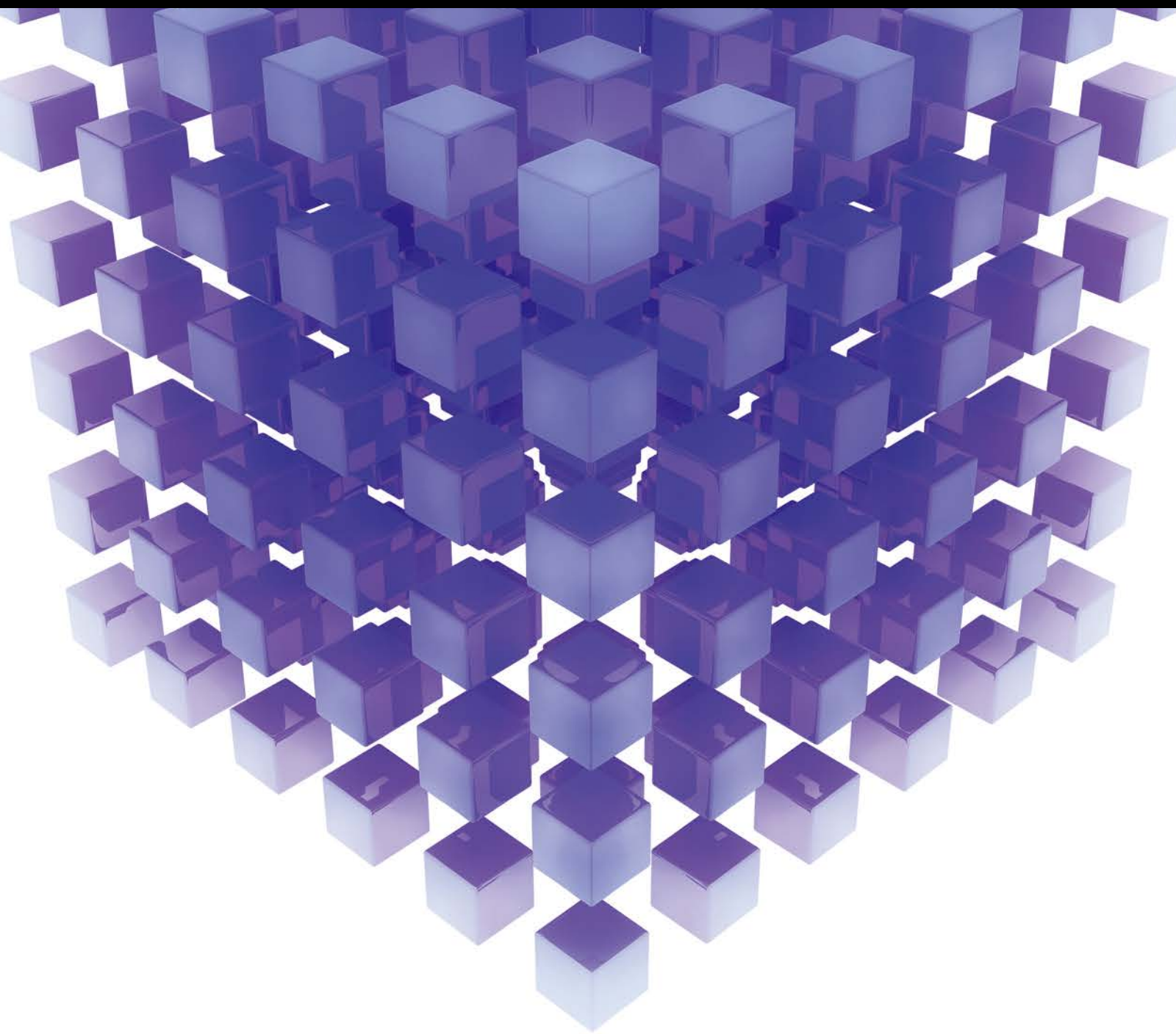
Mathematical Problems in Engineering

Optimization for Detection and Recognition in Images and Videos

Guest Editors: Wonjun Kim, Chanho Jung, and Simone Bianco



Optimization for Detection and Recognition in Images and Videos

Mathematical Problems in Engineering

Optimization for Detection and Recognition in Images and Videos

Guest Editors: Wonjun Kim, Chanho Jung, and Simone Bianco

Editorial Board

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 5190490, 2 pages
http://dx.doi.org/10.1155/2017/5190490

Editorial

Optimization for Detection and Recognition in Images and Videos

Wonjun Kim,[1] Chanho Jung,[2] and Simone Bianco[3]

[1]*Department of Electronics Engineering, Konkuk University, Seoul, Republic of Korea*
[2]*Department of Electrical Engineering, Hanbat National University, Daejeon, Republic of Korea*
[3]*Department of Informatics, Systems and Communication, University of Milan-Bicocca, Milan, Italy*

Correspondence should be addressed to Wonjun Kim; wonjkim@konkuk.ac.kr

Received 27 November 2016; Accepted 28 November 2016; Published 2 March 2017

This special issue aims at providing the new optimization techniques for detection and recognition in computer vision. The optimization techniques have been widely understood and employed to accurately find a solution for a given task in the field of computer vision. For example, visual SLAM (simultaneous localization and mapping), camera calibration, denoising, and segmentation can be robustly implemented based on optimization algorithms [1–3]. Even though their great capability of clearly solving various problems has been proved, most of them are hardly deployed into the mobile and robot platforms due to the heavy computation, which requires quite a lot iterations with complex operations. To cope with this limitation, many researchers have devoted considerable efforts for constructing simple yet powerful optimization frameworks.

The most important thing to resolve a given problem by using the optimization technique is to efficiently construct the sophisticated probabilistic and statistical models. However, those have been restrictively applied to somewhat traditional problems as mentioned above in computer vision. With rapidly increasing demand on the high-level intelligence in mobile and robot platforms, such models need to be applied to more advanced applications such as object detection, classification, and recognition [4, 5]. In this point of view, articles published in this special issue show the plentiful possibilities in computer vision and machine learning. Specifically, some are devoted to solving the problem of face detection and recognition with novel optimization techniques, which is one of the hottest issues in this field. Part of articles attempts to precisely segment salient regions in a given image by optimizing the graph model while another part deals with the more specific problems in images and videos, for example, hyperspectral image classification and highway visibility detection. All things considered, it is thought that articles of the present special issue give a great contribution to object detection and recognition.

X. Lu et al. propose fusing features from two different neural networks for face detection in their article titled as "Feature Extraction and Fusion using Deep Convolutional Neural Networks for Face Detection." To extract features for describing the face, they utilize Clarifai and VGG (16 layers) networks and conduct the feature optimization with the PCA scheme during the training phase. The proposed method is well evaluated based on two benchmark DBs (i.e., FDDB and AFW).

In the article "Customized Dictionary Learning for Subdatasets with Fine Granularity" by L. Ye et al., authors propose customizing the global dictionary, which is already trained based on a variety of subjects, corresponding to the target people for improving the performance of face recognition. To do this, they employ a regularizer penalizing the difference between global and subdataset dictionaries under the sparsity constraint and demonstrate the reconstruction performance on the benchmark dataset.

R. Huang et al. exploit a new neural network structure based on the supervised the autoencoder in their article titled as "Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition." In this method, the

characteristic features from corrupted and clean faces are exactly trained with their labels; thus, the reconstruction is efficiently achieved using the low-dimensional feature vector. A test image can be recognized by comparing its reconstructed image with individual gallery images. Authors show the performance improvement for face recognition by their adaptive scheme on various datasets, for example, AR, PubFig, and extended Yale B.

H. Sima et al. introduce a simple optimization scheme for merging superpixels for image segmentation in their article titled as "Objectness Supervised Merging Algorithm for Color Image Segmentation." Specifically, they build two hierarchy segmentation models, which are based on the region growing and color features, respectively. It is noteworthy that, for smoothing out textures and measuring the objectness, authors propose conducting the total variation-based optimization, which is efficiently yielding the real boundary condition (i.e., object boundary). Both color and objectness features are computed to check the regional similarity between superpixel pairs, and the mixed standard deviation of the union feature is computed to adaptively stop the merging process. Authors demonstrate that this combined approach reliably provides the object boundary on the segmentation spaces in diverse natural images.

In the article "Graph-Based Salient Region Detection through Linear Neighborhoods" by L. Xu et al., authors formulate the problem of salient region detection as the process of graph labeling by learning from partially selected seed (i.e., labeled data) in the graph. Based on the assumption that every node in the graph can be optimally reconstructed by a linear combination of its neighbor nodes, the undirected weight graph is constructed instead of Gaussian-based pairwise weighting, which improves the detection performance in cluttered background. Since the salient region can be directly and efficiently applied to a wide range of computer vision fields, for example, image classification and object recognition, it can be seen as a preprocessing step for further applications.

Y. Tang et al. establish two dictionaries (i.e., background and union) for pixel representation to improve the performance of hyperspectral image classification in their article titled as "Sparse Representation Based Binary Hypothesis Model for Hyperspectral Image Classification." Since the proposed dictionary is built on the neighbor regions centered at each pixel position, the performance is robust to the noise in captured image. Moreover, the kernel method is employed to improve the interclass separability. In the high-dimensional feature space induced by the kernel function, a coding vector is finally computed by the kernel-based orthogonal matching suit (KOMP).

In the article "Visibility Video Detection with Dark Channel Prior on Highway" by J. Zhao et al., authors attempt to remove the different illumination condition for image enhancement under hazy condition on the highway. To do this, they combine many different techniques with the optimization scheme and provide the plentiful experimental results.

In summary, this special issue handles various issues attracting considerable interests in the field of detection and recognition for images and videos. Despite the certain randomness in the submission of manuscripts for publication, we believe that this special issue could be interesting to all those who have to deal with computer vision and machine learning in the vast field of engineering.

Acknowledgments

We would like to thank the authors of articles published in the issue for their contribution. We are grateful to all anonymous reviewers for their valuable work.

Wonjun Kim
Chanho Jung
Simone Bianco

References

[1] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 24–29, Barcelona, Spain, April 2005.

[2] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted schatten p-norm minimization for image denoising and background subtraction," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4842–4857, 2016.

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[4] D. Novotny and J. Matas, "Cascaded sparse spatial bins for efficient and effective generic object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1152–1160, Santiago, Chile, December 2015.

[5] H. Liu, D. Guo, and F. Sun, "Object recognition using tactile measurements: kernel sparse coding methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 656–665, 2016.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2017, Article ID 1376726, 9 pages
http://dx.doi.org/10.1155/2017/1376726

Research Article

Feature Extraction and Fusion Using Deep Convolutional Neural Networks for Face Detection

Xiaojun Lu, Xu Duan, Xiuping Mao, Yuanyuan Li, and Xiangde Zhang

College of Sciences, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Xiangde Zhang; zhangxiangde@mail.neu.edu.cn

Received 12 August 2016; Revised 17 October 2016; Accepted 26 October 2016; Published 24 January 2017

Academic Editor: Wonjun Kim

This paper proposes a method that uses feature fusion to represent images better for face detection after feature extraction by deep convolutional neural network (DCNN). First, with Clarifai net and VGG Net-D (16 layers), we learn features from data, respectively; then we fuse features extracted from the two nets. To obtain more compact feature representation and mitigate computation complexity, we reduce the dimension of the fused features by PCA. Finally, we conduct face classification by SVM classifier for binary classification. In particular, we exploit offset max-pooling to extract features with sliding window densely, which leads to better matches of faces and detection windows; thus the detection result is more accurate. Experimental results show that our method can detect faces with severe occlusion and large variations in pose and scale. In particular, our method achieves 89.24% recall rate on FDDB and 97.19% average precision on AFW.

1. Introduction

Face detection is a classical problem in computer vision, which is widely used for all facial analysis algorithms, including face recognition, face tracking, and facial attribute recognition (e.g., gender, age, and facial expression recognition). However, due to large variations in pose, blur, occlusion, and illumination condition, face detection is still confronted with some challenges.

Since seminal work of Viola and Jones [1], face detection has made great progress in recent years. Viola-Jones detector adopted Adaboost classifier with cascade structure to achieve real-time face detection. Nevertheless, due to simplicity of Haar-like features extracted manually, it fails to detect faces with pose variations, exaggerated expressions, and extreme illumination. Later deformable part models (DPM) [2] detector adopted part models based on pictorial structure for deformation of objects and proposed a detection model to study the parts and their relations. This method is robust to partial occlusion but with higher computational cost. Nowadays, with the availability of massive data and the improvement of computing power, deep convolutional neural network has recently achieved remarkable performance in many computer vision tasks, including image classification, object detection, and face recognition. Farfade et al. [3] propose Deep Dense Face Detector (DDFD) for multiview face detection; however it fails to detect faces with heavy occlusion or blur. Li et al. [4] put forward a cascade structure based on CNN and adjust the location of detection windows by rectification for face detection, but this needs additional computational costs, thus resulting in high computational complexity. Yang et al. [5] exploit scoring facial parts responses by the spatial structure and arrangement for face detection, which can deal with severe occlusion and unconstrained pose variations but with higher computational complexity. Therefore, detection algorithms need a trade-off between detection performance and speed.

This paper proposes a feature extraction and fusion method for face detection by DCNN and achieves the state-of-the-art performance on FDDB [6], AFW [7], and LFW [8] dataset. The rest of this paper is organized as follows. The proposed method is presented in Section 2. Experiments and results are provided in Section 3. Finally, we draw the conclusions in Section 4.

2. The Proposed Method

The framework of the proposed method is shown in Figure 1. In our method, first, we learn and extract feature of input images at fc6 layer of Clarifai net [9] and VGG Net-D (16 layers) [10]. Then, we fuse the features of the two networks. To obtain more compact feature and mitigate computation complexity, PCA is adopted to reduce feature dimension. Finally, we conduct binary classification by SVM to realize face detection on images. The following subsections will discuss the procedure in detail.

2.1. Feature Extraction by DCNN. In this paper, pretrained Clarifai net and VGG Net-D (16 layers) model are used for fine-tuning these two networks. Clarifai net adopts kernels of size 7×7 in the first convolutional layer to filter images to obtain global information, which contains more context information, making it easier to separate faces from nonfaces but harder to handle partial occlusion. VGG Net-D (16 layers) network exploits smaller 3×3 convolution kernels to filter images to obtain local information, which contains higher resolution image information to address face detection under occlusion and blur, but without global superiority; for example, the region extracted from cheek is difficult to be confirmed as a part of face or not. Since both networks have strong ability to learn features and generalize well, we consider feature fusion of them to obtain global and local information simultaneously to distinguish faces from nonfaces more easily and be more robust to faces under partial occlusion, resulting in better performance.

This paper adopts sliding window approach to detect faces with different sizes on each image. We construct image pyramid with max scale of 8 and scaling factor of 0.9057, which is shown in Figure 2. Due to network input (detection window) of size 224×224, we can detect faces as small as size $((224/8) \times (224/8) =) 28 \times 28$.

Due to high computational complexity of original sliding window approach, we convert the fully connected layers into convolutional layers and reshape layer parameters; then we use the fully convolutional network to deal with input images of arbitrary sizes [11]. Figure 3 illustrates that each sliding window of size 6×6 at fc6-conv layer in fully convolutional network corresponds to a detection window of size 224×224 on original image; we can obtain features of all candidate regions by fully convolutional network with just one forward computation.

And similar to the approach introduced by Giusti et al. [12], we adopt multiple starting locations at the last pooling layer with each corresponding to a pooled feature map. We use max-pooling with stride of 2 for the last pooling layer; thus each input feature map generates 4 output feature maps as shown in Figure 4, which contain information of each candidate region on image for denser detection. More details are as follows.

We call each starting location as offset to avoid overlapping with a stride of 2 at the max-pooling layer; there are only $(2 \times 2 =)$ 4 offsets in O, defined as $O = \{(0,0), (1,0), (0,1), (1,1)\}$. Given an input feature map, one output feature map is obtained for each offset o ($o = (ox, oy) \in O$), where o is the coordinate of starting location at the top left on the input feature map for pooling. As shown in Figure 4, applying offset max-pooling with kernel size of 3×3 and stride of 2, one input feature map of size 7×7 can generate output feature maps of sizes 3×3, 3×2, 2×3, and 2×2 by starting at $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$ of the top left, respectively. And four output feature maps above correspond to $(3 \times 3 + 3 \times 2 + 2 \times 3 + 2 \times 2 =)$ 25 detection windows of size 3×3 on the input feature map. However, in case of traditional pooling operation, there is only one feature map of size 3×3, which corresponds to only 9 detection windows of size 3×3 on the input feature map. If we use a max-pooling with kernel size of 3×3 and stride of 1, an input feature map of size 7×7 can generate a feature map of size 5×5, which corresponds to 25 detection windows of size 3×3 on the input feature map. Thus our method is equivalent to reduce the stride by half to conduct denser detection.

2.2. Feature Fusion and Dimensionality Reduction by PCA. After feature extraction of each candidate region by the two networks above, these feature vectors of the same region are catenated to form higher dimensional fusion features. And this can compensate for inadequacy of single network in feature extraction. However, there always exist some correlation and information redundancy among these features, and higher dimensional features lead to higher computation complexity. Therefore, we adopt principle component analysis (PCA) for selection and dimensionality reduction of features. In this paper, we define the eigenvalue statistical rate as the ratio of number of principal components (eigenvalues) retained by PCA to number of all components. And we select the eigenvalue statistical rate as 50%, which means that eigenvectors corresponding to top 50% of principal components (eigenvalues) are selected to build projection direction matrix for dimensionality reduction of features. In Section 3.2, we compare effect on the experiment of different eigenvalue statistical rate in PCA.

Feature fusion helps to learn image features fully for description of their rich internal information, and after dimensionality reduction, we can obtain compact representation of integrated features, thus resulting in lower computational complexity and better performance of face detection with unconstrained environment.

2.3. Binary Classification Using SVM. The features, whose dimension is reduced by PCA after feature fusion, are used to train a SVM classifier for binary classification. And after comparison between polynomial kernel function and RBF kernel functions in Section 3.2, we finally choose polynomial kernel function with better classification results as final kernel function.

By the trained SVM model, we can score feature extracted from each candidate region, which is corresponding to the confidence of a detection box. Comparing confidences of candidate regions with preset threshold, regions with confidence higher than the threshold are labelled as faces, otherwise they are labelled as nonfaces. Despite slow detection speed, SVM classifier can result in smaller risk of wrong classification.

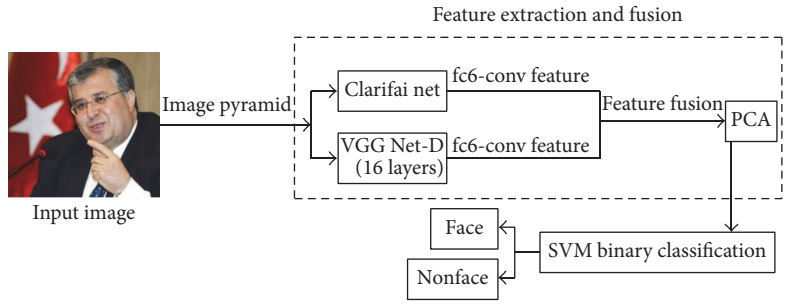


Figure 1: The framework of the proposed method.

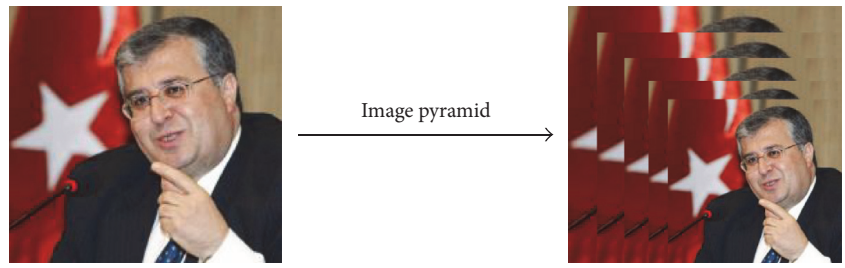


Figure 2: Sketch map of image pyramid.

2.4. Bounding Box Regression. Some methods with deep learning for object detection correct the position of detection box by bounding box regression, resulting in improvement of final detection accuracy [13]. Therefore, bounding box regression is introduced to our method, and comparison between results with/without bounding box regression is shown in Section 3.2. As shown in Figure 5, detection box P is a candidate region extracted by our detector, represented by (x_0, y_0, w_0, h_0), where (x_0, y_0) is the coordinate of top left of the detection box and w_0 and h_0 are defined as width and height of the detection box, respectively. And G is a ground truth bounding box for the face on image. The regression target is to learn a transformation that maps a detection box P to a ground truth bounding box G, and G' is our regression result.

For bounding box regression, candidate regions, whose IOU with ground truth bounding box are greater than a preset threshold, are used for training. After feature extraction and fusion for each candidate region by these two networks above, the features, whose dimension is reduced by PCA, are defined as Φ. And the regression target (t_x, t_y, t_w, t_h) is defined as

$$\begin{aligned} t_x &= \frac{(x - x_0)}{w_0}, \\ t_y &= \frac{(y - y_0)}{h_0}, \\ t_w &= \log\left(\frac{w}{w_0}\right), \\ t_h &= \log\left(\frac{h}{h_0}\right), \end{aligned} \tag{1}$$

where (x_0, y_0, w_0, h_0) represents a candidate region and (x, y, w, h) represents the ground truth bounding box. We learn a set of parameters W $(= (W_x, W_y, W_w, W_h))$ by optimizing the regularized least squares objective as

$$W_* = \arg\min_{W_{0^*}} \sum_{i}^{N} \left(t_*^i - \Delta_*^i\right)^2 + \lambda \left\| W_{0^*} \right\|^2, \tag{2}$$

where i is the number of training samples, $\Delta_*^i = W_{0^*}^T \Phi^i$, and $*$ is one of x, y, w, h, and each transformation corresponds to an optimization objective function.

At testing stage, after scoring each candidate region with SVM classifier, new bounding boxes for regions whose scores are larger than the preset threshold are obtained by bounding box regression with the trained transformation. And the regression result is defined as

$$\begin{aligned} x' &= w_0 \times \Delta_x + x_0, \\ y' &= h_0 \times \Delta_y + y_0, \\ w' &= w_0 \times \exp\left(\Delta_w\right), \\ h' &= h_0 \times \exp\left(\Delta_h\right), \end{aligned} \tag{3}$$

where (x', y', w', h') represents the detection result after bounding box regression.

