, with Hough parameters *NumPeaks* = 100, *Threshold* = 3, *FillGap* = 40, and *MinLength* = 450 (based on [19]).

effect, since similarly in the fovea the acuity is high because of high density of small size receptors. The vertical mean tilts are approximately 5-6° at medium to coarse scales. The diagonal mean tilts (around $D1$ and $D2$ axes) are around 4-5° which can be seen again at medium to coarse scales (σ_c = 20, 24, and 28).

The results of the detected mean tilts at a given scale show slight differences across foveal sample sets, and this is expected because of the Random sampling and the fixed Hough parameters that are not optimized for each scale and sampling size, and they are kept constant here for the consistency of the higher level analysis/model. The tilt detection results are sensible when compared to our angular tilt perception of the pattern, but more accuracy may be achieved by optimizing parameters based on the psychophysical experiments.

Figures 11(a) and 11(b) show the distribution of lines detected from the DoG edge maps at seven scales and around the four reference orientations (H, V, $D1$, $D2$) for the three foveal sample sets and the two sampling methods. The near-diagonal tilted lines (around $D1$ and $D2$ axes) have been graphed together for fairer representation. Figures 11(a) and 11(b) show that the houghlines detected in (b) are more normally distributed around the reference orientations compared to (a). All the graphs indicate the effect of the edge maps at multiple scales on the range of detected mean tilts and the distribution of the lines detected by their deviations in degrees from the reference orientations. The mean tilts cover a wider angular range when the DoG scale increases. We should be reminded that the number of detected lines is highly dependent on the sample size and the *NumPeaks* parameter. We explain the tilt results in Figure 11(b) but the same explanation can be used for part (a).

Figure 11(b) (left-column) shows the near-horizontal lines detected for the three foveal sets. At scale 4 (σ_c = 4), the detected tilt angles are very small, ranging between 2 and 5°, with the peak of 4°. Furthermore, at scale 16, the detection of a high range of variations of tilt angle is not reflected in our perception of the pattern. To detect horizontal tilt cues along the mortar lines, the scale of the center Gaussian in our model should be close to the mortar. At scale 8, the mean tilts are between 3 and 10° with the peak of 7° for most lines, and at scale 12 it is increased to ~14°. At scale 16 the results show a wider range of horizontal lines detected and a fairly broad range of vertical lines, and this fits as a transition stage between the "horizontal groupings" of identically colored tiles with the mortar lines at a focal view and the "zigzag vertical groupings" of the tiles [22] at a more peripheral/global view.

Figure 11(b) (center-column) shows the near-vertical lines detected. Similar to the indications of Figure 10, they start to be detected at fine scales due to some edge effects in a few samples, but as color code indicates, the majority of the near-vertical lines are detected at scales 20 and 24, with the mean tilts in the range of 2–15° and the peak close to the V axis. In Figure 11(b) (right-column) the detected lines with near-diagonal deviations indicate that the dominant scales for detection of the diagonal lines are mainly at coarse scales of 24 and 28 (σ_c = 24, 28) with approximately 5-6° deviation from the diagonal axes ($D1$, $D2$).

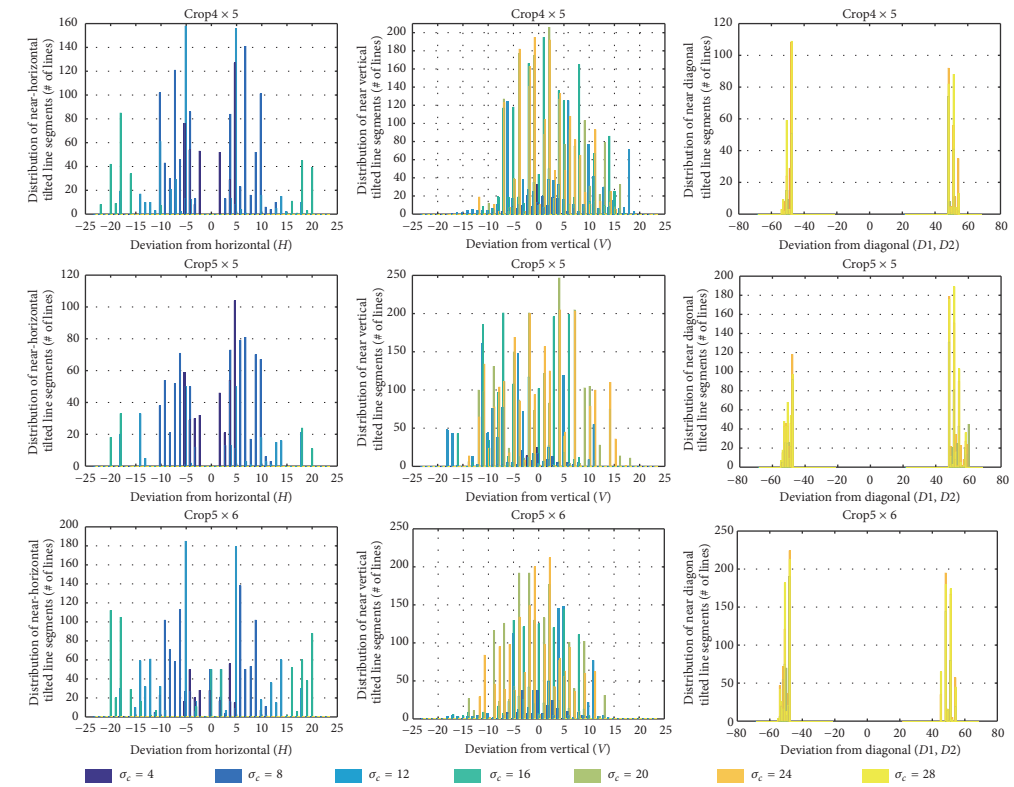
3.2. Global Tilt Investigation

3.2.1. Global Tilts in the Café Wall of 9 × 14 Tiles. The Café Wall illusion is characterized by the appearance of Twisted Cord elements along the mortar lines [23–25], making the tiles seem wedge-shaped [29]. These local tilt elements are believed to be integrated and produce slanted continuous contours along the whole mortar lines by the cortical cells [16, 27] resulting in alternating converging and diverging mortar lines at a global view.

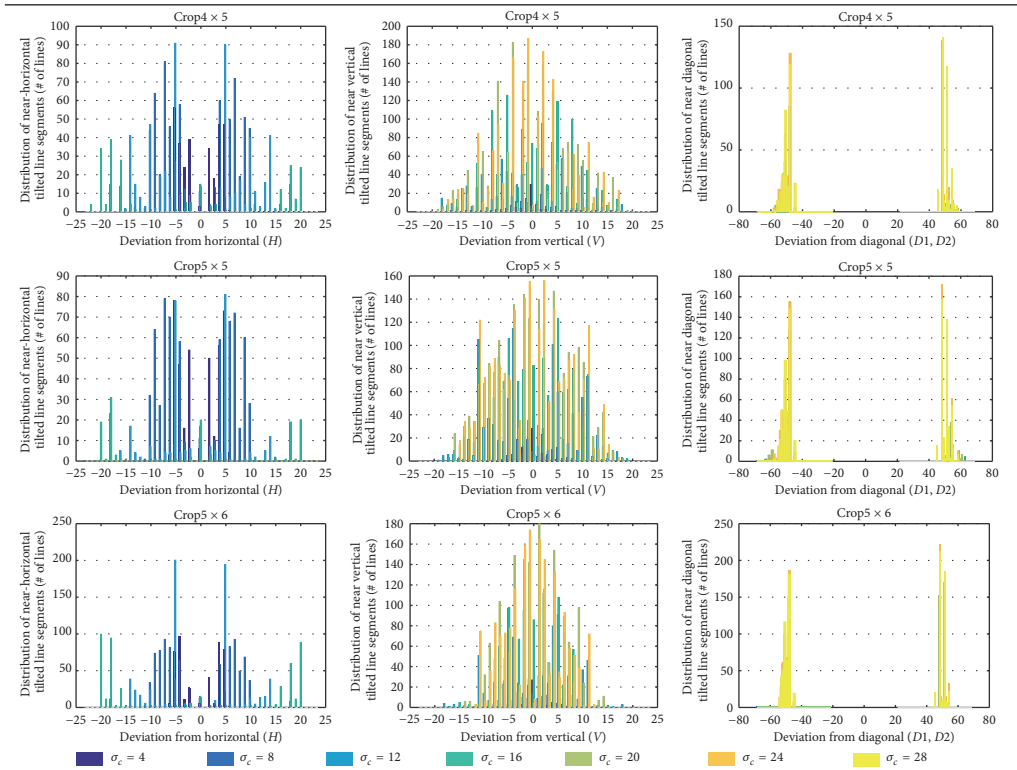
Because the tilt effect in the Café Wall is highly directional, it raises the question of whether lateral inhibition and point spread function (PSF) of retinal cells can explain the tilt effect in the pattern or not. We demonstrated that a bioplausible model [17–22], with a circularly symmetric organization as a simplified model for the retinal ganglion cell responses [32, 33], is able to reveal the tilt cues in the Café Wall illusion across multiple scales of the edge map. To explain the emergence of tilt in the Café Wall, there is no need to utilize complex models of non-CRFs [40–44] implementing the retinal/cortical orientation selective cells.

In this experiment, the intention is to investigate the Gestalt pattern, simulating peripheral awareness across the entire image, and overcome the shortcomings of our previous investigations. We investigate here the global tilts in the Café Wall of 9 × 14 with 200 × 200 px tiles and 8 px mortar (Figure 8(a)) and on its DoG edge map at seven different scales to quantify the tilt angles around the four reference orientations. The DoG scales have a range from $0.5M$ to $3.5M$ with the incremental steps of $0.5M$ the same as the foveal samples in Section 3.1.2.

In our first attempt to examine the robustness of the model for global tilt investigation [18, 19], the analysis was done with the parameters appropriate for local features. We have tested *NumPeaks* = 100 in [18] and *NumPeaks* = 520 in [19], but we have not achieved convincing results. Increasing the value of *NumPeaks* from 100 to 520 did not show any significant change to the mean tilts although it substantially increased the variance. The results showed that the near-horizontal mean tilt was approximately 4° at scale 4 and around 7° at scale 8 nearly the same as the foveal sample sets (Figure 10). The near-vertical mean tilts at medium to coarse scales were around 2°, while they were around 6° in the foveal sets. The near-diagonal mean tilts were approximately 3° and they were in the range of 5-6° in the foveal sample sets. Please refer to [19, Figure 6] for more details. In this work, we perform a global analysis with larger numbers of line segments as appropriate to the large global pattern. *NumPeaks* is a size relevant parameter, and its value is critical for achieving reliable results. Increasing this value for Hough analysis on the foveal sets does not affect the detected houghlines there, but an appropriate value for large samples is essential to detect all the relevant houghlines available in the edge map with smooth variations reflecting our estimation of tilt that is comparable with the detected lines in the foveal sets in our simulations.



(a) Systematic Cropping



(b) Random Cropping

Figure 11: The distribution of the line segments detected from the edge maps of the foveal sets (Figure 8), having either horizontal (left-column), vertical (center-column), or diagonal (right-column) deviations, with the two sampling methods: (a) Systematic and (b) Random Cropping. The edge maps are at seven different scales (σ_c = 4, 8, 12, 16, 20, 24, and 28) and a fixed set of Hough parameters are used as *NumPeaks* = 100, *Threshold* = 3, *FillGap* = 40, and *MinLength* = 450 (based on [19]).

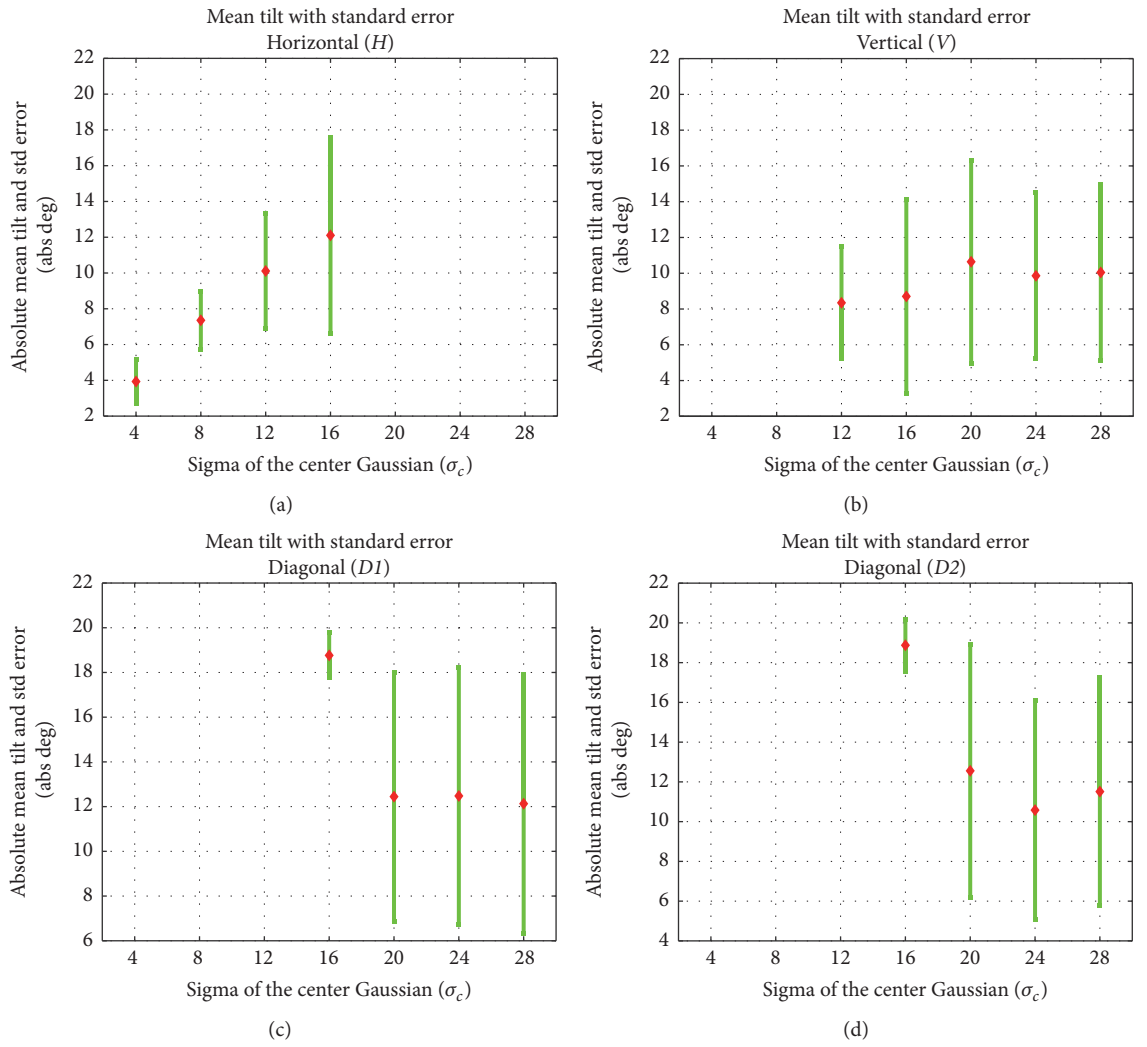


Figure 12: Mean tilts and the standard errors of detected tilt angles around the four reference orientations (H, V, $D1$, and $D2$) from the DoG edge map at seven different scales of the whole Café Wall of 9 × 14 with 200 × 200 px tiles and 8 px mortar. Green error bars correspond to Hough *NumPeaks* = 1000, with mean values shown in Red.

The new experimental results for mean tilts and standard errors of the detected tilt angles have been presented in Figure 12 for the Café Wall of 9 × 14 tiles with *NumPeaks* = 1000. The other parameters are kept the same as Figures 10 and 11 for the foveal sets. As indicated in Figure 12(a), the near-horizontal mean tilt is approximately 4° at scale 4 and 7.5° at scale 8 nearly the same as the foveal sample sets (Figure 10). In the horizontal graph, we see that, by increasing the DoG scale, the mean tilt also increases from 7.5° to ~10° at scale 12 with higher variations compared to the foveal sample sets. The new results for the vertical and diagonal mean tilts at coarse scales have been improved dramatically from our previous reports (explained in previous paragraph) and show a variation of tilt angles for the detected houghlines. The near-vertical mean tilts at medium to coarse scales were around 2° that are quite negligible in the previous report; now they are around 10° while they were around 6° in the foveal sets. The near-diagonal mean tilts at medium to coarse scales were approximately 3° in the previous reports; now the value is ~10° and they were in the range of 5-6° in the foveal sample sets. We have shown here that the new results with periphery appropriate parameterization are reliable and comparable with the previous results for foveal parameterization (Section 3.1.2). We will explain more on these results in Sections 3.2.2 and 3.3.

3.2.2. Variant Sized Café Wall Patterns with the Same Aspect Ratios of Tile Size to Mortar Size. We can assume that the tilt perception of the Café Wall illusion starts by a wholistic view to the pattern, which then extends to a local focusing view along the mortar lines in search for further cues of tilt in the pattern. Both of these local and global views to the Café Wall have their own effect on the strength of our perceptual understanding of tilt in this pattern. We started our investigations of the global tilt analysis in the Café Wall stimulus by first addressing the shortcomings of

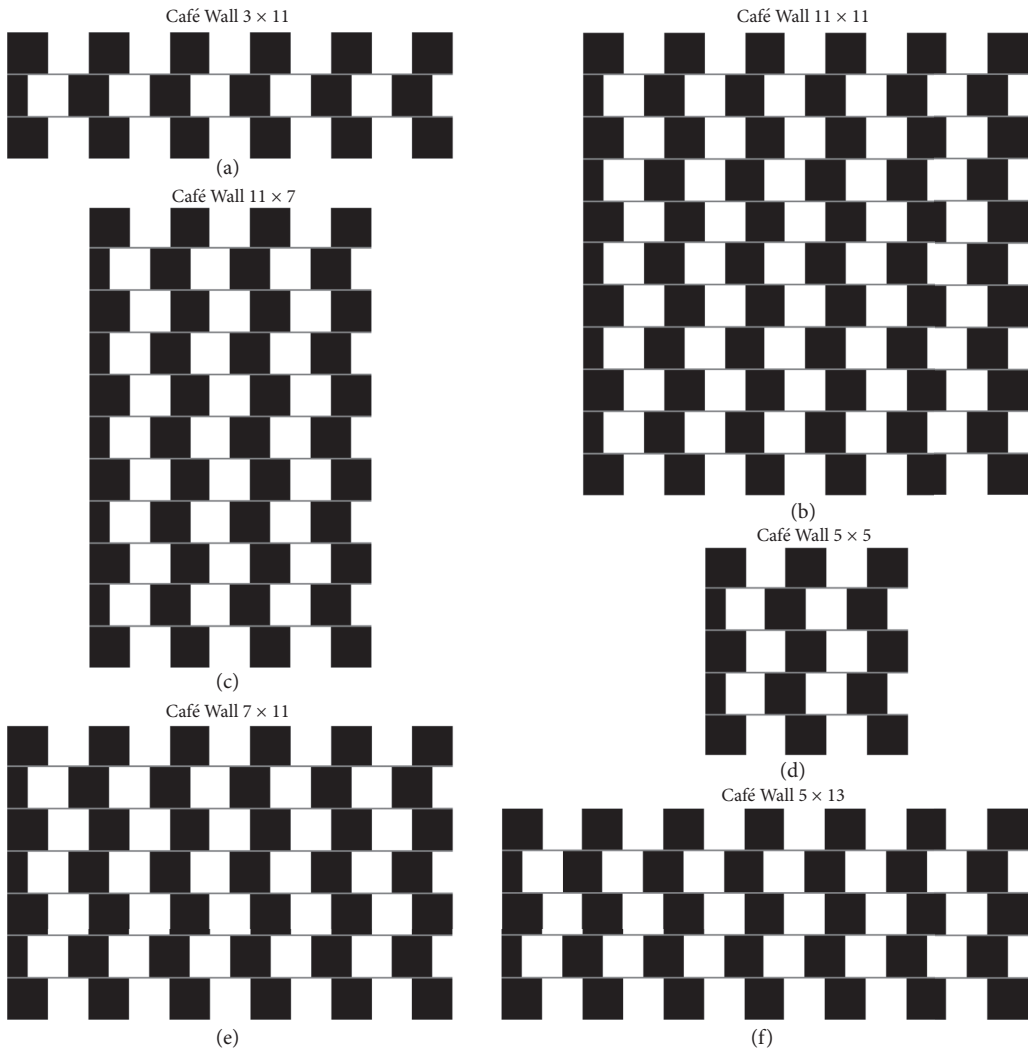


Figure 13: Different configurations of Café Wall pattern, from Café Wall of 3 × 11 tiles on (a) to Café Wall of 5 × 13 tiles at (f). All the patterns have the same tile size of 200 × 200 px and the mortar size of 8 px.

our previous reports [18, 19] as reflected in the previous section and presented reliable tilt results for a specific sample (Café Wall of 9 × 14 tiles) based on periphery appropriate parameterization. We show in this experiment our deep investigations of the global tilts on variations of Café Wall with the same characteristics of mortar lines and tiles but with different arrangements of a whole pattern. We have explored here the correlation between the tilt effect and the layout of the pattern in general (how the tiles are arranged to build the stimulus).

In this experiment, variations of Café Wall pattern have been investigated with the same aspect ratios of tile size (T) to mortar size (M) ($T/M = const.$) in order to check whether # rows and # columns in the overall arrangement of tiles in the Café Wall pattern have an effect on the detected tilts in our simulation results or not. In other words, we check the Gestalt perception of the Café Wall pattern and its relation to visual angle of the whole pattern (not just the visual angle of an individual tile and mortar line investigated so far [18, 19, 21]). We show here that our model can predict slightly different tilt results for these variations, similar to our global perception of illusory tilt in the pattern in the same way as human is affected by the configurations of the pattern. This is being reported for the first time with the quantitative results.

The patterns explored here have the same size of tiles (200 × 200 px) and mortar lines (8 px) with these variations: Café Walls of 3 × 11, 5 × 5, 5 × 13, 7 × 11, 11 × 7, and 11 × 11 tiles, as shown in Figure 13. Looking at these variations, we see, for instance, a stronger tilt effect in the 5 × 13 tiles compared to the 5 × 5 tiles. Similarly, a stronger tilt effect is perceived in the variation of 7 × 11 tiles compared to weaker tilts in the 11 × 7 tiles.

To eliminate the effect of *NumPeaks* on detected houghlines, in this experiment, we have selected *NumPeaks* = 1000 to attain more accurate tilt measurements when the overall sizes of the Café Wall samples are not the same (similar to Section 3.2.1). We have also tested values above 1000 up to 5000 for this parameter, but we found empirically that

there is no significant difference in the mean tilt results above *NumPeaks* = 1000 for the samples tested and around four reference orientations. Increasing this value is computationally expensive and we need to keep a trade-off between the efficiency and the accuracy. The rest of the parameters are kept the same as Figures 10–12 for the local and global investigations of tilt in variations of the Café Wall pattern. The Hough parameters are as follows: *NumPeaks* = 1000, *FillGap* = 40, and *MinLength* = 450 for all scales of the DoG edge maps (σ_c = 4, 8, 12, 16, 20, 24, and 28). Summary tables in Figure 14 present the quantitative mean tilts for the global tilt investigations on these configurations of the pattern. These include the mean tilts and the standard errors of the detected tilt angles around the four reference orientations (H, V, $D1$, and $D2$).

The DoG outputs of these variations are the same across multiple scales, since the tiles and mortar lines have fixed sizes and the same set of parameters for the *Surround* and *Window ratios* (s, h, resp.) have been used in the DoG model. Utilizing the Hough analytical pipeline for quantifying the tilt angles, we have measured slightly different tilts across the multiple scales of the edge maps of these variations around the four reference orientations. This is because Hough analyses more dominant lines (longest lines) first by applying the houghpeaks function prior to detecting lines with the houghlines function (MATLAB functions).

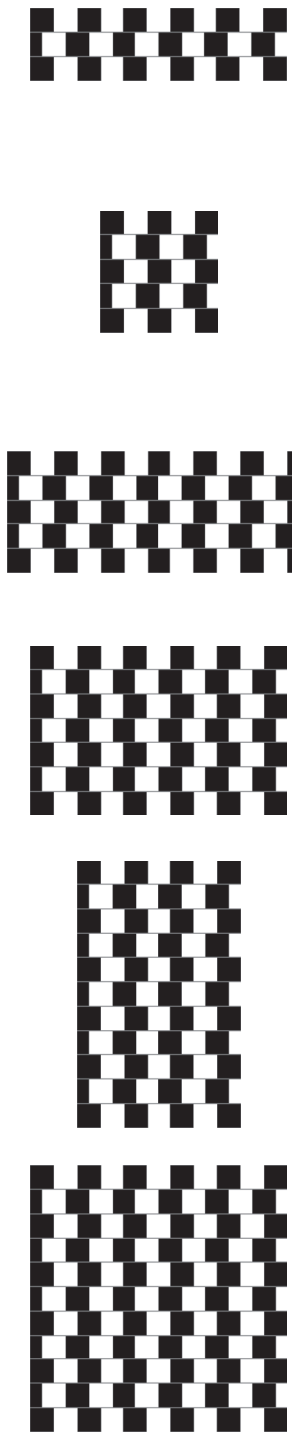
When the pattern is wider in horizontal direction such as the 3 × 11, 5 × 13, or 7 × 11 it seems that we see a stronger tilt effect along the mortar lines compared to the other variations. The quantitative mean tilts near horizontal orientation occur at fine to medium scales (σ_c = 4, 8, and 12) of the edge maps. The near-horizontal lines can be captured until scale 16 (σ_c = 16), with this mortar width (considering the same aspect ratio of the tile size to the mortar size), as well as the midluminance of the mortar lines relative to the luminance of the tiles [21]. This can be seen also for the edge map of Café Wall of 3 × 8 tiles in Figure 2. There is a transient stage at scale 16 (σ_c = 16), connecting the detected near-horizontal lines to the zigzag vertical line segments due to the arrangement of grouping of tiles in the zigzag vertical orientation at medium to coarse scales in the edge maps. The highest tilt range is shown in Figure 14 for the 5 × 13 configuration which is 3–10.6° at fine to medium scale (σ_c = 4, 8, and 12), as expected. Then the variations of 7 × 11 (3.8–10.3°) and 11 × 11 (4–10.5°) come next, followed by the 3 × 11, 5 × 5, and 11 × 7 tiles. Considering the two square patterns (the 5 × 5 and 11 × 11 tiles), there are similar horizontal mean tilts, starting around 4.0° at the finest scale (σ_c = 4) and it is one degree wider in the 11 × 11 variation at scale 12 (σ_c = 12), ~10.5° compared to ~9.5° in the 5 × 5, but the differences are only significant for σ_c = 4.

The near-vertical mean tilts at medium to coarse scales (σ_c = 20, 24, and 28) show good results. The weakest vertical mean tilts correspond to the Café Wall of 3 × 11 tiles which ranges from 7.1° to 7.5°. For the patterns of medium size height such as the 5 × 13 and 7 × 11 tiles, it is ~9-10°. It is in the highest range around 10.5° when the pattern is spread along the vertical orientation such as the 11 × 7 and 11 × 11 variations. This is nearly the same for the 5 × 5 tiles having the ratio of height/width = 1 and with a maximum value for the 11 × 11 tiles (>10.6°). So the Café Wall of 5 × 13 tiles from the samples explored has the *strongest horizontal tilt* range while the 11 × 7 tiles show the *strongest vertical tilt* range. It seems that there is a trade-off in the mean tilts of the vertical and the horizontal orientation, and, for a stronger effect of vertical tilts, we encounter weaker horizontal tilts along the mortar lines. For the diagonal mean tilts the results show roughly similar deviations (in both positive and negative diagonals) at the coarse scales (σ_c = 24, 28) which is ~11-12° across the samples tested.

We note that the results reported here are based on our investigations on the number of lines detected at their angular positions and we have not considered any weights for the length of the lines in the mean tilts calculations. For the horizontal mean tilts this does not affect the results, since at fine to medium scales the local tilted line segments (Twisted Cord elements) are extracted for all of these variations nearly the same with roughly similar size. The detected houghlines at scale 12 (medium scale) for all variations tested have been provided in Figure 15, highlighting local tilts of the nearly horizontal Twisted Cords and small tilt deviations from the vertical orientation. However, if a Café Wall pattern is more spread along the vertical orientation compared to the horizontal, then longer lines are detected with less deviation along the vertical at coarse scales (the whole tiles are present in the edge maps with no mortar cues left at these scales). Figure 16 clarifies this more: the detected houghlines at scale 28 (the coarsest scale) have been presented for these variations, indicating the global tilts of the lines detected with zigzag vertical orientation. In fact, as we expect from the tilt estimation, deviations from the vertical orientation increase as the lines found get shorter.

3.3. Comparison of Local and Global Tilts in Café Wall Illusion. We have shown in the last two sections that the mean tilt results with periphery appropriate parameterization are reliable and comparable with the previous results for foveal parameterization. The results for near-horizontal global tilts in these variations are nearly the same as the local tilts detected in the foveal sample sets (Section 3.1.2) 4° at scale 4 and ~7° at scale 8. At scale 12, we have a higher tilt angle ~9.5–10.5° here compared to the local tilts around 8°. The results of the vertical and diagonal mean tilts are slightly larger than the predicted values for the foveal samples (7–10° for the vertical and 11-12° for the diagonal tilts here compared to ~6° for the vertical and 5-6° for the diagonal tilts in the foveal samples). The results here seem more realistic in our perception of zigzag vertical lines at coarse scales considering the phase shift of rows of tiles in the Café Wall pattern (the deviations from the diagonal axes are more than 5°, considering the geometry of the pattern).