2.5. Postprocessing of Detection Boxes. We have obtained multiscale detection information by image pyramid, and there is high overlap among output detection boxes. Therefore, we adopt non-maximum suppression (NMS) [14] for postprocessing of detection boxes. It aims at ensuring to obtain

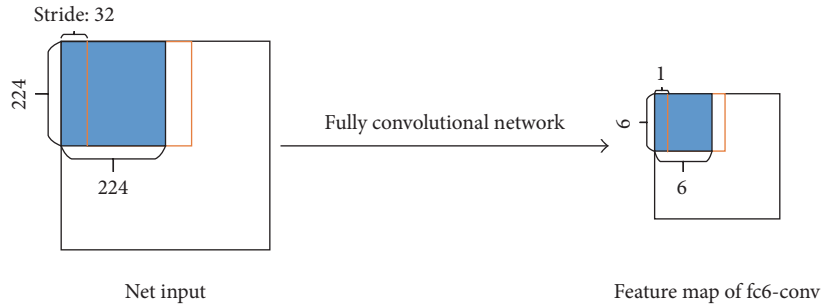


Figure 3: Sketch map of feature map in fully convolutional network.

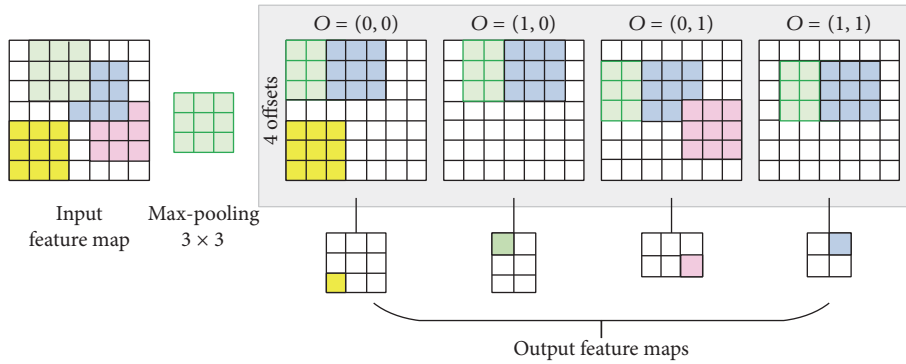


Figure 4: Image forward propagation techniques for max-pooling layers.

only one detection box per object by eliminating redundant overlapping detection boxes that refer to the same object to find optimal detection box for the object. When two objects on the image are in close distance, say, they are occluded by each other, in this case, we keep overlapping detection boxes referring to different objects. Common postprocessing methods include NMS-Max and NMS-Average.

We first apply NMS-Max and later NMS-Average in this paper. As for two detection boxes, IOU is taken as the overlap criterion, and the value of IOU is defined as the intersecting area divided by their union. After selecting the detection box with maximum score, NMS-Max removes the detection boxes whose IOU is larger than an overlap threshold. And then the NMS-Average is used to cluster the rest of detection boxes according to an overlap threshold. Within each cluster, we remove the detection boxes with score less than the maximum score of that cluster and average the locations of the remaining detection boxes to get the optimal detection box. And the maximum score of the cluster is used as the final score of the merged detection box. Figure 6 illustrates results after applying NMS-Max and NMS-Average.

3. Experiments and Results

3.1. Experimental Settings

3.1.1. Training Networks for Feature Extraction. WIDER FACE dataset [15] contains rich annotations, including occlusion, pose, and event categories. We cropped images of WIDER_train, and those are taken as positive samples if IOU between it and the ground truth bounding box is larger than 0.65. Due to larger proportion of small-scale samples in WIDER_train, we cropped WIDER_val dataset using the same standard and selected images whose size is larger than 80 pixels to form positive samples with WIDER_train to expand data for training. And we cropped images of AFLW [16], and those are taken as negative samples if IOU between it and the ground truth bounding box is smaller than 0.3; these cropped colorful images are all resized to network input size. As a preprocessing step, the input image is centered by subtracting the mean image created from a large dataset, and we expanded training set by mirror transformation for training net. Finally we update parameters with a batch size of 128 examples, initializing learning rate at 0.0001, momentum of 0.9, and ratio of 1 : 5 positives to negatives for fine-tuning.

3.1.2. Training SVM Classifier. After cropping WIDER_train and WIDER_val dataset according to ground truth annotations, we select a part of them as positive samples and crop images of AFLW are taken as negative samples if IOU between it and the ground truth bounding box is smaller than 0.3. Then we set the ratio of positive samples and negative ones to 1 : 1 to train SVM classifier.

3.1.3. Testing. We use FDDB, AFW, and LFW dataset as test sets. FDDB dataset is the benchmark of face detection, including faces with variations in occlusion, pose, and scene. Also, faces of out-of-focus are included. Comparisons of experimental results in 3.2 are conducted on FDDB.

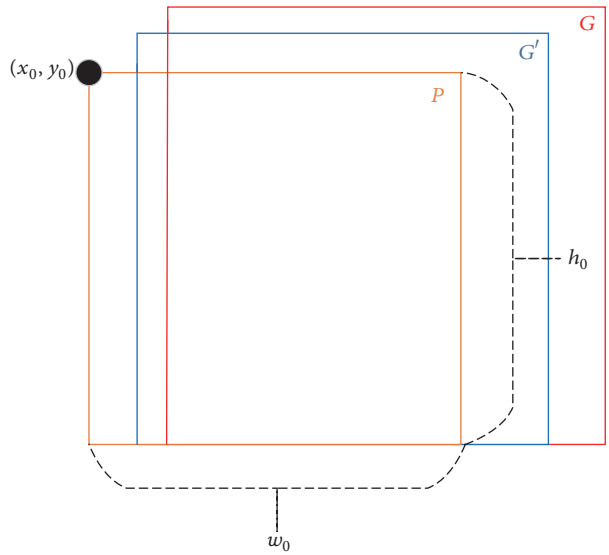
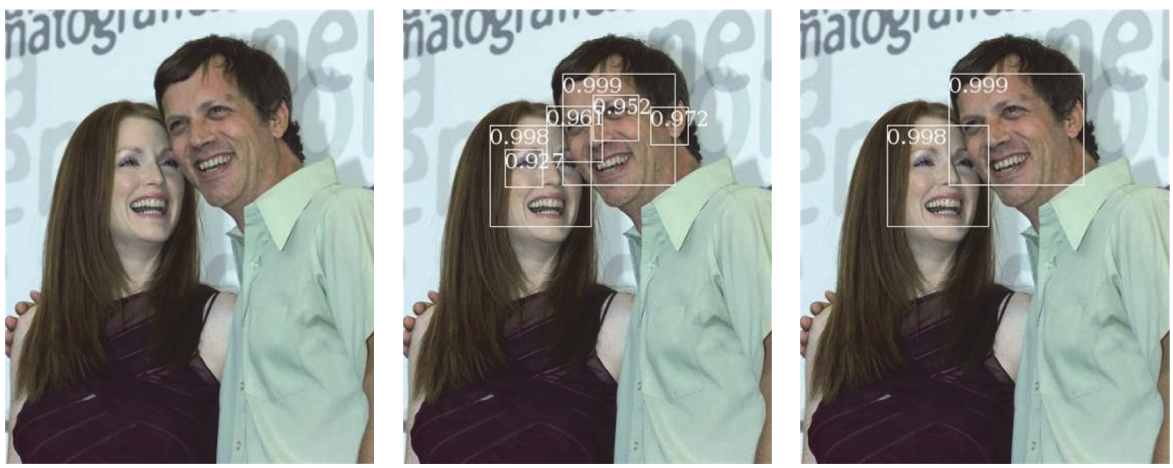


Figure 5: Sketch map of bounding box regression.



(a) (b) (c)

Figure 6: Results after applying NMS-Max and NMS-Average, where (a) is original image, (b) is result of applying NMS-Max, and (c) is result of applying NMS-Average.

AFW is released by Zhu et al., which includes 205 images with cluttered background with large variations in both face viewpoint and appearance (e.g., aging, sunglasses, makeups, skin color, and expression). LFW dataset is a challenge dataset for face verification in the wild. All images of LFW dataset are taken in real scene, which leads to natural variability in light, expressions, pose, and occlusion. People involved in LFW mostly are public figures, which results in more complex interference factor, such as makeup and spotlight. Therefore, we use LFW dataset for evaluating the proposed method. Since LFW dataset is used for following task of face alignment and recognition in the future, and only the central face on each image is needed for face recognition, we take the bounding box nearest the center of image as final detection result, in case there is more than one detected bounding box in an image. This postprocessing method can lead to no false positive and accuracy of 100%. In testing stage, we convert the fully connected layers into convolutional layers and reshape layer parameters and exploit offset max-pooling to extract features with sliding window densely, which leads to better matches of faces and detection windows. Taking each image of image pyramid as input of the fully convolutional network, we extract feature vector of each candidate region at fc6-conv layer and realize feature fusion and dimensionality reduction, and we can obtain a set of bounding boxes with confidence scores by SVM. Then we merge all boxes at each scale and apply NMS to get final detection results.

3.2. Comparisons of Experimental Results

3.2.1. The Effectiveness of Feature Fusion. In order to prove the feasibility of our method, we conduct contrast experiment on

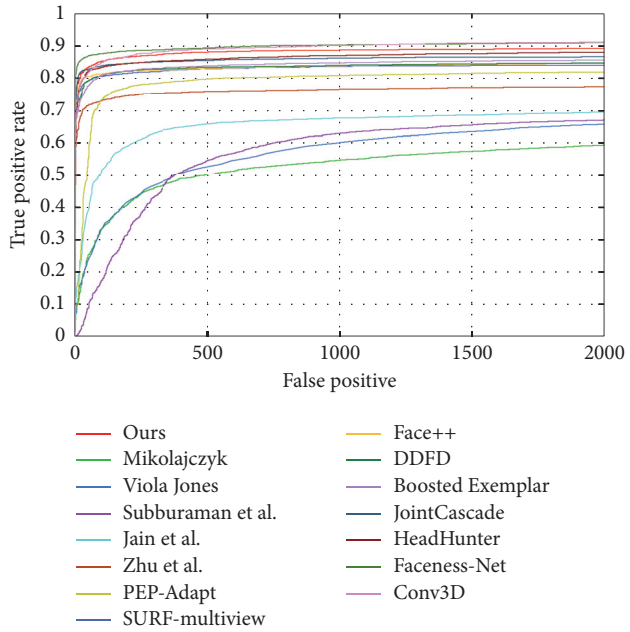


Figure 7: Comparisons of our method with other face detectors on FDDB dataset.

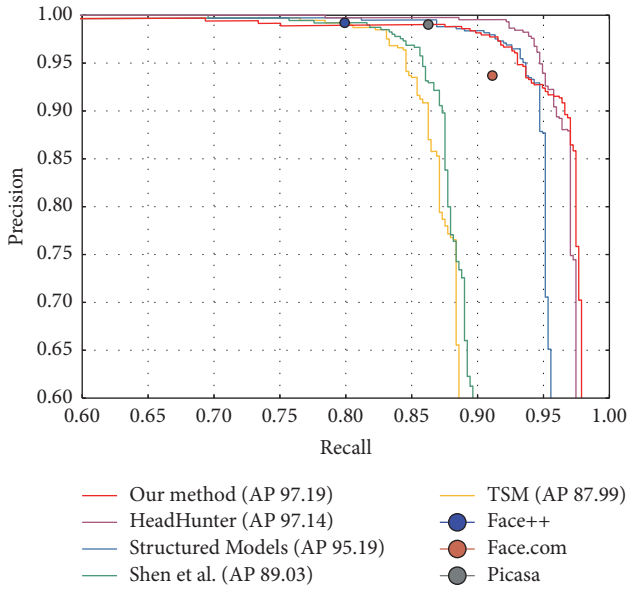


Figure 8: Comparisons of our method with other face detectors on AFW dataset.

Table 1: Comparison between the single net and feature fusion of these two networks on FDDB.

Network	Recall rate (%)	False positives
Clarifai net	86.46	2000
VGG Net-D (16 layers)	86.94	2000
Feature fusion of Clarifai and VGG	89.24	2000

FDDB and AFW before and after feature fusion, as shown in Tables 1 and 2.

Table 2: Comparison between the single net and feature fusion of these two networks on AFW.

Network	Average precision (%)
Clarifai net	96.78
VGG Net-D (16 layers)	96.83
Feature fusion of Clarifai and VGG	97.19

Table 3: Test results of the proposed face detector on FDDB with different eigenvalue statistical rate in PCA.

Eigenvalue statistical rate (%)	Recall rate (%)	False positives
50	89.24	2000
70	89.27	2000
90	88.64	2000

Table 4: Test results of the proposed face detector with two kernel functions of SVM classifier on FDDB.

Kernel function	Recall rate (%)	False positives
Polynomial kernel function	89.24	2000
RBF kernel function	87.25	2000

Table 1 illustrates that Clarifai net achieves recall rate of 86.46% and VGG Net 86.94% with 2000 false positives on FDDB dataset. Notably, our method improves the recall rate to 89.24%, which is 2.3% higher than VGG Net. And the same improvement is observed on AFW dataset, our method achieves 97.19% average precision. Experimental results above demonstrate that the fused features lead to richer representation of images and compensation for defects of feature processing in single net, and it outperforms the single net for face detection on the commonly used face detection datasets. Compared with single net, we note that our method needs additional operations, such as PCA, which result in higher computation complexity and higher memory overhead.

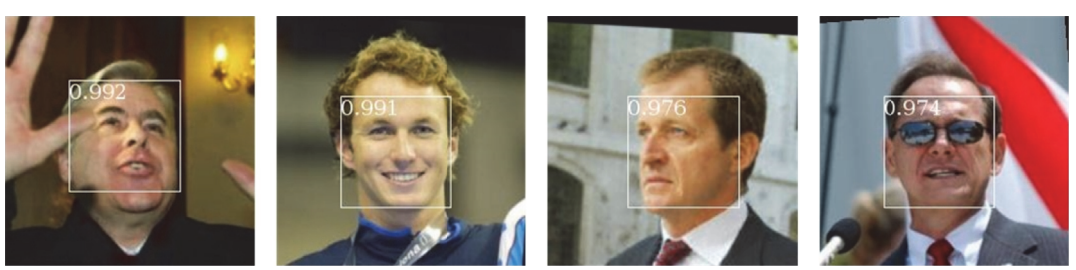
3.2.2. Effect of Different Eigenvalue Statistical Rate in PCA. Table 3 illustrates the performance of different eigenvalue statistical rate on FDDB dataset.

As shown in Table 3, our method achieves recall rate of 89.24% with the eigenvalue statistical rate to 50%, and the recall rate increases slightly to 89.27% when we set the eigenvalue statistical rate to 70% but with much higher dimension. However, when we further increase the eigenvalue statistical rate, recall rate drops to 88.64%, which means that there exists redundancy in the high dimensional features. Obviously, higher eigenvalue statistical rate means higher computational cost. Trading off between the performance and computational cost, we set the eigenvalue statistical rate to 50%.

3.2.3. Comparison between Two Kernel Functions of SVM Classifier. Table 4 illustrates the comparison between different kernel functions of SVM classifier. Compared with RBF kernel function, polynomial kernel function can help to increase recall rate by about 2%; we finally choose polynomial kernel function for our SVM classifier.

(a)

(b)



(c)

Figure 9: Qualitative face detection results of our detector on (a) FDDB, (b) AFW, and (c) LFW.

Table 5: Test results of different classifier on FDDB.

Classifier	Recall rate (%)	False positives
LR	87.50	2000
SVM	89.24	2000

Table 6: Test results of the proposed face detector with/without bounding box regression.

Method	Recall rate (%)	False positives
Ours+bounding box regression	89.51	2000
Ours	89.24	2000

Table 7: Evaluation of performance of other methods.

Method	Recall rate (%)	False positives
DDFD	84.84	2000
Boosted Exemplar	85.65	2000
Joint Cascade	86.68	2000
HeadHunter	88.09	2000
Our method	*89.24*	*2000*
Faceness-Net	90.99	2000
Conv3D	91.16	2000

3.2.4. Comparison between Two Classifiers. In our experiments, besides SVM classifier, we also consider another common and simple classifier, LR (Logistic Regression), to classify face and nonface, whose output represents the confidence of face with cross-entropy loss function based on probability theory, resulting in lower computational complexity. Comparison of different classifiers is shown in Table 5.

Table 5 illustrates that SVM classifier outperforms LR classifier. Experiments indicate that SVM classifier is more time consuming, but with less false positives and higher confidence of detection results, thus achieving better classification.

3.2.5. The Performance of Our Detector with/without Bounding Box Regression. Table 6 compares the performance of our detector with/without bounding box regression.

Table 6 illustrates that bounding box regression slightly improves the performance of our detector but leads to higher computational complexity at stage of data generation and training network. Our method adopts offset max-pooling to extract features with sliding window densely, which leads to better matches of faces and detection windows and gets accurate detection results; therefore, bounding box regression makes little sense in this case.

3.2.6. Comparisons with Other State-of-the-Art Face Detectors on FDDB. We compare the performance of our method with other state-of-the-art methods on FDDB dataset. In particular, we report recall rate of our method with DDFD, Boosted Exemplar et al. [17], Joint Cascade [18], HeadHunter, Faceness-Net, and Conv3D [19] with 2000 false positives in Table 7. Quantitative comparisons of our method with other face detectors on FDDB are displayed in Figure 7.

3.2.7. Comparisons with Other State-of-the-Art Face Detectors on AFW. We compare the performance of our method with other state-of-the-art methods including TSM, Shen et al. [20], Structured Models [21], HeadHunter, Face.com, Face++, and Picasa on AFW dataset. Precision-recall curve is shown in Figure 8, where AP is defined as average precision.

Some detection results are shown in Figure 9.

Figures 7, 8, and 9 illustrate that our method outperforms other state-of-the-art detectors and realizes great improvements of face detection with unconstrained environment. Detection results show that our method can not only cope with faces with small-scale and pose variations, but also perform well for occlusion and blur.

4. Conclusion

In this paper, we propose a face detection method based on two deep convolutional neural networks with SVM classifier; our method has achieved 89.24% recall rate on FDDB and also achieved high accuracy on other datasets. Experimental results show that our method can compensate for defects of feature processing in single deep network by feature fusion of multiple layers and have better performance. In particular, our method is strongly robust to faces with occlusion, blur, and rotation. With using offset max-pooling to extract features, we can obtain better matches of faces and detection windows, and the detection result is more accurate. Further effort will be focused on learning efficient cross-GPU parallelization method, which can take slightly less time to train than the one-GPU net.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61304021).

References

[1] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1–511, Kauai, Hawaii, USA, 2001.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[3] S. S. Farfade, M. Saberian, and L. J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR '15)*, pp. 643–650, Shanghai, China, 2015.

[4] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 5325–5334, June 2015.

[5] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: a deep learning approach," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '15)*, pp. 3676–3684, Santiago, Chile, 2015.

[6] V. Jain and E. Learned-Miller, "FDDB: a benchmark for face detection in unconstrained settings," Tech. Rep., University of Massachusetts Amherst, Amherst, Mass, USA, 2010.

[7] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2879–2886, Providence, RI, USA, June 2012.

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," Tech. Rep., University of Massachusetts, Amherst, Mass, USA, 2007.

[9] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, pp. 818–833, 2014.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," https://arxiv.org/abs/1409.1556.

[11] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "ensenet: implementing efficient convnet descriptor pyramid," https://arxiv.org/abs/1404.1869.

[12] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," https://arxiv.org/abs/1302.1700.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Computer Science*, pp. 580–587, 2014.

[14] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Proceedings of the European Conference on Computer Vision*, pp. 720–735, 2014.

[15] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: a face detection benchmark," https://arxiv.org/abs/1511.06523.

[16] P. M. Roth, M. Kostinger, P. Wohlhart, and H. Bischof, "Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization," in *Proceedings of the IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, pp. 2144–2151, Barcelona, Spain, November 2011.

[17] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua, "Efficient boosted exemplar-based face detection," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1843–1850, Columbus, Ohio, USA, June 2014.

[18] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Proceedings of the European Conference on Computer Vision*, pp. 109–122, Zürich, Switzerland, 2014.

[19] Y. Li, B. Sun, T. Wu, Y. Wang, and W. Cao, "face detection with end-to-end integration of a ConvNet and a 3D model," https://arxiv.org/abs/1606.00850.

[20] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Detecting and aligning faces by image retrieval," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3460–3467, Portland, Ore, USA, June 2013.

[21] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, no. 10, pp. 790–799, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 6795352, 14 pages
http://dx.doi.org/10.1155/2016/6795352

Research Article

Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition

Rongbing Huang,[1,2] Chang Liu,[1] Guoqi Li,[3] and Jiliu Zhou[2]

[1]*Key Laboratory of Pattern Recognition and Intelligent Information Processing, Institutions of Higher Education of Sichuan Province, Chengdu University, Chengdu, Sichuan 610106, China*
[2]*School of Computer and Software, Sichuan University, Chengdu, Sichuan 610065, China*
[3]*School of Reliability and System Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Rongbing Huang; huangrb2006@126.com

Received 3 June 2016; Revised 30 July 2016; Accepted 28 September 2016

Academic Editor: Simone Bianco

Based on a special type of denoising autoencoder (DAE) and image reconstruction, we present a novel supervised deep learning framework for face recognition (FR). Unlike existing deep autoencoder which is unsupervised face recognition method, the proposed method takes class label information from training samples into account in the deep learning procedure and can automatically discover the underlying nonlinear manifold structures. Specifically, we define an Adaptive Deep Supervised Network Template (ADSNT) with the supervised autoencoder which is trained to extract characteristic features from corrupted/clean facial images and reconstruct the corresponding similar facial images. The reconstruction is realized by a so-called "bottleneck" neural network that learns to map face images into a low-dimensional vector and reconstruct the respective corresponding face images from the mapping vectors. Having trained the ADSNT, a new face image can then be recognized by comparing its reconstruction image with individual gallery images, respectively. Extensive experiments on three databases including AR, PubFig, and Extended Yale B demonstrate that the proposed method can significantly improve the accuracy of face recognition under enormous illumination, pose change, and a fraction of occlusion.

1. Introduction

Over the last couple of decades, face recognition has gained a great deal of attention in the academic and industrial communities on account of its challenging essence and its widespread applications. The study of face recognition has a great theoretical value, which involves image processing, artificial intelligence, machine learning, computer vision, and so on, and it also has a high correlation with other biometrics like fingerprints, speech recognition, and iris scans. In the field of pattern recognition, as a classic problem, face recognition mainly covers two issues, feature exaction and classifier design. Currently, most existing works are focusing on these two aspects to promote the performance of face recognition system.

In most real-world applications, it is actually a multiclass classification issue for face recognition. There are many classification methods proposed by researchers. Among them, nearest neighbor classifier (NNC) and its variants like nearest subspace [1] are the most popular methods in pattern classification [2]. In [3], the problem of face recognition was transformed to a binary classification problem through constructing intra- and interfacial image spaces. The intraspace stands for the difference of the same person and the interspace denotes the difference of different people. Then, many binary classifiers such as Support Vector Machine (SVM) [4], Bayesian, and Adaboost [5] can be used.