The quantitative modelling presented for the perceived tilt in the Café Wall illusion considering the foveal/local aspects as well as the peripheral/global view to the pattern leads us to achieve reliable results in our investigations. However, we illustrate some improvements to the current evaluation for future studies on the topic. First, for near-horizontal mean tilts, although the tilt analysis pipeline in our model detects the local Twisted Cord elements as local tilt



σ_c	Café Wall 3 × 11 (mean tilt ± StdErr)			
	H	V	$D1$	$D2$
4	04.50 ± 0.25	NaN	NaN	NaN
8	07.15 ± 0.54	14.00 ± 0.0	NaN	NaN
12	10.00 ± 0.59	10.63 ± 0.49	NaN	NaN
16	12.74 ± 1.07	07.30 ± 0.58	18.71 ± 0.72	19.40 ± 0.45
20	NaN	07.20 ± 0.66	12.84 ± 0.84	12.45 ± 0.82
24	NaN	07.13 ± 0.64	13.06 ± 0.77	12.93 ± 0.70
28	NaN	07.55 ± 0.59	13.69 ± 0.77	14.07 ± 0.72

σ_c	Café Wall 5 × 5 (mean tilt ± StdErr)			
	H	V	$D1$	$D2$
4	04.00 ± 0.0	NaN	NaN	NaN
8	07.33 ± 0.63	NaN	NaN	NaN
12	09.46 ± 1.01	08.66 ± 0.60	NaN	NaN
16	12.94 ± 1.04	08.02 ± 0.70	19.33 ± 0.45	19.00 ± 0.57
20	NaN	09.78 ± 0.72	13.11 ± 0.89	12.89 ± 0.90
24	NaN	10.14 ± 0.63	12.08 ± 0.70	12.37 ± 0.68
28	NaN	10.82 ± 0.68	12.54 ± 0.72	12.74 ± 0.69

σ_c	Café Wall 5 × 13 (mean tilt ± StdErr)			
	H	V	$D1$	$D2$
4	03.00 ± 0.35	NaN	NaN	NaN
8	07.13 ± 0.38	NaN	NaN	NaN
12	10.63 ± 0.43	09.74 ± 0.46	NaN	NaN
16	12.32 ± 0.70	08.41 ± 0.62	19.43 ± 0.34	19.22 ± 0.31
20	NaN	07.63 ± 0.45	11.88 ± 0.71	11.30 ± 0.70
24	NaN	09.27 ± 0.48	11.40 ± 0.56	11.73 ± 0.54
28	NaN	09.08 ± 0.45	11.74 ± 0.61	12.41 ± 0.60

σ_c	Café Wall 7 × 11 (mean tilt ± StdErr)			
	H	V	$D1$	$D2$
4	03.88 ± 0.29	NaN	NaN	NaN
8	07.26 ± 0.35	14.00 ± 0.0	NaN	NaN
12	10.39 ± 0.38	09.10 ± 0.42	NaN	NaN
16	11.79 ± 0.60	07.49 ± 0.46	18.89 ± 0.40	18.73 ± 0.31
20	NaN	08.74 ± 0.41	11.62 ± 0.65	12.01 ± 0.70
24	NaN	09.12 ± 0.40	11.68 ± 0.54	11.54 ± 0.54
28	NaN	09.15 ± 0.40	12.00 ± 0.55	12.05 ± 0.55

σ_c	Café Wall 11 × 7 (mean tilt ± StdErr)			
	H	V	$D1$	$D2$
4	04.00 ± 0.0	NaN	NaN	NaN
8	08.09 ± 0.29	NaN	NaN	NaN
12	09.96 ± 0.42	09.42 ± 0.46	NaN	NaN
16	12.63 ± 0.63	08.27 ± 0.57	18.67 ± 0.36	19.64 ± 0.43
20	NaN	10.63 ± 0.48	12.58 ± 0.63	12.14 ± 0.71
24	NaN	10.85 ± 0.43	12.46 ± 0.54	11.83 ± 0.56
28	NaN	10.62 ± 0.42	12.01 ± 0.56	11.62 ± 0.55

σ_c	Café Wall 11 × 11 (mean tilt ± StdErr)			
	H	V	$D1$	$D2$
4	04.05 ± 0.09	NaN	NaN	NaN
8	07.50 ± 0.33	15.00 ± 0.0	NaN	NaN
12	10.47 ± 0.31	09.39 ± 0.36	NaN	NaN
16	11.96 ± 0.54	06.27 ± 0.47	19.57 ± 0.25	18.79 ± 0.35
20	NaN	10.70 ± 0.40	13.15 ± 0.55	11.44 ± 0.62
24	NaN	10.15 ± 0.37	12.10 ± 0.50	11.16 ± 0.50
28	NaN	10.50 ± 0.35	11.81 ± 0.50	11.20 ± 0.52

(a) (b)

Figure 14: (a) Different configurations of the Café Wall pattern tested (Figure 13) from Café Wall of 3 × 11 tiles on the top to Café Wall of 11 × 11 tiles at the bottom. (b) Mean tilts and the standard errors of tilt angles for each variation are summarized in the mean tilt tables for the four reference orientations of horizontal (H), vertical (V), and diagonals ($D1$, $D2$) orientations at seven scales of the edge maps (σ_c = 4, 8, 12, 16, 20, 24, and 28).

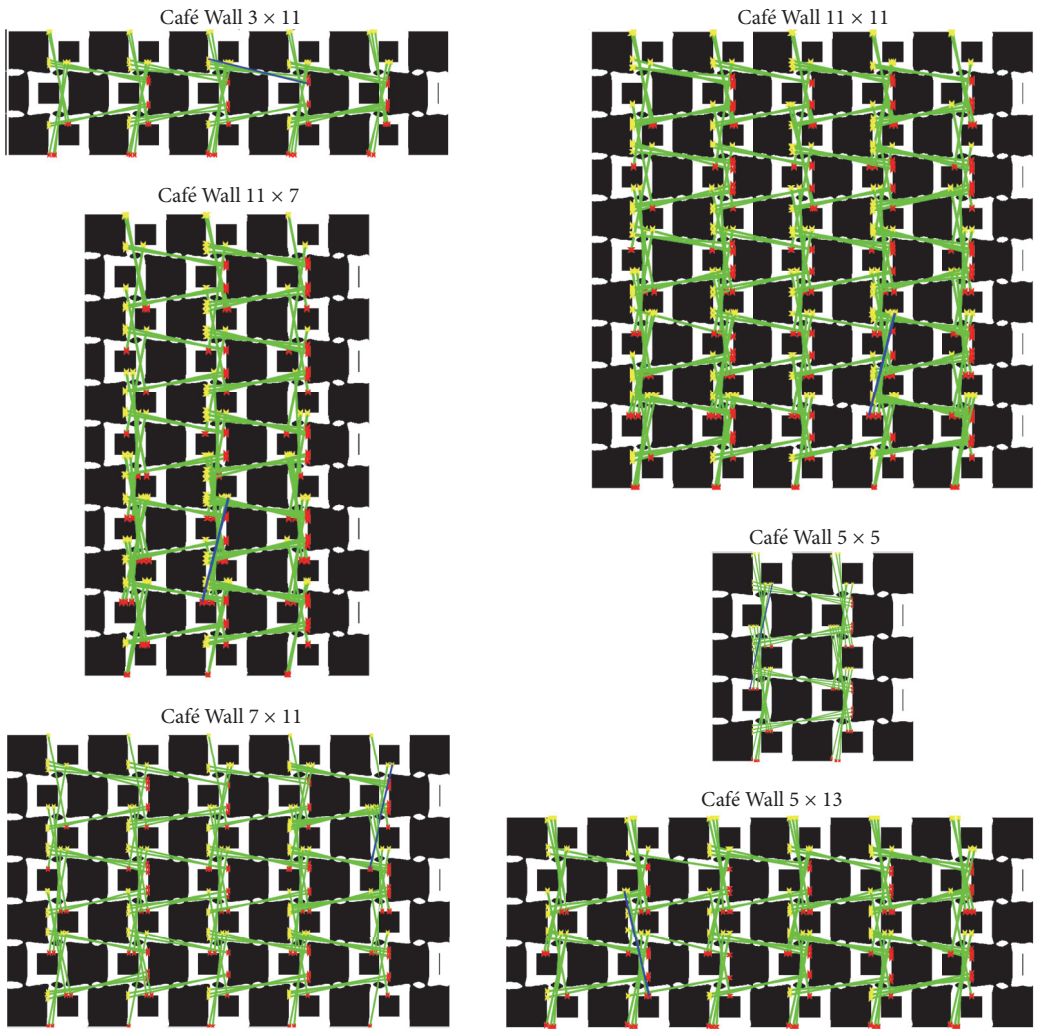


FIGURE 15: Detected houghlines displayed in green, overlaid on the binary edge maps at scale 12 ($\sigma_c = 12$) of different configurations of the Café Wall pattern in Figure 13. Hough parameters are as follows: *NumPeaks* = 1000, *Threshold* = 3, *FillGap* = 40, and *MinLength* = 450.

cues as shown in Figure 15 (for scale 12 for these variations), it seems that, in our perception of tilt, we intend to integrate these local tilt cues to construct a slanted continuous contour along the entire mortar as either diverging or converging [16, 27] tilt. Therefore, an edge integration technique is required for predicting a more precise value for the near-horizontal tilts as we perceive tilts in the Café Wall. Second, in the investigated tilts around the vertical orientation, we expect to see less deviations for the vertically spread configurations compared to the horizontally spread ones. However, the results showed the maximum vertical deviations for the Café Walls of 11 × 7 and 11 × 11 around 10.5° compared to 9° in others, except 7.5° for the Café Wall of 3 × 11. In the 3 × 11 tiles, we see more deviations around the diagonals compared to the rest of the configurations (as it is expected), where the range is ~12.5–14° compared to 11–12.5°, and also less deviations from the vertical orientation due to the groupings of detected lines for the reference orientations. Our explanation of the results is getting clear by looking at the houghlines presented in Figure 16. In the Hough analysis, we have applied the same weight for all the detected lines. Therefore for the patterns that are spread along the vertical orientation, although houghlines detect many longer lines with less deviations in the edge map, houghpeaks let more smaller line segments be detected (up to the maximum value of *NumPeaks*), with more deviations from the vertical orientation. For final validations of these results, psychophysical experiments are required as the priority of our future work. The results from psychophysical experiments lead us to assign weights to each scale and approximate tilt angles in our model based on the perceived tilt in real subjects.

4. Conclusion

A low-level filtering approach [17, 19, 22] has been explored here modelling the retinal/cortical simple cells in our early vision for revealing the tilt cues involved in the local and global perception of the Café Wall stimulus. The model has

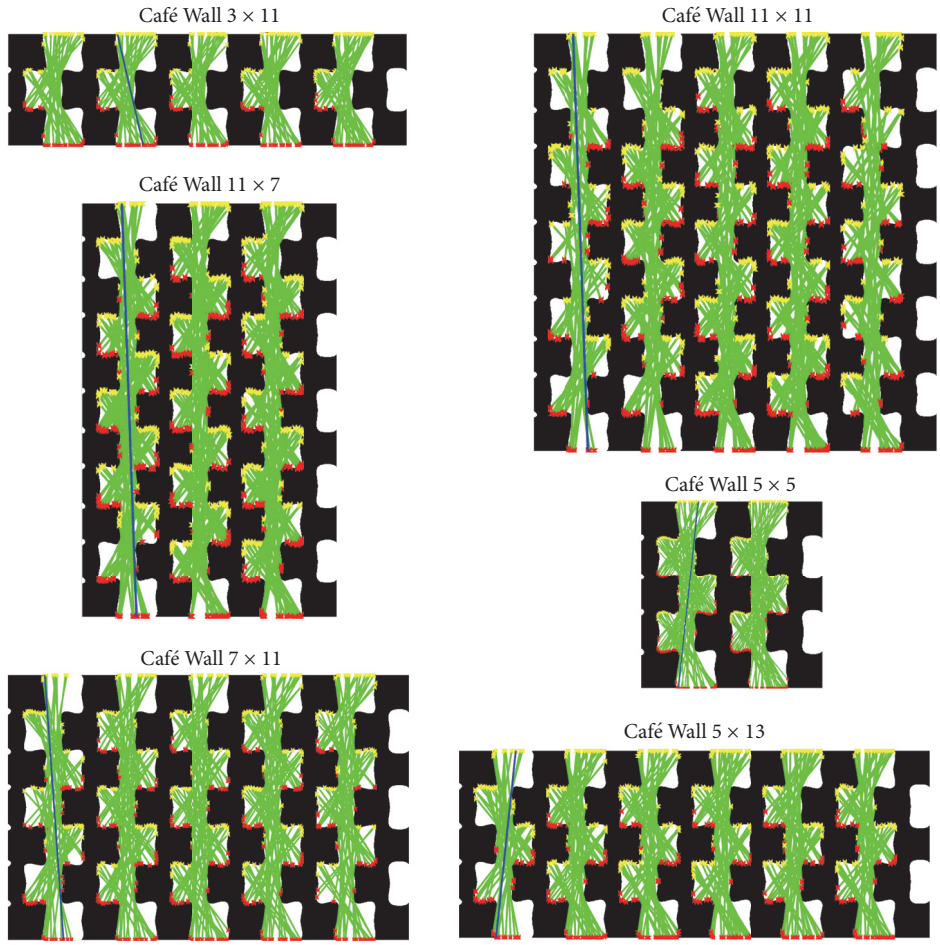


Figure 16: Detected houghlines displayed in green, overlaid on the binary edge maps at scale 28 ($\sigma_c = 28$) of different configurations of the Café Wall pattern in Figure 13. Hough parameters are as follows: *NumPeaks* = 1000, *Threshold* = 3, *FillGap* = 40, and *MinLength* = 450.

an embedded processing pipeline utilizing Hough transform to quantify the degrees of the induced tilts that appeared in the low-level representation for the stimulus in our model, referred to as the *edge map at multiple scales*.

The experiments reported have contributed new understanding of the relationship between the strength of tilt effect perceived in the Café Wall illusion as a function of eccentricity, that is, whether a cell or edge is foveated or perceived in the periphery.

Different size/shape cropped samples of the Café Wall pattern were used to model the role of the shape and size of the fovea and larger samples tended to induce a larger number of longer shallower lines, particularly in the vertical dimension. When we foveate a particular cell we tend to see that as having more horizontal mortar boundaries, while those outside the fovea are perceived as having larger tilts. This is consistent with the larger tilts perceived at lower resolutions, modelling the periphery, and the almost horizontal tilts seen in the foveal region, corresponding to the center of a larger pattern. This makes this a multiple scale model.

It is hypothesized that the multiple scale information from the retina is integrated later in the cortex into a true multiscale model and that the Gestalt illusions result from the angle misperceptions that are already encoded in the retina. The quantitative predictions are based on the analysis of Hough transform of the edge maps here with promising results reported. This tilt investigation can be replaced by any more bioderived techniques, modelling mid-to-high-level tilt integrations, capable of quantifying different degrees of tilt in variations of the Gestalt view of the pattern, as we perceive the tilt differently in those variations.

We regard the publication of the predictions before running experiments to validate them as essential to the integrity of science. A priority in our research is psychophysical experiments to validate the predictions of the model.

Conflicts of Interest

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

Acknowledgments

Nasim Nematzadeh was supported by an Australian Research Training Program (RTP) award for her Ph.D.

References

[1] L. Spillmann, "Receptive fields of visual neurons: The early years," *Perception*, vol. 43, no. 11, pp. 1145–1176, 2014.

[2] P. B. Cook and J. S. McReynolds, "Lateral inhibition in the inner retina is important for spatial tuning of ganglion cells," *Nature Neuroscience*, vol. 1, no. 8, pp. 714–719, 1998.

[3] J. Y. Huang and D. A. Protti, "The impact of inhibitory mechanisms in the inner retina on spatial tuning of RGCs," *Scientific Reports*, vol. 6, article 21966, 2016.

[4] F. Ratliff, B. W. Knight, and N. Graham, "On tuning and amplification by lateral inhibition.," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 62, no. 3, pp. 733–740, 1969.

[5] G. D. Field and E. J. Chichilnisky, "Information processing in the primate retina: Circuitry and coding," *Annual Review of Neuroscience*, vol. 30, pp. 1–30, 2007.

[6] T. Gollisch and M. Meister, "Eye smarter than scientists believed: neural computations in circuits of the retina," *Neuron*, vol. 65, no. 2, pp. 150–164, 2010.

[7] T. Lindeberg and L. Florack, "Foveal scale-space and the linear increase of receptive field size as a function of eccentricity," KTH Royal Institute of Technology, 1994.

[8] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[9] P. H. Schiller, "Parallel information processing channels created in the retina," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 107, no. 40, pp. 17087–17094, 2010.

[10] H. B. Barlow, A. M. Derrington, L. R. Harris, and P. Lennie, "The effects of remote retinal stimulation on the responses of cat retinal ganglion cells.," *The Journal of Physiology*, vol. 269, no. 1, pp. 177–194, 1977.

[11] L. J. Frishman and R. A. Linsenmeier, "Effects of picrotoxin and strychnine on non-linear responses of Y-type cat retinal ganglion cells.," *The Journal of Physiology*, vol. 324, no. 1, pp. 347–363, 1982.

[12] B. Roska and F. Werblin, "Rapid global shifts in natural scenes block spiking in specific ganglion cell types," *Nature Neuroscience*, vol. 6, no. 6, pp. 600–608, 2003.

[13] L. J. Croner and E. Kaplan, "Receptive fields of P and M ganglion cells across the primate retina," *Vision Research*, vol. 35, no. 1, pp. 7–24, 1995.

[14] C. Enroth-Cugell and R. M. Shapley, "Adaptation and dynamics of cat retinal ganglion cells," *The Journal of Physiology*, vol. 233, no. 2, pp. 271–309, 1973.

[15] R. A. Linsenmeier, L. J. Frishman, H. G. Jakiela, and C. Enroth-Cugell, "Receptive field properties of X and Y cells in the cat retina derived from contrast sensitivity measurements," *Vision Research*, vol. 22, no. 9, pp. 1173–1183, 1982.

[16] S. Grossberg and E. Mingolla, "Neural dynamics of form perception. boundary completion, illusory figures, and neon color spreading," *Psychological Review*, vol. 92, no. 2, pp. 173–211, 1985.

[17] N. Nematzadeh, T. W. Lewis, and D. M. W. Powers, "Bioplausible multiscale filtering in retinal to cortical processing as a model of computer vision," in *Proceedings of the 7th International Conference on Agents and Artificial Intelligence, ICAART 2015*, pp. 305–316, January 2015.

[18] N. Nematzadeh and D. M. W. Powers, "A quantitative analysis of tilt in the Café Wall illusion: a bioplausible model for foveal and peripheral vision," in *Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016*, December 2016.

[19] N. Nematzadeh and D. M. W. Powers, "A bioplausible model for explaining Café Wall illusion: foveal vs peripheral resolution," in *Proceedings of the International Symposium on Visual Computing*, pp. 426–438, Springer International Publishing, 2016.

[20] N. Nematzadeh, D. M. W. Powers, and T. W. Lewis, "Quantitative analysis of a bioplausible model of misperception of slope in the Café Wall illusion," in *Proceedings of the Asian Conference on Computer Vision*, pp. 622–637, Springer, 2016.

[21] N. Nematzadeh and D. M. Powers, "A predictive account of Café Wall illusions using a quantitative model," submitted, https://arxiv.org/abs/1705.06846.

[22] N. Nematzadeh, D. M. Powers, and T. W. Lewis, "Bioplausible multiscale filtering in retino-cortical processing as a mechanism in perceptual grouping," *Brain Informatics*, pp. 1–23, 2017.

[23] D. C. Earle and S. J. Maskell, "Fraser cords and reversal of the café wall illusion.," *Perception*, vol. 22, no. 4, pp. 383–390, 1993.

[24] M. J. Morgan and B. Moulden, "The Münsterberg figure and twisted cords," *Vision Research*, vol. 26, no. 11, pp. 1793–1800, 1986.

[25] J. FRASER, "A new visual illusion of direction 1904–1920," *British Journal of Psychology*, vol. 2, no. 3, pp. 307–320, 1908.

[26] G. Westheimer, "Irradiation, border location, and the shifted-chessboard pattern," *Perception*, vol. 36, no. 4, pp. 483–494, 2007.

[27] B. Moulden and J. Renshaw, "The Munsterberg illusion and 'irradiation'," *Perception*, vol. 8, no. 3, pp. 275–301, 1979.

[28] M. E. McCourt, "Brightness induction and the Café Wall illusion.," *Perception*, vol. 12, no. 2, pp. 131–142, 1983.

[29] R. L. Gregory and P. Heard, "Border locking and the Cafe Wall illusion," *Perception*, vol. 8, no. 4, pp. 365–380, 1979.

[30] A. Kitaoka, B. Pinna, and G. Brelstaff, "Contrast polarities determine the direction of Café Wall Tilts," *Perception*, vol. 33, no. 1, pp. 11–20, 2004.

[31] S. W. Kuffler, "Neurons in the retina: organization, inhibition and excitation problems," *Cold Spring Harbor Symposium on Quantitative Biology*, vol. 17, no. 0, pp. 281–292, 1952.

[32] R. W. Rodieck and J. Stone, "Analysis of receptive fields of cat retinal ganlion cells," *Journal of Neurophysiology*, vol. 28, no. 5, pp. 833–849, 1964.

[33] C. Enroth-Cugell and J. G. Robson, "The contrast sensitivity of retinal ganglion cells of the cat," *The Journal of Physiology*, vol. 187, no. 3, pp. 517–552, 1966.

[34] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," *Proceedings of the Royal Society B Biological Science*, vol. 211, no. 1183, pp. 151–180, 1981.

[35] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London—Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[36] D. P. Lulich and K. A. Stevens, "Differential contributions of circular and elongated spatial filters to the Café Wall illusion," *Biological Cybernetics*, vol. 61, no. 6, pp. 427–435, 1989.

[37] D. M. W. Powers, "Jetwhite color map. Mathworks," 2016 https://au.mathworks.com/matlabcentral/fileexchange/48419-jetwhite-colours-996/content/jetwhite.m.

[38] P. V. Hough, "Method and means for recognizing complex patterns," (No. US 3069654), 1962.

[39] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer Vision Graphics and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988.

[40] B. Blakeslee and M. E. McCourt, "A multiscale spatial filtering account of the White effect, simultaneous brightness contrast and grating induction," *Vision Research*, vol. 39, no. 26, pp. 4361–4377, 1999.

[41] M. Carandini, "Receptive fields and suppressive fields in the early visual system," in *Cognitive Neurosciences Iii*, vol. 3, pp. 313–326, 3rd edition, 2004.

[42] E. Craft, H. Schütze, E. Niebur, and R. Von Der Heydt, "A neural model of figure-ground organization," *Journal of Neurophysiology*, vol. 97, no. 6, pp. 4310–4326, 2007.

[43] C. L. Passaglia, C. Enroth-Cugell, and J. B. Troy, "Effects of remote stimulation on the mean firing rate of cat retinal ganglion cells," *The Journal of Neuroscience*, vol. 21, no. 15, pp. 5794–5803, 2001.

[44] H. Wei, Q. Zuo, and B. Lang, "Multi-scale image analysis based on non-classical receptive field mechanism," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7064, no. 3, pp. 601–610, 2011.

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 8718956, 10 pages
https://doi.org/10.1155/2017/8718956

Research Article

CNN-Based Pupil Center Detection for Wearable Gaze Estimation System

Warapon Chinsatit and Takeshi Saitoh

Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, 680–4 Kawazu, Iizuka-shi, Fukuoka 820-8502, Japan

Correspondence should be addressed to Warapon Chinsatit; warapon@slab.ces.kyutech.ac.jp

Received 30 June 2017; Revised 16 October 2017; Accepted 25 October 2017; Published 4 December 2017

Academic Editor: Fouad Slimane

This paper presents a convolutional neural network- (CNN-) based pupil center detection method for a wearable gaze estimation system using infrared eye images. Potentially, the pupil center position of a user's eye can be used in various applications, such as human-computer interaction, medical diagnosis, and psychological studies. However, users tend to blink frequently; thus, estimating gaze direction is difficult. The proposed method uses two CNN models. The first CNN model is used to classify the eye state and the second is used to estimate the pupil center position. The classification model filters images with closed eyes and terminates the gaze estimation process when the input image shows a closed eye. In addition, this paper presents a process to create an eye image dataset using a wearable camera. This dataset, which was used to evaluate the proposed method, has approximately 20,000 images and a wide variation of eye states. We evaluated the proposed method from various perspectives. The result shows that the proposed method obtained good accuracy and has the potential for application in wearable device-based gaze estimation.

1. Introduction

People obtain various information through the human vision system. By observing eyes, we can observe changes in pupil size, eye direction, and changes in eye state, for example, opening, closing, blinking, and crying. This information can be used to estimate emotions, traits, or interests. To analyze the eye, eye image processing is an important task, and the development and availability of wearable cameras and recording devices have made eye image processing, including gaze estimation, increasingly popular.

A gaze estimation system (GES) involves multiple cameras, and such systems can estimate gaze direction and what a user is looking at. Thus, GESs can estimate objects of interest. One type of GES uses an inside-out camera [1, 2], which is comprised of an eye camera and a scene camera. The eye camera captures images of the user's eyes. Such a GES detects the pupil center and maps it to a point in the scene image. Recently, GESs have been used in various applications, such as video summarization [3], daily activity recognition [4], reading [5], human-machine interfaces [6], and communication support [7].

It is difficult to detect the pupil center because the eye is a nonrigid object, users blink frequently, and eyelid or eyelashes can occlude the pupil. Furthermore, the iris has various colors, such as blue, brown, and black. However, when an infrared camera is used to capture eye images, the iris fades out, which makes the pupil clearer. This approach makes the eye image easy to work with. However, blinking remains problematic because it is difficult to detect the pupil center point when a user blinks. Consequently, gaze direction errors can occur.

This research focuses on pupil center detection using infrared eye images captured by a wearable inside-out camera and proposes an accurate detection method that uses a convolutional neural network (CNN). The proposed method is composed of two CNN models. The first determines whether it is possible to detect a pupil in an input image. The second CNN model detects the pupil center in an input eye image. This model outputs the pupil center X- and Y-coordinates.

We evaluated the proposed method using a dataset of infrared eye images captured by our inside-out camera. The results demonstrate that the proposed method demonstrates higher accuracy than other methods.

Typically, CNNs are trained using supervised learning; thus, they require a large training dataset. There are some public datasets of eye images [8, 9]; however, such datasets do not typically include images of eyes in the blink state. We describe a process to capture a sufficiently large image dataset with good distribution and variety of pupil position and eye state.

2. Related Research

Several studies have focused on feature point detection based on eye images [10–13]. Li et al. proposed a hybrid eye-tracking method that integrates feature-based and model-based approaches [10]. They captured eye images using an inexpensive head-mounted camera. Their method detects pupil edge points and uses ellipse fitting to estimate the pupil center. Zheng et al. proposed an algorithm to detect eye feature points, including pupil center and radius, eye corners, and eyelid contours [11]. Moriyama et al. developed a generative eye region model that can meticulously represent the detailed appearance of the eye region for eye motion tracking [12]. Chinsatit and Saitoh proposed a fast and precise eye detection method using gradient value [13]. However, if the eye image contains unexpected objects with a high gradient or intensity, such as an eyelash with mascara or a specular point, it is difficult for such methods to detect the pupil.

CNNs outperform traditional algorithms in various research fields, such as artificial intelligence, image classification, and audio processing. Zhang et al. proposed a CNN-based gaze estimation method in an unconstrained daily life setting [8]. In that method, the input data are an eye image and the 2D head angle, and the output is a 2D gaze angle vector that consists of two gaze angles, that is, yaw and pitch. Fuhl et al. proposed a dual CNN pipeline for image-based pupil detection [14]. Here, the input is an eye image, and the output is an estimated pupil center position. In the first pipeline stage, an input image is downscaled and divided into overlapping subregions. A coarse pupil position is estimated by the first shallow CNN. In the second stage, subregions surrounding the initial estimation are evaluated using a second CNN, and the final pupil center position is detected. Choi et al. proposed a CNN model to categorize driver gaze zones [15]. Here the input image is an eye image, and the outputs are the probabilities of nine gaze zones. As mentioned previously, most related studies that employ CNNs attempt to detect only the center point of a pupil.

The objective of this study is to apply the proposed method to a GES. The proposed method is designed for daily life; thus, it must be robust because it is not always possible to detect the pupil center position, for example, when the eyelid overlays the pupil due to blinking. The proposed method is composed of two CNN models. The first model classifies the input image, as shown in Figure 1. The second model operates in a regression mode [16, 17]. Collectively, this CNN model outputs the X- and Y-coordinates of the pupil center point.

3. Proposed Method

A CNN is composed of a convolutional layer and a fully connected layer. Typically, the fully connected layer is a feed-forward neural network. The effective layer between the input data and the fully connected layer is the convolutional layer, which is used to detect the significant feature point in the input data prior to sending it to the fully connected layer. If the convolutional layer cannot detect the target feature point, it inputs zeros to the fully connected layer. Under this condition, the fully connected layer outputs only the bias effect of each layer. In other words, a CNN outputs a value regardless of the quality of the input data. We employ a CNN model to classify the input data prior to sending it to the detection model.

We describe the classification and detection models in the following subsections.

3.1. Classification Model. There are various CNN classification models, and each model has specific characteristics. AlexNet [18] is a well-known model for classification tasks. We selected this model to classify the eye state. We defined three states in eye images; that is, (1) the image shows the pupil as a full circle (open state), (2) an eyelid overlays the pupil (medium state), and (3) no pupil is observable in the image (closed state).

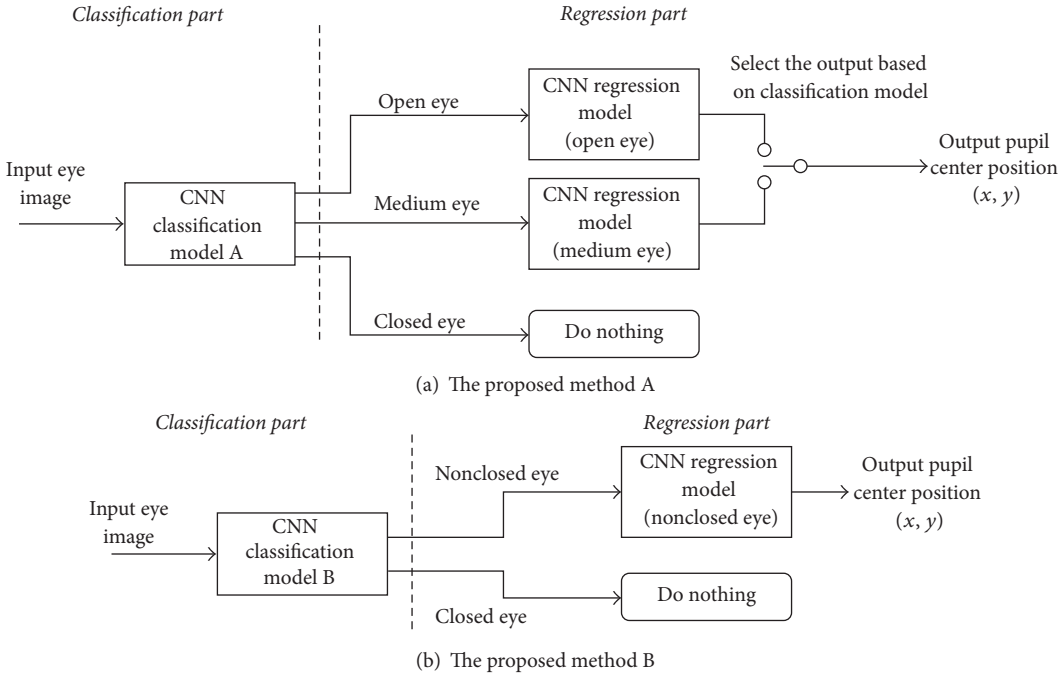
Some studies have used a separate CNN model to perform specific tasks. For example, Sun et al. created multiple models to detect each feature point [16]. We also propose using two methods, which we refer to as methods A and B. For method A, we create a CNN model to classify the input image as open, medium, or closed eye states, as shown in Figure 1(a). For medium and open eye images, we create two CNN regression models to detect the feature points from each image type. The details of method A's classification and regression models are listed in Table 1 (row 1). If the input image is an open eye image, it will be sent to a CNN model trained using only open eye images. Similarly, if the input image is a medium eye image, it is sent to a CNN model trained using only medium eye images.