Besides the classifier design, the other important issue is feature representation. In the real word, face images are usually influenced by variances such as illuminations, posture, occlusions, and expressions. Additionally, there is fact that the difference from the same person would be much larger than that from different people. Therefore, it is crucial to get efficient and discriminant features making the intraspace compact and expanding the margin among different people. Until now, various feature extraction methods

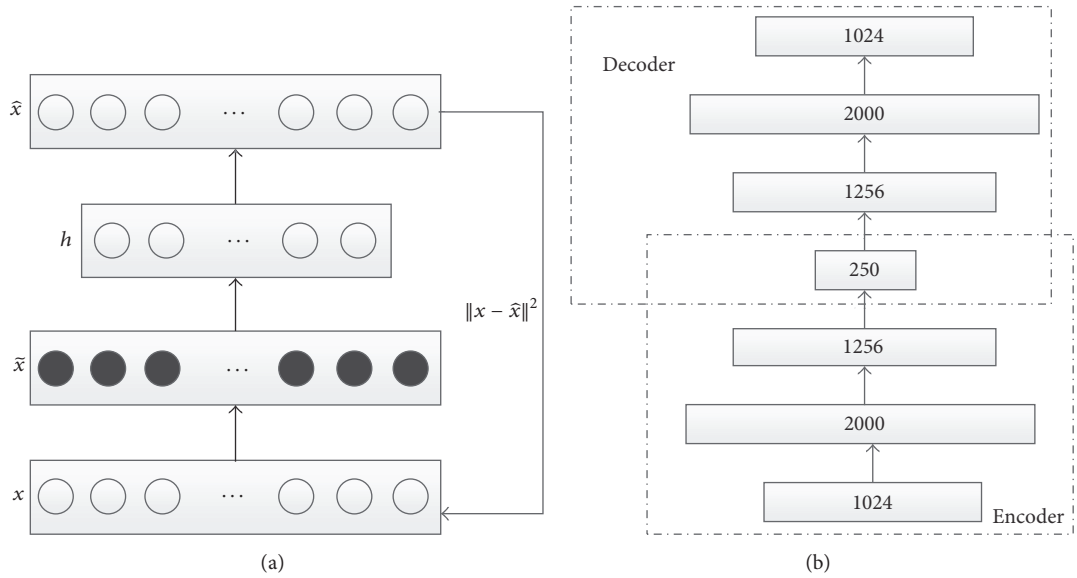


Figure 1: Network architectures. (a) DAE and (b) SDAE.

have been explored, including classical subspace-based dimension reduction approaches like principal component analysis (PCA), fisher linear discriminant analysis (FLDA), independent component analysis (ICA), and so on [6]. In addition, there are some local appearance features extraction methods like Gabor wavelet transform, local binary patterns (LBP), and their variants [7] which are stable to local facial variations such as expressions, occlusions, and poses. Currently, deep learning including deep neural network has shown its great success on image expression [8, 9], and their basic idea is to train a nonlinear feature extractor in each layer [10, 11]. After greedy layer-wise training of a deep network architecture, the output of the network is applied as image feature for latter classification task. Among deep network architectures, as a representative building block, denoising autoencoder (DAE) [12] learns features that is robust to noise by a nonlinear deterministic mapping. Image features derived from DAE have demonstrated good performance in many aspects such as object detection and digit recognition. Inspired by the great success of DAE based deep network architecture, a supervised autoencoder (SAE) [9] was also proposed to build the block, which firstly treated the facial images in some variants like illuminations, expressions, and poses as corrupted images by noises. A face image without the variant through an SAE can be recovered; meanwhile, robust features for image representation are also extracted.

Taking as an example the great success of DAE and SAE based deep learning and inspired by the face recognition under complex environment, in this article, we present a novel deep learning method based on SAE for face recognition. Unlike existing deep stacked autoencoder (AE) which is an unsupervised feature learning approach, our proposed method takes full advantage of the class label information of training samples in the deep learning procedure and tries to discover the underlying nonlinear manifold structures in the data.

The rest of this paper is organized as follows. In Section 2, we give a brief review of DAE and the state-of-the-art face recognition based on deep learning. In Section 3, we focus on the proposed face recognition approach. The experimental results conducted on three public databases are given in Section 4. Finally, we draw a conclusion in Section 5.

2. Related Work

In this section, we briefly review work related to DAE and deep learning based face recognition system.

2.1. Work Related to DAE. DAE is a one-layer neural network, which is a recent variant of the conventional autoencoder (AE). It learns to try to recover the clean input data sample from its corrupted version. The architecture of DAE is illustrated in Figure 1(a). Let there be a total of k training samples and let x denote the original input data. In DAE, firstly, let the input data x be contaminated with some predefined noise such as Gaussian white noise or Poisson noise to obtain corrupted version $\tilde{x}$ such that $\tilde{x}$ is input into an encoder $h = f(\tilde{x}) = u_f(W\tilde{x} + b_f)$. Then an output of the encoder h is used as an input of a decoder $\hat{x} = g(h) = u_g(W'h + b_g)$. Here u_f and u_g are the predefined activation functions such as sigmoid function, hyperbolic tangent function, or rectifier function [13] of encoder and decoder, respectively. $W \in R^{d_h \times d_x}$ and $W' \in R^{d_x \times d_h}$ are the network parameters which denote the weights for the encoder and decoder, respectively. $b_f \in R^{d_h}$ and $b_g \in R^{d_x}$ refer to the bias terms. d_x and d_h present dimensionality of the original data and the number of hidden neurons, respectively. On the basis of the above definition, a DAE learns by solving a regularized optimization problem as follows:

$$\min_{W,W',b_f,b_g} \sum_{i=1}^{k} \|x - \hat{x}\|_2^2 + \frac{\lambda}{2}\left(\sum_j \|W\|_F^2 + \sum_l \left\|W'\right\|_F^2\right). \quad (1)$$

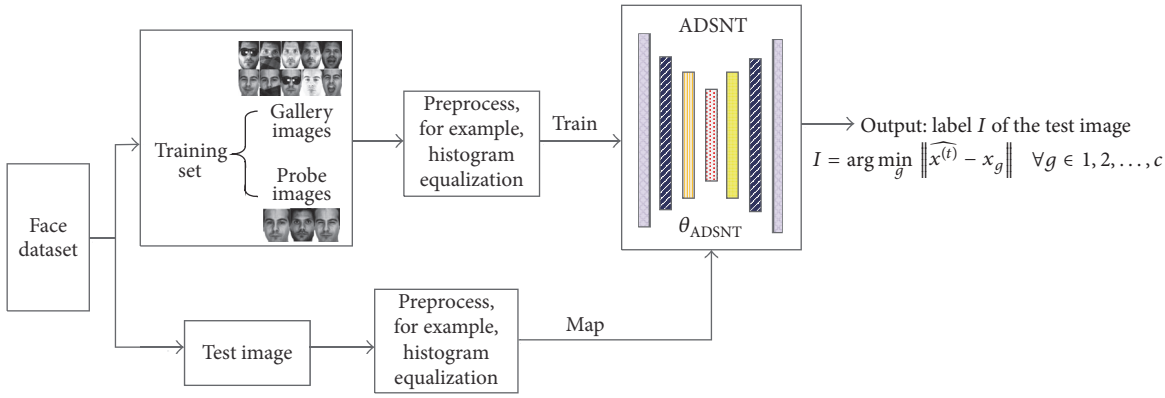


Figure 2: Flowchart of the proposed ADSNT image reconstruction for face recognition.

Here $\|\cdot\|_2^2$ is the reconstruction error and $\|\cdot\|_F$ denotes the Frobenius norm and λ is a parameter that balances the reconstruction loss and weight penalty terms. With reconstructing the clean input data from a corrupted version of it, a DAE can explore more robust features than a conventional AE only simply learning the identity mapping.

To further promote learning meaningful features, sparsity constraints [14] are utilized to impose on the hidden neurons when the number of hidden neurons is large, which is defined in the light of the Kullback-Leibler (KL) divergence as

$$\sum_{j}^{m} \mathrm{KL}\left(\rho \,||\, \overline{\rho_j}\right) = \sum_{j=1}^{m} \rho \log \frac{\rho}{\overline{\rho_j}} + (1-\rho) \log\left(\frac{1-\rho}{1-\overline{\rho_j}}\right), \quad (2)$$

where m is the number of neurons in one hidden layer, $\overline{\rho_j}$ is determined by taking the average activation of a hidden unit j (over all the training set), and ρ is a sparsity parameter (typically a small value).

After finishing f and g learning, the output from encoder h is input to the next layer. Through training such DAE layerwise, stacked denoising autoencoders (SDAE) are then built. Its structure is illustrated in Figure 1(b).

In the real-word application, like face recognition, the faces are usually influenced by all kinds of variances such as expression, illumination, pose, and occlusion. To overcome the effect of variances, Gao et al. [9] proposed supervised autoencoder based on the principle of DAE. They treated the training sample (gallery image) from each person with frontal/uniform illumination, neural expression, and without occlusion as clean data and test faces (probe images) accompanied by variances (expression, illumination, occlusion, etc.) as corrupted data. A mapping capturing the discriminant structure of the facial images from different people is learned, while keeping robust to the variances in these faces. Then robust feature is extracted for image presentation and the performance of face recognition is greatly enhanced.

2.2. Deep Learning Based Face Recognition System. In the early face recognition, there have been various face representation methods including hand-crafted or "shallow" learning ways [6, 7]. In recent years, with the development of big data and computer hardware, feature learning based on deep structure has been greatly successful in image representation field [8, 12, 15, 16]. By means of deep structure learning, the ability of model representation gets great enhancement and we can learn complicated (nonlinear) information from original data effectively. In [16], deep Fisher network was designed through stacking all the Fisher vectors, which greatly performed over conventional Fisher vector representation. Chen et al. [17] proposed marginalized SDAE to learn the optimal closed-form solution, which reduced the computational complexity and improved the scalability of high-dimensional descriptive features. Taigman et al. [18] presented a face verification system based on Convolutional Neural Networks (CNNs), which also obtained high accuracy of verification on the LFW dataset. Zhu et al. [19] designed a network structure that is composed of facial identity-preserving layer and image reconstruction layer, which can reduce intravariance and achieve discriminant information preservation. In [20], Hayat et al. proposed a deep learning framework based on AE with application to image set classification and face recognition, which obtained the best performance comparing with existing state-of-the-art methods. Gao et al. [9] further proposed an SAE which can be used to build the deep architecture and can extract the facial features that are robust to variants. Sun et al. [21] learned multiple convolutional networks (ConvNets) from predicting 10,000 subjects, which generalized well to face verification issue. Furthermore, they improved the ConvNets by incorporating identification and verification missions and enhanced recognition performance [22]. Cai et al. [23] stacked several sparse independent subspace analyses (sISA) to construct deep network structure to learn identity representation.

3. Proposed Method

This section presents our proposed approach whose block diagram is illustrated in Figure 2. Firstly, inspired by stacked DAE and SAE [9], we define Adaptive Deep Supervised Network Template (ADSNT) that can learn an underlying nonlinear manifold structure from the facial images. The basic architecture of ADSNT is illustrated in Figure 3(c) and the corresponding details are depicted in Section 3.1. To make

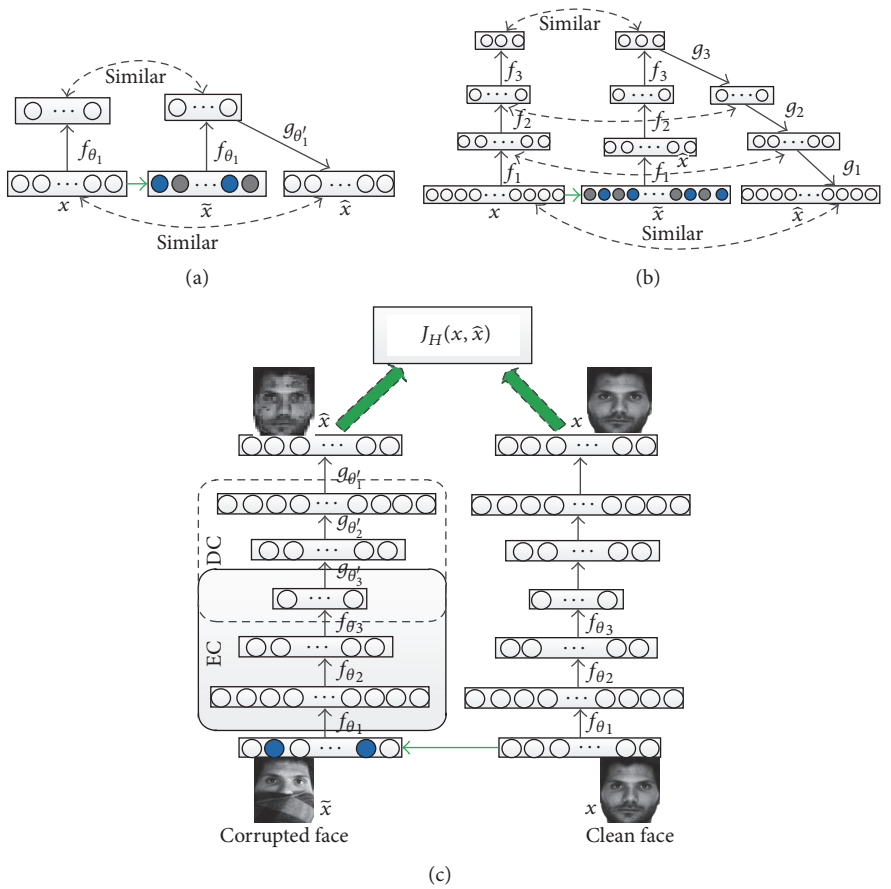


Figure 3: Architecture of SSAE and ADSNT. (a) Supervised autoencoder (SAE) which is comprised of clean/"corrupted" datum, one hidden layer, and one reconstruction layer by using the "corrupted datum"; (b) stacked supervised autoencoder (SSAE); (c) architecture of the Adaptive Deep Supervised Network Template (ADSNT).

the deep network perform well, similar to [20], we need to give it initialization weights. Then, the preinitialized ADSNT is trained to reconstruct the invariant faces which are insensitive to illumination, pose, and occlusion. Finally, having trained the ADSNT, we use the nearest neighbor classifier to recognize a new face image by comparing its reconstruction image with individual gallery images, respectively.

3.1. Adaptive Deep Supervised Network Template (ADSNT). As presented in Figure 3(c), our ADSNT is a deep supervised autoencoder (DSAE) that consists of two parts: an encoder (EC) and a decoder (DC). Each of them has three hidden layers and they share the third layer, that is, the central hidden layer. The features learned from the hidden layer and the reconstructed clean face are obtained by using the "corrupted" data to train the SSAE. In the process of pretraining, we learn a stack of SAE, each having only one hidden layer of feature detectors. Then, the learned activation features of one SAE are used as "data" for training the next SAE in the stack. Such training is repeated a number of times until we get the desired number of layers. Although we use the basic SAE structure which is shown in Figure 3(a) [9] to construct the stacked supervised autoencoder (SSAE), Gao et al.'s stacked supervised autoencoder only used two hidden layers and one reconstruction layer. In this paper, we use three hidden layers to compose the encoder and decoder, respectively, whose structures are shown in Figures 3(b) and 3(c). The encoder part tries best to seek a compact low-dimensional meaningful representation of the clean/"corrupted" data. Following the work [20], the encoder can be formulated as a combination of several layers which are connected with a nonlinear activation function $u_f(\cdot)$. We can use a sigmoid function or a rectified linear unit as nonlinear activation to map the clean/"corrupted" data $x/\tilde{x}$ to a representation h as follows:

$$\begin{aligned} h &= f(h_2) = u_f\left(W_e^{(3)} h_2 + b_e^{(3)}\right), \\ h_2 &= f(h_1) = u_f\left(W_e^{(2)} h_1 + b_e^{(2)}\right), \\ h_1 &= f(x) = u_f\left(W_e^{(1)} x + b_e^{(1)}\right), \\ h &= f(h_2) = u_f\left(W_e^{(3)} h_2 + b_e^{(3)}\right), \\ h_2 &= f(h_1) = u_f\left(W_e^{(2)} h_1 + b_e^{(2)}\right), \\ h_1 &= f(\tilde{x}) = u_f\left(W_e^{(1)} \tilde{x} + b_e^{(1)}\right), \end{aligned} \tag{3}$$

where $W_e^{(i)} \in R^{d_{i-1} \times d_i}$ is a weight matrix of the encoder for the ith layer with d_i neurons and $b_e^{(i)} \in R^{d_i}$ is the bias vector. The encoder parameters learning are achieved by jointly training the encoder-decoder structure to reconstruct the "corrupt" data by minimizing a cost function (see Section 3.2). Therefore, the decoder can be defined as a combination of several layers integrating a nonlinear activation function $u_g(\cdot)$ which reconstructs the "corrupt" data $\widetilde{x}$ from the encoder output h. The reconstructed output $\widehat{x}$ of the decoder is given by

$$\begin{aligned} \widehat{x} &= g\left(\overline{\overline{x}}\right) = u_g\left(W_d^{(3)}\overline{\overline{x}} + b_d^{(3)}\right), \\ \overline{\overline{x}} &= g\left(\overline{x}\right) = u_g\left(W_d^{(2)}\overline{x} + b_d^{(2)}\right), \\ \overline{x} &= g\left(h\right) = u_g\left(W_d^{(1)}h + b_d^{(1)}\right). \end{aligned} \tag{4}$$

So, we can describe the complete ADSNT by its parameter $\theta_{\text{ADSNT}} = \{\theta_W, \theta_b\}$, where $\theta_W = \{W_e^{(i)}, W_d^{(i)}\}$ and $\theta_b = \{b_e^{(i)}, b_d^{(i)}\}$, $i = 1, 2, 3$.

3.2. Formulation of Image Reconstruction Based on ADSNT. Now, we are ready to depict the reconstruction image based on ADSNT. The details are presented as follows.

Given a set of k classes training images that include gallery images (called clean data) and probe images (called "corrupted" data), and their corresponding class labels $y_c = [1, 2, \ldots, k]$, the dataset will be used to train ADSNT for feature learning. Let $\widetilde{x}_i$ denote a probe image, and x_i ($i = 1, 2, \ldots, M$) present gallery images corresponding to $\widetilde{x}_i$. It is desirable that x_i and $\widetilde{x}_i$ should be similar. Therefore, following the work [9, 22], we obtain the following formulation:

$$\begin{aligned} \arg\min_{\theta_{\text{ADSNT}}} J = {} & \frac{1}{M}\sum_i \left\|x_i - \widehat{x}_i\right\|^2 \\ & + \frac{\lambda_{\theta_W}}{M}\sum_i \left\|f\left(x_i\right) - f\left(\widetilde{x}_i\right)\right\|^2 \\ & + \frac{\varphi}{2}\left(\sum_j^3 \left\|W_e^{(i)}\right\|_F^2 + \sum_j^3 \left\|W_d^{(i)}\right\|_F^2\right), \end{aligned} \tag{5}$$

where $\theta_{\text{ADSNT}} = \{\theta_W, \theta_b\}$ (see Section 3.1) are the parameters of ADSNT which is fine-tuned by learning. In this paper, we only explore the tied weights; that is, $W_d^{(3)} = W_e^{(1)^T}$, $W_d^{(2)} = W_e^{(2)^T}$, and $W_d^{(1)} = W_e^{(3)^T}$ (see Figure 3(c)). $\widehat{x}_i$ is the reconstruction image of the corrupted image $\widetilde{x}_i$. Like regularization parameter, λ_{θ_W} balances the similarity of the same person to preserve $f(x_i)$ and $f(\widetilde{x}_i)$ as similarly as possible. $f(\cdot)$ is a nonlinear activation function. φ is a parameter that balances weight penalty terms and reconstruction loss. $\|\cdot\|_F$ presents the Frobenius norm and $\sum_j^3 \|W_e^{(i)}\|_F^2 + \sum_j^3 \|W_d^{(i)}\|_F^2$ ensures small weight values for all the hidden neurons. Furthermore, following the work [9, 14], we impose a sparsity constraint on the hidden layer to enhance learning meaningful features. Then, we can further modify cost function and obtain the following objection formulation:

$$\begin{aligned} & \arg\min_{\theta_{\text{ADSNT}}} J_{\text{reg}} \\ & \quad = J + \gamma\left(\sum_i^3 \text{KL}\left(\rho_x \,||\, \rho_0\right) + \sum_i^5 \text{KL}\left(\rho_{\widetilde{x}} \,||\, \rho_0\right)\right), \end{aligned} \tag{6}$$

where

$$\begin{aligned} \rho_x &= \frac{1}{M}\sum_i \left(\frac{1}{2}f\left(x_i\right) + 1\right), \\ \rho_{\widetilde{x}} &= \frac{1}{M}\sum_i \frac{1}{2}\left(f\left(\widetilde{x}_i\right) + 1\right), \\ & \text{KL}\left(\rho \,||\, \rho_0\right) \\ & \quad = \sum_j \left(\rho_0 \log\left(\frac{\rho_0}{\rho_j}\right) + \left(1 - \rho_0\right)\log\left(\frac{1 - \rho_0}{1 - \rho_j}\right)\right). \end{aligned} \tag{7}$$

Here the KL divergence between two distributions, that is, ρ_0 and ρ_j that present ρ_x or $\rho_{\widetilde{x}}$, is calculated. The sparsity ρ_0 is usually a constant (taking a small value, according to the work [9, 24], it is set to 0.05 in our experiments), whereas ρ_x and $\rho_{\widetilde{x}}$ are the mapping mean activation values from clean data and corrupted data, respectively.

3.3. Optimization of ADSNT. For obtaining the optimization parameter $\theta_{\text{ADSNT}} = \{\theta_W, \theta_b\}$, it is important to initialize weights and select an optimization training algorithm. The training will fail if the initialization weights are inappropriate. This is to say, if we give network too large initialization weights, the ADSNT will be trapped in local minimum. If the initialized weights are too small, the ADSNT will encounter the vanishing gradient problem during backpropagation. Therefore, following the work [20, 24], Gaussian Restricted Boltzmann Machines (GRBMs) are adopted to initialize weight parameters by performing pretraining, which has been already applied widely. For more details, we refer the reader to the original paper [24]. After obtaining the initialized weights, the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm is utilized to learn the parameters as it has better performance and faster convergence than stochastic gradient descent (SGD) and conjugated gradient (CGD) [25]. Algorithm 1 depicts the optimization procedure of ADSNT.

Algorithm 1. (learning adaptive deep supervised network template)

Input. Training images Ω: k classes, and each class is composed of the face with neutral expression, frontal pose, and normal illumination condition (clean data) and random number of variant faces (corrupted data). Number of network layers L. Iterative number I, balancing parameters λ, φ and γ, and convergence error ε.

Output. Weight parameters $\theta_{\text{ADSNT}} = \{\theta_W, \theta_b\}$

(1) Preprocess all images, namely, perform histogram equalization

(2) X: Randomly select a small subset for each individual from Ω

(3) Initialize: Train GRBMs by using X to initialize the $\theta_{\text{ADSNT}} = \{\theta_W, \theta_b\}$

(4) (Optimization by L-BFGS)

For $r = 1, 2, \ldots, R$ do
Calculate J_{reg} using (6)
If $r > 1$ and $|J_r - J_{r-1}| < \varepsilon$, go to Return

Return. θ_W and θ_b.

Since training the ADSNT model aims to reconstruct clean data, namely, gallery images from corrupt data, it might learn an underlying structure from the corrupt data and produce very useful representation. Furthermore, we can learn an overcomplete sparse representation from corrupt data through mapping them into a high-dimensional feature space since the first hidden layer has the number of neurons larger than the dimensionality of original data. The high-dimensional model representation is then followed by a so-called "bottleneck"; that is, the data is further mapped to an abstract, compact, and low-dimensional model representation in the subsequent layers of the encoder. Through such a mapping, the redundant information such as illumination, poses, and partial occlusion in the corrupted faces is removed and only the useful information content for us is kept. In addition, we know that if we use AE with only one hidden layer and jointly linear activation functions, the learned weights would be analogous to a PCA subspace [20]. However, AE is an unsupervised algorithm. In our work, we make use of the class label information to train SAE, so if we also use only one hidden layer with a linear activation function, the learned weights by the SAE are thought to be similar to "LDA" subspace. However, in our structure, we apply the nonlinear activation functions and stack several hidden layers together, and then the ADSNT can adapt to very complicated nonlinear manifold structures. Some of reconstructed images based on ADSNT from AR database are shown in Figure 4(b). One can see that ADSNT can remove the illumination. For those face images with partial occlusion, ADSNT can also imitate the clean faces. This results are not surprising because the human being has the capability of inferring the unknown faces from known face images via the experience (for deep network structure, the experience learned derives from generic set) [9].

3.4. Face Classification Based on ADSNT Image Reconstruction. To better train ADSNT, all images need to be preprocessed. It is a very important step for object recognition including face recognition. The common ways include histogram equalization, geometry normalization, and image smoothing. In this paper, for the sake of simplicity, we only perform histogram equalization on all the facial images to minimize illumination variations. That is, we utilize histogram equalization to normalize the histogram of facial images and make them more compact. For the details about histogram equalization, one can be referred to see [26].

After the ADSNT is trained completely with a certain number of individuals, we can use it to perform on the unseen face images for recognizing them.

Given a test facial image $x^{(t)}$ which is also preprocessed with histogram equalization in the same way as the training images and presented to the ADSNT network, we reconstruct (using (3) and (4)) image $\widehat{x^{(t)}}$ from ADSNT, which is similar to clean face. For the sake of simplicity, the nearest neighbor classification based on the Euclidean distance between the reconstruction and all the gallery images identifies the class. The classification formula is defined as

$$I_k\left(x^{(t)}\right) = \arg\min_g \left\| \widehat{x^{(t)}} - x_g \right\|, \quad \forall g \in 1, 2, \ldots, c, \tag{8}$$

where $I_k(x^{(t)})$ is the resulting identity and x_g is the clean facial image in the gallery images of individual g.

4. Experimental Results and Discussion

In this section, extensive experiments are conducted to present and compare the performance of different methods with the proposed approach. The experiments are implemented on three widely used face databases, that is, AR [27], Extended Yale B [28], and PubFig [29]. The details of these three databases and performance evaluation of different approaches are presented as follows.

4.1. Dataset Description. The AR database contains over 4000 color face images from 126 people (56 women and 70 men). The images were taken in two sessions (between two weeks) and each session contained 13 pictures from one person. These images contain frontal view faces with different facial expression, illuminations, and occlusions (sun glasses and scarf). Some sample face images from AR are illustrated in Figure 5(a). In our experiments, for each person, we choose the facial images with neutral expression, frontal pose, and normal illumination condition as gallery images and randomly select half the number of images from the rest of the images of each person as probe images. The remaining images compose the testing set.

The Extended Yale B database consists of 16128 images of 38 people under 64 illumination conditions and 9 poses. Some sample face images from Extended Yale B are illustrated in Figure 5(b). For each person, we select the faces that have normal light condition and frontal pose as gallery images and randomly choose 6 poses and 16 illumination face images to compose the probe images. The remaining images compose the testing set.

The PubFig database is composed of 58,797 images of 200 subjects taken from the internet. The images of the database were taken in completely uncontrolled conditions with noncooperative people. These images have a very large degree of variability in face expression, pose, illumination, and so forth. Some sample images from PubFig are illustrated in Figure 5(c). In our experiments, for each individual, we select the faces with neutral expression, the frontal or

(a)

(b)

Figure 4: Some original images from AR database and the reconstructed ones. (a) Original face images (corrupted faces) $\tilde{x}$. (b) Reconstructed faces $\hat{x}$.

near frontal pose, and normal illumination as galleries and randomly choose half the number of images from the rest of the images of each person as probes. The remaining images compose the testing set.

4.2. Experimental Settings. In all the experiments, the facial images from the AR, PubFig, and Extended Yale B databases are automatically detected using OpenCV face detector [30]. After that, we normalize the detected facial images (in orientation and scale) such that two eyes can be aligned at the same location. Then, the face areas are cropped and converted to 256 gray levels images. The size of each cropped image is 26×30 pixels. Thus, the dimensionality of the input vector is 780. Figure 6 presents an example from AR database and the corresponding cropped image. Each cropped facial image is further preprocessed with histogram equalization to minimize illumination variations. We train our ADSNT model with 3 hidden layers, where the number of hidden nodes for these layers is empirically set as $[1024 \rightarrow 500 \rightarrow 120]$, because our experiments show that three hidden layers can get a sufficiently good performance (see Section 4.3.3).

In order to show the whole experimental process about parameters setting, we initially use the hyperbolic tangent function as the nonlinear activation function and implement ADSNT on AR. We also choose the face images with neutral expression, frontal pose, and normal illumination as galleries and randomly select half the number of images from the rest of the images of each person as probe images. The remaining images compose the testing set. The mean identification rates are recorded.

Firstly, we empirically set the parameter $\varepsilon = 0.001$ and sparsity target $\rho_0 = 0.05$ and fix the parameters $\lambda = 0.5$ and $\varphi = 0.1$ in ADSNT to check the effect of γ on the identification rate. As illustrated in Figure 6(a), where $\gamma = 0.08$, ADSNT recognition method gets the best performance.

Figure 5: A fraction of samples from AR, PubFig, and Extended Yale B face databases. (a) AR, (b) Extended Yale B, and (c) PubFig.

Then, according to Figure 6(a), we fix the parameters $\gamma = 0.08$ and $\lambda = 0.5$ in ADSNT to check the influence of φ. As showed in Figure 6(b), when $\varphi = 0.6$, our method achieves the best recognition rate. At last, we fix $\gamma = 0.08$ and $\varphi = 0.6$, and the recognition rates are illustrated in Figure 6(c) with different value of λ. When $\lambda = 3$, the recognition rate is the highest. From the plot in Figure 6, one can observe that the parameters λ, φ, and γ cannot be too large or too small. If λ is too large, the ADSNT would be less discriminative of different subjects because it implements too strong similarity preservation entry. But if λ is too small, it will degrade the recognition performance and the significance of similarity preservation entry. Similarly, γ can also not be too large, or the hidden neurons will not be activated for a given input and low recognition rate will be achieved. If γ is too small, we can get poor performance. For the weight decay φ, if it is too small, the values of weights for all hidden units will change very slightly. On the contrary, the values of weights will change greatly.

Using above those experiments, we gain the optimal parameter values used in ADSNT as $\lambda = 3$, $\varphi = 0.6$, and $\gamma = 0.08$ on AR database. The similar experiments also have been performed on Extended Yale B and PubFig databases. We can get the parameters setting as $\lambda = 2.6$, $\varphi = 0.5$, and $\gamma = 0.06$ on Extended Yale B database and $\lambda = 2.8$, $\varphi = 0.52$, and $\gamma = 0.09$ on PubFig database.

In the experiments, we use two measures including the mean identification accuracy μ with standard deviation

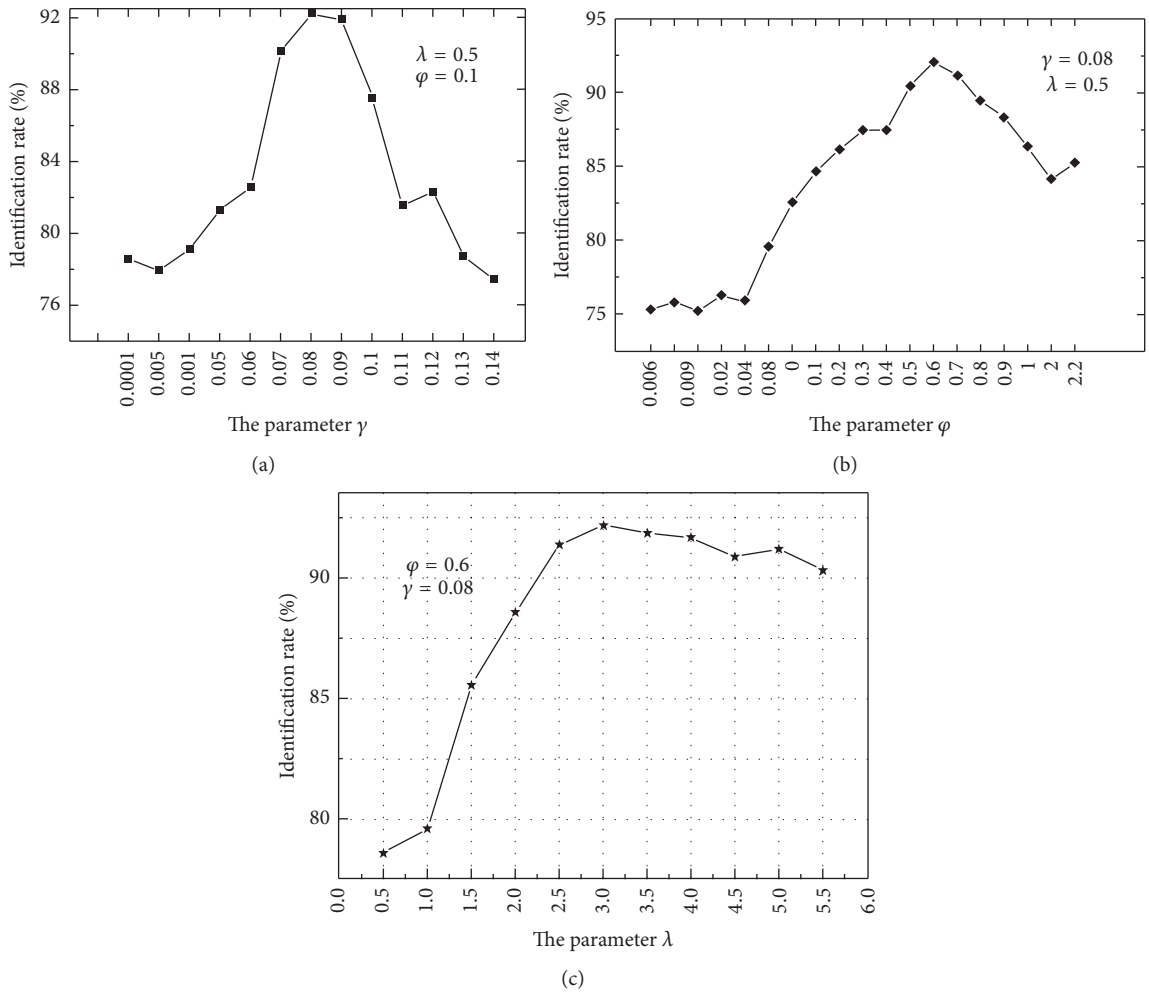


Figure 6: Parameters setting.

$\nu(\mu \pm \nu)$ and the receiving operating characteristic (ROC) curves to validate the effectiveness of our method as well as other methods.

4.3. Experimental Results and Analysis

4.3.1. Comparison with Different Methods. In the following experiments on the three databases, we compare the proposed approach with several recently proposed methods. These compared methods include DAE with 10% random mask noises [12], marginalized DAE (MDAE) [17], Constractive Autoencoders (CAE) [15], Deep Lambertian Networks (DLN) [31], stacked supervised autoencoder (SSAE) [9], ICA-Reconstruction (RICA) [32], and Template Deep Reconstruction Model (TDRM) [20]. We use the implementation of these algorithms that are provided by the respective authors. For all the compared approaches, we use the default parameters that are recommended in the corresponding papers.

The mean identification accuracy with standard deviations of different approaches on three databases is shown in Table 1. The ROC curves of different approaches are illustrated in Figure 7. The results imply that our approach significantly outperforms other methods and gets the best mean recognition rates for the same setting of training and testing sets. Compared to those unsupervised deep learning methods such as DAE, MDAE, CAE, DLN, and TDRM, the improvement of our method is over 30% on Extended Yale B and AR databases where there is a little pose variance. On the PubFig database, our approach can also achieve the mean identification rate of 91.26 ± 1.6%

Table 1: Comparisons of the average identification accuracy and standard deviation (%) of different approaches on different databases.

Method	AR	Extended Yale B	PubFig
DAE [12]	57.56 ± 0.2	63.45 ± 1.3	61.33 ± 1.5
MDAE [17]	67.80 ± 1.3	71.56 ± 1.6	70.55 ± 2.5
CAE [15]	49.50 ± 2.1	55.72 ± 0.8	68.56 ± 1.6
DLN [31]	NA	81.50 ± 1.4	77.60 ± 1.4
SSAE [9]	85.21 ± 0.7	82.22 ± 0.3	84.04 ± 1.2
RICA [32]	76.33 ± 1.7	70.44 ± 1.3	72.35 ± 1.5
TDRM [20]	87.70 ± 0.6	86.42 ± 1.2	89.90 ± 0.9
Our method	92.32 ± 0.7	93.66 ± 0.4	91.26 ± 1.6

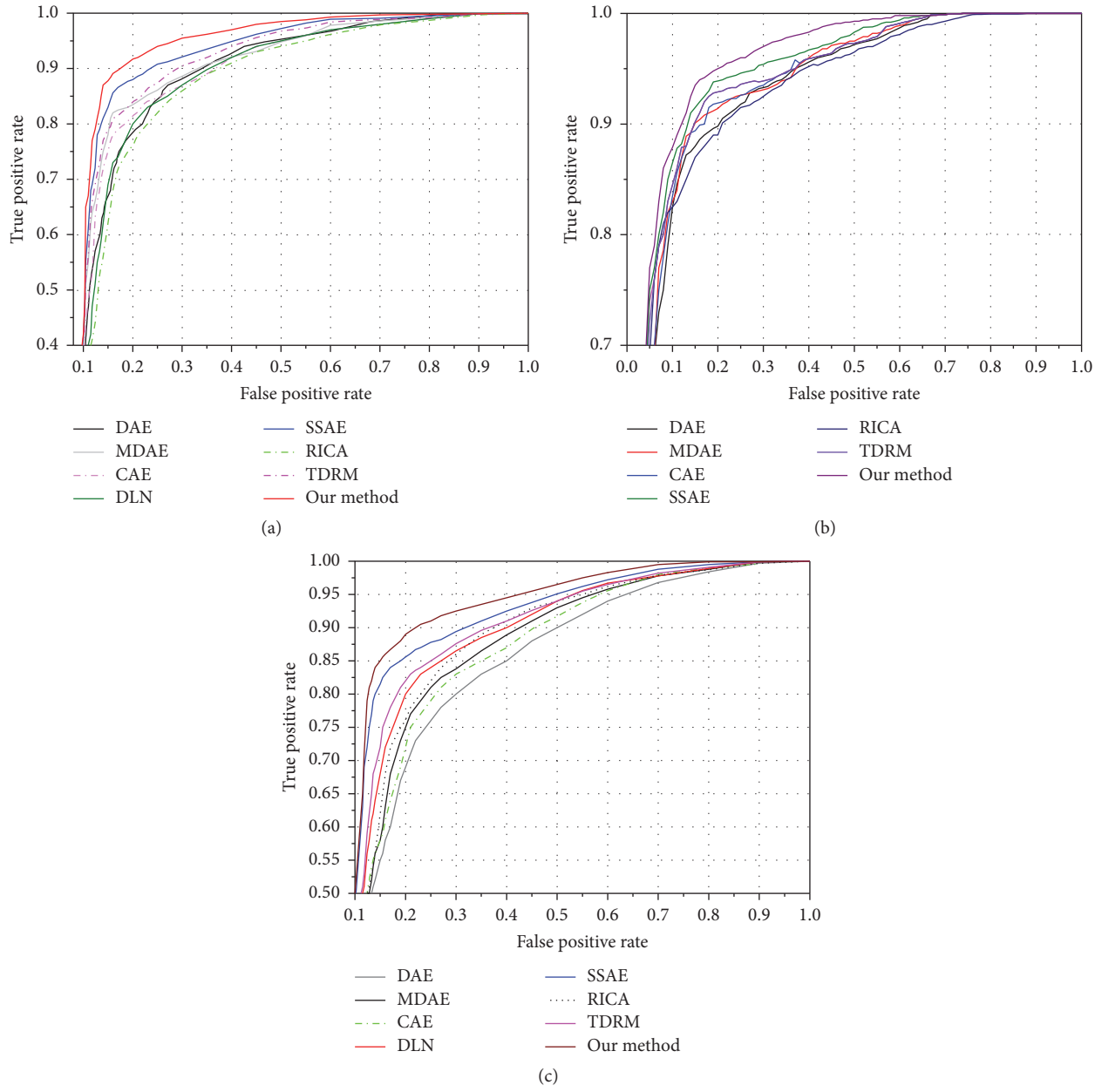


Figure 7: Comparisons of ROC curves between our method and other methods on different databases. (a) AR, (b) Extended Yale B, and (c) PubFig.

and outperforms all compared methods. The reason is that our method can extract discriminative, robust information to variances (expression, illumination, pose, etc.) in the learned deep networks. Compared with a supervised method like RICA, the proposed method can improve over 16%, 19%, and 23% on AR, PubFig, and Extended Yale B databases, respectively. Our method is a deep learning method, which focuses on the nonlinear classification problem with learning a nonlinear mapping such that more nonlinear, discriminant information may be explored to enhance the identification performance. Compared with SSAE method that is designed for removing the variances such as illumination, pose, and partial occlusion, our method can still be better over 6% because of using the weight penalty terms, GRBM to initialize weights, and three layers' similarity preservation term.

4.3.2. Convergence Analysis. In this subsection, we evaluated the convergence of our ADSNT versus a different number of iterations. Figure 8 illustrates the value of the objective function of ADSNT versus a different number of iterations on the AR, PubFig, and Extended Yale B databases. From Figure 8(a), one can observe that ADSNT converges in about 55, 28, and 70 iterations on the three databases, respectively.

We also implement the identification accuracy of ADSNT versus a different number of iterations on the AR, PubFig, and Extended Yale B databases. Figure 8(b) plots the mean

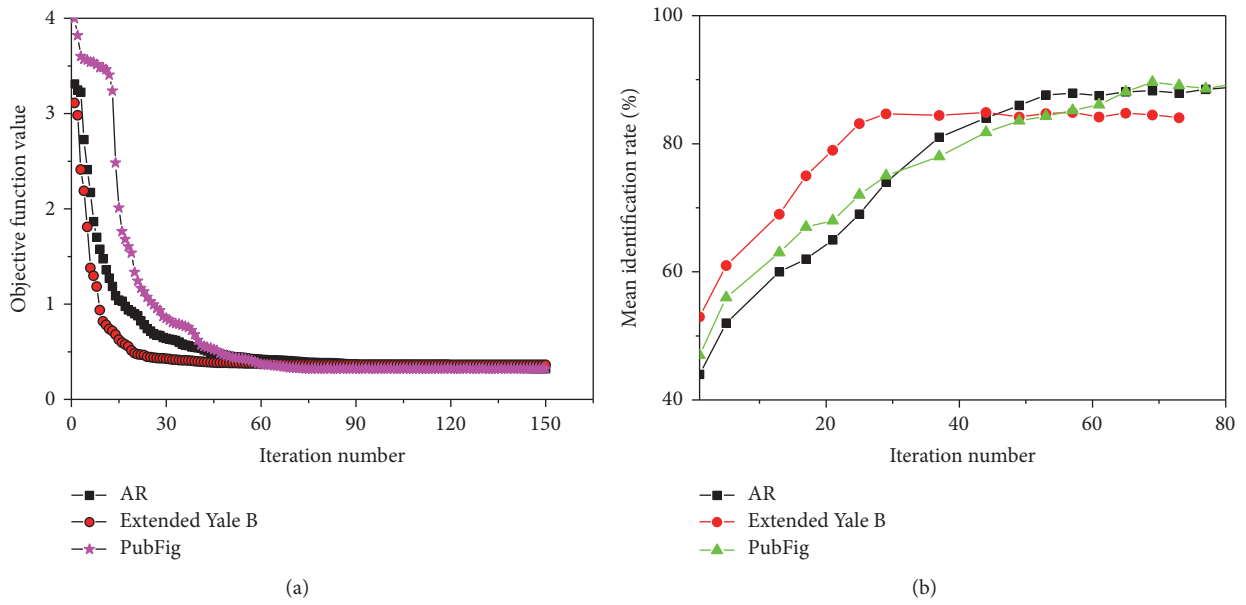


Figure 8: Convergence analysis. (a) Convergence curves of ADSNT on AR, PubFig, and Extended Yale B. (b) Mean identification rate (%) versus iterations of ADSNT on AR, PubFig, and Extended Yale B.