The proposed CNN models can potentially solve multiple problems. Note that most previous studies employed an end-to-end CNN model to solve multiple problems. We use method B (Table 1, row 2) to classify input images as closed or nonclosed eye (i.e., open eye and medium eye images, respectively). This classification model selects only nonclosed eye images and sends those images to the CNN trained using nonclosed eye images, as shown in Figure 1(b). Note that we compare the performance of both methods.

A cost function must be defined prior to training the CNN. The training process attempts to minimize this cost function. In the proposed CNN classification model, we use the mean of the sum of squared errors as the cost function, which is expressed as follows:

$$\text{cost} = \frac{\sum_{i=1}^{N_o} (o_i - d_i)^2}{N_o}, \tag{1}$$

where o_i is an estimation output at i, d_i is a label at i, and N_o is the number of output classification results.



(a) The proposed method A

(b) The proposed method B

Figure 1: Proposed two-part CNN model.

Table 1: Proposed CNN architectures.

Name	Item	Conv1	Conv2	Conv3	Conv4	Conv5	Full1	Full2	Out
Classification model A	Channel	48	128	192	192	128	1024	1024	3 classes
	Filter size	11 × 11	5 × 5	3 × 3	3 × 3	3 × 3	—	—	—
	Pooling size	2 × 2	2 × 2	2 × 2	2 × 2	3 × 3	—	—	—
	Normalization	yes	—	—	—	—	Yes	Yes	—
	Dropout	—	—	—	—	—	Yes	Yes	—
Classification model B	Channel	48	128	192	192	128	1024	1024	2 classes
	Filter size	11 × 11	5 × 5	3 × 3	3 × 3	3 × 3	—	—	—
	Pooling size	2 × 2	2 × 2	2 × 2	2 × 2	3 × 3	—	—	—
	Normalization	yes	—	—	—	—	Yes	Yes	—
	Dropout	—	—	—	—	—	Yes	Yes	—
Regression model	Channel	96	256	512	512	512	4096	4096	2 reg.
	Filter size	7 × 7	5 × 5	3 × 3	3 × 3	3 × 3	—	—	—
	Pooling size	3 × 3	2 × 2	—	—	3 × 3	—	—	—
	Normalization	yes	—	—	—	—	Yes	Yes	—
	Dropout	—	—	—	—	—	Yes	Yes	—

3.2. Regression Model. The proposed CNN regression model (Table 1, row 3) is based on the pose regression ConvNet [17], which consists of five convolutional layers and three fully connected layers. The collection of convolutional layers is followed by pooling and local response normalization layers, and the fully connected layers are regularized using dropout. All hidden weight layers use a rectification activation (i.e., ReLU) function. Most CNN architectures for object localization use five convolutional layers [17, 19–21]. A difference between pose regression ConvNet and the proposed regression model is the normalization layer. ConvNet has a normalization layer after the last convolutional layer (Conv5). However, in a preliminary experiment, we found that training using the eye image dataset does not converge when the normalization layer is applied after the final convolutional layer. Thus, we do not employ this architecture. This difference also applies to the fully connected layers. In our architecture, we use local response normalization [18] for Conv1 and use $L2$ normalization for fully connected layers. $L2$ normalization is defined as follows:

$$x_k' = \frac{x_k}{\sqrt{\sum_{i=1}^{N_i} x_i^2}}, \tag{2}$$

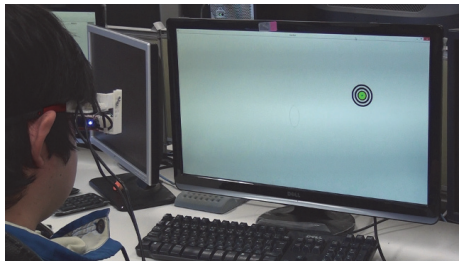


Figure 2: Collection experiment scene.

where k is the index of input nodes, x_k is the input data at node k, x'_k is the output from the normalization process at node k, and N_i is the number of data elements in the layer. This normalization process is required for training to converge.

We remove the activation function to make the output value linear. The input to the proposed CNN is an eye image (120×80 pixels). The error function e of the CNN regression model is defined as follows:

$$e = \sqrt{(P_x - D_x)^2 + (P_y - D_y)^2}. \quad (3)$$

This function is the distance between ground truth D and estimated point P.

4. Experiment

4.1. Dataset. A CNN is a supervised learning method that requires a large dataset to train a model. Moreover, a variety of ground truths are required to make the model more accurate. MPIIGaze [8] is a well-known eye image dataset composed of medial canthus, lateral canthus, and pupil points. However, the pupil points are not center points. For a GES, pupil center points are required to calculate gaze direction. In this study, we developed a system to capture a dataset with appropriate variation and reliability using an inside-out camera [2].

We required a dataset that contains blinking eye images to test the performance of the proposed CNN method. Thus, we had to design a system to capture multiple eye images under appropriate conditions. Note that the center of the pupil's position depends on gaze direction. To create the dataset, subjects wore an inside-out camera and observed a marker displayed on a monitor. Next, the system captures an image from the eye camera. We designed an additional process to ensure that the subject focused on the marker position. This capture system selects an arrow (up, right, down, and left) at random and displays it at the center of the marker. The subjects were tasked with pressing a corresponding arrow key. We asked the subjects to blink approximately five times before pressing the key. If the subject pressed the correct key, the capture system saved the eye images to the dataset. This process improved the variation of eye images in the dataset. The image collection environment is shown in Figure 2. Details about the data collection process are described in the following:

(i) We used a 24-inch widescreen display for this experiment, and the distance between the subject and the display was 60 cm. We captured the images for the dataset in a room with sufficient light from both natural and fluorescent light sources.

(ii) We divided the display area into 49 (7×7) sections and show the marker in that section, respectively. First, we shuffle the order of the marker position, in order to make the unpredictable position. The subject has to gaze at the marker without moving the head.

(iii) Then, the user was asked to blink approximately five times. Next, the subject pressed the direction key corresponding to the arrow shown in the center of the marker. The capture program stored 20 eye images captured approximately one second prior to the subject pressing the key. After the eye images were saved, the marker was moved to the next position automatically. This process was repeated 49 times to collect $49 \times 40 = 1960$ eye images.

After collecting all eye images, we manually annotated the pupil center position by one person for avoiding wrong categorization by multiple persons. We categorized the eye images into three classes: open, medium, and closed eyes. Each class is described as follows:

(i) An open eye image clearly shows the edge of the pupil, which makes it easy to estimate the pupil center position.

(ii) A medium eye image shows the eyelid overlaid on some part of the pupil, which makes it difficult to estimate the pupil position.

(iii) A closed eye image shows no pupil, which makes it impossible to estimate the pupil position.

Figure 3 shows sample eye images. Ten subjects (seven males (a)–(g); three females (h)–(j)) participated, and a total of 19,600 eye images were collected. All subjects were normally sighted and did not wear glasses. This dataset has 6,526 open eye images, 6,234 medium eye images, and 6,840 closed eye images.

The distribution of the pupil center position in our dataset is shown in Figure 4. The distributions of open, medium, and closed eye images are shown in Figures 4(b), 4(c), and 4(d), respectively. These distributions show that the number of image types is approximately equal for each section. Note that the pupil center positions were annotated manually. For medium and closed eye images, the exact pupil center position is unknown. We assume the pupil does not move during blinking; thus, we use the same annotation point from a previous open eye image frame, as shown in Figure 5, where the red dot shows the manually annotated ground truth. At frames one and two, the eye is open and easy to annotate. However, in frames three to five, the eye is in the medium or closed states; therefore, for such images, we used the ground truth from frame two.

4.2. Classification Evaluation. We evaluated the classification problem using leave-one-out cross-validation. We used a

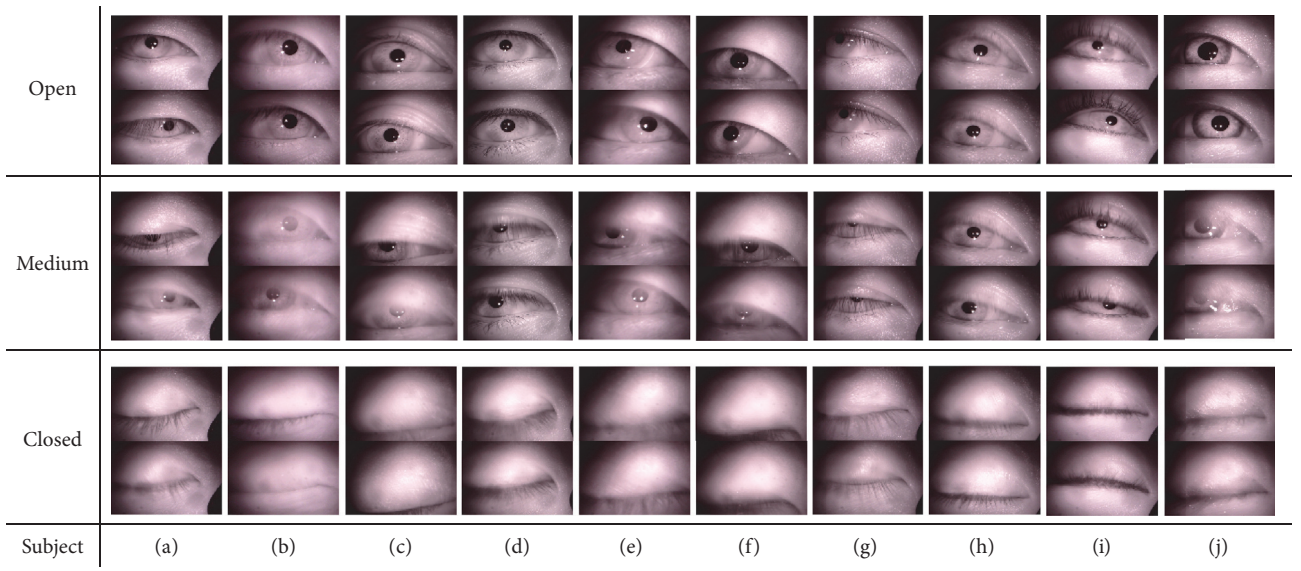


Figure 3: Sample eye images from our dataset.

Horizontal section

Vertical section

	1	2	3	4	5	6	7	8	9	10	11	12	Total
1	0	0	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	11	0	0	0	0	0	0	0	11
3	0	0	39	813	1016	933	750	296	19	0	0	0	3866
4	0	0	34	1419	2314	2985	1383	504	36	0	0	0	8675
5	0	0	22	706	1186	1546	1367	351	0	0	0	0	5178
6	0	0	32	554	324	446	427	82	0	0	0	2	1867
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2	2
total	0	0	127	3493	4851	5910	3927	1233	55	0	0	4	

(a) All images

Horizontal section

Vertical section

	1	2	3	4	5	6	7	8	9	10	11	12	Total
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	6	0	0	0	0	0	0	0	6
3	0	0	0	211	331	278	325	138	11	0	0	0	1294
4	0	0	12	311	707	982	515	208	16	0	0	0	2751
5	0	0	6	254	446	583	537	134	0	0	0	0	1960
6	0	0	21	161	102	100	114	17	0	0	0	0	515
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
total	0	0	39	937	1592	1943	1491	497	27	0	0	0	

(b) Open eye image

Horizontal section

Vertical section

	1	2	3	4	5	6	7	8	9	10	11	12	Total
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	2	0	0	0	0	0	0	0	2
3	0	0	19	257	300	289	161	75	5	0	0	0	1106
4	0	0	13	417	752	1034	406	158	7	0	0	0	2787
5	0	0	10	234	332	496	395	124	1	0	0	0	1592
6	0	0	1	212	116	195	184	39	0	0	0	0	747
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
total	0	0	43	1120	1502	2014	1146	396	13	0	0	0	

(c) Medium eye image

Horizontal section

Vertical section

	1	2	3	4	5	6	7	8	9	10	11	12	Total
1	0	0	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	3	0	0	0	0	0	0	0	3
3	0	0	20	345	385	366	264	83	3	0	0	0	1466
4	0	0	9	691	855	969	462	138	12	0	0	0	3136
5	0	0	6	218	408	467	435	93	0	0	0	0	1627
6	0	0	10	181	106	151	129	26	0	0	0	2	605
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2	2
total	0	0	45	1436	1757	1953	1290	340	15	0	0	4	

(d) Closed eye image

Figure 4: Distributions of our dataset.

Figure 5: Annotation of medium and closed eye image.

Table 2: Confusion matrices of CNN classification model.

(a) Classification result of method A

	Predict			
	Open	Medium	Closed	Accuracy
Actual				
Open	5465	1013	48	83.64%
Medium	784	4595	855	73.67%
Closed	0	707	6133	89.66%

(b) Classification result of method B

	Predict		
	Nonclosed	Closed	Accuracy
Actual			
Nonclosed	6596	244	96.43%
Closed	776	6064	88.65%

pretraining model trained using the ImageNet dataset [22] in order to avoid overfitting. The result from the pretraining model are better than without a pretraining model. The classification results of model A are shown in Table 2(a). The accuracy of this model was 82.58%. This result indicates that the accuracy of closed eye images is greater than that of the other classes. Some images for which classification failed are shown in Figure 6. The accuracy of the medium eye case (73.67%) is less than that of other classes because some of the medium eye images were difficult to classify, as shown in Figures 6(c) and 6(d). However, this level of accuracy is reasonable.

Next, we created a model to classify two classes for method B, which we refer to as classification model B. This model was designed to classify closed and nonclosed eye images. To train model B, we randomly selected nonclosed eye images from medium and open eye images to ensure that the number of nonclosed eye images was the same as closed eye images. The classification results of this model are shown in Table 2(b). The overall accuracy of this model was 92.54%, and the accuracy of nonclosed and closed eye images was 96.43% and 88.65%, respectively. This indicates that the classification accuracy of model B is better than that of model A. Classifying closed and nonclosed eye images is easier than doing so for the three classes of eye images because classification model B only classifies two classes, which improves accuracy compared to classification model A. However, all proposed classification models were designed to identify input images for which it is impossible to detect the pupil center position. Thus, both classification models can potentially identify closed eye images effectively.

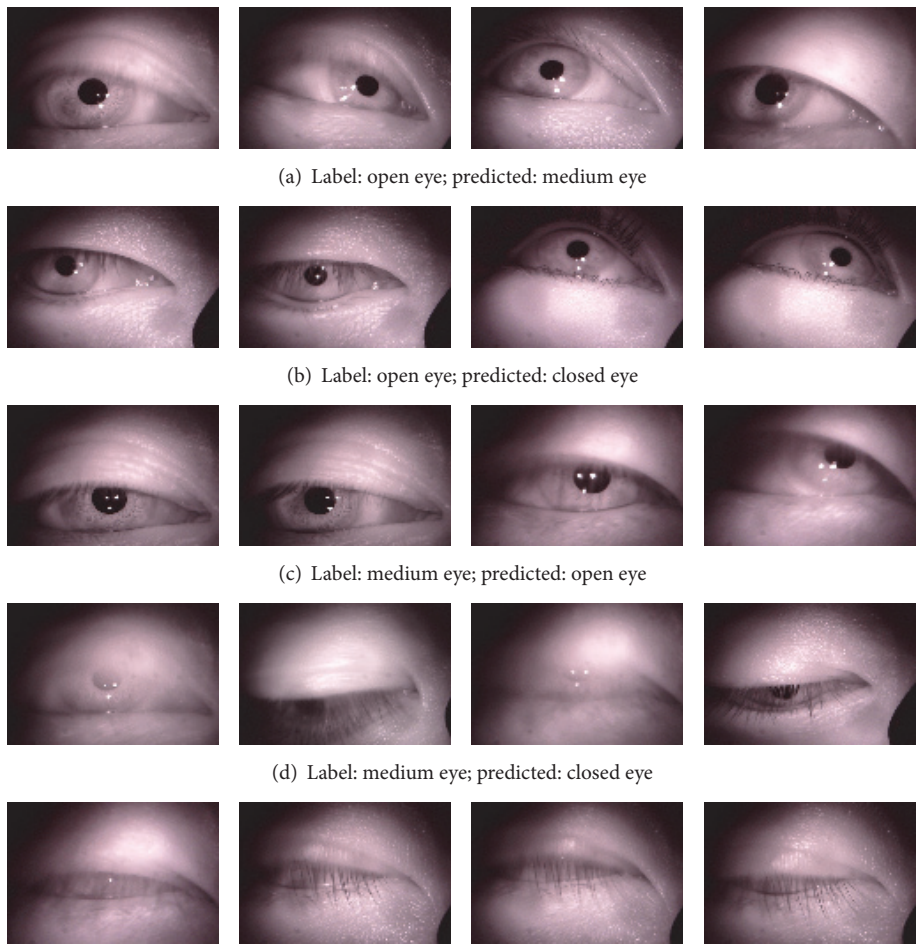
4.3. Regression Model Evaluation. We employed leave-one-out cross-validation to evaluate the regression model. As with the classification model, we used models pretrained using the ImageNet dataset [22] before training with our eye dataset. As discussed in Section 3, the input to the regression model is an eye image selected by the classification model. For the regression model, we had to train and evaluate the model using manually annotated eye images; we called the methods A* and B*. The regression model was trained using methods A* and B* before the regression model was integrated into the CNN classification model. Next, we evaluated the estimated point using an image from the classification model (methods A and B). Methods A and A* have two CNN regression models to estimate the pupil center position in the specific input image (open and medium eye images). The average errors are shown in Table 3.

Table 3: Confusion matrix of CNN classification model.

	Average error [pixel]			
Method	A	B	A*	B*
Open eye	0.79	—	0.80	—
Medium eye	2.19	—	1.21	—
Total	1.49	1.43	1.00	0.97

Methods A* and B* are the situation of classification model having a 100% accuracy. However, when we attempted to detect the pupil position in an image classified by the CNN classification model (methods A and B), the average error was somewhat high. Next, we compared the proposed method to a CNN with no classification model, which we refer to as the simple CNN. This model architecture is the same as the regression model of methods A and B. We trained this model using all eye images in the dataset. Figure 7 shows that the average errors of methods A and B are better than those of the regression model with no classification model. Moreover, we compared the proposed method to other well-known CNNs used in feature point detection research (Sun et al. [16]; Zhang et al. [23]). Sun et al. presented multiple CNN models to detect facial feature points. Zhang et al. presented Coarse-to-Fine Auto-Encoder Networks, which are used to detect multiple facial feature points. We trained the compared models under the same conditions as the simple CNN. The results show that the proposed simple CNN model obtained good accuracy compared to the other models.

Figures 8 and 9 show sample results for the estimated point obtained by method A. Here, the green point is the estimated pupil point, and the blue point is the ground truth



(a) Label: open eye; predicted: medium eye

(b) Label: open eye; predicted: closed eye

(c) Label: medium eye; predicted: open eye

(d) Label: medium eye; predicted: closed eye

(e) Label: closed eye; predicted: medium eye

Figure 6: Sample images from the failed classification model.

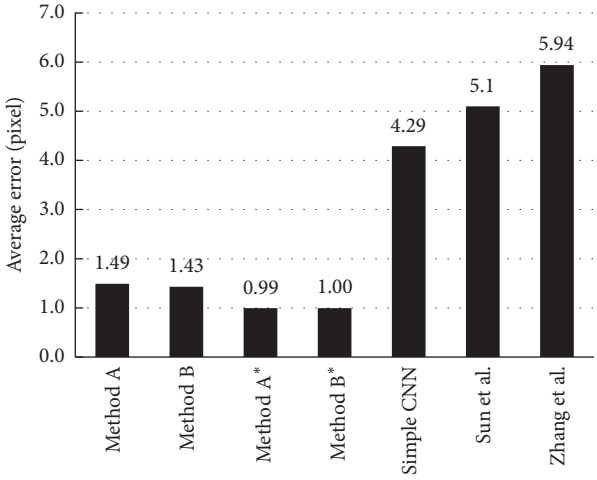


Figure 7: Average error of each CNN model.

from our dataset. As can be seen, these points are very accurate, and the estimated point nearly overlays the ground truth. However, for some difficult images in which the pupil is shown in the small part, the CNN generates more errors, as shown in Figure 10.

5. Discussion

We compared the proposed method to the simple CNN model. We also compared the different effects between method A and method B. Methods A* and B* represent methods A and B when the classification model achieves 100% accuracy. The results shown in Figure 7 indicate that the success rate of method A* is better than that of method B*. This result proves that when we allow the CNN model to learn a specific problem, the model can obtain better results than the single model. However, when we use an input image from the CNN classification, the success rate of method A is slightly less than that of method B because the classification accuracy of method B is better than that of method A. When we consider the difficulty of the classification problem, classifying nonclosed and closed eye images is easier than classifying eye states with three classes (i.e., open, medium, and closed). The single regression model (method B) was trained using both types of image (open and medium). Method B has robustness relative to classification error compared with method A.

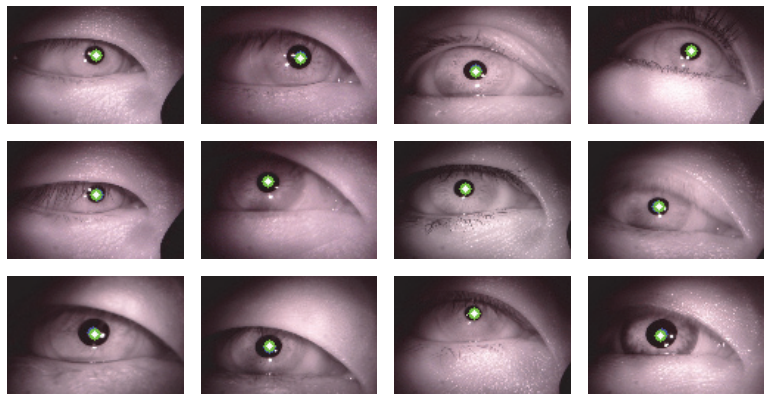


Figure 8: Success samples of open eye image.

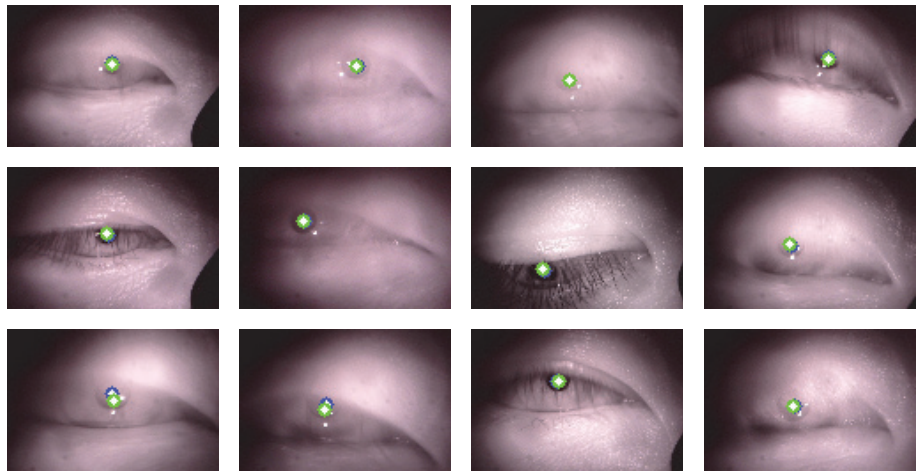


Figure 9: Success samples of medium eye image.

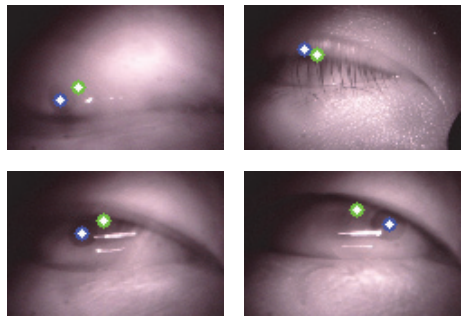


Figure 10: Failure samples.

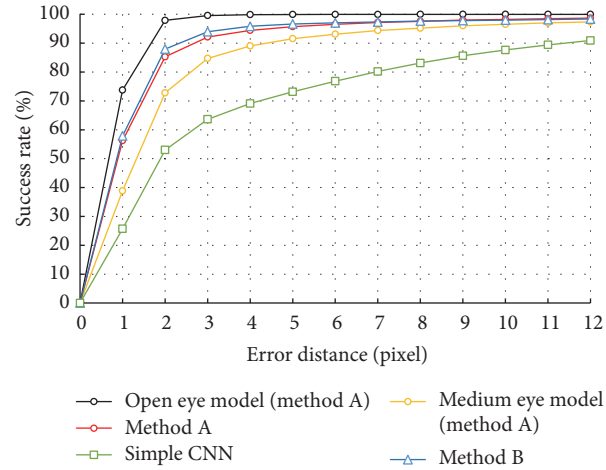


Figure 11: Performance curves.

However, the success rate of both models is better than that of the CNN model with no classification model (i.e., the simple CNN) and the compared models. Figure 11 shows the success rate of the proposed method. These results are the ratio of successful images compared to failed images when the distance between the ground truth and estimated point is less than the error distance. When the error distance is greater than four pixels, the success rate of methods A and B is greater than 90%. This shows that the proposed method has the potential for application in gaze estimation tasks.

6. Conclusion

This paper has presented methods to detect the pupil center position using a CNN model. We have focused on a wearable camera-based GES. When using a GES in daily life, it is

sometimes impossible to detect the pupil center position from an eye image; thus, this paper has considered avoiding this situation, for example, when blinking obscures the pupil. For supervised learning of the CNN, the dataset required specific features, that is, effective variety, appropriate distributions of image types, and sufficient amounts of data, to make the training process successful. Thus, we created a capture system to construct an original dataset. This original dataset provided closed, open, and medium eye images with good distribution. Using pretrained models, the dataset contained approximately 20,000 images, which is sufficient to train the CNN model effectively.

The proposed CNN method has two parts. The first is the CNN model, which is used to classify the eye state, and the other is the CNN regression model, which detects the pupil center position. The results show that the proposed CNN model has the potential to classify the eye state. Moreover, the accuracy of the pupil detection is better than that of the simple CNN model.

Conflicts of Interest

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

References

[1] H. Fujiyoshi, Y. Goto, and M. Kimura, "Inside-out camera for acquiring 3D gaze points," in *Proceedings of the in Proceedings of the Workshop on Egocentric (First-Person) Vision in conjunction with CVPR*, 2012.

[2] J. Iwagami and T. Saitoh, "Easy calibration for gaze estimation using inside-out camera," in *Proceedings of the in Proceedings of the 20th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV2014)*, pp. 292–297, 2014.

[3] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2235–2244, USA, June 2015.

[4] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2847–2854, USA, June 2012.

[5] A. Mazzei, S. Eivazi, Y. Marko, F. Kaplan, and P. Dillenbourg, "3D model-based gaze estimation in natural reading: A systematic error correction procedure based on annotated texts," in *Proceedings of the 8th Symposium on Eye Tracking Research and Applications, ETRA 2014*, pp. 87–90, USA, March 2014.

[6] A. Kiyohiko, N. Yasuhiro, O. Shoichi, and O. Minoru, "A support system for mouse operations using eye-gaze input," *IEEJ Transactions on Electronics, Information and Systems*, vol. 129, no. 9, pp. 11–1713, 2009.

[7] W. Chinsatit, M. Shibuya, K. Kawada, and T. Saitoh, "Character input system using gaze estimation," in *Proceedings of the in Proceedings of the International Conference on Communication Systems and Computing Application Science (CSCAS2016)*, 2016.

[8] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 4511–4520, Boston, Mass, USA, June 2015.

[9] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3D gaze estimation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1821–1828, IEEE, Columbus, Ohio, USA, June 2014.

[10] D. Li, D. Winfield, and D. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pp. 79-79, San Diego, Calif, USA.

[11] Z. Zheng, J. Yang, and L. Yang, "A robust method for eye features extraction on color image," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2252–2261, 2005.

[12] T. Moriyama, T. Kanade, J. Xiao, and J. F. Cohn, "Meticulously detailed eye region model and its application to analysis of facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 738–752, 2006.

[13] W. Chinsatit and T. Saitoh, "Eye detection by using gradient value for performance improvement of wearable gaze estimation system," IEICE Technical Report 115, no. 456, 2016, pp. 149-154.

[14] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci, "Pupilnet: Convolutional neural networks for robust pupil detection," *Computing Research Repository (CoRR)*, 2016, https://arxiv.org/abs/1601.04902.

[15] I.-H. Choi, S. K. Hong, and Y.-G. Kim, "Real-time categorization of driver's gaze zone using the deep learning techniques," in *Proceedings of the International Conference on Big Data and Smart Computing, BigComp 2016*, pp. 143–148, China, January 2016.

[16] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3476–3483, IEEE, Portland, Ore, USA, June 2013.

[17] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9003, pp. 538–552, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proceedings of the in Proceedings of the International Conference on Learning Representations (ICLR2014)*, 2014.

[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1717–1724, IEEE, Columbus, Ohio, USA, June 2014.

[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? - Weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 685–694, June 2015.

[22] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-Fine Auto-encoder Networks (CFAN) for real-time face alignment," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8690, no. 2, pp. 1–16, 2014.

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 5835020, 11 pages
https://doi.org/10.1155/2017/5835020

Research Article

Color Image Denoising Based on Guided Filter and Adaptive Wavelet Threshold

Xin Sun,[1] Ning He,[1] Yu-Qing Zhang,[2] Xue-Yan Zhen,[2] Ke Lu,[3] and Xiu-Ling Zhou[4]

[1]*Smart City College, Beijing Union University, Beijing 100101, China*
[2]*Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China*
[3]*University of Chinese Academy of Sciences, No. 19A Yuquan Road, Beijing 100049, China*
[4]*Beijing City University, Beijing 100083, China*

Correspondence should be addressed to Ning He; xxthening@buu.edu.cn

Received 29 May 2017; Revised 20 August 2017; Accepted 21 August 2017; Published 6 November 2017

Academic Editor: Ridha Ejbali

In the process of denoising color images, it is very important to enhance the edge and texture information of the images. Image quality can usually be improved by eliminating noise and enhancing contrast. Based on the adaptive wavelet threshold shrinkage algorithm and considering structural characteristics on the basis of color image denoising, this paper describes a method that further enhances the edge and texture details of the image using guided filtering. The use of guided filtering allows edge details that cannot be discriminated in grayscale images to be preserved. The noisy image is decomposed into low-frequency and high-frequency subbands using discrete wavelets, and the contraction function of threshold shrinkage is selected according to the energy in the vicinity of the wavelet coefficients. Finally, the edge and texture information of the denoised color image are enhanced by guided filtering. When the guiding image is the original noiseless image itself, the guided filter can be used as a smoothing operator for preserving edges, resulting in a better effect than bilateral filtering. The proposed method is compared with the adaptive wavelet threshold shrinkage denoising algorithm and the bilateral filtering algorithm. Experimental results show that the proposed method achieves superior color image denoising compared to these conventional techniques.

1. Introduction

During their acquisition and transmission, images are adversely affected by noise. Color images contain better visual effects than gray image in terms of visual perception, and the edge information of color images is more abundant than in gray images. Ideally, when removing the additive noise from an image, as many of the important features as possible should be retained. The denoising of color images often results in the loss of some edge and texture information, making the image blurred and creating a poor visual effect.