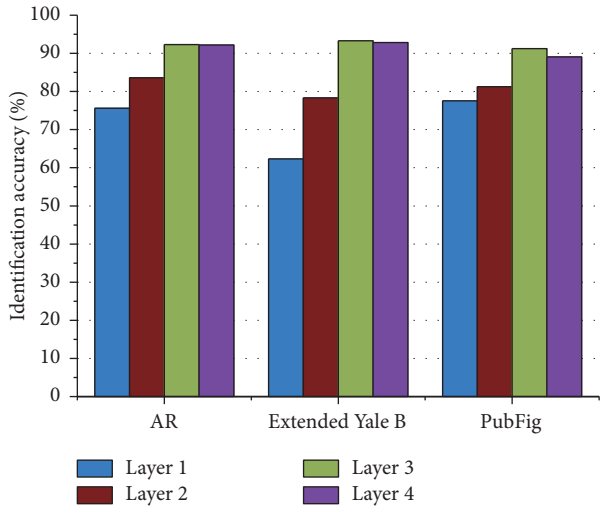


Figure 9: The results of ADSNT with different network depth on the different datasets.

identification rate of ADSNT. From Figure 8(b), one can also observe that ADSNT achieves stable performance after about 55, 70, and 28 iterations on AR, PubFig, and Extended Yale B databases, respectively.

4.3.3. The Effect of Network Depth. In this subsection, we conduct experiments on the three face datasets with different hidden layer of our proposed ADSNT network. The proposed method achieves an identification rate of 92.3 ± 0.6%, 93.3 ± 1.2%, and 91.22 ± 0.8% by three-hidden layer ADSNT network, that is, 1024 → 500 → 120, respectively, on AR, Extended Yale B, and PubFig datasets. Figure 9 illustrates the performance of different layer ADSNT. One can observe that three-hidden layer network outperforms 2-layer network, and the result of 3-layer ADSNT network is very nearly equal to those of 4-layer network on AR and Extended Yale B databases. We also observe that the performance of 4-layer network is a bit lower than that of 3-layer network on the PubFig database. In addition, the deeper ADSNT network is, the more complex its computational complexity becomes. Therefore, the 3-layer network depth is a good trade-off between performance and computational complexity.

4.3.4. Activation Function. Following the work in [9], we also estimate the performance of ADSNT with different activation functions such as sigmoid, hyperbolic tangent, and rectified linear unit (ReLU) [33] which is defined as $f(x) = \max(0, x)$. When the sigmoid $f(x) = 1/(1 + e^{-x})$ is used as activation function, the objective function (see (6)) is rewritten as follows:

$$\begin{aligned}
&\arg\min_{\theta_{\text{ADSNT}}} J \\
&= \frac{1}{M}\sum_i \left\|x_i - \widehat{x}_i\right\|^2 + \frac{\lambda_{\theta_W}}{M}\sum_i \left\|f(x_i) - f(\widetilde{x}_i)\right\|^2 \\
&\quad + \frac{\varphi}{2}\left(\sum_j^3 \left\|W_e^{(i)}\right\|_F^2 + \sum_j^3 \left\|W_d^{(i)}\right\|_F^2\right) \\
&\quad + \gamma\left(\sum_i^3 \mathrm{KL}\left(\rho_x \,||\, \rho_0\right) + \sum_i^5 \mathrm{KL}\left(\rho_{\widetilde{x}} \,||\, \rho_0\right)\right),
\end{aligned} \tag{9}$$

where $\rho_x = (1/M)\sum_i f(x_i)$, $\rho_{\widetilde{x}} = (1/M)\sum_i f(\widetilde{x}_i)$.

Table 2: Comparisons of the ADSNT algorithm with different activation functions on the AR, PubFig, and Extended Yale B databases.

Dataset	Sigmoid	Tanh	ReLU
AR	88.66 ± 1.4	92.32 ± 0.7	93.22 ± 1.5
Extended Yale B	90.55 ± 0.6	93.66 ± 0.4	94.54 ± 0.3
PubFig	87.40 ± 1.2	91.26 ± 1.6	92.44 ± 1.1

If ReLU is adopted as activation function, (6) is formulated as

$$\begin{aligned}\arg\min_{\theta_{\mathrm{ADSNT}}} J &= \frac{1}{M}\sum_i \left\|x_i - \widehat{x}_i\right\|^2 + \frac{\lambda_{\theta_W}}{M}\sum_i \left\|f(x_i) - f(\widetilde{x}_i)\right\|^2 \\ &+ \frac{\varphi}{2}\left(\sum_j^3 \left\|W_e^{(i)}\right\|_F^2 + \sum_j^3 \left\|W_d^{(i)}\right\|_F^2\right) \\ &+ \gamma\left(\sum_i^3 \left\|f(x_i)\right\|_1 + \sum_i^5 \left\|f(\widetilde{x}_i)\right\|_1\right).\end{aligned} \tag{10}$$

Table 2 shows the performance of the proposed ADSNT based on different activation functions conducted on the three databases. From Table 2, one can see that ReLU achieves the best performance. The key reason is that we use the weight decay term φ to optimize the objective function.

4.3.5. Timing Consumption Analysis. In this subsection, we use a HP Z620 workstation with Intel Xeon E5-2609, 2.4 GHz CPU, 8 G RAM and conduct a series of experiments on AR database to compare the time consumption of different methods which are tabulated in Table 3. The training time (seconds) is shown in Table 3(a) while the time (seconds) needed to recognize a face from the testing set is shown in Table 3(b). From Table 3, one can see that the proposed method requires comparatively more time for training because of initialization of ADSNT and performing image reconstruction. However, the procedure of training is offline. When we identity an image from testing set, our method requires less time than other methods.

Table 3: (a) Training time (seconds) for different methods. (b) Testing time (seconds) for different methods. The proposed method costs the least amount of testing time comparing with other methods.

(a)

Methods	Time
DAE [12]	8.61
MDAE [17]	10.54
CAE [15]	7.86
DLN [31]	23.43
SSAE [9]	53.51
RICA [32]	13.44
TDRM [20]	110.2
Our method	122.32

(b)

Methods	Time
DAE [12]	0.27
MDAE [17]	0.3
CAE [15]	0.26
DLN [31]	0.35
SSAE [9]	0.22
RICA [32]	0.19
TDRM [20]	0.18
Our method	0.13

5. Conclusions

In this article, we present an adaptive deep supervised autoencoder based image reconstruction method for face recognition. Unlike conventional deep autoencoder based face recognition method, our method considers the class label information from training samples in the deep learning procedure and can automatically discover the underlying nonlinear manifold structures. Specifically, a multilayer supervised adaptive network structure is presented, which is trained to extract characteristic features from corrupted/clean facial images and reconstruct the corresponding similar facial images. The reconstruction is realized by a so-called "bottleneck" neural network that learns to map face images into a low-dimensional vector and to reconstruct the respective corresponding face images from the mapping vectors. Having trained the ADSNT, a new face image can then be recognized by comparing its reconstruction image with individual gallery images during testing. The proposed method has been evaluated on the widely used AR, PubFig, and Extended Yale B databases and the experimental results have shown its effectiveness. For future work, we are focusing on applying our proposed method to other application fields such as pattern classification based on image set and action recognition based on the video to further demonstrate its validity.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is partially supported by the research grant for the Natural Science Foundation from Sichuan Provincial Department of Education (Grant no. 13ZB0336) and the National Natural Science Foundation of China (Grant no. 61502059).

References

[1] J.-T. Chien and C.-C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pp. 1644–1649, 2002.

[2] Z. Lei, M. Pietikäinen, and S. Z. Li, "Learning discriminant face descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 289–302, 2014.

[3] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, no. 11, pp. 1771–1782, 2000.

[4] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, New York, NY, USA, 2004.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2001.

[6] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[7] B. Zhang, S. Shan, X. Chen, and W. Gao, "Histogram of Gabor phase patterns (HGPP): a novel object representation approach for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 57–68, 2007.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, pp. 1527–1554, 2012.

[9] S. Gao, Y. Zhang, K. Jia, J. Lu, and Y. Zhang, "Single sample face recognition via learning deep supervised autoencoders," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2108–2118, 2015.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *American Association for the Advancement of Science. Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[11] Y. Bengio, "Practical recommendations for gradient-based training of deep architecturesm," in *Neural Networks: Tricks of the Trade*, pp. 437–478, Springer, Berlin, Germany, 2012.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research (JMLR)*, vol. 11, no. 5, pp. 3371–3408, 2010.

[13] V. Nair and G. E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 807–814, Haifa, Israel, June 2010.

[14] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS '11)*, pp. 215–223, Sardinia, Italy, 2010.

[15] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, pp. 833–840, Bellevue, Wash, USA, July 2011.

[16] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep fisher networks for large-scale image classification," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*, pp. 163–171, Lake Tahoe, Nev, USA, December 2013.

[17] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proceedings of the 29th International Conference on Machine Learning (ICML '12)*, pp. 767–774, Edinburgh, UK, July 2012.

[18] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1701–1708, June 2014.

[19] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 113–120, Sydney, Australia, December 2013.

[20] M. Hayat, M. Bennamoun, and S. An, "Deep reconstruction models for image set classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 713–727, 2015.

[21] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1891–1898, Columbus, Ohio, USA, June 2014.

[22] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," Tech. Rep., https://arxiv.org/abs/1406.4773.

[23] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Deep nonlinear metric learning with independent subspace analysis for face verification," in *Proceedings of the 20th ACM International Conference on Multimedia (MM '12)*, pp. 749–752, November 2012.

[24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[25] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, pp. 265–272, Bellevue, Wash, USA, July 2011.

[26] C. Zhou, X. Wei, Q. Zhang, and X. Fang, "Fisher's linear discriminant (FLD) and support vector machine (SVM) in non-negative matrix factorization (NMF) residual space for face recognition," *Optica Applicata*, vol. 40, no. 3, pp. 693–704, 2010.

[27] A. Martinez and R. Benavente, "The AR face database," CVC Tech. Rep. #24, 1998.

[28] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[29] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 365–372, Kyoto, Japan, October 2009.

[30] M. Rezaei and R. Klette, "Novel adaptive eye detection and tracking for challenging lighting conditions," in *Computer Vision—ACCV 2012 Workshops*, J.-I. Park and J. Kim, Eds., vol. 7729 of *Lecture Notes in Computer Science*, pp. 427–440, Springer, Berlin, Germany, 2013.

[31] Y. Tang, R. Salakhutdinov, and G. H. Hinton, "Deep Lambertian networks," in *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, pp. 1623–1630, Edinburgh, UK, July 2012.

[32] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '11)*, pp. 1017–1025, Granada, Spain, December 2011.

[33] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 315–323, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 5376087, 13 pages
http://dx.doi.org/10.1155/2016/5376087

Research Article

Customized Dictionary Learning for Subdatasets with Fine Granularity

Lei Ye,[1] Can Wang,[2] Xin Xu,[1] and Hui Qian[2]

[1]*College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, China*
[2]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*

Correspondence should be addressed to Lei Ye; yelei@nudt.edu.cn

Received 21 June 2016; Accepted 18 October 2016

Academic Editor: Simone Bianco

Sparse models have a wide range of applications in machine learning and computer vision. Using a learned dictionary instead of an "off-the-shelf" one can dramatically improve performance on a particular dataset. However, learning a new one for each subdataset (subject) with fine granularity may be unwarranted or impractical, due to restricted availability subdataset samples and tremendous numbers of subjects. To remedy this, we consider the dictionary customization problem, that is, specializing an existing global dictionary corresponding to the total dataset, with the aid of auxiliary samples obtained from the target subdataset. Inspired by observation and then deduced from theoretical analysis, a regularizer is employed penalizing the difference between the global and the customized dictionary. By minimizing the sum of reconstruction errors of the above regularizer under sparsity constraints, we exploit the characteristics of the target subdataset contained in the auxiliary samples while maintaining the basic sketches stored in the global dictionary. An efficient algorithm is presented and validated with experiments on real-world data.

1. Introduction

Sparse models are of great interest in machine learning and computer vision, owing to their applications for image denoising [1], face recognition [2–4], traffic sign recognition [5], visual-tactile fusion [6, 7], and so forth. In sparse coding, samples or signals are represented as sparse linear combinations of the column vectors (called atoms) of a redundant dictionary. This dictionary can be a predefined one, such as the DCT bases and wavelets [8], or a learned one based on a specific task or dataset of interest.

With sufficient samples, learning a specialized dictionary instead of using the "off-the-shelf" one has been shown to dramatically improve the performance. Generally, the dictionary and the coefficients are estimated by minimizing the sum of least squared errors under the sparsity constraint. Batch algorithms such as MOD [9] and K-SVD [10] and nonparametric Bayesian methods [11] have shown state-of-the-art performance. Further, Mairal et al. [12] developed an online approach to handle large amounts of samples.

Recently, theoretical analysis of sparse dictionary learning has attracted much attention. Schnass [13] presented theoretical results of the dictionary identification problem. Sample complexity has been estimated in [14, 15]. Gribonval et al. [16] analyzed the local minima of dictionary learning. Moreover, to extend the capacity, dictionary learning with specific motivations [17–19] has also attracted lots of interests. For instance, robust face recognition [3] is dedicated to particular applications, and Hawe et al. [20] require the dictionary to have a separable structure. While the learned dictionary has significant effects on a given dataset, attaining further specialized dictionaries for subdatasets with fine granularity is an interesting and useful concept as well. For instance, with a dictionary corresponding to facial images of all humans, we want to gain a customized dictionary for each particular individual. However, in this case, standard dictionary learning approaches may be unwarranted or impractical: on one hand, samples for a particular individual (subject) are restricted and insufficient in most cases; on the other hand, even with enough data, learning so many

dictionaries becomes inefficient for computation and storage. We demonstrate further examples, such as customizing handwritings to different styles, matching images of flower to various species, or matching paper corpora to specific proceedings.

In terms of classification tasks, approaches such as Yang et al. [18] and Ma et al. [2] learn a structured dictionary which consists of N subdictionaries on behalf of N different subjects. However they are often unfeasible: firstly, as a part of the global dictionary, the coding performance of the subdictionary is always worse than the global one. Secondly, the subdictionaries for N subjects must be learned together, which becomes inflexible and exacting for a huge N. Thirdly, once the global dictionary is obtained, specialization for a new $(N+1)$th subdataset would be impossible.

In this paper, we are looking for an effective, economic, and flexible dictionary customization approach, which are supposed to have the following characteristics:

(i) We specialize an existing global dictionary by utilizing auxiliary samples obtained from the target subdataset, valid for finer granularity and a small quantity of examples (hence less computations).

(ii) Compared with the global one, the customized dictionary has the same size but smaller reconstruction errors and better representation of the target subdataset.

(iii) The customization for each subdataset is independent; thus we can customize an arbitrary number of subdatasets or attain a particular one alone.

As depicted in Figure 1, we first observed that the corresponding dictionary atoms of the global and the particular subjects often look "similar." This is reasonable, as the dictionary atoms describe the sketches of the object and the basic shapes of all the subjects are consistent. For a more rigorous theoretical analysis, we further considered dictionary identifiability [13] for mixed bounded signal models, that is, signals that are generated from more than one source (reference dictionary). And we proved that if reference dictionaries were close in the sense of the Frobenius norm, the global dictionary learned from mixed signals would be close to each of them. In fact, the global dictionary grabs the common basic shapes of all the subdatasets, regarding characteristics of the subjects as noise and discarding them.

Thus, formulating the dictionary customization problem, we introduced a regularizer penalizing the difference between the global and the customized dictionary. By minimizing the sum of the reconstruction error and above regularizer under sparsity constraints, we exploit the characteristic of the target subdataset contained in the auxiliary samples while maintaining the basic shapes stored in the global dictionary. As a result, a better dictionary, closer to the global one, is obtained. The solution is an asymptotic unbiased estimation of the underlying dictionary and can be seen as a trade-off between learning a new one from data and using an existing one.

To minimize the object function, we considered a general strategy the same as dictionary learning, that is, coding the samples and updating the atoms alternately in each iteration. Further, we present an algorithm that shares the idea with K-SCVD [10], which we call C-Ksvd. The flow chart of our methods is demonstrated in Figure 2. Experiments on tasks such as denoising and superresolution illustrate that our approach can handle the customization problem effectively and efficiently, outperforming both the global one and the normal dictionary learning approach. In addition, our model is also promising for more tasks such as enhancing an insufficient learned dictionary.

2. Notations

Throughout this paper, we write matrices as uppercase letters and vectors as lowercase letters. Given $p > 0$, the l_p-norm of the vector $v \in \mathbb{R}^n$ is defined as $\|v\|_p \triangleq (\sum_{i=1}^n |v_i|^p)^{1/p}$. In particular, the l_0-norm $\|v\|_0$ counts the nonzero entries of v. Let $\operatorname{sign}(v)$ denote the vector such that its jth entry $[\operatorname{sign}(v)]_j$ is equal to zero if $v_j = 0$ and to one (resp., minus one) if $v_j > 0$ (resp., $v_j < 0$).

The *Frobenius norm* of the matrix M is denoted as $\|M\|_F \triangleq [\sum_{i=1}^m \sum_{j=1}^n |m_{ij}|^2]^{1/2}$ and *matrix* l_1-norm as $\|M\|_1 \triangleq \sum_{i=1}^m \sum_{j=1}^n |m_{ij}|^1$. Define the operator norm

$$\|M\|_{p,q} \triangleq \sup_{\|x\|_p \le 1} \|Mx\|_q, \tag{1}$$

where m_i denotes the ith column vector of M.

3. Dictionary Learning with Mixed Signals

Dictionary identifiability [13], that is, recovering a reference dictionary that is assumed to generate the observed signals, is important for the interpretation of the learned atoms. In particular, Gribonval et al. [16] proved that the loss function of dictionary learning admits a local minimum in the neighborhood of the dictionary generating the signals.

In this section, we consider that there are multiple reference dictionaries and that the signals generated from them are mixed. Further, we prove that if reference dictionaries are close to each other in the sense of the Frobenius norm, dictionary learning with mixed signals admits a local minimum near both reference dictionaries simultaneously.

Without loss of generality, we analyze the case of two signal sources S_1, S_2. In particular, for the signal source S_i ($i = 1$ or 2), assume its signals $x^i \in \mathbb{R}^m$ are generated by model

$$S_i\colon\ x^i = D^i \alpha^i + \varepsilon^i, \tag{2}$$

where $D^i \in \mathbb{R}^{m \times p}$ is the reference dictionary of S_i, $\alpha^i \in \mathbb{R}^p$ is the coefficient, and $\varepsilon^i \in \mathbb{R}^p$ is the noise.

Particularly, the coefficient α^i is drawn on index set $J \subset \{1, 2, \ldots, p\}$ such that $\alpha^i_{J^c}$ is a zero vector and α^i_J is a random vector. Assume α^i_J and ε^i satisfy the following assumptions similar to [15], where we denote $\xi^i = \operatorname{sign}(\alpha^i)$.

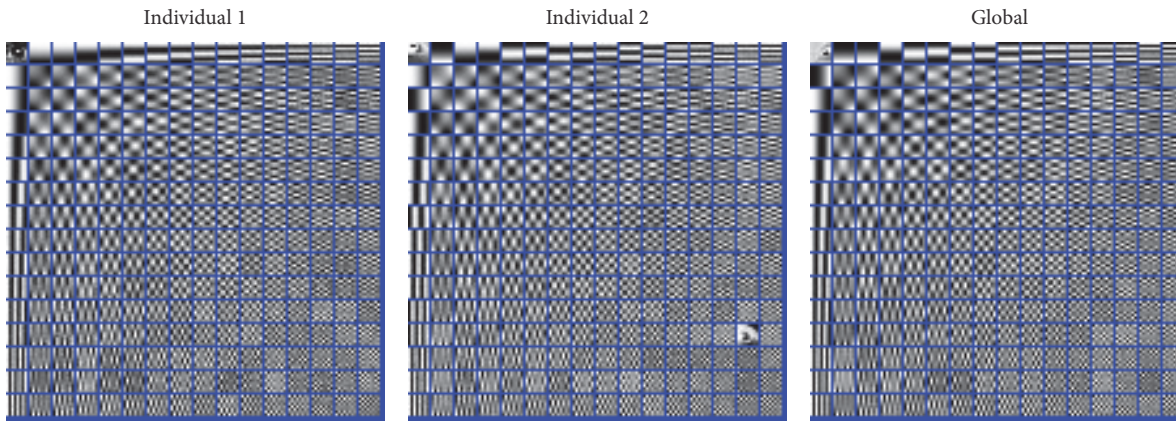


Figure 1: Three sorted dictionaries corresponding to two individual ones and the global one are demonstrated as images. Each one has a size of 64×256. The dictionaries for individuals 1 and 2 are trained from 40,000 patches picked from 24 of their corresponding facial images. The global one is trained from 80,000 patches sampled from 200 facial images belonging to 50 different individuals.

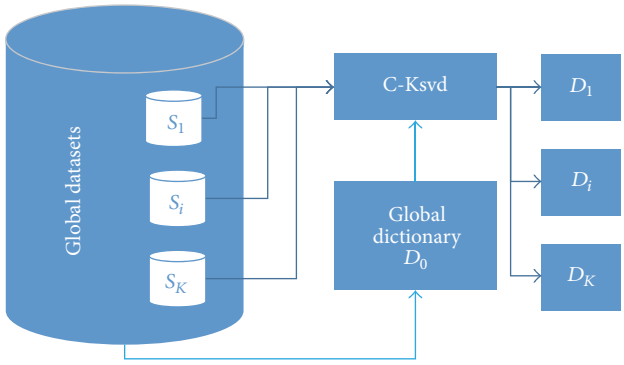


Figure 2: Flow chart of our methods.

Assumption 1 (basic and bounded signal assumption). There exist random variables α, ε, values $\underline{\alpha}$, M_α, and M_ε, such that

$$\begin{aligned}
&\mathbb{E}\left\{\alpha_J^i \left[\alpha_J^i\right]^T \mid J\right\} = \mathbb{E}\left\{\alpha^2\right\} \cdot I,\\
&\mathbb{E}\left\{\xi_J^i \left[\xi_J^i\right]^T \mid J\right\} = I,\\
&\mathbb{E}\left\{\alpha_J^i \left[\xi_J^i\right]^T \mid J\right\} = \mathbb{E}\left\{|\alpha|\right\} \cdot I,\\
&\mathbb{E}\left\{\varepsilon^i \left[\varepsilon^i\right]^T \mid J\right\} = \mathbb{E}\left\{\varepsilon^2\right\} \cdot I,\\
&\mathbb{E}\left\{\varepsilon^i \left[\alpha_J^i\right]^T \mid J\right\} = \mathbb{E}\left\{\varepsilon^i \left[\xi_J^i\right]^T \mid J\right\} = 0.\\
&\mathbb{P}\left(\min_{j\in J} \left|\alpha_j^i\right| < \underline{\alpha} \mid J\right) = 0,\\
&\mathbb{P}\left(\left\|\alpha^i\right\|_2 > M_\alpha\right) = 0,\\
&\mathbb{P}\left(\left\|\varepsilon^i\right\|_2 > M_\varepsilon\right) = 0.
\end{aligned} \tag{3}$$

$$(i = 1, 2).$$

Remark 2. Almost all sparse signal models such as k-sparse Gaussians and Laplacians satisfy the first five formulas, which can be seen as a kind of abstract and generalizing of the basic sparse signal model.

Further, the additional assumptions that the signal is upper-bounded and lower-bounded are standard and mainly used to make the analysis simple and clear [15]. In practice, as digital data is gathered with sensors with limited dynamics and stored in float format with limited precision, the boundedness assumption seems to be reasonably relevant.

The index set J is called the support of α^i and the sparsity s is defined as the number of elements in J. Thus the signal model is parameterized by the sparsity s, the expected coefficient energy $\mathbb{E}\alpha^2$, the minimum coefficient magnitude $\underline{\alpha}$, maximum norm M_α, and the flatness $\kappa_\alpha \triangleq \mathbb{E}|\alpha|/\sqrt{\mathbb{E}\alpha^2}$.

Note these assumptions can be generalized to multiple sources case easily, and thus we have the following definition.

Definition 3 (mixed bounded signal source). A mixed signal source S_O is defined as the union set of several signal sources $S_1, S_2, S_3, \ldots$; that is,

$$S_O = S_1 \cup S_2 \cup S_3 \cup \cdots, \tag{4}$$

where each source generates the signals by the way described in (2). Further, if $S_1, S_2, S_3, \ldots$ satisfy the basic and bounded signal assumptions (3) simultaneously, we say that S_O is a *mixed bounded signal source* or satisfies a *mixed bounded signal model*.

Further, for the two signal sources' case, assume D^1 and D^2 are close in the sense of Frobenius distance; that is, there is a small $\zeta \in \mathbb{R}$, s.t. $\|D^1 - D^2\|_F \leq \zeta$. (As discussed in [15], a dictionary is invariant by sign flips and permutations of the atoms, and we simply assume the atoms have been tuned to attain the minimum distance.) Denoting d_j the jth column

of D, the cumulative coherence of a dictionary D is defined as

$$\mu_k(D) \triangleq \operatorname*{supsup}_{|J|\le k}\ \max_{j\notin J} \left\| D_J^T d_j \right\|_1. \tag{5}$$

The term $\mu_k(D)$ gives a measure of the level of correlation between columns of D. Moreover, the lower restricted isometry constant of a dictionary D, $\underline{\delta}(D)$, is the smallest number, for any $\alpha \in \mathbb{R}^m$, satisfying

$$\left(1 - \underline{\delta}(D)\right) \|\alpha\|_2^2 \le \|D\alpha\|_2^2. \tag{6}$$

Recall that, for a set $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$, the loss function of dictionary learning is

$$F_X(D) \triangleq \frac{1}{n}\sum_{i=1}^{n} \inf_{\alpha_i\in\mathbb{R}^p} \frac{1}{2}\|x_i - D\alpha_i\|_2^2 + g(\alpha_i), \tag{7}$$

where $g(\alpha)$ is a penalty function promoting sparsity. Now consider a set of mixed signals $Y = [X_1, X_2] = [x_1^1, \ldots, x_n^1, x_{n+1}^2, \ldots, x_{2n}^2]$, where $X_1 \subset S_1$ and $X_2 \subset S_2$; the dictionary learning can be formulated as

$$\min_{D\in\mathscr{D}} F_Y(D) = \frac{1}{2}F_{X_1}(D) + \frac{1}{2}F_{X_2}(D), \tag{8}$$

where $\mathscr{D}$ denotes the set of dictionaries with unit l_2 norm atoms. Further, we have the following asymptotic result.

Theorem 4. *S_O is a mixed bounded signal source described above which consists of two signal sources S_1 and S_2. Without loss of generality, let $\|D^1\|_{2,2}^2 \ge \|D^2\|_{2,2}^2$. And assume the cumulative coherence $\mu_s(D^i)$ and the sparsity level s satisfy*

$$\mu_s\left(D^i\right) \le \frac{1}{4},$$
$$s \le \frac{p}{16\left(\left\|D^i\right\|_{2,2}+1\right)^2}. \tag{9}$$
$$(i = 1, 2).$$

Further, we define

$$\begin{aligned} C_{\min}^i &\triangleq 24\kappa_\alpha^2 \frac{s}{p}\left(\left\|D^i\right\|_{2,2}+1\right)\left\|\left[D^i\right]^T D^i - I\right\|_F, \\ C_{\min} &\triangleq \max\left(C_{\min}^1, C_{\min}^2\right), \\ C_{\max}^i &\triangleq \frac{2}{7}\cdot\frac{E|\alpha|}{M_\alpha}\left(1 - 2\mu_s\left(D^i\right)\right), \\ C_{\max} &\triangleq \min\left(C_{\max}^1, C_{\max}^2\right). \end{aligned} \tag{10}$$

And assume $C_{\min} < C_{\max}$. Define $t = \mathbb{E}\|\varepsilon\|_2^2/\mathbb{E}\|\alpha^0\|_2^2$. Moreover, let $g(\alpha) = \lambda\|\alpha\|_1$ with a regularization parameter $\lambda \le (1/4)\,\underline{\alpha}$ and denote $\overline{\lambda} \triangleq \lambda/E|\alpha|$, $\pi = \pi(D^1 - D^2)$, $\underline{\delta} = \max(\underline{\delta}(D^1), \underline{\delta}(D^2))$. Then there exists a radius r which satisfies $C_{\min}\overline{\lambda} + \zeta < r < C_{\max}\overline{\lambda}$, $M_\varepsilon/M_\alpha < (7/2)(C_{\max}\overline{\lambda} - r)$, and

$$1 + t < \frac{\sqrt{(1-\underline{\delta})}}{8\pi\sqrt{s}\left\|D^2\right\|_{2,2}^2}\left(\frac{r}{\zeta} - 1\right)\left(r - \zeta - C_{\min}\overline{\lambda}\right), \tag{11}$$

such that the expectation of the function $F_Y(D)$ admits a local minimum $\widehat{D}$ that $\|\widehat{D} - D^1\|_F < r$, $\|\widehat{D} - D^2\|_F < r$.

Let us consider in more detail the assumptions in Theorem 4.

(i) $\mu_s(D^i) \le 1/4$ and $s \le p/16(\|D^i\|_{2,2}+1)^2$ assume upper bounds of the correlation level between columns of D^i and the sparsity s. This is common in the analysis of sparse learning [21].

(ii) The condition $C_{\min} < C_{\max}$ would be satisfied with small s/p. The smaller s/p is, the larger $C_{\max} - C_{\min}$ would be.

(iii) $\lambda \le (1/4)\,\underline{\alpha}$ impose an upper limit on admissible regularization parameters. Note that limits on regularization parameters are also frequent [22].

(iv) $M_\varepsilon/M_\alpha < (7/2)(C_{\max}\overline{\lambda} - r)$ requires the level of noises. In particular, noiseless situation, that is, $M_\varepsilon = 0$, is a special case. Besides, t would be particularly small; for example, if the noise level is 30 dB, then $1 + t = 1.001$.

(v) Consider $C_{\min}\overline{\lambda} + \zeta < r < C_{\max}\overline{\lambda}$ and

$$1 + t < \frac{\sqrt{(1-\underline{\delta})}}{8\pi\sqrt{s}\left\|D^2\right\|_{2,2}^2}\left(\frac{r}{\zeta} - 1\right)\left(r - \zeta - C_{\min}\overline{\lambda}\right), \tag{12}$$

and we can rewrite them as

$$\begin{aligned} \zeta &\le \left(C_{\max} - C_{\min}\right)\overline{\lambda}, \\ \zeta &\lesssim \frac{\sqrt{(1-\underline{\delta})}}{8\pi\sqrt{s}\left\|D^2\right\|_{2,2}^2}, \end{aligned} \tag{13}$$

so the conditions would be satisfied for small ζ, in line with that D^1 and D^2 are close.

To conclude, the assumptions will hold for small cumulative coherence $\mu_s(D^i)$, sparsity s, noise level M_ε, dictionary distance ζ, and chosen regularization parameter λ.

Remark 5. For the radius r, it is lower-bounded by $C_{\min}$, $\overline{\lambda}$, and ζ. While ζ is fixed, if the sparsity s is particularly small, $C_{\min}\overline{\lambda}$ will be very small as well and the lower bound of r will be close to ζ. While s is fixed, and ζ tends to zero, that is, the mixed signal model degenerated into one single source case, then

$$1 + t < \frac{\sqrt{(1-\underline{\delta})}}{8\pi\sqrt{s}\left\|D^2\right\|_{2,2}^2}\left(\frac{r}{\zeta} - 1\right)\left(r - \zeta - C_{\min}\overline{\lambda}\right) \tag{14}$$

will be held forever and Theorem 4 degenerated into the case in [15], implying that the discussion in [15] can be seen as a special case of ours.

Moreover, the upper bound of r is implied to be less than 0.15, which can be concluded by a discussion similar to [15].

Remark 6. Theorem 4 can be generated to n ($n > 2$) sources case easily by considering a loss

$$\min_{D\in\mathscr{D}} nF_Y(D) = F_{X_1}(D) + F_{X_2}(D) + F_{X_3}(D) + \cdots + F_{X_n}(D), \tag{15}$$

and the proof is similar.

Proof. Define the closed ball of a dictionary D with radius r as

$$\mathscr{B}(D,r) = \left\{D' \in \mathscr{D} : \left\|D' - D\right\|_F \le r\right\}. \tag{16}$$

□

Now consider D^1 and D^2, as $\|D^1 - D^2\|_F \le \zeta$ and $\zeta < \zeta + C_{\min}\overline{\lambda} < r$, and the two balls $\mathscr{B}(D^1, r)$ and $\mathscr{B}(D^2, r)$ have intersection $U = \mathscr{B}(D^1, r) \cap \mathscr{B}(D^2, r)$ with D^1, D^2 contained. Denote $\mathscr{T}$ as the boundary of $\mathscr{U}$.

Further, for a set of samples X and two dictionaries D, D', define

$$\begin{aligned} f_X(D) &\triangleq \mathbb{E}F_X(D), \\ \Delta f_X\left(D, D'\right) &\triangleq f_X(D) - f_X\left(D'\right), \\ \Delta f_X\left(\mathscr{T}, D'\right) &\triangleq \inf_{D\in\mathscr{T}} \Delta f_X\left(D, D'\right). \end{aligned} \tag{17}$$

Note that $2f_Y(D) = f_{X_1}(D) + f_{X_2}(D)$; then we have

$$\begin{aligned} 2\Delta f_Y\left(\mathscr{T}, D^1\right) &= \Delta f_{X_1}\left(\mathscr{T}, D^1\right) + \Delta f_{X_2}\left(\mathscr{T}, D^1\right) \\ &= \Delta f_{X_1}\left(\mathscr{T}, D^1\right) + \Delta f_{X_2}\left(\mathscr{T}, D^2\right) \\ &\quad - \Delta f_{X_2}\left(D^1, D^2\right). \end{aligned} \tag{18}$$

When $g(\alpha) = \lambda\|\alpha\|_1$, the function $F_X(D)$ is Lipschitz continuous with respect to Frobenius metric on the compact constraint set $\mathscr{D} \subset \mathbb{R}^{m\times p}$ [16]. Thus by choosing a radius r such that $\Delta f_Y(\mathscr{T}, D^1) > 0$, the compactness of the closed set $\mathscr{U}$ will then imply the existence of a local minimum $\widehat{D}$ of $F_Y(D)$ such that $\|\widehat{D} - D^1\|_F < r$, $\|\widehat{D} - D^2\|_F < r$. Now let us bound each item of (18).

First note that assuming $\mu_s(D^1) \le 1/4$, $s \le p/16(\|D^1\|_{2,2} + 1)^2$, $\lambda \le (1/4)\underline{\alpha}$, and $M_\varepsilon/M_\alpha < (7/2)(C_{\max}\overline{\lambda} - r)$, then, by the proof of theorem 1 in [15], for any radius $d \in (C_{\min}\overline{\lambda}, C_{\max}\overline{\lambda})$ and any dictionary D that $\|D - D^1\|_F = d$, we have

$$\Delta f_{X_1}\left(D, D^1\right) \ge \frac{\mathbb{E}\alpha^2}{8} \cdot \frac{s}{p} \cdot d\left(d - C^1_{\min}\overline{\lambda}\right) > 0. \tag{19}$$

Further, $\mathbb{E}\alpha^2/8 \cdot s/p \cdot d(d - C^1_{\min}\overline{\lambda})$ is monotonically increasing for $d > C_{\min}\overline{\lambda}$ and for $D \in \mathscr{T}$, we have $\|D - D^1\|_F \ge r - \zeta > C_{\min}\overline{\lambda}$. Thus

$$\Delta f_{X_1}\left(\mathscr{T}, D^1\right) \ge \frac{\mathbb{E}\alpha^2}{8} \cdot \frac{s}{p} \cdot (r - \zeta)\left(r - \zeta - C_{\min}\overline{\lambda}\right). \tag{20}$$

For the second item $\Delta f_{X_2}(\mathscr{T}, D^2)$, we have the same lower bound similarly.

Moreover, for the dictionary D^2 and any coefficient α with sparsity s, we have

$$\frac{1 - \underline{\delta}}{s}\|\alpha\|_1^2 \le (1 - \underline{\delta})\|\alpha\|_2^2 \le \left\|D^2\alpha\right\|_2^2. \tag{21}$$

Then by the theorem 2 and lemma 6 in [16], when $g(\alpha)$ is l_1 norm, we have

$$\left|F_{X_2}\left(D^1\right) - F_{X_2}\left(D^2\right)\right| \le L_{X_2}\left\|D^1 - D^2\right\|_{1,2}, \tag{22}$$

where $L_{X_2} = 2\sqrt{s}/(n\sqrt{1 - \underline{\delta}}) \cdot \|X_2\|_F^2$. Thus

$$\begin{aligned} &\left|F_{X_2}\left(D^1\right) - F_{X_2}\left(D^2\right)\right| \\ &\le \frac{2\sqrt{s}}{n\sqrt{1 - \underline{\delta}}} \cdot \frac{\pi}{p} \cdot \|X_2\|_F^2\left\|D^1 - D^2\right\|_F. \end{aligned} \tag{23}$$

Assume x is a sample in X_2 and its sparse coefficient and noise coefficient are α^0 and ε. As $x = D^2\alpha^0 + \varepsilon$, we have

$$x^2 = \left(D^2\alpha^0\right)^2 + \varepsilon^2 + 2\left(D^2\alpha^0\right)^T\varepsilon; \tag{24}$$

taking expectation on each side of it, by assumptions in (2), as $\mathbb{E}\|\alpha^0\|_2^2 = s\mathbb{E}\alpha^2$ and $\mathbb{E}(D^2\alpha^0)^T\varepsilon = 0$, then

$$\begin{aligned} \mathbb{E}\|x\|_2^2 &\le \left\|D^2\right\|_{2,2}^2 \mathbb{E}\left\|\alpha^0\right\|_2^2 (1 + t) \\ &\le s\left\|D^2\right\|_{2,2}^2 \mathbb{E}\alpha^2 (1 + t); \end{aligned} \tag{25}$$

thus $\mathbb{E}\|X_2\|_F^2 \le ns\|D^2\|_{2,2}^2\mathbb{E}\alpha^2(1 + t)$. Taking expectation on each side of (23), we have

$$\left|\Delta f_{X_2}\left(D^1, D^2\right)\right| < \frac{2s\sqrt{s}(1 + t)}{\sqrt{1 - \underline{\delta}}} \cdot \frac{\pi}{p} \cdot \mathbb{E}\alpha^2\left\|D^2\right\|_{2,2}^2 \cdot \zeta. \tag{26}$$

By (18), (20), and (26), as long as

$$1 + t < \frac{\sqrt{(1 - \underline{\delta})}}{8\pi\sqrt{s}\left\|D^2\right\|_{2,2}^2}\left(\frac{r}{\zeta} - 1\right)\left(r - \zeta - C_{\min}\overline{\lambda}\right), \tag{27}$$

we have

$$\begin{aligned} 2\Delta f_Y\left(\mathscr{T}, D^1\right) &\ge 2 \cdot \frac{s\mathbb{E}\alpha^2}{8p} \cdot (r - \zeta)\left(r - \zeta - C_{\min}\overline{\lambda}\right) \\ &\quad - \frac{2\pi s\sqrt{s}}{p\sqrt{1 - \underline{\delta}}} \cdot \mathbb{E}\alpha^2\left\|D^2\right\|_{2,2}^2 \cdot \zeta > 0, \end{aligned} \tag{28}$$

which means that $\mathbb{E}F_Y(D)$ admits a local minimum $\widehat{D}$ in $\mathscr{U}$; that is, $\|\widehat{D} - D^1\|_F < r$, $\|\widehat{D} - D^2\|_F < r$.

The result is reasonable, as when those reference dictionaries are similar, the dictionary learned from the mixed signals should be similar to each of them, in order to get less reconstruction errors for each subdataset and hence a lower total loss.

4. The Regularizer and Dictionary Customization Problem

Now we turn back to the dictionary customization problem. In particular, the dataset S_O consists of several separable subdatasets $S_A, S_B, S_C, \ldots$; that is, $S_O = S_A \cup S_B \cup S_C \cup \cdots$. Further, $D_0 \in \mathbb{R}^{m\times p}$ ($p \gg m$) is an existing global dictionary corresponding to S_O. This is common, as the dictionary for facial images would always be well trained, and the corresponding dataset can be divided by different individuals. Then we would like to customize D_0 with some auxiliary samples $X = \{x_i \mid x_i \in \mathbb{R}^m\}_{i=1}^n \subset S_A$, requiring that the customized dictionary D has the same size but behaves better on S_A.

Obviously, D should have sparse representations and small reconstruction errors on X, which corresponds to minimizing $\sum_{i=1}^n \|x_i - Dw_i\|$ under sparsity constraint $\|w_i\|_0 \le s$. Further, noting that $S_A, S_B, S_C, \ldots$ can be regarded as several signal sources and, hence, S_O as a mixed bounded model. Moreover, accounting for the fine granularity, the differences between those subdatasets $S_A, S_B, S_C, \ldots$ are small and the basic sketches of them are consistent, implying that the underlying dictionaries for all subdatasets are similar. Thus, D_0 should, according to Theorem 4, be close to our customized dictionary D as well, which is also in accordance with the practical observation. Considering the distance induced by the Frobenius norm, this leads directly to a regularizer $\|D - D_0\|_F$.

Denote $E = D - D_0$; then the customization model can be formulated as a sum of the reconstruction errors and the above regularizer; that is,

$$\begin{aligned} \arg\min_{E,W} \quad & \left\|X - (D_0 + E)W\right\|_F^2 + \gamma \|E\|_F^2, \\ \text{s.t.} \quad & \forall i,\ \|w_i\|_0 \le s, \end{aligned} \tag{29}$$

where w_i represents the ith column vector of W, $s \ll m$ is the sparsity number, and $\gamma \ge 0$ is the parameter balancing the prior knowledge of D_0 and the information in X.

It is worth noting that problem (29) is connected with the matrix version of total least squares (TLS) problems [23], which generalized the least squares by assuming noises in both dependent and independent variables. This is interpretable: as mentioned above, the atoms of the global dictionary only grab the main sketches. They regard the characteristics belonging to different subdatasets as noise and discard them. As a result, when considering a particular subdataset, the characteristic information is absent and thus the corresponding atoms of D_0 can be seen as noisy. Different from TLS, the tuning parameter γ is necessary, as noises in D_0 and X are different and should be balanced. We further depict model (29) with the following properties.

Theorem 7. *Consider customization problem (29), where $X = \{x_i\}_{i=1}^n$ is the auxiliary data, D_0 is the global dictionary, D_* is the true one corresponding to the target subdataset, and $\overline{D}$ is the customized one attained from (29); then*

(1) *denote $\mathscr{W} = \{w \in \mathbb{R}^p \mid \|w\|_0 \le s\}$; for any $\gamma \ge 0$,*

$$\inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - \overline{D}w_i\right\|_F \le \inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - D_0 w_i\right\|_F; \tag{30}$$

(2) *for a fixed γ, when n tends to infinity, $\overline{D}$ will converge to D_*; in other words, the minimizer of (8) is an asymptotically unbiased estimator of D_*;*

(3) *the tuning parameter γ reflects the confidence in D_0; in particular, if $\gamma \to \infty$, $\overline{D} = D_0$; if $\gamma = 0$, (29) will degrade into a common dictionary learning problem.*

Proof. For 1, as $\overline{D}$ is the optimal solution of problem (29), then, for any $\gamma \ge 0$ and $D \in \mathbb{R}^{m\times p}$, we have

$$\begin{aligned} &\inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - \overline{D}w_i\right\|_F + \gamma \left\|\overline{D} - D_0\right\|_F^2 \\ &\quad \le \inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - Dw_i\right\|_F + \gamma \left\|\overline{D} - D_0\right\|_F^2. \end{aligned} \tag{31}$$

Let $D = D_0$; then we have

$$\begin{aligned} \inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - \overline{D}w_i\right\|_F &\le \inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - \overline{D}w_i\right\|_F \\ &\quad + \gamma \left\|\overline{D} - D_0\right\|_F^2 \\ &\le \inf_{w_i\in\mathscr{W}} \sum_{i=1}^n \left\|x_i - D_0 w_i\right\|_F, \end{aligned} \tag{32}$$

and the equality holds only when $\overline{D} = D_0$.

For 2, reshape the loss function as

$$\arg\min_{E,\{w_i\}} \frac{1}{n}\sum_{i=1}^n \left\|x_i - (D_0 + E)w_i\right\|_2^2 + \frac{\gamma}{n}\|E\|_F^2. \tag{33}$$

When n tends to infinity, the penalty will tend to zero and thus the loss function will degenerate into the common dictionary learning form.

For 3, it is easy to see that γ reflects the weight of the penalty in the loss function and the conclusion is reasonable. □

According to the third property of Theorem 8, customization can be seen as a trade-off between learning a dictionary and using an existing one, which fills the void between them and implies a more flexible dictionary selection strategy. In particular, for datasets with coarse granularity, select dictionary learning with large amounts of samples. For subjects with fine granularity, customize the existing one with some auxiliary samples, and use a predefined dictionary if no sample is available.

We also emphasize that our model (29) will be valid as long as the assumption is satisfied (i.e., $\|D_0 - D_*\|_F \le \zeta$). As demonstrated in the experiments, there are more applications, such as improving an insufficient learned dictionary or correcting a contaminated one. In addition, for the

regularizer, more matrix norms can be selected as well. For example, consider the distance induced by *matrix* l_1-norm; then E will be sparse and the consumptions in storage and transmission will be reduced greatly.

5. Optimization

In this section, we first introduce a general optimization strategy and then devise a more straightforward dictionary updating strategy similar to K-SVD [10].