Denoising methods of color image commonly, Wiener filter and Gaussian filter denoising, have edge blurred situation. Bilateral filtering [1] is the most intuitive nonlinear smoothing filter, although it suffers from the gradient inversion effect, which uses a histogram-based approximation to calculate the weight, and it is computationally complex. Recently, Zhang et al. [2] develop an improved bilateral filter based framework which is capable of effectively removing universal noise. Bilateral filter takes spatial information and grayscale similarity into account and achieves both denoising and edge-preserving. Bilateral filter preserves too much high-frequency information but, however, does not denoise the high-frequency noise in color images. Thus, bilateral filter has better denoising effect for low-frequency noise merely. Sometimes, bilateral filter suffers from gradient reverse. The reason is that when an edge pixel has few similar pixels around it, the Gaussian weighted average is unstable. In this paper, proposed method based on local linear model has good edge-preserving smoothing properties like bilateral filter, but it does not suffer from the gradient reverse.

Wavelet threshold denoising [3–8] is a simple and effective denoising method. This technique effectively involves the decomposition of a signal into a set of independent, spatially oriented frequency channels. The discrete wavelet threshold [9] can be used to decompose the original image

into a sequence of images of different spatial resolutions. Dong and Ding [10] proposed a method that can always achieve better performance with lower computation cost and fewer decomposition scales than a high-frequency denoising method. Elyasi and Zermchi [11] proposed several adaptive wavelet denoising methods: Bayes Shrink, Modified Bayes Shrink, and Normal Shrink.

In recent years, many algorithms have improved and studied the wavelet threshold denoising. Like, Bhandari et al. [12] developed an optimized adaptive thresholding function which selects the appropriate threshold values to separate noise from the actual image and preserve edge details. In this paper, a new adaptive wavelet shrink denoising algorithm is proposed. Compared with other threshold algorithms, the proposed approach improves the denoising performance and has lower complexity than existing adjacent pixels methods [13, 14].

In the process of color image denoising, we compared the proposed method with the algorithms mentioned above (bilateral filter, Wiener filter, Gaussian filter, and wavelet threshold methods). Proposed method achieves superior color image denoising to these conventional algorithms. The reason is that classic algorithms could suppress the Gaussian noise effectively, but, at the same time, these methods fail to maintain the quality of denoised color images (like, texture) and may blur edges in the image. To address these short comings, this paper proposes a method based on image structure using adaptive wavelet threshold and guided filter to maintain edges when denoising. It makes edges continuous and the color of image more brightly. Because guided filter using a local linear model to enhance the image, the edge details remain. In particular, the details in color image, like texture, are more abundant and saturation is more greater.

On this basis, this paper presents a new color image denoising method based on the adaptive wavelet threshold shrinkage algorithm combined with image structure-based guided filtering [15]. The method uses the discrete wavelet transform to calculate the energy near the wavelet coefficients and then uses the adaptive threshold shrinkage function to denoise the image. The threshold function depends on the energy of adjacent pixels. Further using guided filter enhances the image after denoising. Experiments show that the proposed technique enables better preservation of edge information during the denoising process.

The rest of the paper is organized as follows. Section 2 reviews the related work. Algorithm analysis and the structure of proposed method are presented in Section 3. Then experimental results and analysis are shown in Section 4. Section 5 concludes the paper.

2. Related Work

Numerous works have been proposed for image denoising. In this part, we review previous and related work about wavelet threshold algorithms and guided filter.

2.1. Wavelet Threshold Shrinkage Algorithm. Wavelet threshold denoising is done by Donoho in 1994, which is based on thresholding the discrete wavelet transform (DWT) of the signal. Hard threshold and soft threshold are traditional threshold algorithm. Donoho [5] proposed wavelet soft threshold denoising and the threshold VisuShrink algorithm. This method for image denoising obtains a series of wavelet coefficients from the wavelet transform and applies a threshold to determine the smaller coefficients (which correspond to noise). Denoising and an inverse wavelet transformation yield the reconstructed image with reduced noise. Hard and soft threshold functions are defined as follows.

The hard threshold function is expressed in

$$Y = \begin{cases} X, & |X| \geq \lambda \\ 0, & |X| < \lambda. \end{cases} \tag{1}$$

The soft threshold function on the other hand is expressed in

$$Y = \begin{cases} \operatorname{sgn}(X)(|X| - \lambda), & |X| \geq \lambda \\ 0, & |X| < \lambda, \end{cases} \tag{2}$$

where λ is a threshold value, X is wavelet coefficients value after the DWT of images, and Y is output value using wavelet threshold shrinkage function.

Normally, hard threshold function can preserve the wavelet coefficients well generated by the useful information from images, but it is discontinuous at $|X| = \lambda$ after reconstruction. An alternative approach to hard threshold is the soft threshold, which has advantages of continuity. Soft threshold function is smooth and continuous relatively at the threshold. But sometimes, there are defects that decreased the wavelet coefficients generated by the effective signal.

An appropriate threshold λ is the most important role of discrete wavelet denoising. In the process of denoising, if the threshold λ is too small, the wavelet coefficients contain too many noise components and cannot denoise effectively. Otherwise, the threshold λ is particularly large resulting in the loss of useful components that causes distortion. Thus, new methods are proposed and some of them have been delivered to real applications.

Adaptive wavelet threshold method first assigns zeroes when the wavelet coefficients are smaller than the given threshold. As the threshold increases, the number of coefficients below the threshold will increase rapidly. When the number of nonzero coefficients reaches a certain value, the threshold is further enlarged and the number of nonzero values slowly decreases; this method can remove most of the noise and improve the compression efficiency. Nasri and Nezamabadi-pour [16] proposed a new thresholding function to be further used in a new subband-adaptive thresholding neural network to improve the efficiency of the denoising procedure. Liu et al. [17] found that a wavelet denoising using neighbor coefficients and level dependency was proposed to separate spikes from background noise.

2.2. Guided Filter. The guided filter is based on a dual integral image architecture VLSI [18] (Very Large Scale Integration). The filtering method computes the output image based on the input guiding image, which preserves the edges of the image well; this is an accurate and fast edge-preserving filtering

algorithm. He et al. [15] proposed a new explicit local linear guided filtering model, which is a fast and nonapproximate linear time algorithm. Their model senses the edges and enhances the texture detail. Gadge and Agrawal [19] used guided filtering to color images.

Guided filter of image is a linear transformable filtering process, where the guidance image I needs to be preset according to the specific application and I could be identical to the input image p. The output value at pixel i is calculated as follows:

$$q_i = \sum_j W_{ij}(I)\, p_j, \tag{3}$$

where i and j are pixel indexes. W_{ij} is filter kernel function, which defined in [15] expressed by

$$W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \varepsilon}\right), \tag{4}$$

where ω_k is the local window that center of k, $|\omega|$ is count of pixels in the local window, μ_k and σ_k^2 are mean and variance of guidance image I in the window, and, finally, ε is a smoothing factor.

A local linear model (3) is used in guided filter. This linear relationship ensures that output q_i has the same edge information with guidance image I. Thus, guided filter has a better edge-preserving smoothing and avoids the gradient reverse. In addition, its algorithm computed efficiently and nonapproximately.

3. Proposed Algorithm

In this paper, we describe an adaptive wavelet transform method to remove noise from a color image and use the inverse discrete wavelet transform to obtain the denoised image. The guided filter is then applied for edge and texture recovery and enhancement, producing a better color image effect.

3.1. Overview of Method Structure. The framework of proposed method contains two main stages (Figure 1). The first step is to obtain preliminary denoised image p using adaptive wavelet threshold shrinkage algorithm, which is based on image structure feature to shrinkage wavelet coefficients. The wavelet coefficients are decomposed by two-level discrete wavelet transform (DWT). The second step is further denoising and enhancing by using guided filter to the previous result p. In guided filter, the guidance image I should be preset. Setting I and p is identical and can perverse edge and texture of image.

In the ideal case, the wavelet threshold shrinkage algorithm subtracts Gaussian noise from the image, and its denoising effect is obvious. Natural image denoising using the wavelet threshold is very effective because it can capture the energy of the converted images.

Proposed denoising algorithm has the following steps in detail:

(1) Transform the noisy image into the frequency domain using DWT.

(2) Apply the adaptive wavelet threshold shrinkage algorithm to the local window on each subband and then use inverse DWT to obtain preliminary denoising image p.

(3) Apply guided filter on image p to obtain further denoise image q.

(4) Enhance q and output image.

3.2. Adaptive Wavelet Threshold Algorithm. The discrete wavelet transform (DWT) applied to image processing has two main components: decomposition and reconstruction. We use DWT to decompose the noisy image into a sequence of images of different spatial resolutions. Two-dimensional images can be decomposed in two-degree directions, resulting in different frequency bands: LL (Low-Frequency), LH (Horizontal High-Frequency), HL (Vertical High-Frequency), and HH (Diagonal High-Frequency).

In Figure 2, using DWT to decompose the noisy image into a sequence of images of different frequency bands. Low-Frequency (LL) is decomposed using DWT. And decomposing it produces four different frequency subbands (LL2, HL2, LH2, and HH2) using two-level wavelet decomposition function. On the basis of these frequency subbands, the different wavelet threshold shrinkage algorithm can be used to denoise from the image.

After the two-level wavelet decomposition of the image (Figure 2), an adaptive wavelet transform is used to extract the structure information in the multiresolution image, and the corresponding shrinkage function is applied to the structure features of the image. In fact, the natural image structure of the wavelet coefficients in the resolution scale exhibits a certain similarity, so there is a certain degree of redundancy in the wavelet decomposition scale. For example, the wavelet coefficients of the edge region are usually concentrated together, indicating that there is a certain degree of dependency in the adjacent wavelet coefficients corresponding to the edge region.

The structure information of the image can be obtained by calculating the energy of the local area in the wavelet domain. The smoother the image, the lower the energy. A threshold range is determined based on the local energy calculated by the wavelet decomposition, and a different function is used within the corresponding threshold. The specific algorithm takes the average of the square of each pixel value in the local window to calculate the energy of the center pixel of the window. The appropriate shrinkage factor α, β is then selected according to the corresponding shrink function (6).

In practice, we select the local window $R * R$ (i.e., $R = 5$) using (5) to calculate the local window center pixel energy value $S_{j,k}^2$ and then use (6) to calculate the local shrinkage function:

$$S_{j,k}^2 = \frac{1}{R^2} \sum_{m=-R}^{m=R} \sum_{n=-R}^{n=R} d_{m,n}^2, \tag{5}$$

$$\widehat{d}_{j,k} = \begin{cases} d_{j,k}\left(1 - \alpha * \dfrac{\lambda^2}{S_{j,k}^2}\right), & \text{if } S_{j,k}^2 \geq \beta * \lambda^2 \\ 0 & \text{else,} \end{cases} \tag{6}$$

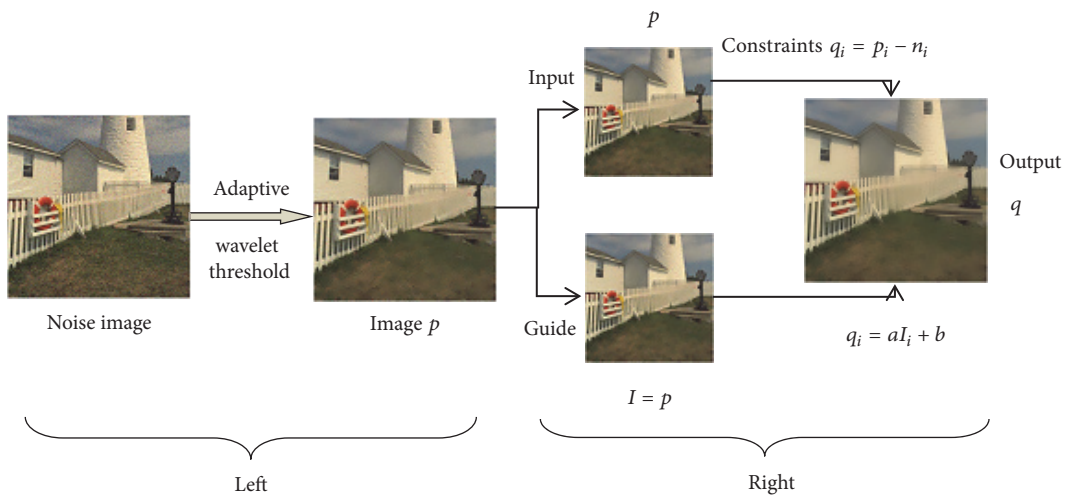


Figure 1: Illustration of the proposed method. It contains two main parts. Left part is denoising by adaptive wavelet threshold algorithm to obtain denoise image p. Right part is using guided filter to enhance edges of image p.

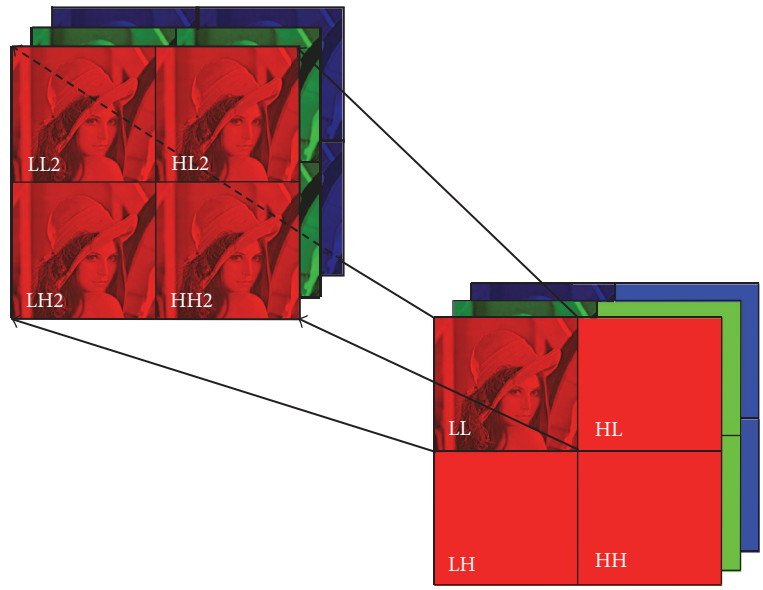


Figure 2: Two-level decomposition of the image. Obtaining Low-Frequency image (LL) from the noise image and decomposing it resulting in subbands LL2, HL2, LH2, and HH2. The adaptive wavelet shrink algorithm is applied to each subband to denoise the image.

$$\sigma^2 = \left[\frac{\text{Median}\left(\left|Y_{i,j}\right|\right)}{0.6745}\right] \quad \left(Y_{i,j} = \text{subband HH}\right), \tag{7}$$

where $\lambda^2 = (4\sigma^2 \log R)$. Equation (7) gives the noise variance σ^2, and $d_{j,k}$ is the central pixel of the local window. If $d_{j,k}$ is at the boundary of the second wavelet coefficients, a boundary condition is required. In the experiments, we set $\alpha = 0.1$, $\beta = 0.3$. Finally, the reconstructed image is denoised.

This denoising algorithm uses the DWT to calculate the energy near the wavelet coefficients and then applies the adaptive wavelet threshold shrink function to denoise the image. In the experiments, we added Gaussian noise with variances of $\sigma^2 = 0.01$ and $\sigma^2 = 0.03$ to the images. Adaptive wavelet threshold shrink function denoising noise and getting denoised image p. From the experimental results, it can be seen that when the noise variance is large, the image p is not very clear after denoising; in particular, the effect of texture is not obvious. Therefore, we need to enhance the image p after this processing step. Thus, the proposed technique uses the guided filter to enhance the image after denoising.

3.3. Guided Filter to Further Processing. Guided filtering is a spatial enhancement technique for the spatial domain, and the filtered output is a linear transformation of the localized image. The filtering algorithm uses a guiding image to process the edges of the noisy image. The guiding image can be the image itself. At this time, their structures are the same; that is, the edges of the original image are the same as the edges of the guiding image. The output pixel values take into account

the statistics of the local spatial neighborhood in the guided image. Hence, using guided filtering, the output image is more structured. This can be used for image dehazing and so on. The guided filter adopts an exact linear algorithm. The algorithm is efficient and fast and is considered to be one of the fastest edge-preserving filters.

For both grayscale and color images, the guided filtering algorithm has $o(N)$ time complexity, regardless of the local window radius (r).

3.3.1. Algorithm of Guided Filter

Guided Filter. The guide image can be a separate image or the original input image; when the guiding image is the original input image, the guided filter retains the edges for image reconstruction.

Assume that the input image is p, the output image is q, and the guiding image is I. q and I have a local linear relationship in the window ω_k centered on pixel k:

$$q_i = a_k I_i + b_k, \quad \forall i \in \omega_k, \tag{8}$$

where a_k, b_k are linear (constant) coefficients in the local window, and the window radius is r. Equation (8) calculates the guidance of the guiding image to be $\nabla q = a\nabla I$. Therefore, the linear equation is only guaranteed if I is in the presence of an edge, and the output image q will contain edges. To determine the coefficients, we require constraints from the input image p. The output q is the input p with a number of noise components n_i subtracted, that is,

$$q_i = p_i - n_i. \tag{9}$$

To minimize the difference between the output q and the input p and ensure the linearity of (8), the function $E(a_k, b_k)$ is minimized in the window ω_k, where ε is the regularization parameter:

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left(\left(a_k I_i + b_k - p_i \right)^2 + \varepsilon a_k^2 \right). \tag{10}$$

In order to minimize the value of the model $E(a_k, b_k)$, we can derive a_k and b_k, respectively, and (10) can be solved using linear regression:

$$\begin{aligned} a_k &= \frac{\left((1/|\omega|) \sum_{i \in \omega_k} I_i p_i \right) - \mu_k \overline{p}_k}{\sigma_k^2 + \varepsilon}, \\ b_k &= \overline{p}_k - a_k \mu_k, \end{aligned} \tag{11}$$

where μ_k and σ_k^2 represent the mean and variance in the local window ω_k, respectively; $|\omega|$ is the number of pixels in the window; and $\overline{p}_k$ represents the mean value in the window ω_k of p. After obtaining a_k and b_k, in Figure 3, the output q_i is

$$q_i = \frac{1}{|\omega|} \sum_{k | i \in \omega_k} \left(a_k I_i + b_k \right) = \overline{a}_i I_i + \overline{b}_i, \tag{12}$$

where $\overline{a}_i = (1/|\omega|) \sum_{k \in \omega_i} a_k$, $\overline{b}_i = (1/|\omega|) \sum_{k \in \omega_k} b_k$. Convert a_k and b_k into weights forms, which is the general form of filtering.

Figure 3: Illustration of (12). When the window ω is sliding in the image, a pixel is involved in many windows that covers pixel. For instance, windows ω_1 and ω_2 are different local windows that cover the same pixel i. Therefore, the output q_i should average all the values of pixel i in different windows.

Therefore, the guide filter algorithm proceeds as follows. Traverse the entire image via each local window, implementing the following calculation, where f is the mean filter with local window radius r; input image is p; guidance image is I; corr is the correlation coefficient; var is the variance; ε is smoothing factor; and cov is the covariance:

$$\begin{aligned} \text{mean}_I &= f_{\text{mean}}(I, r) \\ \text{mean}_p &= f_{\text{mean}}(p, r) \\ \text{corr}_I &= f_{\text{mean}}(I. * I, r) \\ \text{corr}_{Ip} &= f_{\text{mean}}(I. * p, r) \\ \text{var}_I &= \text{corr}_I - \text{mean}_I. * \text{mean}_I \\ \text{cov}_{Ip} &= \text{cov}_{Ip} - \text{mean}_I. * \text{mean}_p \\ a &= \frac{\text{cov}_{Ip}.}{(\text{var}_I + \varepsilon)} \\ b &= \text{mean}_p - a. * \text{mean}_I \\ \text{mean}_a &= f_{\text{mean}}(a, r) \\ \text{mean}_b &= f_{\text{mean}}(b, r) \\ q &= \text{mean}_a. * I + \text{mean}_b. \end{aligned} \tag{13}$$

3.3.2. Edge Preservation. When $I = p$, the guided filter becomes an edge-preserving filter. At this point, we have

$$\begin{aligned} a_k &= \frac{\sigma_k^2}{(\sigma_k^2 + \varepsilon)}, \\ b_k &= (1 - a_k) \mu_k. \end{aligned} \tag{14}$$

Therefore, when the local window variance is large, the center pixel value remains unchanged. In smoother areas, the average value of the neighboring pixels is used as the center pixel value.

3.3.3. Guided Filter for Color Image. When the guided filter is applied independently to the three color channels of the color image, (8) can be rewritten as

$$q_i = a_k^T I_i + b_k, \quad \forall i \in \omega_k, \tag{15}$$

where I_i is a 3×1 color vector, a_k is a 3×1 coefficient vector, and q_i, b_k are scalars. Thus, the color image of the guided filter is

$$\begin{aligned} a_k &= \left(\sum_k + \varepsilon U \right)^{-1} \left(\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \overline{p}_k \right), \\ b_k &= \overline{p}_k - a_k^T \mu_k, \\ q_i &= \overline{a}_i^T I_i + \overline{b}_i, \end{aligned} \tag{16}$$

where $\sum_k$ is the 3×3 covariance matrix of I in ω_k and U is the 3×3 identity matrix.

Because the local linear model is more effective in the color space, the edges of gray images cannot be identified but through the color image of the guided filter it can be well-preserved. Thus, the image edges have a significant effect.

3.4. Image Enhancement. For images with more texture (i.e., Image 5), the above denoising method may cause some regions to appear too smooth. Therefore, it is necessary to further enhance the image texture detail. Using (17), the denoised image q is subtracted from the original noiseless image to yield the details of the loss in q. Positive number c is to control and balance the degree of its stacking:

$$q\_\text{enhanced} = (I - q) * c + q. \tag{17}$$

The proposed algorithm uses the characteristics of the image structure. In the wavelet domain, threshold shrinkage is used to denoise the image, and then the guided filter enhances the edges of the denoised image to better reflect the texture details of the image.

4. Experimental Results and Analysis

We conducted a series of experiments using MATLAB R2015b and images in Figure 4. Images in Figure 4, Image 9, Image 10, Image 11, and Image 12, are rich in texture. First, variance of 0.01 and 0.03 Gaussian noise was added to the color images in Figure 4. This produced the noisy images shown in Figures 5(a), 6(a), 7(a), 8(a), 9(a), 10(a), 11(a), and 12(a). The proposed method based on image characteristics and the adaptive wavelet threshold shrinkage algorithm was then applied to obtain the denoised images p shown in Figures 5(c), 6(c), 7(c), 8(c), 9(c), 10(c), 11(c), and 12(c). We used the "sym4" two-level wavelet decomposition function to decompose the noisy images, with a local window $R = 5 * 5$, $\beta = 0.3$, and $\alpha = 0.1$ to control the degree of shrinkage in the wavelet coefficients.

The guided filter was used to denoise the image p with the guiding image I set to the original image without noise. This was intended to give the image a better edge effect after denoising. The local window radius was set to $r = 8$ and $\varepsilon = 0.02^2$. The image p was denoised after the adaptive wavelet threshold algorithm. In image p, the r, g, b channels were individually subjected to the guide filter. The texture and edge information of the images were enhanced and we obtained the output q. The parameter c in (17) was found to give better texture effects and undistorted images when $c = 1.5$. The final results q_enhanced images are presented in Figures 5(f), 6(f), 7(f), 8(f), 9(f), 10(f), 11(f), and 12(f). From Figures 9(f), 10(f), 11(f), and 12(f), it can be concluded that the texture and the edge of images had a good performance in proposed method.

The peak signal-to-noise ratio *(PSNR)* of the proposed method is presented in Table 1. These *PSNRs* indicate that the method proposed in this paper is superior to bilateral filtering, the adaptive wavelet threshold algorithm, nonlocal means [20] algorithm, and BM3D [21] method. When the noise level increases, the denoising effect of all five methods decreases, but the proposed method gives better performance than the other four.

The *PSNR* is calculated as follows:

$$\begin{aligned} \text{PSNR} &= 10 * \log_{10} \left(\frac{255^2}{\text{MSE}} \right), \\ \text{MSE} &= \frac{1}{3} \left(\text{mse}_r + \text{mse}_g + \text{mse}_b \right), \\ \text{mse}_k &= \frac{1}{m * n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(I(i,j) - q(i,j) \right)^2, \\ & \qquad k = r, g, b, \end{aligned} \tag{18}$$

where the input image is I and the output image is q_enhanced. The mean square error *(MSE)* is the average mean square error of the three channels (r, g, b) in the color image.

Wells [22] proposed a quality factor to evaluate the edges of an image after denoising. This quality factor (Pratt's figure of merit) takes into account three kinds of error: the loss of the effective edge, the edge of the positioning error, and noise misjudged as the edge. Pratt's figure of merit provides a quantitative evaluation of image edges and can be used to compare the edges in the denoised image and the original image. From Figures 13 and 14, it is obvious that there are differences between the texture parts. Pratt's figure of merit was measured in four images. The results in Figure 15 confirm that the edge quality factor (red curve) of the proposed method is higher than that of the other four methods for different degrees of noise variance. Thus, our method achieves better edge effects.

From a subjective point of view, we find that not only can the proposed method preserve edge better, it also reduces noise well when compared with the other four methods. From Figures 5–12, the result images (f) using proposed method look clearer than the others because we use a guided filter to filter the image and eliminate noise. Moreover, in Figures 13 and 14, the edge of the result image that uses the proposed method is smooth and coherent, which greatly helps to enhance edge.

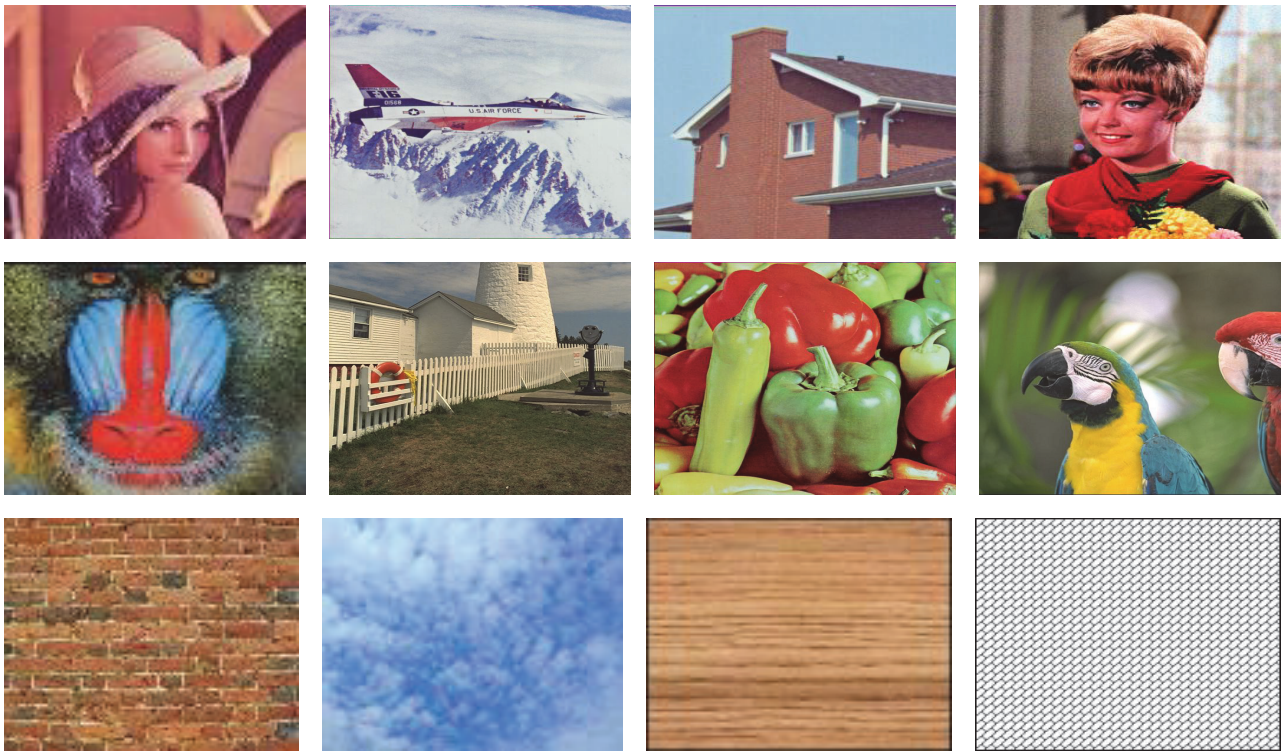
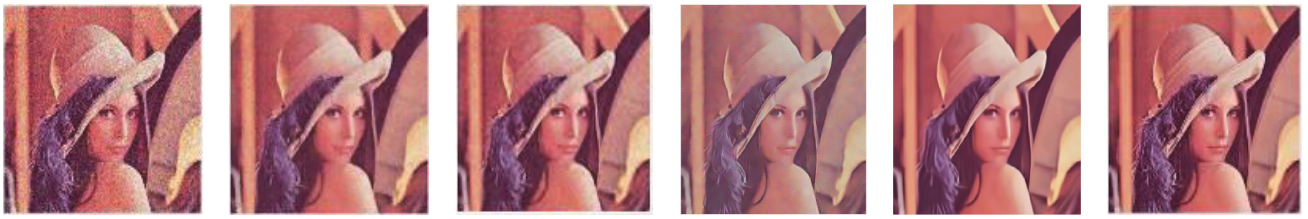


Figure 4: Test image: Images 1–12 (from left to right, top to bottom).



(a) $\sigma^2 = 0.01$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

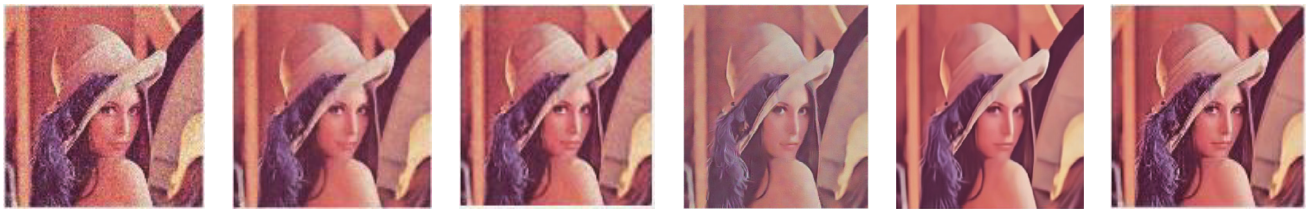
Figure 5: Image 1: comparison of various experimental results under noise variance $\sigma^2 = 0.01$.

To objectively evaluate the results, we use the Pratt's figure of merit and PSNR for image quality evaluation. Pratt's figure of merit results are listed in Figure 15. y-axis is the Pratt factor results and x-axis is noise variance. The curves showed that Pratt factor decreases when the noise variance increases, but proposed method (red curves) is much higher than others. Curves results suggested that edge of image using proposed method is effective and abundant. Table 1 is result of *PSNR*. The proposed method is superior to the other four methods with respect to image processing.