5.1. A General Strategy. A general optimization strategy, not necessarily leading to a global optimum, can be found by splitting the problem into two parts which are alternately solved within an iterative loop. The two parts are as follows.

5.1.1. Sparse Coding. Keeping E fixed, find W by

$$\begin{aligned} \min_{W} \quad & \left\| X - (D_0 + E) W \right\|_F^2 \\ \text{s.t.} \quad & \forall i, \ \left\| w_i \right\|_0 \le s. \end{aligned} \tag{34}$$

This can be solved by pursuit algorithms such as OMP [24], FOCUSS [25], or relaxed to Lasso [26].

5.1.2. Dictionary Updating. Keeping W fixed, find E by

$$\min_{E} \left\| X - (D_0 + E) W \right\|_F^2 + \gamma \left\| E \right\|_F^2. \tag{35}$$

This is a quadratic programming problem with a closed-form solution $E = (X - D_0 W) W^T (\gamma I + W W^T)^{-1}$.

5.2. C-Ksvd Algorithm. We now turn to a more involved dictionary updating strategy: rather than freezing the coefficient matrix W, we update $D = D_0 + E$ together with the nonzero coefficients (i.e., only the support is fixed).

In particular, assume that both W and E are fixed except for one column e_k in the correction matrix E and the coefficients that correspond to it, the kth row in W, denoted as w_T^k. Then the loss function can be rewritten as

$$\begin{aligned} & \left\| X - (D_0 + E) W \right\|_F^2 + \gamma \left\| E \right\|_F^2 \\ & = \left\| X - \sum_{i \ne k}^{p} \left(d_i^0 + e_i \right) w_T^i - \left(d_k^0 + e_k \right) w_T^k \right\|_F^2 \\ & \quad + \gamma \left\| e_k \right\|_2^2 + C \\ & = \left\| M_k - \left(d_k^0 + e_k \right) w_T^k \right\|_F^2 + \gamma \left\| e_k \right\|_2^2 + C, \end{aligned} \tag{36}$$

where d_k^0 is the kth column of D_0, $C = \gamma \sum_{i \ne k}^{p} \left\| e_i \right\|_2^2$ is a constant, and $M_k = X - \sum_{i \ne k}^{p} (d_i^0 + e_i) w_T^i$ represents the error when the kth dictionary atom is removed.

Now we shrink the loss function to the support of row vector w_T^k. Define δ_k as the group of indices pointing to samples $\{x_i\}$ that use the atom $d_k = d_k^0 + e_k$; that is, $\delta_k = \{i \mid 1 \le i \le p, w_T^k(i) \ne 0\}$. Further, define $\Omega_k \in \mathbb{R}^{m \times |\delta_k|}$ as ones on the $(\delta_k(i), i)$th entries and zeros elsewhere. Then problem (29) is transformed to

$$\min_{e_k, w_R^k} \left\| M_k^R - \left(d_k^0 + e_k \right) w_R^k \right\|_F^2 + \gamma \left\| e_k \right\|_2^2, \tag{37}$$

where $M_k^R = M_k \Omega_k$ and $w_R^k = w_T^k \Omega_k$. For this subproblem, we have the following results.

Theorem 8. *Suppose the largest singular value and the corresponding singular vectors of matrix* $[\sqrt{\gamma} d_k^0, M_k^R] \in \mathbb{R}^{m \times (n+1)}$ *are* σ_1, u_1, *and* v_1. v_{11} *is the first element of* v_1. *Then, the unique solution for problem (37) is*

$$\begin{aligned} e_k &= \frac{\sigma_1 v_{11}}{\sqrt{\gamma}} u_1 - d_k^0, \\ w_R^k &= \frac{d_k^T M_k^R}{\left\| d_k \right\|_2^2}, \end{aligned} \tag{38}$$

where $d_k = d_k^0 + e_k = (\sigma_1 v_{11} / \sqrt{\gamma}) u_1$.

Proof. Denote $d_k = d_k^0 + e_k$; then

$$\begin{aligned} & \left\| M_k^R - \left(d_k^0 + e_k \right) w_R^k \right\|_F^2 + \gamma \left\| e_k \right\|_2^2 \\ & = \left\| M_k^R - d_k w_R^k \right\|_F^2 + \left\| \sqrt{\gamma} \left(d_k^0 - d_k \right) \right\|_2^2 \\ & = \left\| \left[\sqrt{\gamma} d_k^0, M_k^R \right] - \left[\sqrt{\gamma} d_k, d_k w_R^k \right] \right\|_F^2. \end{aligned} \tag{39}$$

□

As $[\sqrt{\gamma} d_k, d_k w_R^k] = d_k [\sqrt{\gamma}, w_R^k]$ is the product of two vectors, its rank is one. Then problem (37) can be rewritten as

$$\min_{d_k, w_R^k} \left\| \left[\sqrt{\gamma} d_k^0, M_k^R \right] - d_k \left[\sqrt{\gamma}, w_R^k \right] \right\|_F^2. \tag{40}$$

And thus $d_k [\sqrt{\gamma}, w_R^k]$ is the best rank-one approximation of $[\sqrt{\gamma} d_k^0, M_k^R]$. By Eckart-Young-Mirsky Theorem [27], we have $d_k [\sqrt{\gamma}, w_R^k] = \sigma_1 u_1 v_1$; thus $d_k = (\sigma_1 v_{11} / \sqrt{\gamma}) u_1$, and $e_k = d_k - d_k^0$. Taking e_k back into the original problem (37), it becomes a least squares problem and we have

$$w_R^k = \frac{d_k^T M_k^R}{\left\| d_k \right\|_2^2}. \tag{41}$$

Thus, problem (37) has a closed-form solution and the main computation is top SVD of $[\sqrt{\gamma} d_k^0, M_k^R]$. In the dictionary updating stage, we can suggest minimization with respect to each column d_k (for simplicity, omitting e_k, we directly use d_k in updating) and corresponding w_T^k in sequence, forcing the support of the coefficients fixed. The complete algorithm, named "C-Ksvd", is described as Algorithm 9. Noting that while K-SVD computes the top SVD of matrix $M_k^R \in \mathbb{R}^{n \times |\delta_k|}$ for the kth column, C-Ksvd computes that of $[\sqrt{\gamma} d_k^0, M_k^R] \in \mathbb{R}^{m \times (|\delta_k| + 1)}$.

Assuming that the sparse coding stage is performed perfectly, a local minimum is guaranteed, as the loss function is guaranteed to be nonincreasing at the update step for d_k and a series of such steps ensures a monotonic reduction. Compared with the general strategy, the updating for d_k is more straightforward as it allows tuning of the values of the corresponding coefficients. In addition, each atom can have a unique parameter γ_i as well, on behalf of the confidence level of the ith atom d_i.

Algorithm 9 (C-Ksvd algorithm).

Initialization: a global dictionary D_0, samples $\{x_i\}_{i=1}^n$.

Repeat:

(i) Sparse coding stage: use any sparse recovery algorithm to compute the coefficients w_i for each sample x_i by approximating the solution of

$$\begin{aligned} \min_{\{w_i\}} \quad & \left\|x_i - (D_0 + E)\, w_i\right\|_2^2 \\ \text{s.t.} \quad & \left\|w_i\right\|_0 \le s. \end{aligned} \tag{42}$$

(ii) Dictionary updating stage: for each column $k = 1, 2, \ldots, p$ in D, update it by

(a) Compute M_k by $M_k = X - \sum_{i \neq k} d_i w_T^i$.

(b) Define the group of samples that use this atom as δ_k. Restrict M_k and w_T^k by choosing the columns corresponding to δ_k, obtain M_k^R and w_R^k.

(c) Apply top SVD decomposition to $[\sqrt{\gamma} d_k^0, M_k^R]$, obtain σ_1, u_1, v_1. Update

$$\begin{aligned} d_k &= \frac{\sigma_1 v_{11}}{\sqrt{\gamma}} u_1, \\ w_R^k &= \frac{d_k^T M_k^R}{\left\|d_k\right\|_2^2}. \end{aligned} \tag{43}$$

Until convergence (stopping rule).

Output: a better dictionary D.

6. Experiments

We first showed the effectiveness of our approach on the denoising task, with analysis of the customized dictionary and the tuning parameter γ. Further, a novel superresolution experiment was illustrated, sharing the idea of transferring knowledge from a related auxiliary data source. In addition, we conducted an experiment that enhances an insufficient learned dictionary by C-Ksvd, illustrating that our model was also valid for more tasks.

6.1. Denoising. We demonstrated the customization results by denoising tasks on facial images drown from PIE Database [28]. The denoising process was similar to [1], which included sparse coding of each patch of the noisy image. As the coding performance relied heavily on the dictionary, we could assess the dictionary by the denoising results, which were evaluated by PSNR (Peak Signal Noise Ratio).

Table 1: Denoising results (PSNR, dB) on facial images of different individuals with the noise level $\sigma = 30$. For each image, three kinds of D_0 were considered.

Individual	Original	Type of D_0	D_0	K-SVD	Customized
1	18.77	Global I	26.62	26.63	27.34
		Global II	26.37	26.62	27.12
		DCT	25.42	26.34	26.61
2	18.52	Global I	26.73	26.24	27.39
		Global II	26.39	26.06	27.10
		DCT	25.72	25.85	25.82
3	18.63	Global I	26.83	27.16	27.78
		Global II	26.41	26.82	27.31
		DCT	25.84	26.49	26.57

In particular, the noisy images were produced by adding Gaussian noises with mean zero and different standard deviation σ. The patch size and the redundant factor were set as 16×16 and 4. (We chose them for the best visual effect while similar comparisons can be attained for different value.) OMP was used for coding and atoms were accumulated until the average error passed the threshold, chosen empirically to be $\varepsilon = 1.15 \cdot \sigma$. Results corresponding to three dictionaries were compared, that is, the global dictionary D_0, the one generated by K-SVD, and the one produced by our customization approach. In K-SVD and customization, D_0 was used as initialization and the iteration number was set to 10. Moreover, three kinds of D_0 were considered, denoted as "global I", "global II," and "DCT," respectively: (1) a dictionary learned by K-SVD, with 40,000 noiseless patches picked from 100 individuals; (2) similar to (1), but learned with noisy patches ($\sigma = 20$); (3) predefined DCT (discrete cosine transform).

Each experiment was repeated 5 times and results are depicted in Table 1 and Figure 3. It was seen that customization outperformed the global dictionary and K-SVD on both PSNR and visual effects, accounting for the fact that both the common sketches in D_0 and characteristics in X had been utilized. Particularly, note that denoising by D_0 tended to be too smooth, and results by K-SVD were likely to be too rough. Regarding DCT as a suboptimal global dictionary, the results also showed that our customization is valid for a wide range of D_0. Conducted on a i7-3770 CPU and processed with the same dataset X, the average running time for K-SVD and customization were 173.34 s and 48.21 s, respectively, showing that our approach is competitive. In particular, for K-SVD, 119.31 s were used for removing identical atoms. We also display the three dictionaries as images in Figure 4, showing that the customized one was similar to the global one, while the one corresponding to K-SVD was not.

In addition, we plotted the relations of the tuning parameter γ, the average number (AN) of coefficients for patches, and the PSNR after denoising in Figure 5. It was shown that γ could be chosen as the one attaining the minimum average number of coefficients by a quick one-dimensional search.

Figure 3: Examples of denoising different facial images. For the parameters of three rows, D_0 was chosen as global I, II, and DCT, and $\delta = 30, 20, 25$, respectively.

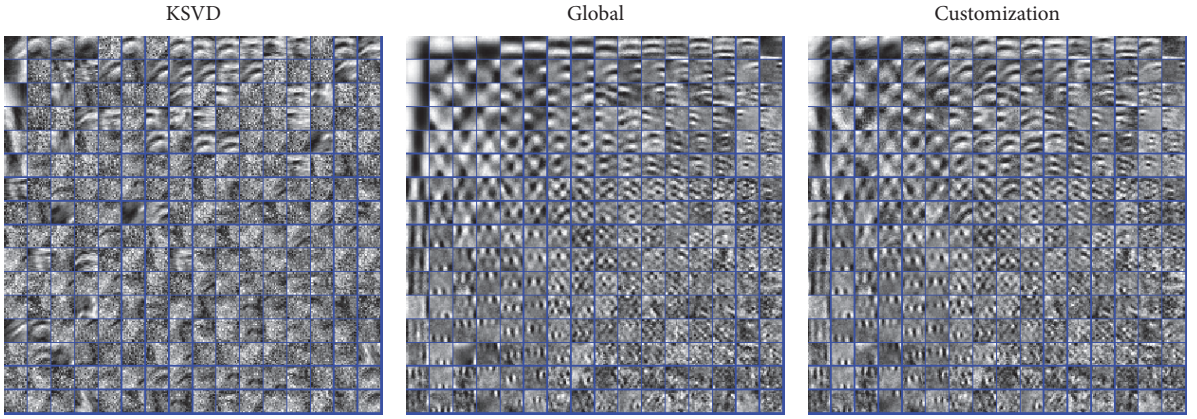


Figure 4: Three dictionaries were sorted and quarters at the top left corner were demonstrated while denoising a noisy image with $\sigma = 30$. For the convenience of comparison, the global dictionary learned by noisy patches is placed in the center.

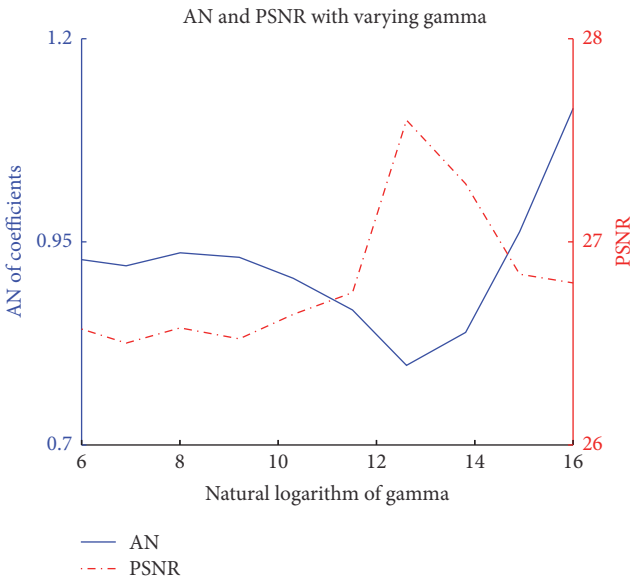


Figure 5: γ with highest PSNR was attained by the minimum average number of coefficients.

What is more, experimentally, for a fixed D_0, the best γ for different individuals was the same, which implies we only need to tune it once while customizing.

6.2. Superresolution. Yang et al. [29] proposed a scale-up algorithm via sparse signal representation, which contains two steps: dictionary learning and patch-pairs construction. To reduce the dimension and speed-up processing, Elad [30] applied PCA on the samples and used K-SVD for training. However, this learned dictionary was still a global dictionary, which means that we can further improve the performance of superresolution by customization.

Consider a global dictionary D_0 and patches $X = \{x_i\}_{i=1}^n$ sampled from related high-resolution images. We can customize this dictionary to a finer granularity. In particular, by substituting K-SVD, the low-resolution dictionary D_l and the coefficient W was customized by C-Ksvd, and the corresponding high-resolution dictionary was attained by

$$D_h = D_{h_0} + \left(X - D_{l_0} W\right) W^T \left(\gamma I + W W^T\right)^{-1}, \tag{44}$$

where D_{h_0} and D_{l_0} denoted the initial high-resolution and low-resolution dictionaries, respectively.

In this experiment, similar to the settings in [31], we evaluated the proposed approach on the Yale Face Database [32], which contains 11 different 100×100 facial images for each of the 15 individuals. A downscaled image was taken as the low-resolution object, and the down-scale factor was set to 3. Further, other images of the same individual were considered as high-resolution auxiliary data. The patch size was set to 3×3. The global dictionary D_0 was trained by 34,650 patches sampled from the 80% downscaled total dataset. (This is to highlight the relevance of auxiliary data and simulate real conditions, as the total training set is relatively small and clean in our experiment.) Results produced by D_0, K-SVD (i.e., the original version with D_0 as initialization and X as training data), and customization were compared. For customization, 225 patches were taken from each auxiliary image. For K-SVD, the total number of sampled patches was fixed to 6,000, to gain the best results.

Table 2: PSNR for superresolution on test images.

Task	Bicubic	D_0	DL3	Cus3	DL6	Cus6	DL9	Cus9
1	32.5	34.4	33.1	**35.0**	34.1	**35.4**	35.6	**35.7**
2	33.8	36.2	36.1	**36.9**	35.0	**36.8**	37.0	**37.2**
3	32.9	35.8	32.4	**36.6**	31.8	**37.0**	36.7	**37.3**
4	32.0	34.0	34.0	**34.7**	32.1	**34.9**	33.8	**35.1**
5	36.2	38.1	37.6	**38.7**	36.7	**38.9**	38.7	**38.9**

Varying the number of the auxiliary images and repeating the experiments on different individuals, the performance was evaluated by PSNR. Some of the results are summarized in Table 2. "DL" and "Cus" represent K-SVD and customization, respectively, and "3," "6," or "9" denote the number of auxiliary images. "Bicubic," that is, simple Bicubic interpolation, is shown as a baseline method.

It is seen that if the number of auxiliaries is small, results produced by K-SVD are worse than the common dictionary, implying that the learning is meaningless. However, even when the auxiliary data is small (675 patches from 3 images), superresolutions by customization have significant improvements. Further, customization still outperforms or is no worse than the learning approach when new data is added. Remember that the number of patches which K-SVD needs is much larger than that needed for customization, meaning more computations and time are required. Also note that once the dictionary has been customized, it is valid for all the images of the person.

6.3. Enhancing. As was mentioned above, model (29) can be applied to more tasks, as long as the assumption that D_0 and the reference dictionary D_* are close. In this subsection, we consider the case of enhancing an existing dictionary by C-Ksvd and evaluate the performance on classification.

In particular, LC-KSVD [33], one of the state-of-the-art methods for image classification, introduced a triple model (D, A, W), where D represents the dictionary, A stands for parameters of the label consistent term, and W denotes the linear classifier. Regarding $(D^T, \sqrt{\alpha}A^T, \sqrt{\beta}W^T)^T$ as a new dictionary, the following object function can be solved by K-SVD:

$$\begin{aligned} \arg\min_{D,W,A,X} \quad & \|Y - DX\|_2^2 + \alpha \|Q - AX\|_2^2 \\ & + \beta \|H - WX\|_2^2, \\ \text{s.t.} \quad & \forall i, \ \|x_i\|_0 \le T. \end{aligned} \tag{45}$$

Sometimes the model $M_0 = (D_0, A_0, W_0)$ learned is likely not good enough, due to the fact that the training data may be insufficient or too noisy. Moreover, over time the past training information often becomes unavailable. In this case, we can further enhance it by our customization model, simply replacing the K-SVD procedure with C-Ksvd.

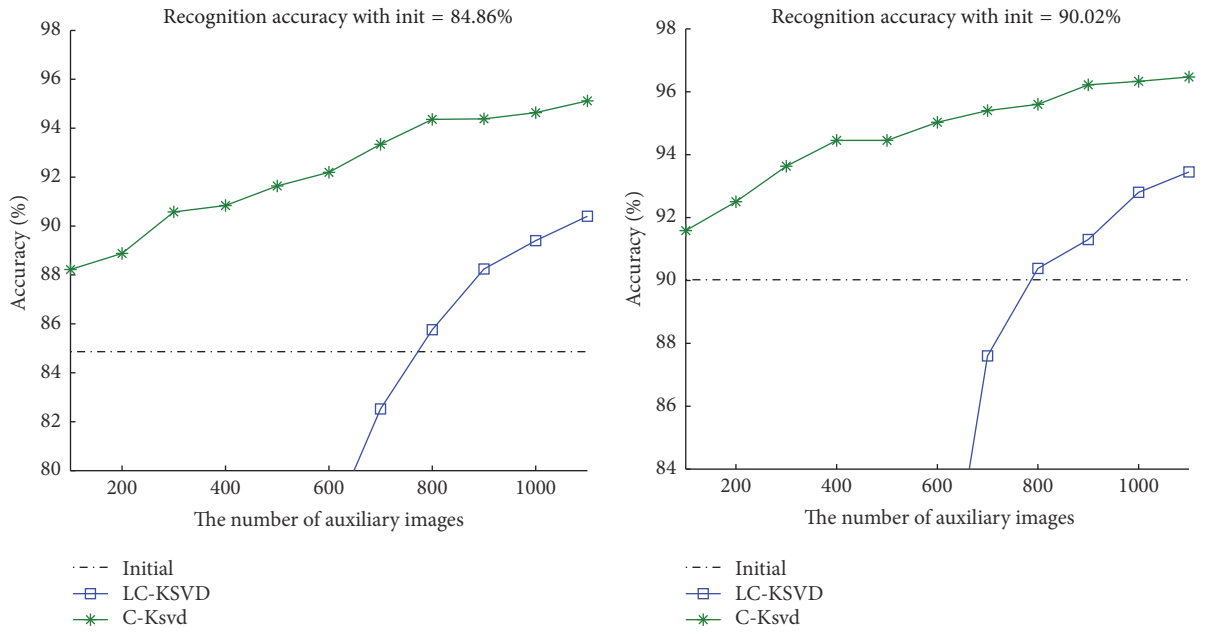


Figure 6: Varying the number of auxiliary images with fixed initial models.

Table 3: Accuracy (%) for different level initial and fixed number auxiliary images.

Init	76.76	79.32	84.86	88.04	90.41	93.30
LC-KSVD	85.32	85.36	85.76	88.69	90.23	90.83
C-Ksvd	**91.78**	**92.20**	**94.36**	**94.94**	**95.79**	**97.38**

Table 4: Accuracy (%) on enhanced classes, remainder classes, and all classes.

Classes	Init	LC-KSVD	C-Ksvd
Enhanced	83.90	92.05	**93.40**
Remainder	84.86	0.0	**82.83**
All	84.38	46.02	**88.11**

In accordance with [33], we used the "Extended Yale-B" dataset [34] to demonstrate that the performance and the data were divided into three parts: training data to obtain the initial model, auxiliary data for model enhancement, and test data to evaluate it. Parameters α and β were tuned while training the initial models and then kept fixed. The initial model M_0, LC-KSVD, and C-Ksvd were compared, where LC-KSVD used M_0 as initialization and X as training data. Results were analyzed in three ways.

(1) Initial models of different levels were obtained by tuning the number of training samples, and we tried to promote the model with 800 auxiliary images. After repeating the experiment 5 times for each level, the averaged recognition accuracies are summarized in Table 3.