From the data in Table 1 and the quality factor curves in Figure 15, we can conclude that the proposed method is superior to bilateral filtering, the adaptive wavelet threshold denoising algorithm, nonlocal means algorithm, and BM3D algorithm on color image denoising. From a visual point of view, the proposed method not only gives a good denoising effect but also achieves better edge retention.

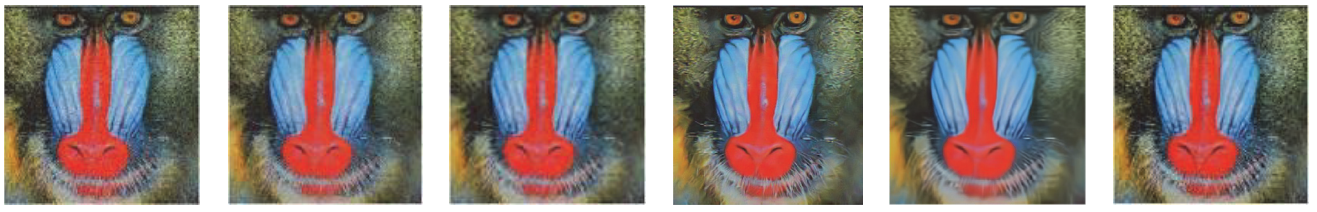
5. Conclusion

In this paper, a new denoising method based on the adaptive wavelet threshold denoising algorithm and edge-guided filtering has been proposed. The image denoising is performed according to the local structure of the image. In comparative experiments against bilateral filtering, the adaptive wavelet denoising method, nonlocal means algorithm, and BM3D algorithm, the proposed method exhibited the best denoising and edge preservation performance. The proposed approach removes Gaussian noise in the frequency domain and then uses linear guided filtering to further enhance the image recovery, resulting in better denoising and edge effects. As color images display more detailed textures, this filtering overcomes the gradient inversion effect in the edge regions. As the linear model (see (8)) is a block-unsupervised learning method, it can be combined with other models to obtain new



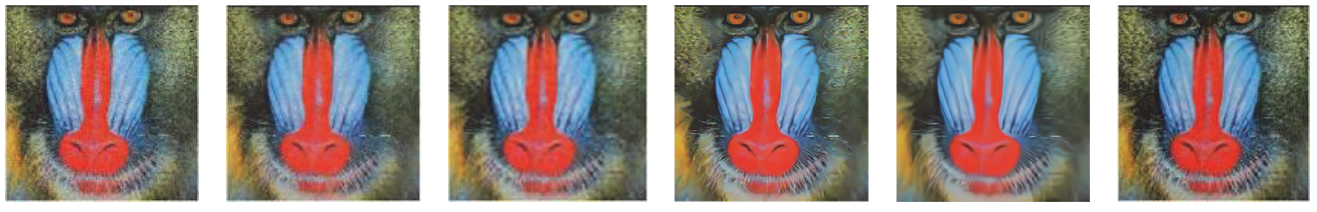
(a) $\sigma^2 = 0.03$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 6: Image 1: comparison of various experimental results under noise variance $\sigma^2 = 0.03$.



(a) $\sigma^2 = 0.01$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 7: Image 5: comparison of various experimental results under noise variance $\sigma^2 = 0.01$.



(a) $\sigma^2 = 0.03$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 8: Image 5: comparison of various experimental results under noise variance $\sigma^2 = 0.03$.

(a) $\sigma^2 = 0.01$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

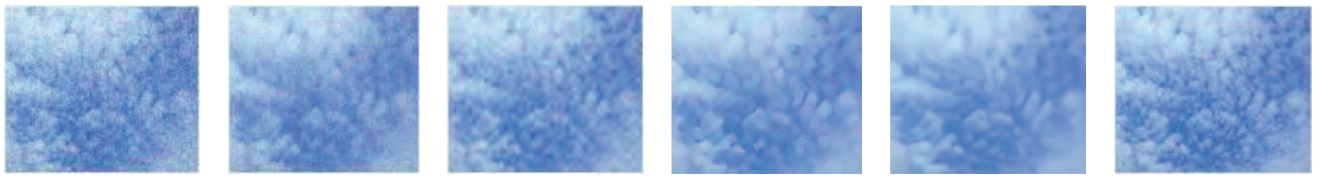
Figure 9: Image 9: comparison of various experimental results under noise variance $\sigma^2 = 0.01$.

(a) $\sigma^2 = 0.03$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 10: Image 9: comparison of various experimental results under noise variance $\sigma^2 = 0.03$.

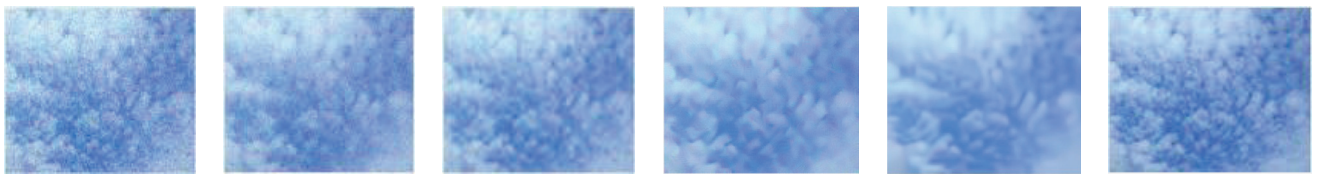
Table 1: PSNR of the proposed method, bilateral filtering, and adaptive wavelet.

Noise variance	0.01	0.03	0.01	0.03	0.01	0.03
	Image 1 (256 ∗ 256)		Image 2 (512 ∗ 512)		Image 3 (256 ∗ 256)	
Adaptive wavelet	26.7048	25.4507	28.0801	26.4821	27.7967	26.1699
Bilateral filtering	22.5751	22.0167	23.1551	22.6118	22.7560	22.1802
Nonlocal means	29.9765	27.4871	30.1415	27.6777	30.1910	27.6823
BM3D	28.9235	26.9276	30.4215	27.8901	30.9457	28.0957
Proposed method	**39.5634**	**35.3129**	**41.5647**	**36.0267**	**40.8101**	**35.5930**
	Image 4 (256 ∗ 256)		Image 5 (256 ∗ 256)		Image 6 (512 ∗ 512)	
Adaptive wavelet	25.2375	24.2724	23.4663	22.7332	25.6576	24.5766
Bilateral filtering	22.6074	21.9415	22.1076	21.5494	22.3016	21.7254
Nonlocal means	26.6512	25.3287	25.3588	24.3120	27.8813	26.2105
BM3D	25.5157	24.4968	24.0394	23.2556	28.0621	26.3334
Proposed method	**37.0265**	**34.1344**	**33.8535**	**32.1364**	**36.8294**	**33.9588**
	Image 7 (512 ∗ 512)		Image 8 (512 ∗ 512)		Image 9 (200 ∗ 150)	
Adaptive wavelet	27.5706	25.9130	29.0039	26.9908	21.4746	20.8466
Bilateral filtering	23.2116	22.4080	23.1373	22.4268	21.4834	21.0023
Nonlocal means	29.4155	27.0736	31.3181	28.2534	23.7273	23.0137
BM3D	29.1443	26.9120	32.0590	28.6197	21.7645	21.2712
Proposed method	**39.1414**	**34.8926**	**41.2256**	**35.7943**	**31.4041**	**30.0693**
	Image 10 (200 ∗ 150)		Image 11 (187 ∗ 171)		Image 12 (187 ∗ 171)	
Adaptive wavelet	29.4247	27.5268	26.7443	25.4201	15.9075	16.9781
Bilateral filtering	23.6728	23.0581	22.2984	21.7973	22.7751	22.6673
Nonlocal means	30.3106	27.9794	28.7439	26.6227	17.4105	17.2673
BM3D	30.8146	28.3780	31.8744	28.4922	31.0366	30.1990
Proposed method	**40.2191**	**35.9371**	**38.7096**	**35.0295**	**24.4256**	**25.7756**



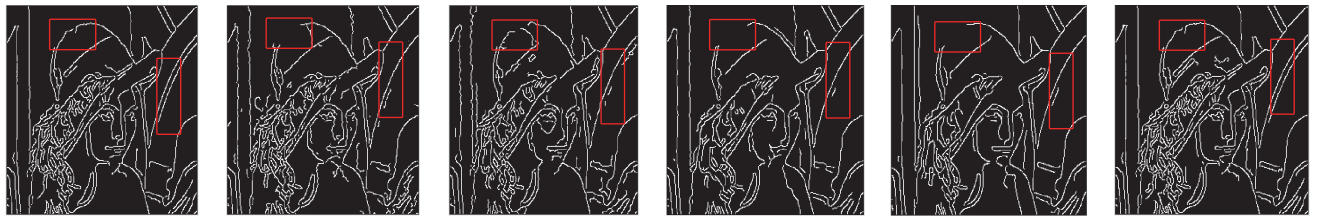
(a) $\sigma^2 = 0.01$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 11: Image 10: comparison of various experimental results under noise variance $\sigma^2 = 0.01$.



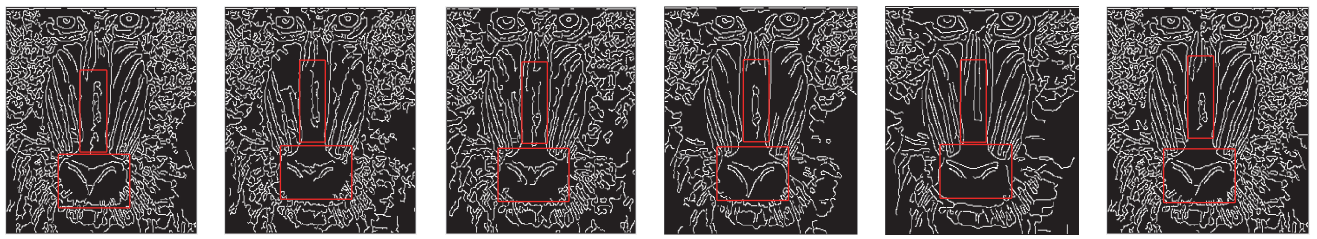
(a) $\sigma^2 = 0.03$ (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 12: Image 10: comparison of various experimental results under noise variance $\sigma^2 = 0.03$.



(a) Image 1 edge (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 13: Image 1: edge comparison of various experimental results under noise variance $\sigma^2 = 0.03$.



(a) Image 5 edge (b) Bilateral filtering (c) Adaptive wavelet (d) Nonlocal means (e) BM3D (f) Proposed method

Figure 14: Image 5: edge comparison of various experimental results under noise variance $\sigma^2 = 0.01$.

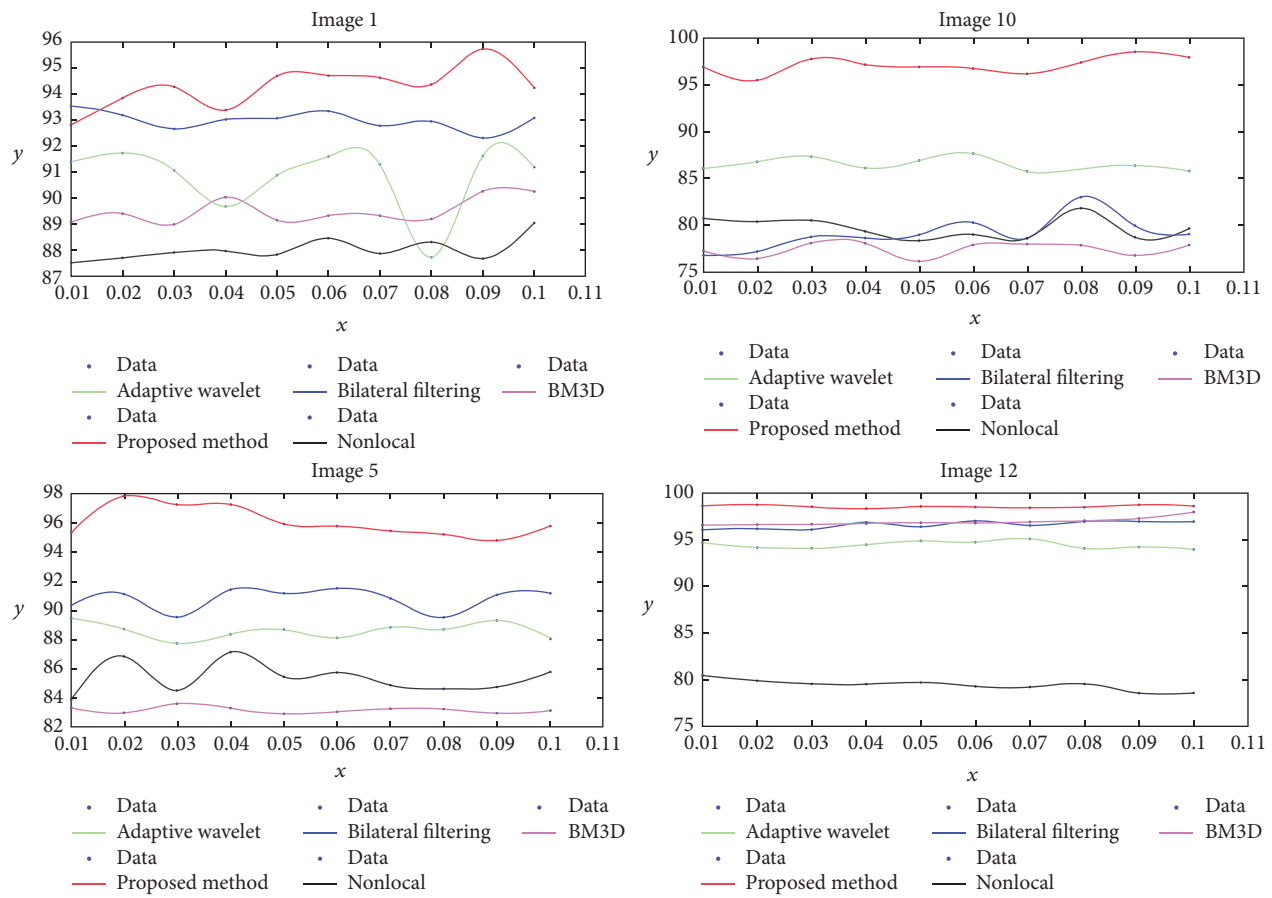


Figure 15: Pratt's figure of merit.

denoising techniques. This will be the focus of future research and exploration on color image denoising.

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 (nos. 61370138, 61572077, and U1301251) and the Beijing Municipal Natural Science Foundation (no. 4162027); the Project of Oriented Characteristic Disciplines (no. KYDE40201701); the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality (no. IDHT20170511).

References

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the International Conference on Computer Vision IEEE Computer Society*, 839 pages, 1998.

[2] Y. Zhang, X. Tian, and P. Ren, "An adaptive bilateral filter based framework for image denoising," *Neurocomputing*, vol. 140, pp. 299–316, 2014.

[3] J. S. Lim, *Two-dimensional signal and image processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[4] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[5] D. L. Donoho, "De-noising by soft-thresholding," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[6] T. D. Bui and G. Chen, "Translation-invariant denoising using multiwavelets," *IEEE Transactions on Signal Processing*, vol. 46, no. 12, pp. 3414–3420, 1998.

[7] C. Srisailam, P. Sharma, and S. Suhane, "Color image denoising using wavelet soft thresholding," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 7, pp. 474–478, 2014.

[8] K. K. Gupta and R. Gupta, "Feature adaptive wavelet shrinkage for image denoising," in *Proceedings of the 2007 International Conference on Signal Processing, Communications and Networking, ICSCN 2007*, pp. 81–85, ind, February 2007.

[9] E. Ordentlich, G. Seroussi, S. Verdu, M. Weinberger, and T. Weissman, "A discrete universal denoiser and its application to binary images," in *Proceedings of the International Conference on Image Processing (ICIP 2003)*, vol. 1, pp. 20–117, 2003.

[10] W. Dong and H. Ding, "Full frequency de-noising method based on wavelet decomposition and noise-type detection," *Neurocomputing*, vol. 214, pp. 902–909, 2016.

[11] I. Elyasi and S. Zermchi, "Elimination noise by adaptive wavelet threshold," *World Academy of Science Engineering & Technology*, vol. 3, no. 8, pp. 1541–1545, 2009.

[12] A. K. Bhandari, D. Kumar, A. Kumar, and G. K. Singh, "Optimal sub-band adaptive thresholding based edge preserved satellite image denoising using adaptive differential evolution algorithm," *Neurocomputing*, vol. 174, pp. 698–721, 2016.

[13] T. T. Cai and B. W. Silverman, "Incorporating information on neighbouring coefficients into wavelet estimation, Sankhyā," *The Indian Journal of Statistics, Series B (1960-2002)*, vol. 63, no. 2, pp. 127–148, 2001.

[14] G. Y. Chen, T. D. Bui, and A. Krzyzak, "Image denoising using neighbouring wavelet coefficients," *Integrated Computer-Aided Engineering*, vol. 12, no. 1, pp. 99–107, 2005.

[15] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[16] M. Nasri and H. Nezamabadi-pour, "Image denoising in the wavelet domain using a new adaptive thresholding function," *Neurocomputing*, vol. 72, no. 4–6, pp. 1012–1025, 2009.

[17] X. Liu, H. Wan, Z. Shang, and L. Shi, "Automatic extracellular spike denoising using wavelet neighbor coefficients and level dependency," *Neurocomputing*, vol. 149, pp. 1407–1414, 2015.

[18] C.-C. Kao, J.-H. Lai, and S.-Y. Chien, "VLSI architecture design of guided filter for 30 frames/s full-HD Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 513–524, 2014.

[19] A. Gadge and S. S. Agrawal, "Guided filter for color image," *IJIREEICE*, vol. 4, no. 6, pp. 250–252, 2016.

[20] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 60–65, June 2005.

[21] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[22] P. N. T. Wells, *Digital Image Processing*, W. K. Pratt John Wiley, Chichester, UK, 1991, (Medical Engineering & Physics, vol. 16 no. 1, p. 83, 1994).

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 2520301, 9 pages
https://doi.org/10.1155/2017/2520301

Research Article

A Regular k-Shrinkage Thresholding Operator for the Removal of Mixed Gaussian-Impulse Noise

Han Pan, Zhongliang Jing, Lingfeng Qiao, and Minzhe Li

School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, China

Correspondence should be addressed to Han Pan; hanpan@sjtu.edu.cn

Received 5 April 2017; Accepted 12 June 2017; Published 12 July 2017

Academic Editor: Ridha Ejbali

The removal of mixed Gaussian-impulse noise plays an important role in many areas, such as remote sensing. However, traditional methods may be unaware of promoting the degree of the sparsity adaptively after decomposing into low rank component and sparse component. In this paper, a new problem formulation with regular spectral k-support norm and regular k-support ℓ_1 norm is proposed. A unified framework is developed to capture the intrinsic sparsity structure of all two components. To address the resulting problem, an efficient minimization scheme within the framework of accelerated proximal gradient is proposed. This scheme is achieved by alternating regular k-shrinkage thresholding operator. Experimental comparison with the other state-of-the-art methods demonstrates the efficacy of the proposed method.

1. Introduction

Image restoration [1–4] attempts to recover a clear image from the observations of real scenes. As a fundamental procedure, it has been applied to various application areas, such as image fusion [5] and action recognition [6]. However, typically, the noise characteristics of imaging camera is completely or partially unknown. Among these, the removal of mixed noise has not been investigated because the noise model is not easy to establish accurately.

Recently, a patch based method [7] for video restoration has attracted much attention [8–10]. This method also is extended to video in-painting for archived films. However, the mechanisms of modeling the sparsity level of the grouping patches remain unclear.

To deal with the lack of adaptivity in sparsity level [7], a robust video restoration algorithm is proposed. The main idea of the proposed method is to model the sparsity levels of the low rank component by regular spectral k-support norm and sparse component by regular k-support ℓ_1 norm. Specially, a new problem formulation is presented, where the objective function is minimized under an upper bound constraint on the regularization term. However, it is not easy to solve the resulting problem. Some recent progress [11] in the theory of optimization on iterative shrinkage thresholding method is considered. And, an efficient alternating minimization scheme is proposed to solve the new objective.

1.1. Related Works. Recently, the problem of denoising image corrupted by mixed Gaussian-impulse noise has been studied in many different contexts [8–10, 12, 13]. These methods fall into three categories: variational methods [9, 12], sparse representation [8, 10], and patch based method [7].

Variational methods are a new class of the solutions to promote edge-preservation, such as total variation [14]. These methods first utilized some spatial filters to detect and remove the corrupted pixels, for example, adaptive center-weighted median filter [15] (ACWMF) or rank order absolute differences [16] (ROAD) detector. In [12], Cai et al. employed Mumford-Shah regularization term to encourage sparsity in gradient domain. In [9], Rodríguez et al. presented a novel optimization method for the generalized total variation regularization method. It can be seen that the denoised performance of these methods relies on the detection for the damaged candidates. The adaptivity of sparsity level of the regularization terms has not been investigated carefully.

Sparse representation based methods have been extended to this problem. In the main idea of this scheme, it is assumed

that the signal can be described by linear combination of a spare number of elements or atoms of an overcomplete dictionary. In [8], an efficient image reconstruction method by posing ℓ_1 norm on the error, and ℓ_0 norm on image patches in learned dictionary, was proposed. In [10], Filipovic and Jukic reformulated a new problem formulation by enforcing ℓ_0-ℓ_1 sparsity constraints. The resulting problem is solved by a mixed soft-hard thresholding method. However, it should be noted that these methods are time-consuming.

Patch based method is proven to be a state-of-the-art denoising scheme. In [7], Ji et al. approximated the patch stack by reformulating the problem as a low rank matrix completion problem. Despite its efficacy, one of the limitations of patch based method in [7] is that the degree of sparsity has not been considered carefully. When the underlying sparsity level is unknown, we may obtain a bias estimate, considerably. To alleviate these issues in a unified formulation, a new problem formulation is proposed.

1.2. Contributions. The main idea of this paper is to deal with the weakness of the approach in [7]. Existing methods, such as ℓ_1 norm and trace norm, can not promote the sparsity level of all two components adaptively. The details or local fine content can not be represented and described well. To deal with these issues, a new problem formulation incorporating correlated and adaptive sparsity is proposed.

Our contributions can be summarized as follows.

(1) A new problem formulation to model the sparsity level of the patches is proposed. A new norm extended from k-support norm and ordered ℓ_1 norm is presented.

(2) An efficient minimization scheme with regular k-shrinkage thresholding operator is proposed, which is based on the optimization framework of accelerated proximal gradient (APG) method.

(3) Numerical experiments, compared to other state-of-the-art methods, demonstrate that the proposed method outperforms the related restoration methods.

1.3. Organization. The remainder of this paper is presented as follows. In Section 2, some basic notations are provided. In Section 3, a detailed description about the proposed objective function is given. In Section 4, an efficient minimization scheme within the framework of APG is proposed. Then, some experiments are conducted to validate the effectiveness of the proposed method in Section 5. Finally, we conclude the paper in Section 6.

2. Preliminaries

There are some notations presented for the simplicity of discussions. Frobenius norm and ℓ_1 norm of a matrix $X \in \mathbb{R}^{m\times n}$ are defined by $\|X\|_F$ and $\|X\|_1$, respectively. For a scalar τ, the shrinkage operator [17] $S_\tau(x)$ for ℓ_1 norm minimization problem is defined as follows:

$$S_\tau(x) = \operatorname{sgn}(x)\max(|x| - \tau, 0), \tag{1}$$

where sgn is a signum function; $|\cdot|$ calculates the absolute value.

Assuming that X is of rank r, the singular value decomposition (SVD) of X with nonnegative singular values is defined by $X = U\Sigma V^T$, where Σ denotes a diagonal matrix with the singular values. Based on the SVD computation, the nuclear norm is defined in the following way:

$$\|X\|_* = \operatorname{tr}\left(\sqrt{X^tX}\right) = \operatorname{tr}\left(\sqrt{V\Sigma^2V^T}\right) = \sum_{i=1}^{r}|\sigma_i|, \tag{2}$$

where σ_i is the ith largest singular value of X. A solution with shrinkage operator to the nuclear norm is singular value shrinkage operator [18] $D_\tau(X)$, which can be expressed as follows:

$$D_\tau(X) = U\Sigma_\tau V^T, \tag{3}$$

where $\Sigma_\tau : \mathbb{R} \to \mathbb{R}$ is defined as follows:

$$\Sigma_\tau = \operatorname{diag}\left(\operatorname{sgn}(\sigma_i)\max\left(|\sigma_i| - \tau, 0\right)\right). \tag{4}$$

However, it should be noted that the shrinkage operator for ℓ_1 norm is different from singular value shrinkage operator for nuclear norm. There operators play an important role in joint sparse and low rank matrix approximation.

In this paper, ordered ℓ_1-norm [19] is provided as follows:

$$\ell_{O1} = \sum_{i=1}^{n} w|x|_i, \tag{5}$$

where x is sorted in decreasing order. $|x|_i$ denotes the ith largest element of the magnitude vector $|x| = (|x_1|, \ldots, |x_n|)^T$. w is a trade-off vector in nonincreasing order. When w is a constant vector, (5) reduces to ℓ_1 norm. When $w_1 > 0$ and $w_{2\le i\le n} = 0$, then (5) reduces to ℓ_∞-norm.

k-support norm [20] is defined as follows:

$$k_s(x) = \left\{x \in \mathbb{R}^n : \|x\|_0 \le S,\ \|x\|_2 \le 1\right\}, \tag{6}$$

where S denotes the bound of the sparsity level of x, which is a positive integer. The details of k-support norm are in traduced in [20, 21]. It has been extended to the case of matrix, which named by regular spectral k-support norm. Although this norm provides the number of elements of the sparsity level, it lacks of efficient mechanism to promote the sparsity adaptively.

After taking advantage of both ordered ℓ_1 norm and k-support norm, regular k-support ℓ_1 norm is defined as follows:

$$\lambda\|x\|_{1,k} = \sum_{i=1}^{k}\lambda_i|x|_i, \tag{7}$$

where λ is a positive regularization vector in nonincreasing order. And $1 \le k \le n$, where n is the size of vector x.

For the case of matrix, regular spectral k-support norm is proposed, which can be expressed as follows:

$$\lambda\|X\|_k^* = \sum_{i=1}^{k}\lambda_i|\sigma_i|, \quad X \in \mathbb{R}^{m\times n}, \tag{8}$$

where $1 \le k \le \min(m, n)$. It can be noted that the singular values are arranged in nonincreasing order.

3. Problem Setup

This section introduces the objective function in detail. For each reference patch p, similar patches in the spatial- and temporal-domain are obtained by utilizing the patch matching algorithm. The matched patches are denoted as $\{p_i\}_{i=1}^{m}$, where m stands for the size. It can be noted that each patch p_i is rearranged as a vector with size $\mathbb{R}^{n^2}$ through concatenating all columns into a column vector. At last, a matrix $O \in n^2 \times m$ is generated after considering all the m patches, which can be represented as follows:

$$O = (p_1, p_2, \ldots, p_m). \tag{9}$$

In this paper, we assumed that the observed patch matrix O can be decomposed into three components:

$$O = L + S + N, \tag{10}$$

where L stands for low rank component, S is sparse component, and N is additive noise. There are some regularization methods for (10), such as L with nuclear norm (also known as the trace norm) and S with ℓ_1 norm. The problem formulation in [7] can be expressed as follows:

$$\begin{aligned} \min_{L,S} \quad & \|L\|_* + \lambda \|S\|_1, \\ \text{s.t.} \quad & \|O - L - S\|_F \le \epsilon, \end{aligned} \tag{11}$$

where $\|\cdot\|_*$ is nuclear norm, and $\|\cdot\|_1$ for ℓ_1 norm.

However, these norms may lead to a large estimation bias [19] but can not promote the sparsity level adaptively. For example, the limitations of ℓ_1 norm have been investigated in [22, 23]. Similarly, some alternative cases, such as the ℓ_p quasinorm, also have been discussed in [24–26]. Thus, a suitable solution is required to recover these components.

To alleviate these limitations, a new problem formulation is proposed to model the sparsity levels both on L and S. Moreover, a unified formulation to describe the correlated variables is considered. To estimate the underlying structures of L and S, we focus on the following minimization function:

$$\begin{aligned} \min_{L,S} \quad & \lambda_1 \|L\|_k^* + \lambda_2 \|S\|_{1,k}, \\ \text{s.t.} \quad & \|O - L - S\|_F \le \epsilon, \end{aligned} \tag{12}$$

where λ_1 and λ_2 are two positive regularization vectors in a nondecreasing order. $\lambda_1\|L\|_k^*$ stands for regular spectral k-support norm on L. $\lambda_2\|S\|_{1,k}$ denotes regular k-support ℓ_1 norm on S. And ϵ is the standard deviation of noise N.

The above formulation amounts to the constraint $\|O - L - S\|_F \le \epsilon$, which is considered more natural than usual formulation because it stands for the tolerance on the error.

After choosing a suitable ϵ, (12) can be reformulated as follows:

$$\min_{L,S} \quad \lambda_1 \|L\|_k^* + \lambda_2 \|S\|_{1,k} + \frac{1}{2\mu} \|O - L - S\|_F^2, \tag{13}$$

where μ is a suitable positive value. It can be seen that these are some challenges to solve (13). First, regular k-support ℓ_1-norm is posed on sparse component S. Second, the low rank component L is penalized by regular spectral k-support norm.

There are several properties of our problem formulation in (13). First, the proposed regular spectral k-support norm on L and regular k-support ℓ_1 norm on S aim to reconstruct the local structures clearly. It should be noted that these modeling strategies can adaptively promote the sparsity level with an upper bound. Second, to the best of our knowledge, this is the first time of combining the advantages of both ordered ℓ_1 norm and k-support norm to yield a robust subspaces estimation against noise. Third, although the optimization method in [27] is very similar to the proposed method, the proposed method can deal with more complex situations. Moreover, the proposed method can adopt the more challenging situations, such as the removal of mixed Gaussian, salt-and-pepper noise, and random value impulse noise. It should be noted that this noisy situation has not been explored in [7].

4. Proposed Method

4.1. Proposed Framework. In this section, an optimization framework using regular k-shrinkage thresholding operator is presented. First, accelerated proximal gradient method (APG) is applied to the resulting problem because of its simplicity and popularity in imaging applications [28, 29]. Second, the proposed regular k-shrinkage thresholding operator is applied to the two resulting subproblems. As showed in [22, 23], nonconvex regularization functions have been shown both theoretically and experimentally to provide better results than ℓ_1 norm. Then, some explicit proximal mappings are developed.

APG based scheme aims to solve an unconstrained minimization problem by

$$\min_{X} \quad g(X) + f(X), \tag{14}$$

where g is assumed to be a nonsmooth function and f for a smooth function. Here, L_f denotes the Lipschitz constant of the gradient of f.

Applying the framework of APG to problem (13), we have the following expressions:

$$\begin{aligned} X &= (S, L), \\ g(X) &= \mu\lambda_1 \|L\|_k^* + \mu\lambda_2 \|S\|_{1,k}, \\ f(X) &= \frac{1}{2} \|O - L - S\|_F^2. \end{aligned} \tag{15}$$

Require: $L_0 = L_{-1} = 0; S_0 = S_{-1} = 0; t_0 = t_{-1} = 1;$
$\mu_0 > \overline{\mu} > 0, \rho < 1;$
repeat
$Y_k^L = L_k + \frac{t_{k-1} - 1}{t_k}(L_k - L_{k-1});$
$Y_k^S = S_k + \frac{t_{k-1} - 1}{t_k}(S_k - S_{k-1});$
$G_k^L = Y_k^L - \frac{1}{2}\left(Y_k^L + Y_k^S - O\right);$
$G_k^S = Y_k^S - \frac{1}{2}\left(Y_k^L + Y_k^S - O\right);$
$(U, \Sigma, V) = \text{svd}\left(G_k^L\right);$
$L_{k+1} = U\text{RK}_{k,\lambda_1\mu/2}(\Sigma)V^T$
$S_{k+1} = \text{RK}_{k,\lambda_2\mu/2}\left(G_k^S\right);$
$t_{k+1} = \frac{1 + \sqrt{4t_k^2 + 1}}{2};\ \mu_{k+1} = \max\left(\rho\mu_k, \overline{\mu}\right);\ k \longleftarrow k + 1;$
until converged

Algorithm 1: An efficient minimization scheme with regular k-shrinkage thresholding operator.

Setting $L_f = 1$, then we have the following objective function:

$$\min_{L,S} \quad \mu\lambda_1 \|L\|_k^* + \mu\lambda_2 \|S\|_{1,k} + \left\|L - G_k^L\right\|_F^2 + \left\|S - G_k^S\right\|_F^2, \tag{16}$$

where the proximal points G_k^L and G_k^S in the framework of APG are defined in Algorithm 1. It can be noted that both the variables L and S are separable. Thus, there are two subproblems, which can be represented as follows:

$$\min_{L} \quad \mu\lambda_1 \|L\|_k^* + \left\|L - G_k^L\right\|_F^2, \tag{17}$$

$$\min_{S} \quad \mu\lambda_2 \|S\|_{1,k} + \left\|S - G_k^S\right\|_F^2. \tag{18}$$

It can be seen that (17) is a minimization problem with regular spectral k-support norm and (18) with regular k-support ℓ_1-norm. To deal with these problems, regular k-shrinkage thresholding operator (RK) is defined as follows:

$$\text{RK}_{k,\tau}(x) = \text{sgn}(x)\max(|x| \ominus \tau, 0), \quad \|x\|_0 \le k, \tag{19}$$

where $\ominus$ denotes an operation of direct minus in nonincreasing order. k denotes the sparsity level of the input vector. τ also is a vector in a nonincreasing order.

Remark 1. There are some differences between regular k-shrinkage thresholding operator and shrinkage operator [17]. First, the proposed operator models the sparsity level by the procedure of regular shrinkage adaptively. Second, the introduction of k-support constraint can bound the degree of the sparsity. Third, the combination of regular shrinkage and k-support leads to the modeling of the correlated variables robustly.

When applying the proposed operator to G_k^L and G_k^S, we have

$$L_{k+1} = U\text{RK}_{k,\lambda_1\mu/2}(\Sigma)V^T, \quad S_{k+1} = \text{RK}_{k,\lambda_2\mu/2}\left(G_k^S\right), \tag{20}$$

where Σ denotes the eigenvalues of G_k^L. It should be noted that the solution to L_{k+1} can be viewed as a generalization of singular value shrinkage operator. Based on the framework of APG, an optimization framework for the objective function (16) is presented in Algorithm 1. The detailed procedures for the two subproblems are provided in Algorithm 1.

4.2. Some Implementation Details. In our implementation, the sampled image patches with overlapping regions are considered. Then, each frame of the restored video may be replaced by the recovered patches. For the synthesis process, the outcome of each selected pixel is accomplished by calculating the average of multiple estimates from the related patches. This procedure could deal with the artifacts along the boundaries of patches and restore fine details locally.

5. Experiments and Discussion

5.1. Experimental Settings. To demonstrate the effectiveness and efficacy of the proposed method, some experiments are conducted. We focus on the removal of mixed Gaussian-impulse noise. Two types of noisy situations, including mixed Gaussian and random value impulse noise (GRV) denoted by (σ, si) and mixed Gaussian, salt-and-pepper noise, and random value impulse noise (GSPRV) by $(\sigma, \text{sp}, \text{si})$, are tested. Some samples of three videos (http://trace.eas.asu.edu/yuv/) are displayed in Figure 1. The sizes of *coastguard*, *flower*, and *news* in our experiments are $176 \times 144 \times 100$, $352 \times 240 \times 150$, and $352 \times 288 \times 150$, respectively. The parameter k of the

(a) (b)

Figure 1: Two samples used in the experiments ((a) sample for *flower*, (b) sample for *news*).

Table 1: Numerical results by the removal of MGRV, measured by PSNR and FSIM.

Noise level	Indexes	Methods	*Coastguard*	*Flower*	*News*
(10, 10%)	PSNR	VBM3D	28.75	19.62	30.36
		RPCA	30.75	22.11	33.49
		ℓ_1-ℓ_0	28.49	19.45	28.40
		Ours	**31.07**	**23.01**	**34.86**
(15, 20%)	PSNR	VBM3D	27.37	18.72	28.10
		RPCA	28.98	19.66	30.24
		ℓ_1-ℓ_0	26.15	18.12	25.56
		Ours	**29.58**	**21.41**	**31.32**
(20, 30%)	PSNR	VBM3D	25.56	17.97	25.64
		RPCA	25.69	18.23	26.34
		ℓ_1-ℓ_0	23.06	16.66	24.20
		Ours	**27.17**	**19.30**	**27.41**
(10, 10%)	FSIM (%)	VBM3D	87.71	81.64	95.52
		RPCA	90.01	87.57	**96.47**
		ℓ_1-ℓ_0	91.35	84.94	90.11
		Ours	**92.04**	**92.14**	96.35
(15, 20%)	FSIM (%)	VBM3D	82.24	75.90	92.74
		RPCA	89.32	79.93	93.47
		ℓ_1-ℓ_0	84.33	79.25	87.10
		Ours	**90.13**	**87.58**	**94.06**
(20, 30%)	FSIM (%)	VBM3D	77.19	70.42	89.18
		RPCA	86.69	74.30	88.35
		ℓ_1-ℓ_0	79.47	73.35	82.41
		Ours	**87.82**	**77.09**	**89.48**

proposed method for the *coastguard* is set to 1000. For the other videos, that is, *flower* and *news*, k is set to 2000. All experiments are performed in MATLAB R2014 running on a desktop with Intel Core i7 at 3.2 GHz.

Three related methods are compared with the proposed method, including VBM3D [30], RPCA based method [7], and ℓ_0-ℓ_1 based method [8]. VBM3D based method is not originally designed for the removal of mixed Gaussian-impulse noise. To remedy this problem, adaptive center-weighted median filter [15] (ACWMF) is used to detect and remove the impulse noise firstly. Two indexes are taken for assessing the denoised performance of all competing methods, that is, peak-signal-to-noise ratio (PSNR) and feature-similarity (FSIM) index [31].

5.2. Mixed Gaussian and Random Value Impulse Noise. In this subsection, the denoising results for three different scenarios are presented, including (σ, si) = (10, 10%), (15, 20%), and (20, 30%). Numerical results on three videos are presented in Table 1. It can be observed that the proposed

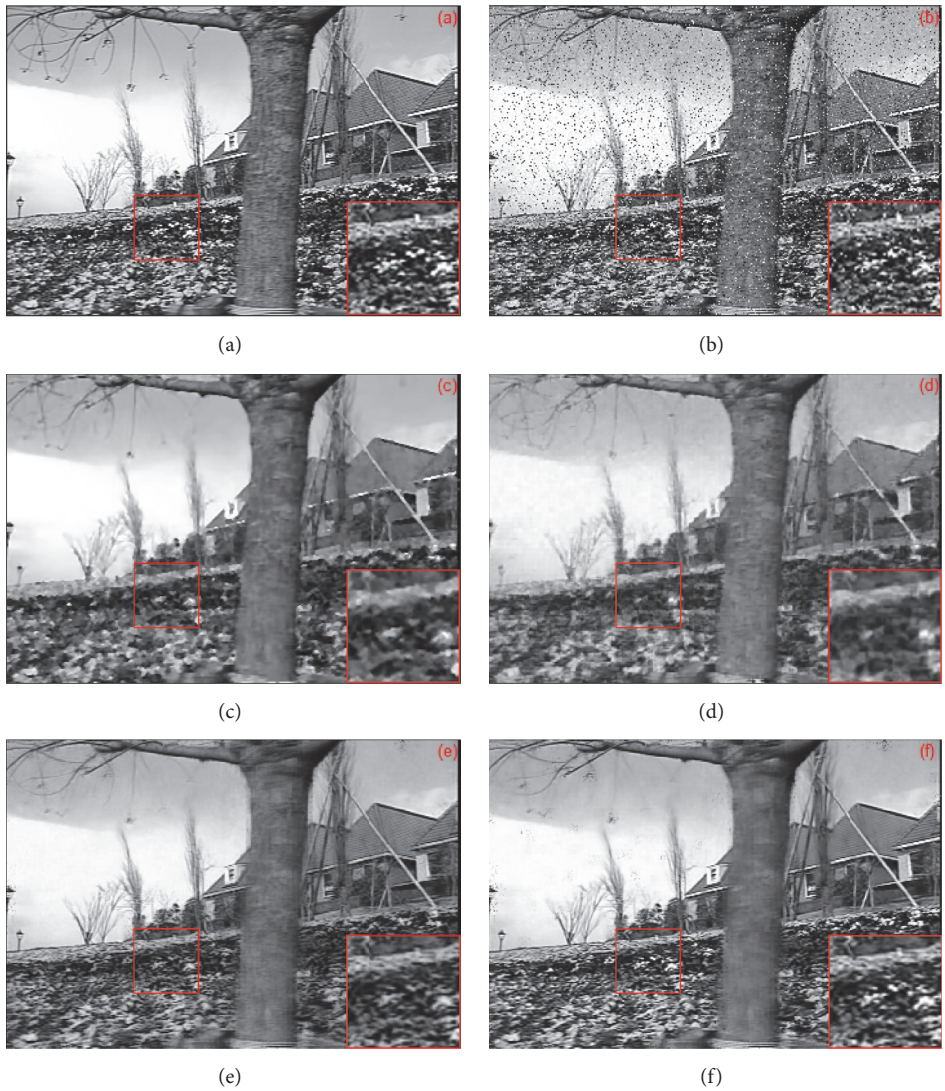


FIGURE 2: Visual comparison of denoising results on flower by noise level $(\sigma, \text{si}) = (10, 10\%)$. (a) Original sample *(flower)*, (b) noisy image, (c) VBM3D based method (PSNR = 19.62), (d) RPCA based method (PSNR = 22.11), (e) ℓ_1-ℓ_0 based method (PSNR = 19.45), and (f) the proposed method (PSNR = 23.01, $k = 2000$).

method outperforms all the competing methods with respect to FSIM and PSNR. Visual outcomes are demonstrated in Figure 2. The recovered result of proposed method is presented in Figure 2(f). To examine the recovered details, the selected parts in the visual results are enlarged. It can be noted that the proposed method can reconstruct more local details.

5.3. Mixed Gaussian, Salt-and-Pepper Noise, and Random Value Impulse Noise. In this subsection, the experimental results by the removal of Gaussian, salt-and-pepper, and random value impulse noise are demonstrated. Two noisy levels are assessed. The numerical results are presented in Table 2. It can be noted that the proposed method outperforms other methods. A visual assessment of the reconstruction performance of both algorithms is shown in Figure 3. As shown in the enlarged parts, the proposed method presented in Figure 3(f) recovers more local details than other methods.

5.4. Discussion. In this paper, an efficient image restoration scheme for hybrid Gaussian-impulse noise is proposed. The denoising performance of our method is examined in various noisy scenarios. When the strength of noisy levels increased, our method performed more efficiently than other methods. The outcomes of all experiments verified the effectiveness of the proposed method. The difference may be related to the modeling method and optimization strategy we taken. Moreover, the intrinsic sparsity structure of each decomposition component is explored. It should be noted that some limitations may be observed, such as being oversmooth on the local region.

In this paper, an alternating minimization method with regular k-shrinkage thresholding operator is proposed.

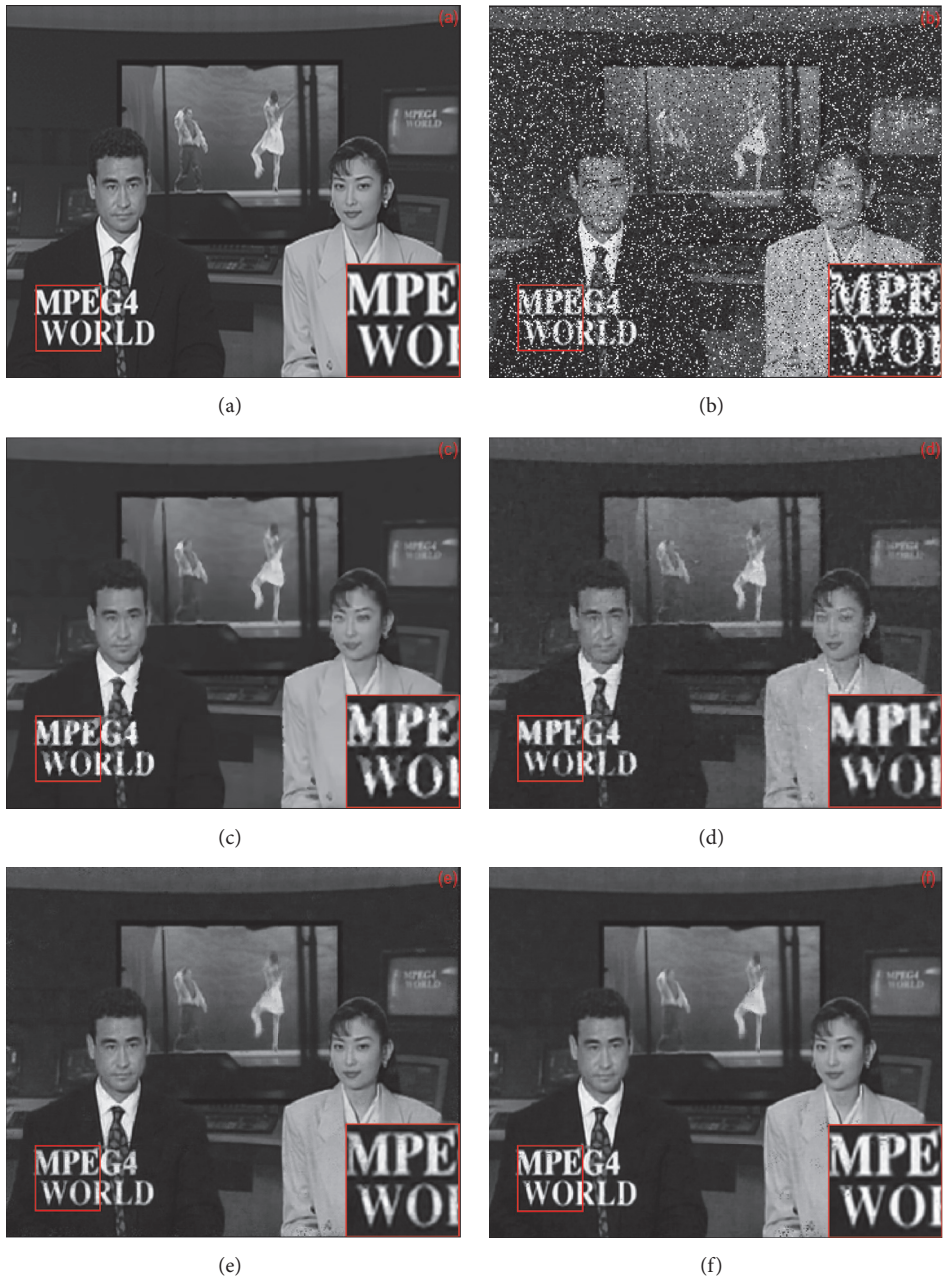


FIGURE 3: Visual comparison of denoising results by noise level (15, 15%, 15%). (a) Original sample *(news)*, (b) noisy image, (c) VBM3D based method (PSNR = 26.23), (d) RPCA based method (PSNR = 24.60), (e) ℓ_1-ℓ_0 based method (PSNR = 24.87), and (f) the proposed method (PSNR = 28.01, $k = 2000$).

Specially, a universal modeling strategy by exploiting the adaptivity of sparsity structure leads to higher quality reconstructions. The proposed method may provide a new class of denoising methods to deal with mixed Gaussian-impulse noise. The numerical results from various experiments validated the effectiveness of the proposed method again.

6. Conclusion

In this paper, an efficient video restoration scheme is proposed for the removal of mixed Gaussian-impulse noise. Unlike traditional ℓ_1 norm based methods, which treat all the values equally, the proposed method tries to explore the additional structure by regular spectral k-support norm on low rank component and regular k-support ℓ_1 norm on sparse component. Then, the special structure can be promoted on the sparsity level of the decomposition matrices adaptively. To overcome the nonconvex problem, a solution with alternating regular k-shrinkage thresholding operator is proposed. The proposed method has good practical performance with appropriate structures. The numerical results, compared to

Table 2: Numerical results by the removal of GSPRV, measured by PSNR and FSIM.

Noise level	Indexes	Methods	*Coastguard*	*Flower*	*News*
(10, 10%, 10%)	PSNR	VBM3D	24.54	18.13	27.22
		RPCA	24.72	18.98	28.40
		ℓ_1-ℓ_0	**26.27**	18.35	27.05
		Ours	25.27	**20.51**	**29.32**
(15, 15%, 15%)	PSNR	VBM3D	23.79	17.62	26.23
		RPCA	22.11	16.65	24.60
		ℓ_1-ℓ_0	24.21	17.37	24.87
		Ours	**24.26**	**18.67**	**28.01**
(10, 10%, 10%)	FSIM (%)	VBM3D	85.32	78.72	93.52
		RPCA	**89.50**	82.94	93.75
		ℓ_1-ℓ_0	88.62	79.44	90.66
		Ours	89.43	**87.64**	**94.13**
(15, 15%, 15%)	FSIM (%)	VBM3D	80.05	74.39	90.87
		RPCA	83.33	75.13	82.78
		ℓ_1-ℓ_0	83.57	75.16	84.61
		Ours	**87.72**	**79.92**	**91.83**

some state-of-the-art methods, demonstrate the advantages of the proposed method.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is jointly supported by National Natural Science Foundation of China (Grants nos. 61603249 and 61673262) and key project of Science and Technology Commission of Shanghai Municipality (Grant no. 16JC1401100).

References

[1] R. Das, S. Thepade, S. Bhattacharya, and S. Ghosh, "Retrieval architecture with classified query for content based image recognition," *Applied Computational Intelligence and Soft Computing*, vol. 2016, Article ID 1861247, 9 pages, 2016.

[2] H. Pan, Z. Jing, M. Lei, R. Liu, B. Jin, and C. Zhang, "A sparse proximal Newton splitting method for constrained image deblurring," *Neurocomputing*, vol. 122, pp. 245–257, 2013.

[3] M. Kumar, S. K. Mishra, and S. S. Sahu, "Cat swarm optimization based functional link artificial neural network filter for gaussian noise removal from computed tomography images," *Applied Computational Intelligence and Soft Computing*, vol. 2016, Article ID 6304915, 6 pages, 2016.

[4] G. Zhang, P. Jiang, K. Matsumoto, M. Yoshida, and K. Kita, "Reidentification of persons using clothing features in real-life video," *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 5834846, 9 pages, 2017.

[5] H. Pan, Z. Jing, R. Liu, and B. Jin, "Simultaneous spatial-temporal image fusion using Kalman filtered compressed sensing," *Optical Engineering*, vol. 51, no. 5, pp. 23–29, 2012.

[6] X. Li, Y. Zhang, and D. Liao, "Mining key skeleton poses with latent svm for action recognition," *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 5861435, 11 pages, 2017.

[7] H. Ji, S. Huang, Z. Shen, and Y. Xu, "Robust video restoration by joint sparse and low rank matrix approximation," *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1122–1142, 2011.

[8] Y. Xiao, T. Zeng, J. Yu, and M. K. Ng, "Restoration of images corrupted by mixed Gaussian-impulse noise via l1-l0 minimization," *Pattern Recognition*, vol. 44, no. 8, pp. 1708–1720, 2011.

[9] P. Rodríguez, R. Rojas, and B. Wohlberg, "MIxed Gaussian-impulse noise image restoration via total variation," in *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2012*, pp. 1077–1080, March 2012.

[10] M. Filipovic and A. Jukic, "Restoration of images corrupted by mixed gaussian-impulse noise by iterative soft-hard thresholding," in *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pp. 1637–1641, IEEE, 2014.

[11] J. Woodworth and R. Chartrand, "Compressed sensing recovery via nonconvex shrinkage penalties," *Mathematics*, vol. 2012, no. 7, article 075004, 2015.

[12] J.-F. Cai, R. H. Chan, and M. Nikolova, "Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise," *Inverse Problems and Imaging*, vol. 2, no. 2, pp. 187–204, 2008.

[13] Y.-R. Li, L. Shen, D.-Q. Dai, and B. W. Suter, "Framelet algorithms for de-blurring images corrupted by impulse plus Gaussian noise," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1822–1837, 2011.

[14] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D. Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[15] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1–3, 2001.

[16] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1747–1754, 2005.

[17] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[18] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[19] M. Bogdan, E. V. D. Berg, W. Su, and E. Candes, "Statistical estimation and testing via the sorted l1 norm," *Statistics*, 2013.

[20] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the *k*-support norm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1466–1474, 2012.

[21] A. Eriksson, T. T. Pham, T.-J. Chin, and I. Reid, "The k-support norm and convex envelopes of cardinality and rank," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 3349–3357, June 2015.

[22] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3320–3325, 2003.

[23] 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.

[24] R. Saab, R. Chartrand, and Ö. Yilmaz, "Stable sparse approximations via nonconvex optimization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 3885–3888, April 2008.

[25] X. Chen, F. Xu, and Y. Ye, "Lower bound theory of nonzero entries in solutions of ell_2-ell_p minimization," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2832–2852, 2010.

[26] M. J. Lai and L. Y. Liu, "A new estimate of restricted isometry constants for sparse solutions," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 402–406, 2011.

[27] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using Low rank matrix completion," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pp. 1791–1798, June 2010.

[28] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[29] Z. Shen, K.-C. Toh, and S. Yun, "An accelerated proximal gradient algorithm for frame-based image restoration via the balanced approach," *SIAM Journal on Imaging Sciences*, vol. 4, no. 2, pp. 573–596, 2011.

[30] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Proceedings of the 15th European Signal Processing Conference*, vol. 1, p. 7, 2007.

[31] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: a feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 3048181, 9 pages
https://doi.org/10.1155/2017/3048181

Research Article

Sliding Window Based Machine Learning System for the Left Ventricle Localization in MR Cardiac Images

Abdulkader Helwan and Dilber Uzun Ozsahin

Department of Biomedical Engineering, Near East University, Near East Boulevard, 99138 Nicosia, Northern Cyprus, Mersin 10, Turkey

Correspondence should be addressed to Abdulkader Helwan; abdulkader.helwan90@gmail.com

Received 9 March 2017; Revised 12 April 2017; Accepted 30 April 2017; Published 4 June 2017

Academic Editor: Mourad Zaied

The most commonly encountered problem in vision systems includes its capability to suffice for different scenes containing the object of interest to be detected. Generally, the different backgrounds in which the objects of interest are contained significantly dwindle the performance of vision systems. In this work, we design a sliding windows machine learning system for the recognition and detection of left ventricles in MR cardiac images. We leverage on the capability of artificial neural networks to cope with some of the inevitable scene constraints encountered in medical objects detection tasks. We train a backpropagation neural network on samples of left and nonleft ventricles. We reformulate the left ventricles detection task as a machine learning problem and employ an intelligent system (backpropagation neural network) to achieve the detection task. We treat the left ventricle detection problem as binary classification tasks by assigning collected left ventricle samples as one class, and random (nonleft ventricles) objects are the other class. The trained backpropagation neural network is validated to possess a good generalization power by simulating it with a test set. A recognition rate of 100% and 88% is achieved on the training and test set, respectively. The trained backpropagation neural network is used to determine if the sampled region in a target image contains a left ventricle or not. Lastly, we show the effectiveness of the proposed system by comparing the manual detection of left ventricles drawn by medical experts and the automatic detection by the trained network.

1. Introduction

Machine learning (ML) is a form of artificial intelligence (AI) which gives computers the skills to learn without being specifically programmed. It focuses on building computer programs which are subject to change when exposed to new data. Machine learning can be classified as either supervised or unsupervised. Supervised algorithms can apply past knowledge to new data whereas unsupervised algorithms make conclusions from datasets [1–5].

The field of medical imaging has witnessed a delay in embracing the novel ML techniques as compared to other fields. Despite machine learning being virtually new, its concept has been applied to medical imaging for years, particularly in areas of computer-aided diagnosis (CAD) and functional brain mapping [6]. Components of medical imaging (image analysis and reconstruction) tend to benefit from the merger of machine learning with medical imaging. From this perspective, new methods for image reconstruction and exceptional performance in both clinical and preclinical applications will be achieved [6]. A study [7] sees machine learning as a major tool for current computer-aided analysis (CAD). Previous knowledge acquired from examples provided by medical experts has helped in areas like image registration, image fusion, segmentation, and other analyses steps towards describing accurately the initial data and CAD goals. Other applications of machine learning in medical imaging include but are not limited to tumour classification, tumour diagnosis, image segmentation, image reconstruction, and prediction [3, 6, 7].

In this research we focus on detection tasks employing artificial systems (machines). Such systems are required to "look" at an image and determine if a particular object of interest is contained anywhere in the image, in addition to detecting it. Medical object detection is a task that traditionally belongs to the class of computer vision problems.

It is noteworthy that while humans are very effective and efficient in detecting various complex objects irrespective of scene constraints such as varying background, object scale, object positional translation, object orientation, and object illumination; machines strive to achieve near human performance on object detection. Furthermore, it is stressed that object detection is quite more challenging for machines as compared to object recognition. In object detection, the object of interest to be detected can be positioned in any region of an image, while, in object recognition, the objects of interest to be recognized is usually already segmented, hence, making the recognition less challenging. In order to succeed in a task such as object detection, developed vision models or systems should be capable of coping with the aforementioned scene constraints. More important is that, in robotic systems, medical objects detection tasks are very delicate, requiring utmost accuracy. Since robotic systems are usually interacting with its environments in a somewhat real-time fashion, the consequence of wrongly detecting objects of interest can be very grave or serious.

In this paper, we design a sliding window based machine learning system for the detection of left ventricles in MRI slices. It is important to note that while any other object could have been used to demonstrate the effectiveness of the designed system, we found the detection of left ventricles in highly varying and unconstrained images sufficient. Furthermore, it will be seen later on in this work that the approach implemented for the detection of left ventricles in images can be easily extended and modified to realize the detection of other objects in images.

2. Sliding Window Machine Learning Approach

In the sliding window approach, a window of suitable size, say $m \times n$, is chosen to perform a search over the target image [8, 9]. First, a classifier is trained on a collection of training samples spanning the object of interest for detection as one class and random objects as the other class. Formally, samples belonging to the object of interest for detection are referred to as positive examples, while random object samples of no interest are referred to as negative examples. For a single object detection task, the idea is to train a binary classifier, which determines if the presented object is "positive" or "negative." The trained classifier can then be used to "inspect" a target image by sampling it, starting from the top-left corner. It is noteworthy that the input dimension of the trained classifier is generally a fraction of the size or dimension of the target image; hence, sampling of target images can be achieved.

Some of the classifiers that have found applications within the context of object detection include deep neural network (DNN), convolutional neural network (CNN), and decision trees (DT) [2, 10, 11]. Considering the aforementioned considerations for selecting a suitable classifier for object detection, the support vector machine (SVM), which is a maximum margin classifier, would be an obvious choice, but for the long training time as compared to backpropagation neural network and decision trees. In view of required training time, decision trees (DTs) usually have the least training time as compared to the support vector machine (SVM) and backpropagation neural network (BPNN); however, decision trees tend to quickly overfit or "memorize" the training data. The consequence is such that the performance of decision trees on the test set (unseen examples) is not competitive. The backpropagation neural network (BPNN) seems to be the modest trade-off between training time and generalization power, since the BPNN has a training time that is in between that of the support vector machine and decision tree and a generalization performance that is better than that of decision trees and competitive with that of the support vector machine [12, 13]. Hence, in this project work, the backpropagation neural network has been used as the classifier for the object detection task.

3. The Proposed Automatic Left Ventricle Detection System

The aim of this work is to develop an artificial vision system that can perform the task of detecting left ventricle in images. In this work, considering challenges such as object illumination, scale, translation, and rotation, which make the detection a complex problem for such an open detection problem, we resolve to implementing an intelligent system which can somewhat graciously cope with the aforementioned detection constraints. Neural network, namely, the backpropagation neural network (BPNN), has been used in this work as the 'brain' behind the detection.

This research is achieved in two phases. First is the left ventricle recognition phase by training a backpropagation neural network (BPNN). The second phase is the detection of left ventricle objects in MRI slices using the backpropagation neural network. The flowchart for the system is shown in Figure 1 and both phases are briefly described below.

3.1. Phase 1: Left Ventricle Object Recognition. In this phase, a backpropagation neural network is trained to recognize left ventricle objects and nonleft ventricle objects. In order to achieve this binary classification task, training data is collected to span both left ventricle images and nonleft ventricle images. The data used for training and testing data are obtained from Sunnybrook Cardiac Data (SCD) [14]. The dataset contains 45 cine-MRI slices collected from a mix of patients and different pathologies such as healthy, hypertrophy, heart failure with infarction, and heart failure without infarction. A subset of 100 images was used for the proposed system training and testing purposes for both phases. Since the actual interest is to develop a system that recognizes left ventricle objects, MRI slices were cropped to have only the left ventricle and are referred to as positive examples or samples. Conversely, images containing random nonleft ventricle images are referred to as negative examples or samples. Note that, for earlier training phases, there was no constraint on the contents of the negative examples except that they do not contain left ventricle objects. However, it was discovered that the negative images can be collected by cropping the other parts of the whole MRI cardiac slice by excluding the left ventricle. This seems to improve the robustness of the system in distinguishing left ventricle and

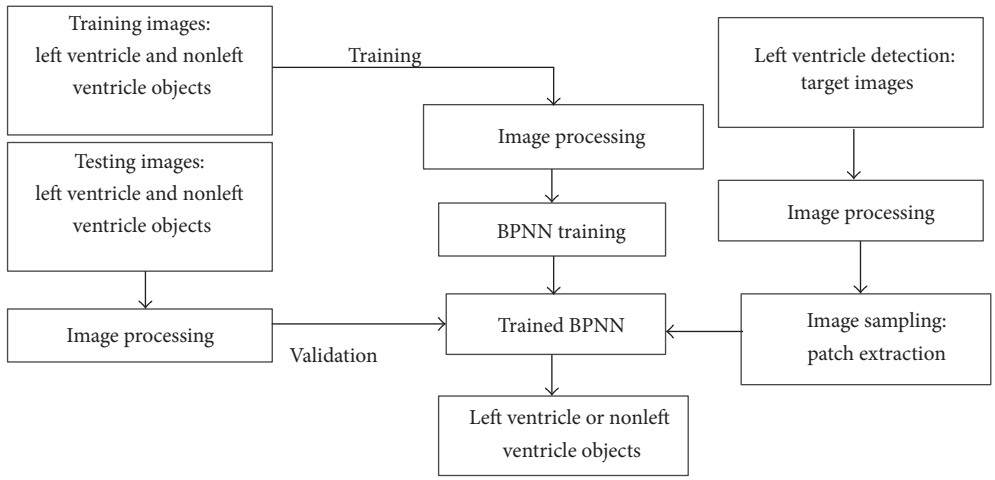


Figure 1: Flowchart for developed system.

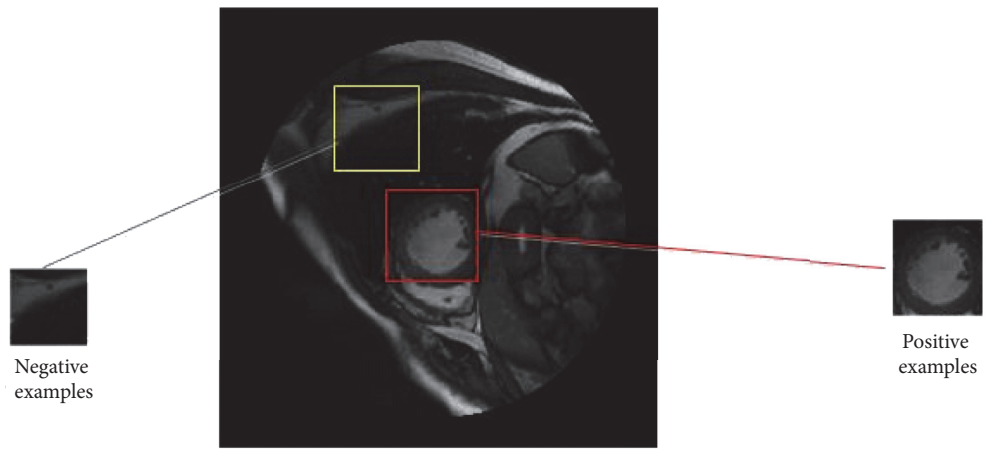


Figure 2: Positive and negative examples extraction from the MR images.

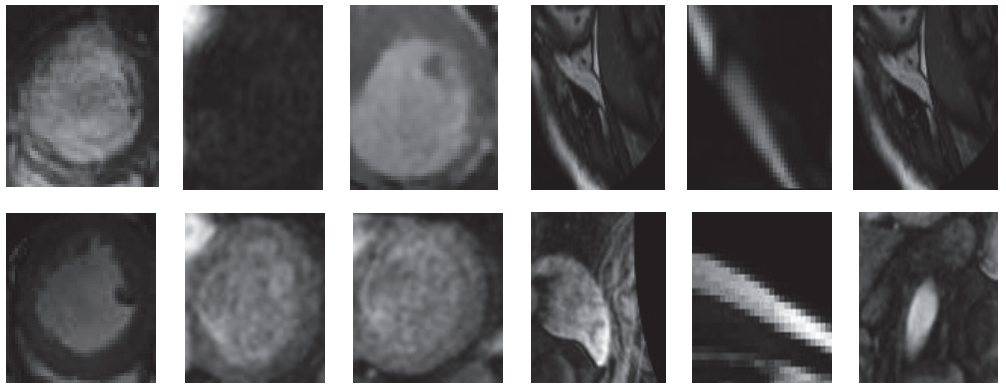
nonleft ventricle images. Cropping ventricle and nonventricle images from cardiac MR image is shown in Figure 2.

3.1.1. Image Processing. Since the positive and negative examples are cropped manually, they are of different sizes. Thus, in order to make the images consistent, they are all resized to 40×40 pixels (1600 pixels). Samples of positive and negative examples are shown in Figure 3.

3.1.2. Backpropagation Neural Network (BPNN) Design, Training, and Testing. A backpropagation neural network is trained on the collected samples spanning both positive and negative examples. For the positive examples (left ventricles), 100 samples cropped from different cardiac MRI slices are used, while, for the negative examples (nonleft ventricles), 200 samples are used. The negative images are more because one MR image can provide many negative images where the left ventricle is not included. The positive and negative samples form the training and testing data for the designed backpropagation neural network (BPNN). All images are first rescaled to 40×40 pixels (1600 pixels). The whole dataset is then divided into training and testing data. The testing data allows the observation of performance of the trained BPNN on unseen or new data. It is very desirable that trained ANNs can perform well on unseen data, that is, generalization. 75 left ventricles and 275 nonleft ventricles are used for training, while 25 left ventricles and 25 nonleft ventricles are used for testing the trained BPNN. Hence, there are a total of 250 training images and 50 testing images.

(a) Input Data and Neurons. Considering that the training images are now 40×40 pixels, the designed BPNN has 1600 input neurons, where each input attribute or pixel is fed into one of the input neurons. Also, note that the input neurons are nonprocessing. That is, they basically receive input pixels and supply them to the hidden layer neurons which are processing neurons.

(b) Hidden Layer Neurons. The hidden layer is where the extraction of input data features that allows the mapping of input data to corresponding target classes is achieved. Unlike the input layer neurons, the hidden layer neurons are processing. Also, each hidden layer neuron receives inputs from all the input layer neurons. In this work, several



(a) Positive examples (left ventricles) (b) Negative examples (nonleft ventricles)

Figure 3: Samples for (a) left ventricles (positive examples) and (b) nonleft ventricles (negative examples).

Table 1: Final training parameters for BPNN.

Network parameters	Values
Number of training samples	250
Number of input neurons	1600
Number of hidden neurons	80
Activation function at hidden and output layers	Log-Sigmoid
Learning rate (η)	0.11
Momentum rate (α)	0.80
Required error (MSE)	0.01
Epochs	1,215
Training time	40 secs

experiments are carried out to determine the suitable number of hidden layer neurons. Finally, the number of suitable hidden neurons was obtained as 80 during network training.

(c) Output Layer Coding. Considering that we aim to classify all images as left ventricle object or nonleft ventricle object, the BPNN has two output neurons. The output of the BPNN is coded such that output neurons activations are as shown in the following:

(i) $[1 \;\; 0] \rightarrow$ a left ventricle object

(ii) $[0 \;\; 1] \rightarrow$ a nonleft ventricle object.

Figure 4 shows the designed BPNN. The BPNN is trained on the processed images described in Figure 3. The final training parameters are shown in Table 1.

The Log-Sigmoid activation function allows neuron's output to be in the range of 0 to 1. From Table 1, it is seen that the BPNN achieve the required error of value 0.01 in 40 secs, with 1,215 epochs. The learning curve for the BPNN is shown in Figure 5.

The trained BPNN is then tested using the training and testing data. Table 2 shows the recognition rates of the BPNN on the training andtesting data.

Table 2: Recognition rates for BPNN.

Parameter	Training	Testing
Number of samples	250	50
Number of samples correctly classified	255	44
Recognition rates	100%	88%

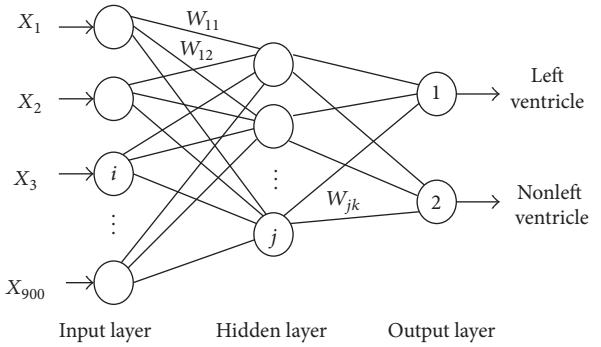


Figure 4: Designed backpropagation neural network (BPNN).

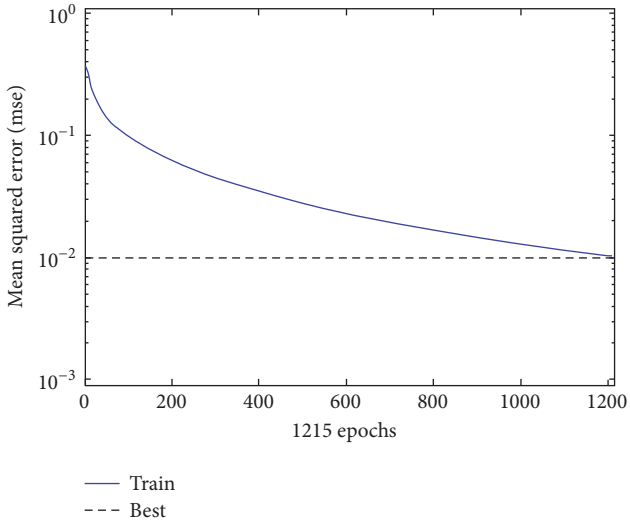


Figure 5: Error versus epochs curve for BPNN.

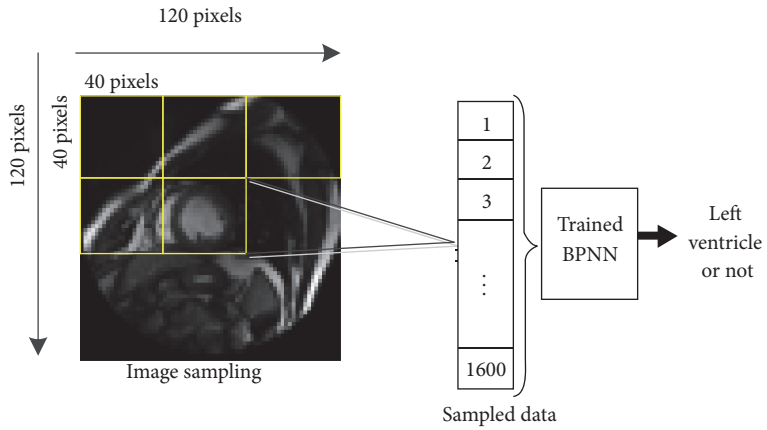


Figure 6: Sampling of image for left ventricle detection using trained BPNN.

It is seen in Table 2 that the BPNN achieved a recognition rate of 100% and 88% on the training and testing data, respectively. Note that a testing recognition rate of 88% is enough to show that the BPNN can generalize well on unseen data (images), that is, classifying new images as left ventricle or nonleft ventricle.

3.2. Phase 2: Left Ventricle Detection from Images. In this phase, the trained BPNN is used to detect left ventricles in images containing various objects, background, illumination, scale, and so on. In order to detect left ventricles in new images, the new images are sampled in a nonoverlapping fashion using a sliding window or mask. Firstly, all images in which left ventricles are to be detected are rescaled to 120×120 pixels; this significantly reduces the required number of samplings and therefore computations. Note that the new size of images containing left ventricle for detection is selected such that input field (40×40 pixels) of the earlier trained BPNN can fit in without falling off image edges.

It therefore follows that if the new images containing left ventricle for detection is rescaled to 120×120 pixels, and a sliding window of size 40×40 pixels is used for nonoverlapping sampling, 3 samplings are obtained in the x-pixel coordinate, and 3 samplings are obtained in the y-pixel coordinate; this makes a total of 9 samplings for an image. Figure 6 shows the analogy of the sampling technique.

The sampling outcomes using a sliding window of size 40×40 pixels (1600 pixels) is supplied as the input of the trained BPNN as shown in Figure 6. It is expected that, for windows containing a left ventricle, the BPNN gives an output of $[1 \ \ 0]$, as coded during the BPNN training. Also, it is expected that, for windows not containing left ventricles, the trained BPNN gives an output of $[0 \ \ 1]$. From the sampling approach described above, it will be observed that 9 samplings (patches) and therefore predictions are made for any target image. The BPNN output with the closest match with the desired output for left ventricle output, $[1 \ \ 0]$, is selected as containing a left ventricle, that is, with maximum activation value for neuron 1 in Figure 4. It is seen that to achieve the complete detection of left ventricles in images, both phases 1 and 2 are sandwiched together as one module.

4. Performance Evaluation

An example of left ventricle detection for the image shown in Figure 6 is shown in Figure 7 using the developed system. More examples of the left ventricle localizations of different types of MR images are shown in Figures 8 and 9.

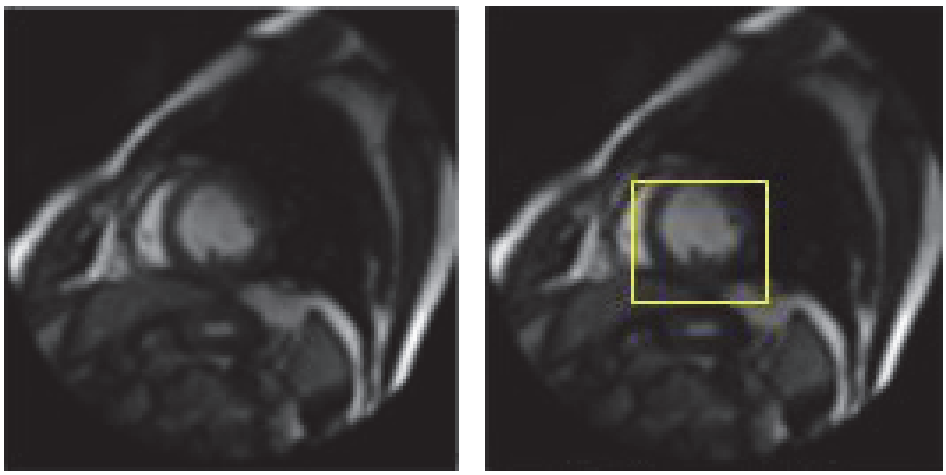
The detected left ventricle is highlighted in a rectangular bounding box.

Also, samples of other target images for left ventricle detection using the developed system within this work are shown in Figures 8, 9, and 10. The detected left ventricle objects are highlighted as a rectangular bounding box.

Also, some instances where the developed failed to achieve the correct detection of left ventricle in images are shown in Figure 10.

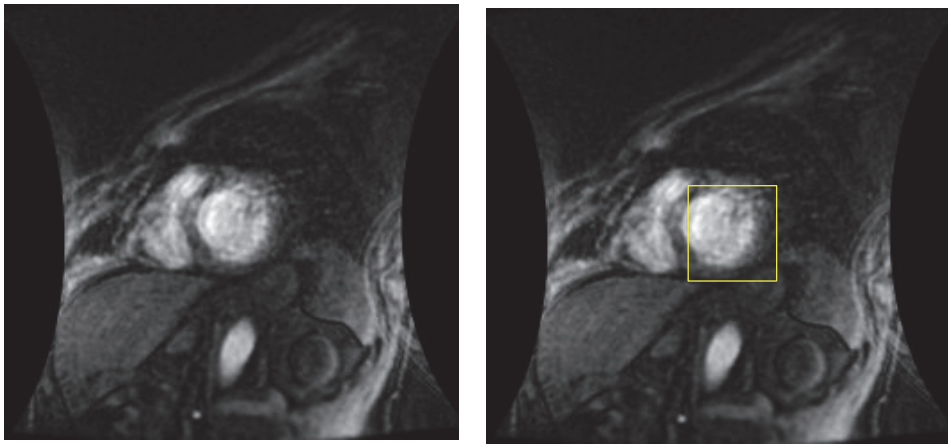
Generally, most of the approaches provided in the state of the art of left ventricle detection can be considered as some variation of the active contour and segmentation models [15–19]. These models are meant to segment the endocardium and epicardium areas of the left ventricle. In contrast, our proposed model is a general squared detection or localization of the left ventricle in an MRI slice. This model is mainly a machine learning approach that aims to evaluate the effectiveness and capability of a simple backpropagation neural network in sampling an MRI slice for the purpose of finding and detecting a left ventricle object based on a sliding window's approach. The approach here is not to accurately segment the edges of the left ventricle; however, it is to find and localize the left ventricle as an object in the image. Therefore, in some images, a small part of the left ventricle can be undetected, and still this can be considered as a correct detection due to the application type which is to find or localize the left ventricle even though a small part of it is missing. Hence, our results cannot be compared to other research findings since the approach and the techniques used are totally different.

In order to show the effectiveness of the developed system, some left ventricles were manually detected by medical experts to validate our system capability in detecting the left ventricle in MRI slices. The idea is to compare both detections, that is, the network detection and the medical



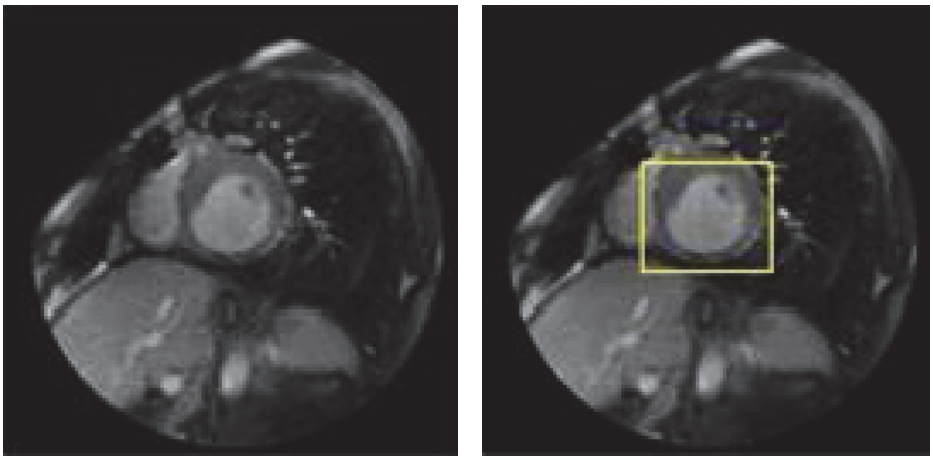
Target image Left ventricle detection

Figure 7: Detection outcome using the developed system (gradient echo sequence, short axis image).



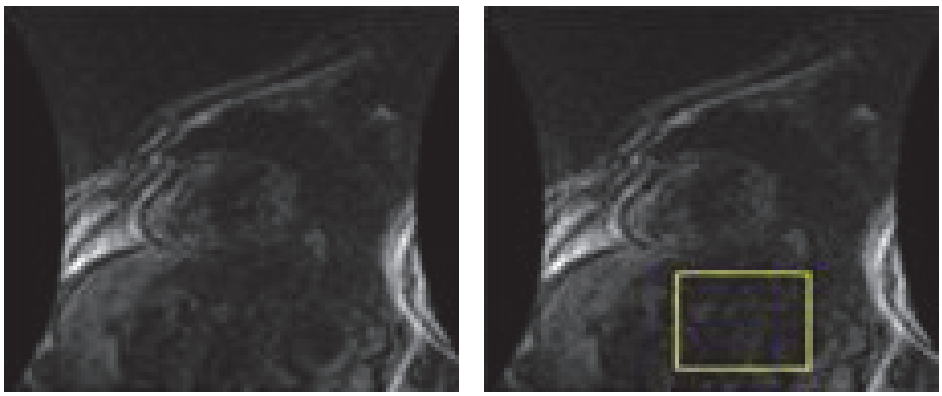
Target image Left ventricle detection

Figure 8: Detection outcome using the developed system (short axis image).



Target image Left ventricle detection

Figure 9: Detection outcome using the developed system (short axis image).



Target image Incorrect left ventricle detection

Figure 10: Wrong detection outcome using the developed system (fast spin echo sequence, FSE).

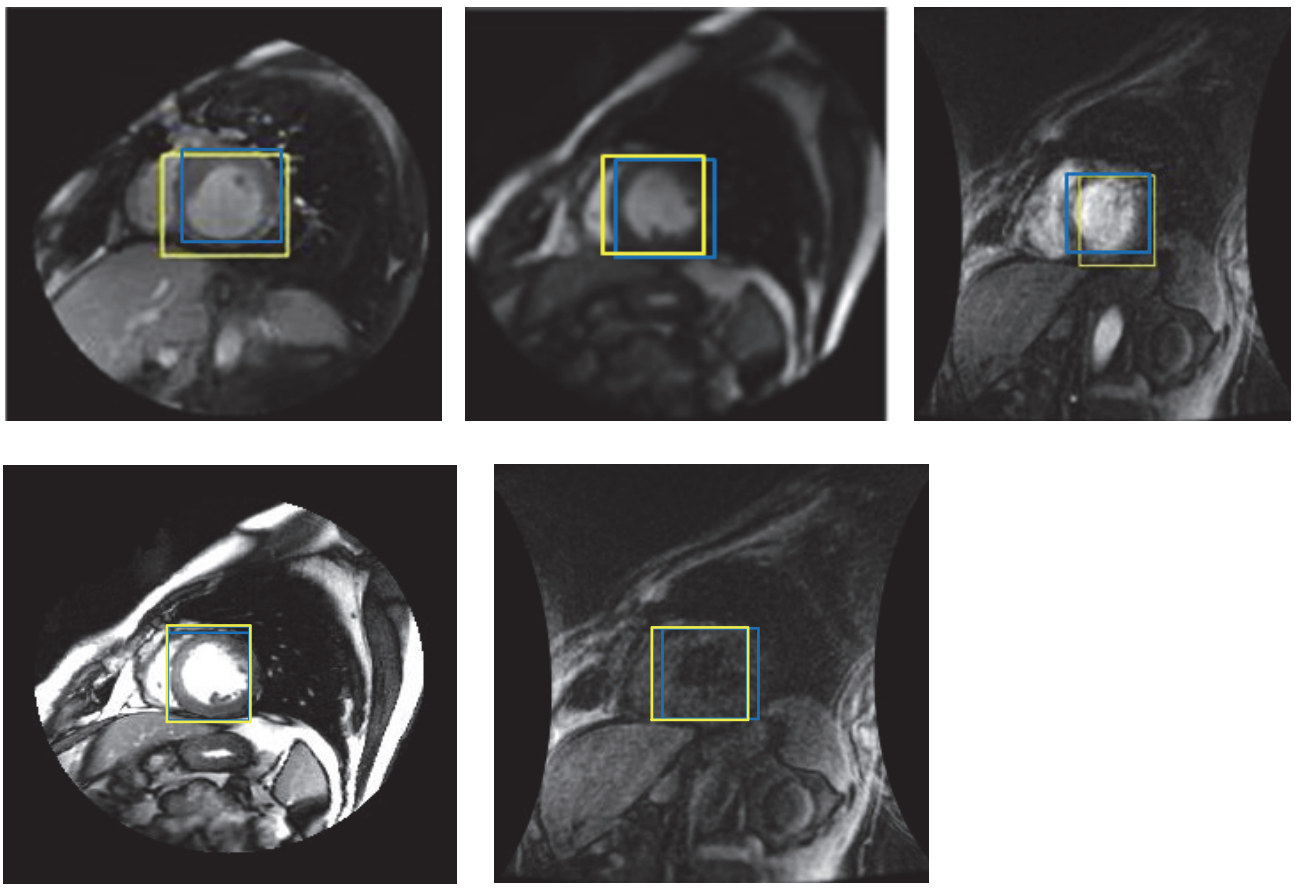


Figure 11: Manual and network detection results. Yellow color: network's detection; blue color: expert's manual detection.

expert manual detection of the left ventricles to check if the left ventricles were fit into the detected 40 ∗ 40 square in target images. In other words, it is to check how accurately the system was capable to detect the left ventricle by comparing the detected area to the left ventricle highlighted by the medical expert. Figure 11 illustrates some images where the manual and network's detection of the left ventricle are shown.

5. Results Discussion

Since artificial neural network weights are usually randomly initialized at the start of training, it therefore follows that trained BPNN is not always guaranteed to converge to the global minimum or good local minima. Consequently, the learning of left ventricles and nonleft ventricles can be negatively affected; this therefore affects the detection phase,

where the trained BPNN may wrongly predict a sampling window or patch as containing a left ventricle. In order, to solve this problem, the MATLAB program written contains instructions to retrain the BPNN till a testing recognition (relating to BPNN generalization capability) of greater than 80% is obtained. This greatly reduces the BPNN's probability of wrongly predicting a sampling window (patch) as containing a left ventricle. In this project, we have allowed for a maximum of 30 retraining schedules of the BPNN. Therefore, when the MATLAB script for the developed whole detection system is run, it is possible that the BPNN may be automatically retrained a couple of times before the detection task is then executed.

Moreover, another challenge encountered is that even after the BPNN achieves a testing recognition rate of greater than 88%, it is still possible that sampling windows are wrongly classified, though the probability of this happening is quite small. In this project, it is found that when the BPNN achieves a testing recognition of greater than 88%, a maximum of 3 retraining schedules is required to correctly detect a left ventricle in the target image.

This work describes a highly challenging task in computer vision, medical object detection. We show that backpropagation neural network (BPNN) can be employed to learn the robust recognition/classification of left ventricles and nonleft ventricles as positive and negative training examples, respectively. The trained BPNN is then used in a nonoverlapping sampling fashion to "inspect" target images containing left ventricles for detection. The developed system is tested and found to be very effective in the detection of left ventricles in images containing other objects. Also it is important that the developed system is intelligent such that image scene constraints such as translation and scale only slightly affect the overall efficiency of the system.

6. Conclusion

In this research, an artificial vision system for left ventricle detection has been developed. It is important to note that the work in itself is broader than the detection of only left ventricles, since the same insight and approach presented within this work can be used to realize the detection of other objects. Also, considering the broadness of the scenes and environments in which the developed system will be deployed, we opt to reformulating the detection task as that of a machine learning problem. This allows some robustness to the aforementioned scene constraints which may render the developed system quite erroneous on the detection task. A backpropagation neural network (BPNN) has been used as the learning system in this research. The BPNN is trained on samples of left ventricles and nonleft ventricles (random) collected from the same samples. For the detection of left ventricles in target images, a window size (40×40 pixels) corresponding to the size of the input to the BPNN is used to sample the target image in a nonoverlapping fashion. The developed system tested some randomly collected target images containing left ventricles without any scene constraints such as the scale, translation, illumination, and orientation of left ventricles contained in the target images. The developed system is seen to perform quite well in detecting cub objects, including scenarios, where left ventricles are even partially occluded. Furthermore, to show the effectiveness and robustness of the developed system, the left ventricles were contoured or detected by medical experts beside the system detection to show the effectiveness and accuracy of the proposed system of performing the left ventricles detection.

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 thank Dr. Hadi Sasani for comments on earlier versions of this paper.

References

[1] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference (ICML '09)*, pp. 609–616, ACM, Montreal, Canada, June 2009.

[2] O. K. Oyedotun, E. O. Olaniyi, A. Helwan, and A. Khashman, "Hybrid auto encoder network for iris nevus diagnosis considering potential malignancy," in *Proceedings of the International Conference on Advances in Biomedical Engineering (ICABME '15)*, pp. 274–277, September 2015.

[3] A. Helwan and R. H. Abiyev, "ISIBC: an intelligent system for identification of breast cancer," in *Proceedings of International Conference on Advances in Biomedical Engineering (ICABME '15)*, pp. 17–20, September 2015.

[4] A. Helwan and D. P. Tantua, "IKRAI: intelligent knee rheumatoid arthritis identification," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 1, article 18, 2016.

[5] A. Helwan, A. Khashman, E. O. Olaniyi, O. K. Oyedotun, and O. A. Oyedotun, "Seminal quality evaluation with RBF neural network," *Bulletin of the Transilvania University of Brasov, Series III: Mathematics, Informatics, Physics*, vol. 9, no. 2, 2016.

[6] M. Argyrou, D. Maintas, C. Tsoumpas, and E. Stiliaris, "Tomographic image reconstruction based on artificial neural network (ANN) techniques," in *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC '12)*, pp. 3324–3327, November 2012.

[7] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016.

[8] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," in *Proceedings of the 20th International Conference on Processing (ICIP '13)*, pp. 4034–4038, September 2013.

[9] H. G. Gouk and A. M. Blake, "Fast sliding window classification with convolutional neural networks," in *Proceedings of the 29th International Conference on Image and Vision Computing*, pp. 114–118, Hamilton, New Zealand, November 2014.

[10] D. Xie, L. Zhang, and L. Bai, "Deep learning in visual computing and signal processing," *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 1320780, pp. 1–13, 2017.

[11] M. Ranzato, Y.-L. Boureau, and Y. Le Cun, "Sparse feature learning for deep belief networks," in *Advances in Neural Information Processing Systems*, pp. 1185–1192, 2008.

[12] P. Kumar, D. K. Gupta, V. N. Mishra, and R. Prasad, "Comparison of support vector machine, artificial neural network, and spectral angle mapper algorithms for crop classification using LISS IV data," *International Journal of Remote Sensing*, vol. 36, no. 6, pp. 1604–1617, 2015.

[13] K. Dimililer, "Backpropagation neural network implementation for medical image compression," *Journal of Applied Mathematics*, vol. 2013, Article ID 453098, 8 pages, 2013.

[14] P. Radau, Y. Lu, K. Connelly, G. Paul, A. J. Dick, and G. A. Wright, "Evaluation framework for algorithms segmenting short axis cardiac MRI," *The MIDAS Journal—Cardiac MR Left Ventricle Segmentation Challenge*, 2009, http://www.midasjournal.org/browse/publication/658.

[15] C. Constantinides, E. Roullot, M. Lefort, and F. Frouin, "Fully automated segmentation of the left ventricle applied to cine MR images: description and results on a database of 45 Subjects," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '12)*, pp. 3207–3210, San Diego, Calif, USA, September 2012.

[16] S. Huang, J. Liu, L. Lee et al., "Segmentation of the left ventricle from cine MR images using a comprehensive approach," in *Proceedings of the MICCAI 2009 Workshop on Cardiac MR Left Ventricle Segmentation Challenge. MIDAS Journal*, London, UK, September 2009.

[17] S. Huang, J. Liu, L. C. Lee et al., "An image-based comprehensive approach for automatic segmentation of left ventricle from cardiac short axis cine MR images," *Journal of Digital Imaging*, vol. 24, no. 4, pp. 598–608, 2011.

[18] M. Jolly, "Fully automatic left ventricle segmentation in cardiac cine MR images using registration and minimum surfaces," *The MIDAS Journal—Cardiac MR Left Ventricle Segmentation Challenge*, 2009, http://www.midasjournal.org/browse/publication/684.

[19] L. Marak, J. Cousty, L. Najman, and H. Talbot, "4D Morphological segmentation and the MICCAI LV-segmentation grand challenge," in *Proceedings of the MICCAI 2009 Workshop on Cardiac MR Left Ventricle Segmentation Challenge. MIDAS Journal*, pp. 1–8, MIDAS, France, November 2009, http://www.midasjournal.org/browse/publication/677.

Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 9635348, 8 pages
https://doi.org/10.1155/2017/9635348

Research Article

Deep Hashing Based Fusing Index Method for Large-Scale Image Retrieval

Lijuan Duan,[1,2] Chongyang Zhao,[1,3] Jun Miao,[4] Yuanhua Qiao,[5] and Xing Su[1]

[1] *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2] *Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, Beijing, China*
[3] *National Engineering Laboratory for Critical Technologies of Information Security Classified Protection, Beijing 100124, China*
[4] *School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China*
[5] *College of Applied Science, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Xing Su; xingsu@bjut.edu.cn

Received 31 March 2017; Accepted 26 April 2017; Published 24 May 2017

Academic Editor: Ridha Ejbali

Hashing has been widely deployed to perform the Approximate Nearest Neighbor (ANN) search for the large-scale image retrieval to solve the problem of storage and retrieval efficiency. Recently, deep hashing methods have been proposed to perform the simultaneous feature learning and the hash code learning with deep neural networks. Even though deep hashing has shown the better performance than traditional hashing methods with handcrafted features, the learned compact hash code from one deep hashing network may not provide the full representation of an image. In this paper, we propose a novel hashing indexing method, called the Deep Hashing based Fusing Index (DHFI), to generate a more compact hash code which has stronger expression ability and distinction capability. In our method, we train two different architecture's deep hashing subnetworks and fuse the hash codes generated by the two subnetworks together to unify images. Experiments on two real datasets show that our method can outperform state-of-the-art image retrieval applications.

1. Introduction

With the rapidly growing of images on the Internet, it is extremely difficult to find relevant images according to different people's needs. For example, nowadays the volume of images is becoming larger and larger, and a database having millions of images is quite common. Thus, a great deal of time and memory would be used in a linear search through the whole database. Moreover, images are always represented by real-valued features, so that the curse of dimension often occurred in many content-based image search engines and applications.

To address the inefficiency and the problem of memory cost of real-valued features, the ANN search [1] has become a popular method and a hot research topic in recent years. Among existing ANN techniques, hashing approaches are proposed to map images to compact binary codes that approximately preserve the data structure in the original space [2–6]. Due to the high query speed and low memory cost, the hashing and image binarization techniques have become the most popular and effective techniques to enhance identification and retrieval of information using content-based image recognition [4, 7–16]. Instead of real-valued features, images are represented by binary codes so that the time and memory costs of search can be greatly reduced [17]. However, the retrieval performance of most existing hashing methods heavily depends on the features they used, which are basically extracted in an unsupervised manner, thus more suitable for dealing with the visual similarity search than the semantic similarity search.

As we all know, the Convolutional Neural Network (CNN) has demonstrated its impressive learning power on image classification [5, 18–20], object detection [21], face recognition [22], and many other vision tasks [23–25]. The CNN used in these tasks can be regarded as a feature extractor guided by the objective function, specifically designed for the individual task [5]. The successful applications of CNN in various tasks imply that the features learned by CNN can well

capture the underlying semantic structure of images in spite of significant appearance variations. Moreover, hashing with the deep learning network has shown that both feature representation and hash coding can be learned more effectively.

Inspired by the robustness of CNN features and the high performance of deep hashing methods, we propose a binary code generating and fusing framework to index large-scale image datasets, named Deep Hashing based Fusing Index (DHFI).

In our method, firstly, we train two different deep pairwise hashing networks which take image pairs along with labels to indicate whether the two images are similar as training inputs and produce binary codes as outputs. Then, we merge the hash codes produced by the two subnetworks together and regard the merged hash code as a fingerprint or binary index of an image. Under these two stages, images can be easily encoded by forward propagating through the network and then merging the network outputs to binary hash code representation.

The rest of the paper is organized as follows: Section 2 discusses the related work to the method. Section 3 describes DHFI method in detail. Section 4 extensively evaluates the proposed method on two large-scale datasets. Section 5 gives concluding remarks.

2. Related Work

Existing learning methods can be divided into two categories: data-independent methods and data-dependent methods [8, 24, 26, 27].

The hash function in data-independent methods is typically randomly generated and is independent of any training data. The representative data-independent methods include locality-sensitive hashing (LSH) [1] and its variants. Data-dependent methods try to learn the hash function from some training data, which is also called learning to hash (L2H) methods [15, 26]. L2H methods can achieve comparable or better accuracy with shorter hash codes when compared to data-independent methods. In real applications, L2H methods have become more popular than data-independent methods.

Existing L2H methods can be further divided into two categories: unsupervised hashing and supervised hashing refer to a comprehensive survey [28].

Unsupervised hashing methods use the unlabeled training data only to learn hash functions and encode the input data points to binary codes. Typical unsupervised hashing methods include reconstruction error minimization [29, 30], graph based hashing [3, 31], isotropic hashing (IsoHash) [9], discrete graph hashing (DGH) [32], scalable graph hashing (SGH) [33], and iterative quantization (ITQ) [8].

Supervised hashing utilizes information, such as class labels, to learn compact hash codes. Representative supervised hashing methods include binary reconstruction embedding (BRE) [7], Minimal Loss Hashing (MLH) [34], Supervised Hashing with Kernels (KSH) [4], two-step hashing (TSH) [35], fast supervised hashing (FastH) [12], and latent factor hashing (LFH) [36]. In the pipelines of these methods, images are first represented by handcrafted visual descriptor feature vectors (e.g., GIST [37], HOG [38]), followed by separate projection and quantization steps to encode vectors into binary hash codes. However, such handcrafted feature represents the low level information of a picture whose construction process is independent of the hash function learning process, and the resulting features might not be optimally compatible with hash codes.

Recently, as the deep learning has shown its effective image representation power on high level semantic information in a picture, then, a lot of feature learning based deep hashing methods have recently been proposed and have shown their better performance than traditional hashing methods with handcrafted features, such as convolutional neural network hashing (CNNH) [39], network in network hashing (NINH) [40], deep hashing network (DHN) [41], and deep pairwise supervised hashing (DPSH) [15]. CNNH is proposed by Xia et al. The CNNH method first learns the hash codes from the pairwise labels and then tries to learn the hash function and feature representation from image pixels based on hash codes. Lai et al. improved the two-stage CNNH by proposing NINH. NINH uses a triplet ranking loss to preserve relative similarities and the hash codes of images are encoded by dividing and encoding modules. Moreover, this method is a simultaneous feature learning and hash coding deep network so that image representations and hash codes can improve each other in the joint learning process. DHN further improves NINH by controlling the quantization error in a principled way and devising a more principled pairwise cross entropy loss to link the pairwise Hamming distances with the pairwise similarity labels, while DPSH learns hash codes by learning features and hash codes simultaneously with pairwise labels. Due to the fact that different components in deep pairwise supervised hashing (DPSH) can give feedback to each other, DPSH outperforms other methods in image retrieval application as far as we know.

In this work, we further improve the retrieval accuracy by two steps: (1) training two different architecture's deep hashing subnetworks and (2) fusing the hash codes generated by the two subnetworks to unify images so that the merged codes can represent more semantic information and support each other. These two important stages constitute the DHFI approach.

3. The Proposed Approach

In this section, we describe our method in detail. We first train two different architecture's deep hashing subnetworks. Then, we perform each image through the subnetworks to generate binary hash codes and fuse the hash codes generated by the same image together. For the first step discussed in Section 3.1, we follow the simultaneous feature learning and hash code learning method of [15]. The major novelty of our method is training two deep hashing subnetworks and fusing the hash codes generated by the two subnetworks together to index images.

3.1. Subnetwork Training. We have n images (feature points) $\chi = \{x_1, x_2, \ldots, x_n\}$ and the training set of supervised hashing

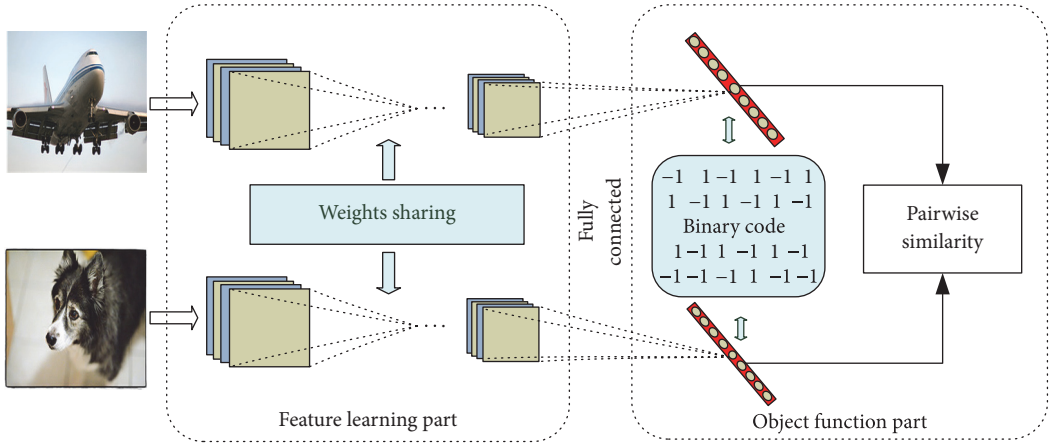


Figure 1: The end-to-end deep hash network learning architecture.

with pairwise labels also contains a set of pairwise labels $S = \{s_{ij}\}$ with $s_{ij} \in \{0, 1\}$, where $s_{ij} = 1$ means that x_i and x_j are similar and $s_{ij} = 0$ means that x_i and x_j are dissimilar. Here, the pairwise labels typically refer to semantic labels provided with manual efforts.

The goal of supervised hashing with pairwise labels is to learn a binary code $b_i \in \{-1, 1\}^c$ for each point x_i, where c is the code length. The binary code $B = \{b_i\}_{i=1}^n$ should preserve the similarity in S. More specifically, if $s_{ij} = 1$, then binary codes b_i and b_j should have a low Hamming distance; if $s_{ij} = 0$, the binary codes b_i and b_j should have a high Hamming distance. In general, we can write the binary code as $b_i = h(x_i) = [h_1(x_i), h_1(x_i), \ldots, h_c(x_i)]^T$, where $h(x_i)$ is the hash function to learn. For the subnetworks training step, we use the model and learning method called deep pairwise supervised hashing (DPSH) from Li et al. The model is an end-to-end deep learning method, which consists of two parts: the feature learning part and the objective function part.

The feature learning part has seven layers, which are the same as those of fast architecture's Convolutional Neural Network (CNN-F) in [42, 43].

As for the objective function part, given the binary codes $B = \{b_i\}_{i=1}^n$ for all the images, the likelihood of pairwise labels $S = \{s_{ij}\}$ can be defined as that of LFH [36]:

$$p\left(s_{ij} \mid B\right) = \begin{cases} \sigma\left(\Omega_{ij}\right), & s_{ij} = 1 \\ 1 - \sigma\left(\Omega_{ij}\right), & s_{ij} = 0, \end{cases} \tag{1}$$

where $\Omega_{ij} = (1/2) b_i^T b_j$ and $\sigma(\Omega_{ij}) = 1/(1 + e^{-\Omega_{ij}})$. Please note that $b_i \in \{-1, 1\}^c$. When taking the negative log-likelihood of the observed pairwise labels in S, the problem becomes an optimization problem:

$$\begin{aligned} \min_B \quad J_1 &= -\log p\left(S \mid B\right) = -\sum_{S_{ij} \in S} \log p\left(s_{ij} \mid B\right) \\ &= -\sum_{S_{ij} \in S} \left(s_{ij} \Omega_{ij} - \log\left(1 - e^{\Omega_{ij}}\right)\right). \end{aligned} \tag{2}$$

The optimization problem above can make the Hamming distance between two similar images (points) as small as possible and make the Hamming distance between two dissimilar images (points) as large as possible simultaneously. While the problem is a discrete optimization problem, which is difficult to solve, we follow the strategy designed by Li et al., to reformulate the problem as follows:

$$\begin{aligned} \min_{B,u} \quad & J_2 = -\sum_{s_{ij} \in S} \left(s_{ij} \theta_{ij} - \log\left(1 + e^{\theta_{ij}}\right)\right) \\ \text{s.t.} \quad & u_i = b_i, \quad \forall i = 1, 2, \ldots, n, \\ & u_i \in R^{c \times 1}, \quad \forall i = 1, 2, \ldots, n, \\ & b_i \in \{-1, 1\}^c, \quad \forall i = 1, 2, \ldots, n, \end{aligned} \tag{3}$$

where $\theta_{ij} = (1/2) u_i^T u_j$ and $U = \{u_i\}_{i=1}^n$. And the problem can be continually optimized by moving the equality constraints in the equation to the regularization terms.

$$\begin{aligned} \min_{B,u} \quad & J_3 \\ &= -\sum_{s_{ij} \in S} \left(s_{ij} \theta_{ij} - \log\left(1 + e^{\theta_{ij}}\right)\right) + \eta \sum_{i=1}^{n} \left\| b_i - u_i \right\|_2^2, \end{aligned} \tag{4}$$

where η is the regularization term.

A fully connected hash layer is designed between the two parts to integrate them to a whole framework. The framework is shown in Figure 1. Please note that two images are input into the framework at each training time, and the loss function is based on pair labels of images.

For the hash layer, we set

$$u_i = W^T \phi\left(x_i; \theta\right) + v, \tag{5}$$

where θ denotes all the parameters of the first seven layers in the feature learning part, $\phi(x_i; \theta)$ denotes the output of the

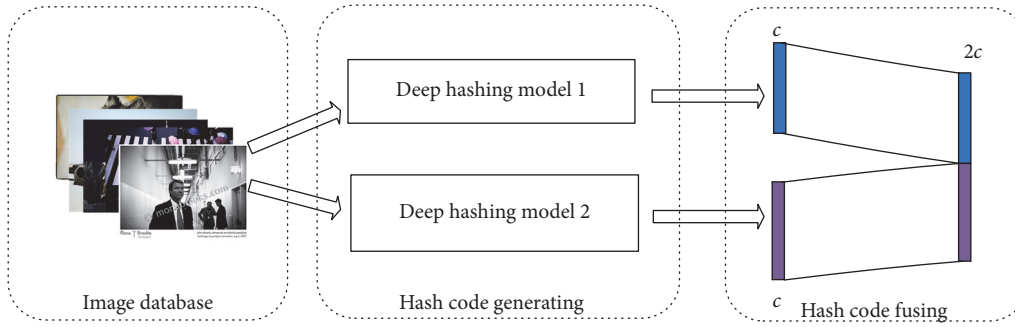


Figure 2: The deep hashing based fusing index learning architecture.

seventh layer associated with image (point) x_i, $W \in \mathbb{R}^{4096\times c}$ denotes a weight matrix, and $v \in \mathbb{R}^{c\times 1}$ is a bias vector.

After connecting the feature learning part and the objective function together, the problem of learning becomes

$$\min_{B,U} \quad J_3 - \sum_{s_{ij}\in S}\left(s_{ij}\theta_{ij} - \log\left(1 + e^{\theta_{ij}}\right)\right) + \eta\sum_{i=1}^{n}\left\|b_i - \left(W^T\phi\left(x_i;\theta\right) + v\right)\right\|_2^2. \tag{6}$$

In each subnetwork, following Li et al., we also adopt the minibatch based strategy and alternating method to learn the parameters containing W, v, θ, and B. We sample a minibatch of images (points) from the whole training set and each subnetwork learns based on these sampled images (points). Then, we optimize one parameter with other parameters fixed. b_i can be directly optimized as follows:

$$b_i = \operatorname{sgn}\left(u_i\right) = \operatorname{sgn}\left(W^T\phi\left(x_i;\theta\right) + v\right). \tag{7}$$

We use the back-propagation method to learn other parameters W, v, and θ. Specially, we can compute the derivatives of the loss function with respect of u_i as follows:

$$\frac{\partial J}{\partial u_i} = \frac{1}{2}\sum_{j:s_{ij}\in S}\left(a_{ij} - s_{ij}\right)u_j + \frac{1}{2} \cdot \sum_{j:s_{ji}\in S}\left(a_{ji} - s_{ji}\right)u_j + 2\eta\left(u_i - u_j\right), \tag{8}$$

where $a_{ij} = \sigma((1/2)u_i^T u_j)$. Then, we can update the parameters W, v, and θ by back-propagation:

$$\frac{\partial J}{\partial W} = \phi\left(x_i;\theta\right)\left(\frac{\partial J}{\partial u_i}\right)^T, \quad \frac{\partial J}{\partial V} = \frac{\partial J}{\partial u_i}, \quad \frac{\partial J}{\phi\left(x_i;\theta\right)} = W\frac{\partial J}{\partial u_i}. \tag{9}$$

In our method, we trained two deep hashing subnetworks by utilizing the learning algorithm in [15]. More specially, the CNN-F and the Caffe-alex [18] pretrained networks are separately used in the feature learning part of the different subnetworks.

3.2. Hash Codes Generating and Fusing. After we have successfully completed the training of subnetworks, we can only get the hash codes for images in the training data. We still have to predict the hash codes for other images which did not appear in the training set. For any image $x_q \in X$, we let it through each subnetwork to predict its hash codes just by forward propagation:

$$b_q = h\left(x_q\right) = \operatorname{sgn}\left(W^T\phi\left(x_q;\theta\right) + v\right). \tag{10}$$

Thus we can get two hash codes related to x_q. We concatenate the two different hash codes learned from the two different subnetworks together in a vector way and use the concatenated code as the latest hash code of x_q. The hash code generating and fusing process is shown in Figure 2.

4. Experiments

4.1. Experimental Settings. All our experiments for DHFI are completed with MatConvNet [43] on a NVIDIA K40 GPU server.

In this section, we conduct extensive evaluations of the proposed method on two widely used benchmark datasets with different kinds of images: CIFAR-10 and NUS-WIDE. (1) The CIFAR-10 [44] dataset consists of 60K 32×32 color tiny images which are categorized into 10 classes (6K tiny images per class). It is a single-label dataset in which each image belongs to one of the 10 classes. (2) The NUS-WIDE dataset [45, 46] has nearly 270K images collected from the web. It is a multilabel dataset in which each image is annotated with one or multiple class labels in 81 semantic concepts. Following [15, 40], we only use the images from the 21 most frequent classes. For these classes, the number of images in each class is at least 5K.

The experimental protocols in [15] are also employed in our experiments. In CIFAR-10, 1000 images (100 images per class) are randomly selected as the query set. For the

Table 1: Accuracy in terms of MAP compared to two different deep DPSH models.

Method	CIFAR-10				NUS-WIDE			
	24 bits	32 bits	48 bits	64 bits	24 bits	32 bits	48 bits	64 bits
DPSH1	0.727	0.744	0.757	0.768	0.822	0.838	0.845	0.850
DPSH2	0.686	0.714	0.745	0.736	0.828	0.838	0.846	0.849
DHFI	**0.750**	**0.768**	**0.774**	**0.788**	**0.836**	**0.854**	**0.860**	**0.864**

unsupervised methods, we use the rest images as the training set. For the supervised methods, we randomly select 5000 images (500 images per class) from the rest of the images as the training set. The pairwise label set S is constructed based on the image class labels, where two images will be considered to be similar if they share the same class label.

In NUS-WIDE, 2100 query images from 21 most frequent labels (100 images per class) are randomly sampled as the query set by following the strategy used in [15, 39, 40]. For the supervised methods, we randomly select 500 images per class from the rest images as the training set. The pairwise label set S is constructed based on the image class labels. It means that two images will be considered to be similar if they share at least one common label.

Following [15], we compare our method to several state-of-the-art hashing methods, including SH [31], ITQ [8], SPLH [47], KSH [4], FastH [12], LFH [36], SDH [13], DPSH [15], CNNH [39], DHN [41], DSH [5], and NINH [40]. Note that SH and ITQ are unsupervised hashing methods and the other methods are supervised hashing methods. DPSH, CNNH, DHN, and DSH are four deep hashing methods with pairwise labels, while NINH is a triplet-based method. Beyond this, we also evaluate the nondeep hashing methods with deep features extracted by the CNN-F.

For hashing methods which use handcrafted features, we represent each image in CIFAR-10 by a 512-dimensional GIST vector. And we represent each image in NUS-WIDE by a 1134-dimensional low level feature vector, including 64-D color histogram, 144-D color correlogram, 73-D edge direction histogram, 128-D wavelet texture, 225-D block-wise color moments, and 500-D SIFT features.

For deep hashing methods, we first resize all images to 224×224 pixels and then directly use the raw image pixels as input and adopt the CNN-F network which has been pretrained on the ImageNet dataset to initialize the layers of feature learning part. Similar initialization strategy has also been adopted by other deep hashing methods [48].

For our method, we learn the hash codes separately from different architecture's pretrained networks; we use the fast architecture's Convolutional Neural Network (CNN-F) and Caffe-alex network to initialize the parameters.

4.2. Results and Discussion. The mean average precision (MAP) is often used to measure the accuracy in large-scale image retrieval applications. As most existing hashing methods, the MAP is used to measure the accuracy of the proposed method. For fair comparison, all of the methods use identical training and test sets. In this paper, the MAP value is calculated based on the top 5000 returned neighbors for NUS-WIDE dataset. The best MAP for each category in the tables are shown in boldface.

Firstly, to verify the effectiveness of deep binary hash code fusing, we compare our method to two different architecture's deep pairwise supervised hashing models; one uses the CNN-F pretrained model in the feature learning part and the other uses the Caffe-alex pretrained model in the feature learning part. The MAP results are listed in Table 1. Please note that DPSH1 uses CNN-F and DPSH2 uses Caffe-alex pretrained model. By comparing DHFI to DPSH1 and DPSH2, we find that DHFI can dramatically outperform both of them. It means that the integrated hash codes learned from different architecture's deep hashing subnetworks can get a better solution than hash codes generated from independent subnetwork.

Secondly, the MAP results of all methods are listed in Tables 2 and 3. Please note that, in Table 2, DPSH, DSH, DHN, NINH, and CNNH are deep hashing methods, and all the other methods are nondeep methods with handcrafted features. The results of NINH, CNNH, KSH, and ITQ are from [15, 39, 40], the results of DPSH are from [15], the results of DSH are from [5], and the results of DHN are from [41]. Please note that the above experimental settings and evaluation metrics are exactly the same as that in [15, 39, 40]. Hence, the comparison is reasonable. We can find that our method dramatically outperforms other baselines, including unsupervised methods, supervised methods with handcrafted features, and deep hashing methods with feature learning.

To further verify the effectiveness of the deep binary hash code fusing, we compare DHFI to other nondeep methods with deep features extracted by the fast architecture's Convolutional Neural Network (CNN-F). The results are shown in Table 3, where the notation of "+CNN" denotes that the methods use deep features as input. We can find that our method outperforms all the other nondeep baselines with deep features.

5. Conclusion

In this paper, we proposed a "two-stage" deep hashing based fusing index method for image retrieval. In the proposed method, we train two different architecture's deep hashing networks at first and then merge the hash codes generated from separate networks together to unify an image. Due to the fact that hash codes are learned from different networks and they may provide different information and supplement each other, the proposed method can learn better codes than other hashing methods. Experiments on real datasets show

Table 2: Accuracy in terms of MAP compared to hashing methods.

Method	CIFAR-10				NUS-WIDE			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
SH	0.127	0.128	0.126	0.129	0.454	0.406	0.405	0.400
ITQ	0.162	0.169	0.172	0.175	0.452	0.468	0.472	0.477
SPLH	0.171	0.173	0.178	0.184	0.568	0.589	0.597	0.601
LFH	0.176	0.231	0.211	0.253	0.571	0.568	0.568	0.585
KSH	0.303	0.337	0.346	0.356	0.556	0.572	0.581	0.588
SDH	0.285	0.329	0.341	0.356	0.568	0.600	0.608	0.637
FastH	0.305	0.349	0.369	0.384	0.621	0.650	0.665	0.687
CNNH	0.439	0.476	0.472	0.489	0.611	0.618	0.625	0.608
NINH	0.552	0.566	0.558	0.581	0.674	0.697	0.713	0.715
DHN	0.555	0.594	0.603	0.621	0.708	0.735	0.748	0.758
DSH	0.616	0.651	—	0.661	0.548	0.551	—	0.562
DPSH	**0.713**	0.727	0.744	0.757	0.747	0.822	0.838	0.845
DHFI	0.613	**0.750**	**0.768**	**0.774**	**0.807**	**0.836**	**0.854**	**0.860**

Table 3: Accuracy in terms of MAP compared to nondeep methods with deep features.

Method	CIFAR-10				NUS-WIDE			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
SH + CNN	0.183	0.164	0.161	0.161	0.621	0.616	0.615	0.612
ITQ + CNN	0.237	0.246	0.255	0.261	0.719	0.739	0.747	0.756
SPLH + CNN	0.299	0.330	0.335	0.330	0.753	0.775	0.783	0.786
LFH + CNN	0.208	0.242	0.266	0.339	0.695	0.734	0.739	0.759
KSH + CNN	0.488	0.539	0.548	0.563	0.768	0.786	0.790	0.799
SDH + CNN	0.478	0.557	0.584	0.592	0.780	0.804	0.815	0.824
FastH + CNN	0.553	0.607	0.619	0.636	0.779	0.807	0.816	0.825
DHFI	**0.613**	**0.750**	**0.768**	**0.774**	**0.807**	**0.836**	**0.854**	**0.860**

that our method has superior performance over state-of-the-art image retrieval applications.

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 Natural Science Foundation of China [Grant nos. 61370113, 61572004, 61650201, and 91546111], the Beijing Municipal Natural Science Foundation [Grant nos. 4152005 and 4162058], the Key Project of Beijing Municipal Education Commission [Grant no. KZ201610005009]; the Science and Technology Program of Tianjin [Grant no. 15YFXQGX0050], and the Science and Technology Planning Project of Qinghai Province [Grant no. 2016-ZJ-Y04].

References

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Foundations of Computer Science Annual Symposium on 51.1*, pp. 117–122, 2008.

[2] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 12, pp. 2393–2406, 2012.

[3] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[4] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2074–2081, Providence, RI, USA, June 2012.

[5] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR, 1b)*, pp. 2064–2072, 2016.

[6] K. Zhan, J. Guan, Y. Yang, and Q. Wu, "Unsupervised discriminative hashing," *Journal of Visual Communication & Image Representation*, vol. 40, pp. 847–851, 2016.

[7] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *NIPS*, pp. 1042–1050.

[8] Y. Gong et al., "Iterative quantization: a Procrustean approach to learning binary codes for large-scale image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 817–824, 2011.

[9] W. Kong and W. J. Li, "Isotropic hashing," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1646–1654, 2012.

[10] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, and L. Davis, "Predictable dual-view hashing," in *Proceedings of 30th International Conference on Machine Learning*, pp. 1328–1336, 2013.

[11] K. He, F. Wen, and J. Sun, "*K*-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proceedings of 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'*, pp. 2938–2945, 2013.

[12] G. Lin, C. Shen, Q. Shi, A. Van Den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proceedings of 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'*, pp. 1971–1978, usa, 2014.

[13] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'*, pp. 37–45, June 2015.

[14] K. Wang-Cheng, L. Wu-Jun, and Z. Zhi-Hua, "Column sampling based discrete supervised hashing," in *AAAI*, 2016.

[15] L. Wujun, W. Sheng, and K. Wangcheng, "Feature learning based deep supervised hashing with pairwise labels," in *IJCAI*, 2016.

[16] R. Das, S. Thepade, S. Bhattacharya, and S. Ghosh, "Retrieval Architecture with classified query for content based image recognition," *Applied Computational Intelligence and Soft Computing*, vol. 2016, 2 pages, 2016.

[17] Y. Xu, F. Shen, X. Xu, L. Gao, Y. Wang, and X. Tan, "Large-scale image retrieval with supervised sparse hashing," *Neurocomputing*, vol. 229, pp. 45–53, 2017.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks.," in *Proceedings of International Conference on Neural Information Processing Systems Curran Associates Inc*, pp. 1097–1105, 2012.

[19] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1026–1034, IEEE, Santiago, Chile, December 2015.

[21] C. Szegedy, A. Toshev, and D. Erhan, "Deep Neural Networks for object detection," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2553–2561, 2013.

[22] Y. Sun, X. Wang, and X. Tang, *Deep Learning Face Representation by Joint Identification-Verification*, vol. 27, 2015.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 3431–3440, IEEE, Boston, Mass, USA, June 2015.

[24] Y. Liu, Y. Pan, H. Lai, C. Liu, and J. Yin, "Margin-based two-stage supervised hashing for image retrieval," *Neurocomputing*, vol. 214, pp. 894–901, 2016.

[25] D. Xie, L. Zhang, and L. Bai, "Deep Learning in Visual Computing and Signal Processing," *Applied Computational Intelligence and Soft Computing*, vol. 2017, pp. 1–13, 2017.

[26] J. Deng, N. Ding, Y. Jia et al., "Large-scale object classification using label relation graphs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8689, pp. 48–64, 2014.

[27] Z. Songhao et al., "Integration of semantic and visual hashing for image retrieval," *Journal of Visual Communication & Image Representation*, 2016.

[28] W. Kong and W. J. Li, "Isotropic hashing," *Advances in Neural Information Processing Systems*, vol. 2, no. 2012, pp. 1646–1654, 2012.

[29] W. Jingdong et al., "Hashing for similarity search: a survey," *Computer Science*, 2014.

[30] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," *Journal of Machine Learning Research*, vol. 2, pp. 412–419, 2007.

[31] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December DBLP*, pp. 1753–1760, 2008.

[32] L. Wei, "Discrete graph hashing," in *Proceedings of International Conference on Neural Information Processing Systems MIT Press*, pp. 3419–3427, 2014.

[33] Q. Jiang Yuan and W. J. Li, "Scalable graph hashing with feature transformation," in *Inproceeding of International Conference on Artificial Intelligence AAAI Press*, pp. 2248–2254, 2015.

[34] N. Mohammad Emtiyaz and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proceedings of nternational Conference on Machine Learning*, pp. 353–360, Bellevue, Washington, USA, 2011.

[35] G. Lin, C. Shen, D. Suter, and A. V. D. Hengel, "A general two-step approach to learning-based hashing," in *Proceedings of 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 2552–2559, aus, December 2013.

[36] P. Zhang, W. Zhang, W.-J. Li, and M. Guo, "Supervised hashing with latent factor models," in *Proceedings of 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2014*, pp. 173–182, aus, July 2014.

[37] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

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

[39] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," *AAAI Conference on Artificial Intelligence*, pp. 2156–2162, 2014.

[40] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'*, pp. 3270–3278, June 2015.

[41] Z. Han, "Deep hashing network for efficient similarity retrieval," in *Proceedings of Thirtieth AAAI Conference on Artificial Intelligence AAAI Press*, pp. 2415–2421, 2016.

[42] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," *Computer Science*, 2014.

[43] A. Vedaldi and K. Lenc, "MatConvNet: convolutional neural networks for matlab," in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 689–692, Brisbane, Australia, October 2015.

[44] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2012, Learning Multiple Layers of Features from Tiny Images.

[45] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from national university of singapore," in *Proceedings of ACM International Conference on Image and Video Retrieval, CIVR 2009*, pp. 368–375, grc, July 2009.

[46] X. Zhao, X. Li, and Z. Zhang, "Multimedia retrieval via deep learning to rank," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1487–1491, 2015.

[47] J. Wang, S. Kumar, and S. F. Chang, "Sequential Projection Learning for Hashing with Compact Codes," in *Inproceeding of International Conference on Machine Learning DBLP*, pp. 1127–1134, 2010.

[48] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 1556–1564, June 2015.