It is seen that C-Ksvd is valid in a wide range of initial models and always significantly outperforms LC-KSVD. Besides, influences of the initial models on LC-KSVD are relatively small, in accordance with our previous analysis.

(2) For fixed initial models, varying the number of auxiliary images from 100 to 1100, we plotted the corresponding recognition results in Figure 6 and found that the accuracy had a significant increase, even when the auxiliary number was relatively small. To gain a competitive result for LC-KSVD, large amounts of images were required, which was unaffordable.

(3) In the previous discussion, the auxiliary images were uniformly sampled from all 38 individuals. Then we considered the nonuniform case where only images of several classes (named "enhanced classes") were available. After setting the number of enhanced classes as 19, and getting 31 images from each class, the results are reported in Table 4.

While C-Ksvd improved the accuracy on enhanced classes, the accuracy on the remainder was slightly reduced, owing to the similarity of the original and new dictionaries. LC-KSVD presented a sharp contrast.

7. Conclusion

In this paper, we considered the dictionary customization problem, which can be seen as a trade-off between learning a new dictionary from data and using an existing one. We investigated our hypothesis with theoretical analysis and formulated a model by raising a specific regularizer. An efficient algorithm was proposed, and experiments on real-world data demonstrate that our approach is promising.

Competing Interests

The authors declare that they have no competing interests.

References

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

[2] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2586–2593, Providence, RI, USA, June 2012.

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[4] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2691–2698, June 2010.

[5] H. Liu, Y. Liu, and F. Sun, "Robust exemplar extraction using structured sparse coding," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 26, no. 8, pp. 1816–1821, 2015.

[6] H. Liu, D. Guo, and F. Sun, "Object recognition using tactile measurements: kernel sparse coding methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 656–665, 2016.

[7] H. Liu, Y. Yu, F. Sun, and J. Gu, "Visual-tactile fusion for object recognition," *IEEE Transactions on Automation Science and Engineering*, 2016.

[8] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[9] K. Engan, S. O. Aase, and J. H. Husoy, "Method of Optimal Directions for frame design," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, vol. 5, pp. 2443–2446, March 1999.

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

[11] M. Zhou, H. Chen, J. Paisley et al., "Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th International Conference on Machine Learning (ICML '09)*, pp. 689–696, ACM, Montreal, Canada, June 2009.

[13] K. Schnass, "Local identification of overcomplete dictionaries," https://arxiv.org/abs/1401.6354.

[14] K. Schnass, "On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD," *Applied and Computational Harmonic Analysis*, vol. 37, no. 3, pp. 464–491, 2014.

[15] R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert, "Sample complexity of dictionary learning and other matrix factorizations," https://arxiv.org/abs/1312.3790.

[16] R. Gribonval, R. Jenatton, and F. Bach, "Sparse and spurious: dictionary learning with noise and outliers," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6298–6319, 2015.

[17] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 415–422, Portland, Ore, USA, June 2013.

[18] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 543–550, Barcelona, Spain, November 2011.

[19] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Proceedings of the Advances in Neural Information Processing Systems 22 (NIPS '09)*, pp. 1033–1040, 2009.

[20] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 438–445, IEEE, Portland, Ore, USA, June 2013.

[21] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[22] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 1348–1356, Vancouver, Canada, December 2009.

[23] I. Markovsky and S. V. Huffel, "Overview of total least-squares methods," *Signal Processing*, vol. 87, no. 10, pp. 2283–2302, 2007.

[24] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Asilomar Conference on Signals, Systems & Computers*, pp. 40–44, IEEE, Pacific Grove, Calif, USA, November 1993.

[25] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[26] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B. Methodological*, vol. 58, no. 1, pp. 267–288, 1996.

[27] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[28] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, pp. 46–51, IEEE, 2002.

[29] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[30] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, NY, USA, 2010.

[31] Y. Guo, "Robust transfer principal component analysis with rank constraints," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS '13)*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 1151–1159, 2013.

[32] A. Georghiades, *Yale Face Database*, Center for Computational Vision and Control at Yale University, 1997.

[33] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proceedings of the 2011 IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR '11), pp. 1697–1704, IEEE, Colorado Springs, Colo, USA, June 2011.

[34] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 3180357, 11 pages
http://dx.doi.org/10.1155/2016/3180357

Research Article

Objectness Supervised Merging Algorithm for Color Image Segmentation

Haifeng Sima,[1] Aizhong Mi,[1] Zhiheng Wang,[1] and Youfeng Zou[2]

[1] *The School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China*
[2] *Henan Polytechnic University, Jiaozuo, China*

Correspondence should be addressed to Aizhong Mi; miaizhong@hpu.edu.cn

Received 17 June 2016; Accepted 14 September 2016

Academic Editor: Wonjun Kim

Ideal color image segmentation needs both low-level cues and high-level semantic features. This paper proposes a two-hierarchy segmentation model based on merging homogeneous superpixels. First, a region growing strategy is designed for producing homogenous and compact superpixels in different partitions. Total variation smoothing features are adopted in the growing procedure for locating real boundaries. Before merging, we define a combined color-texture histogram feature for superpixels description and, meanwhile, a novel objectness feature is proposed to supervise the region merging procedure for reliable segmentation. Both color-texture histograms and objectness are computed to measure regional similarities between region pairs, and the mixed standard deviation of the union features is exploited to make stop criteria for merging process. Experimental results on the popular benchmark dataset demonstrate the better segmentation performance of the proposed model compared to other well-known segmentation algorithms.

1. Introduction

Color image segmentation is an important task in image analysis and understanding. It is widely used in many image applications as a critical step. The primary purpose of segmentation is to divide an image into meaningful and spatially connected regions (such as object and background) on the basis of diverse properties. Existing segmentation approaches can predominantly be divided into the following four categories: region-based methods, feature-based methods, boundary-based methods, and model-based methods. Most existing image segmentation methods consider only the low-level features [1–4] and it is difficult to obtain ideal segmentation results without high-level visual features. Therefore, many researchers aim to build a bridge between low-level features and visual cognitive behaviors of human. Most segmentation methods integrated high-level knowledge, which are learning cognitive information from underlying features, and they obtain the corresponding label of elements and textons of image. Researchers aim to find high-level summary tags or labels to represent all the underlying features. With all these elements and textons, the segmentation can employ affinity computing to make decision of pixels labels of semantic objects and regions. This procedure employs knowledge-assisted multimedia analysis model to bridge the gap between semantics and low-level visual features [5, 6].

Many segmentation methods based on objects detection have shown their capability by utilizing low-level visual features [7, 8]. They integrated basic visual features to detect homogeneous regions for image segmentation. Even though these low-level features have shown favorable results, they are unbefitting in many complex scenes, such as texture, luminance, and overlapping. To deal with this problem, some researchers attempted to combine different features together [9]. However, most of them integrated these features in a simple way. Few studies evaluated the significance of object features to the overall segmentation. Besides, since most previous segmentation methods lack high-level knowledge about objects, it is difficult for them to discriminate semantic regions from cluttered images.

The objectness is a pixel-level characteristic obtained from training underlying characteristics, so we will combine it with color and texture to achieve better segmentation in this paper. The objectness is proposed by Alexe et al. [10]

as a novel high-level property estimated from multiple low-level features, and it is widely used for salience computing, object detection, cosegmentation, object tracking, and so forth. It is designed for measuring the likelihood of a window containing a complete object in it at first. This measurement employs four cues in the calculation of objectness: multiscale saliency map, color contrast, edges density, and superpixels straddling. The four cues have been proved to be useful for detecting whole object from background. Jiang et al. [11] proposed a strategy for computing regional objectness by computing average objectness values from salient regions in slide windows. Through the above analysis, we find that the objectness is an effective feature to describe the integrity and semantic regions of objects in the image.

In order to tackle those issues remaining in existing segmentation methods, we developed a novel color image segmentation method based on merging superpixels supervised by regional objectness and underlying characteristics in this paper. Firstly, we adopt seeds growing strategy for superpixels computing. The seed pixels are selected from the blurring image from total variation (TV) diffusion image. Next, we introduce the growing method to expand these seed pixels to superpixels according to similarity of color and texture features at local areas. In the merging procedure, we define a combined feature consisting of color, texture, and objectness to measure the similarity between adjacent superpixels and implement merging procedure to form the segmentation results.

The main contributions of this paper are summarised as follows: (1) we develop efficacious superpixels extraction method to detect homogeneous regions with hybrid features, (2) we propose a combined histogram descriptor of color and texture feature for identifying semantic regions, and (3) we design a fast and less sampling objectness computing model and integrate it with low-level features for reasonable merging.

The rest of the paper is organized as follows: our segmentation method is shown in Section 2 including superpixels computing, objectness computing, combined feature definition of superpixels, and merging rules. The experiments on the Berkeley Segmentation Datasets are performed in Section 3 and the conclusion is made in Section 4. The segmentation process and intermediate results are shown in Figure 1.

2. Objectness Supervised Superpixels Merging Segmentation

In our method, the segmentation process consists of the following two stages. First, the input image is segmented into hundreds of uniform homogeneous superpixels with hybrid features including RGB channels and diffusion image. Second, we construct a descriptor for superpixels with invariant color-texture histograms as low-level features and define the objectness of all pixels based on sparse sampling of objectness windows as high-level features. Then, we develop the merging rules combined with two levels of features for reasonable segmentation. The detailed procedure of our method is as follows.

2.1. Compact Superpixels Computing. In the color image segmentation, to obtain homogeneous superpixels is an ideal preprocessing technique to simplify the segmentation task. A superpixel is defined as a connected homogeneous image region often called oversegmentation computed by certain strategies. An image is represented by hundreds of superpixels instead of many redundant pixels which can simplify the image representation and segmentation. There are many segmentation strategies for superpixels computing such as SLIC [12], N-cut [13], watershed [14], and Mean-Shift [15], and various strategies result in different property and quality of superpixels. Each method has its own advantages; however, all these methods did not perform well when encountering texture areas.

For segmentation task, the real boundary keeping and homogeneous superpixels extraction are critical for ideal final segmentation. Texture feature is consisted of group pixels that requires enough pixels to represent the texture patterns in superpixels. How to retain the superpixels boundaries surrounded complete texture is a difficult problem for superpixels computing algorithms. On the other hand, smaller scale superpixel is conducive to saving the real and accuracy boundaries. An important branch of superpixel extraction methods is to predefine segments number before segmentation and it is easy to control the running time and segmentation quality by adjusting segments number in these methods. It is also convenient to integrate multifeatures in these methods. The other data driven superpixels extraction methods are randomness in contrast. So we devolve our method based on the above analysis. We select seeds in the smooth area in the uniform grids of images and grow them to be superpixels with similarity computing. Implementation details are as follows.

2.1.1. Texture Smoothing. In order to enhance the descriptive ability of local regions, we detect texture information by integrating the diffusion feature and original channels into the region growing procedure. Perona and Malik proposed [16] an anisotropic diffusion filtering method for multiscale smoothing and edge detection; this diffusion process is a powerful image processing method which encourages intraregion smoothing in preference to smoothing across the boundaries. This filtering method tries to estimate homogeneous region of image with surface degradation and the edge strengths of local structure. To compute texture features, we adopt nonlinear diffusion to extract texture areas and enhance the edges simultaneously. It is easier to smooth regions and detect real boundary of image by TV diffusion model in texture area [17]. Therefore, the TV flow diffusion method is used for blurring local area of color image for extracting nonoverlapped superpixels. The implicit equation of nonlinear diffusion technique of TV is computed by

$$\frac{\partial I_c^i(x,y,t)}{\partial t} = \operatorname{div}\left(g\left(\sum_{k=1}^{3}\left|\nabla I_k\right|\right)\left|\nabla I^i\right|\right). \tag{1}$$

The initial value of I_c is single channel c of the original image in RGB space, and t is the iteration times. g is the diffusion function defined in [17]

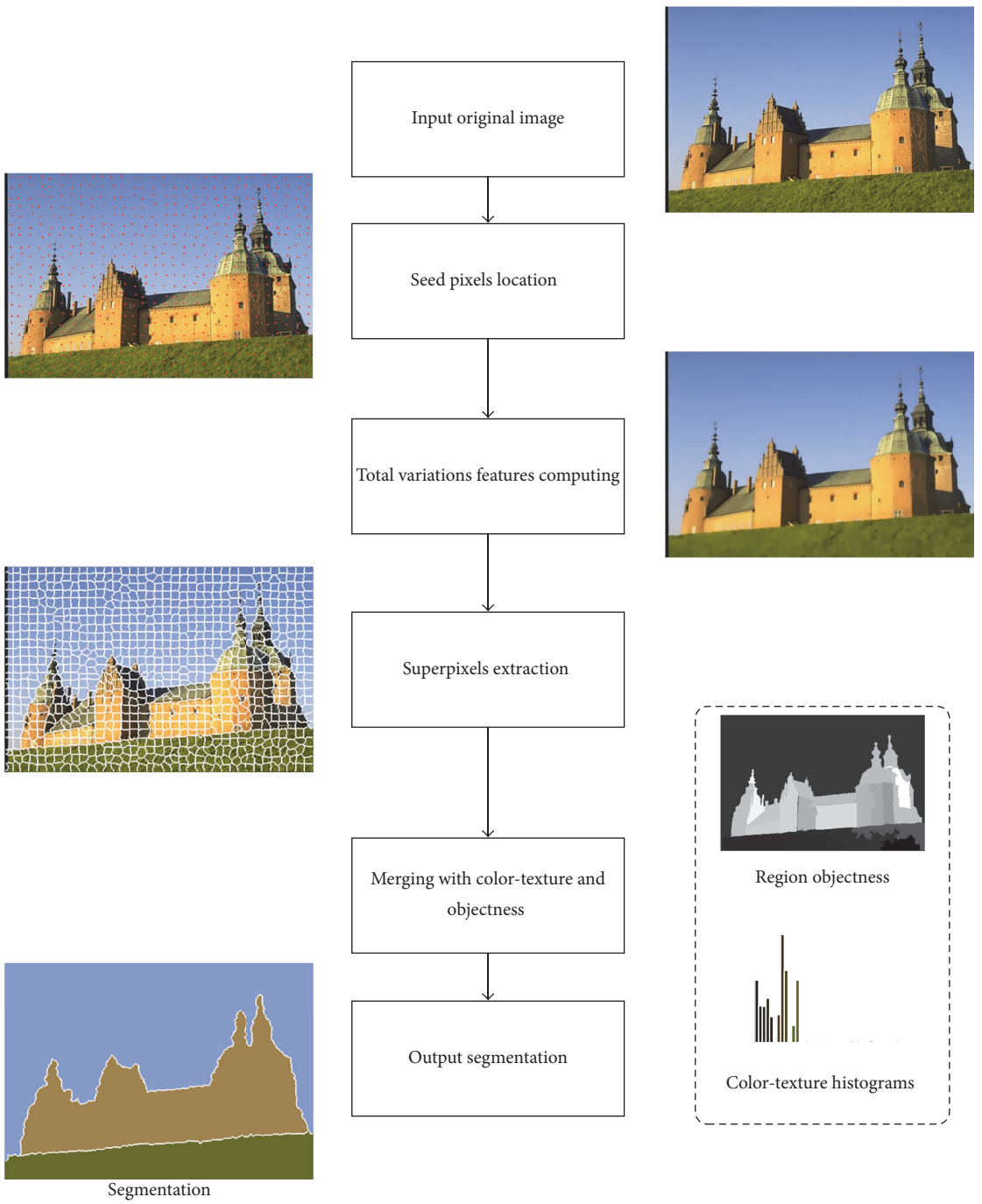


Figure 1: The segmentation process and intermediate results of our method.

$$g\left(\left|\nabla I_k\right|\right) = e^{-\left|\nabla I_k\right|/2\lambda^2}. \quad (2)$$

The iteration times rely on the superpixels scale for proper smoothing of local area. The reasonable iteration times of diffusion are in proportion to the radius of superpixels ≈13 in this paper.

2.1.2. Superpixels Growing Scheme. In this paper, we adopt a seed growing method to obtain compact superpixels with complete homogeneous pixels. The number of superpixels is preset in SLIC [12] and Turbo [18]. Seed pixels selection method has been developed by choosing minimum gradients in divided uniform grids [12], which means to locate seed pixels at the smoothest areas of the grids. It is an effective method to find the optimal seed pixels for superpixels, so we employ the seed pixel extraction strategy and improve it by choosing seeds in the TV diffusion image as the foundation of growing superpixels.

The smoothed image is divided into a number of blocks and located gradient minimum point as the seed pixels. The color image after filtering provides a robust feature

Figure 2: Some superpixels segmentation results of our method.

for computing homogeneous superpixels. We locate the initial seeds at the local minimum in gradient map of smoothed image in average grids at first. The grids setting method is in [12]. In the growing process, both original image and filtered image are integrated into a combined feature $F[R, G, B, R_F, G_F, B_F]$, and the similarity measurement between pixels X and Y, $\mathrm{Sim}(X, Y)$ is calculated by Euclidean distance. The growing procedure is limited to local regions $(1.2r)$ for uniform superpixels, and r is the average radius of grids. Let $s_1, s_2, \ldots, s_i$ denote the initial seeds of the superpixels and A_i denote the growing area corresponding to s_i. The growing algorithm is thus outlined as follows:

(1) Initialize the selected seed pixels as areas A_i, and assign a label p_i for each area A_i.

(2) Compute average F_N of neighbor regions $(1.2r)$ F_N and F_i of A_i; check the neighbor pixels of A_i, and if $\mathrm{Sim}(N[i], F_i) < \mathrm{Sim}(N[i], F_N)$, merge neighbor pixels $N[i]$ to A_i, until no pixels can be labeled.

(3) Connect all scatter regions <20 pixels to adjacent and similar regions.

We have shown some superpixels of test images in Figure 2. Since precise boundary is necessary for successful work, we adopt a standard measure of boundary recall. The average performance of boundaries precision with regard to superpixels numbers on BSD500 is shown in Figure 3; the smallest distance between ground truth and superpixels boundaries is 2 pixels in the experiment. We can see that more superpixels are conducive to more accurate boundaries, and 900 superpixels are big enough for merging segmentation in the next stage.

2.2. Combined Features and Superpixels Merging. Reliable region merging technique is critical for hierarchical segmentation. In this section, the similarity measurement of regions and corresponding stopping criteria are proposed based on the color-texture features and objectness of image. The merging process begins from the primitive superpixels of image until termination criterion is achieved, and the segmentation is finished.

To improve the segmentation accuracy and dependability, we define combined feature CF(His, Obj) to describe the oversegmentation regions. This feature integrates both color-texture information and objectness values. His is a kind of histograms containing color information and local color distribution feature of pixels. As high-level visual information, Obj is the objectness value or probability of a region belonging to an identifiable object.

2.2.1. Color-Texture Histograms. The number of colors that can be discriminated by eyes is limited, no more than 60

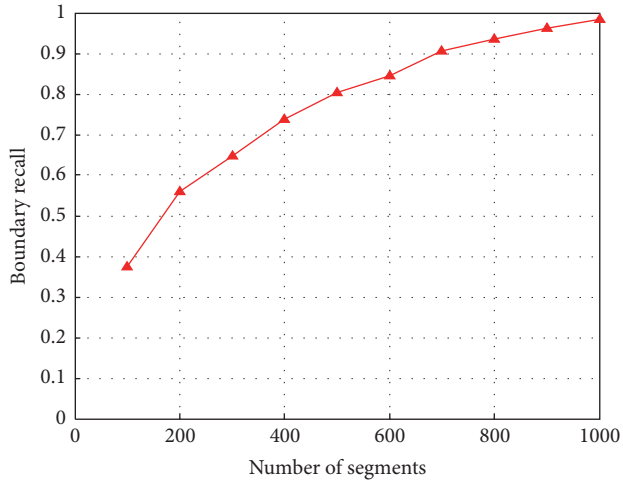


FIGURE 3: Boundary precision with regard to superpixel density.

kinds. Therefore, a quantized image is introduced to compute the color distribution of superpixels for segmentation and an image is quantized to 64 colors by the minimum variance method that leads to least distortion as well as reduces the computing time [19]. After quantization, the connection property of pixels in superpixels of same colors is computed as the texture features of superpixels. We assume Con as the connection relation between pixel C_p and its neighbor pixel Neigh(j), and eight-neighbor pixels Neigh($j = 0$–7) are coded by ($2^{0-7} * \text{Con}_j$) as local color pattern (LCP). If color difference C_d between Neigh(j) and C_p is greater than $C_{d\text{th}}$ (standard deviation of all pixels in quantized image), then $\text{Con}_j = 0$, otherwise $\text{Con}_j = 1$. After coding all pixels, sum up all LCP values of pixels with same color in the superpixel as color distribution weights for corresponding color bins. Thus, the composite color-texture histograms feature (CF) is computed by multiplying the probability of quantitative colors and pixel distribution in the superpixels. The bins of color-texture histograms are defined as follows:

$$\text{Bin}(i) = C_{\text{Bin}} \times C_{\text{LCP}}, \quad i = 1, 2, 3, \ldots, 64. \tag{3}$$

2.2.2. Objectness Based on Sparse Sampling of Regions. As mentioned in [10], the objectness is an attribute of pixels computed with multicues for a set of sample slide windows, and the objectness map is biased towards rectangle intensity response. It seems that objectness has no ability to discriminate regions, but it provides a potential and useful value of all pixels in the rectangle. Jiang et al. [11] proposed a novel method to assign objectness value to every pixel based on the definition of objectness.

Inspired by the objectness computing method proposed by Jiang et al. [11], we propose an improved objectnenss computing method based on sparse sampling of slide windows for all pixels. We also employ four cues for estimation of objectness characteristics of all pixels in an image: the multiscale saliency, color contrast, edge density, and superpixels straddling. The multiscale saliency (MS cues) in the original paper [10] is calculated based on the residual spectrum of image. This method highlights fewer regions of an image, and it is not suitable for segmentation. To expand the global significance of objectness, we use histogram-based contrast (HC) [20] salient map as the MS cues. The histogram-based contrast (HC) method is to define the saliency values for image pixels using color statistics of the input image. In the HC model, a pixel is defined by its color contrast to all other pixels in the image and sped up by sparse histograms. The saliency [20] value of pixel $I_{x,y}$ in image I is defined as

$$S\left(I_{(x,y)}\right) = \sum_{j=1}^{n} f_j D\left(c_{(x,y)}, c_j\right), \tag{4}$$

where $c_{(x,y)}$ is the color value of pixel $I_{(x,y)}$, n is the number of pixel colors after histogram quantization, and f_j is the statistic value of pixel color c_j that appears in image I. In order to reduce the noise caused by the randomness of histogram quantization, the saliency results are smoothed by the average saliency of the nearest $n/4$ neighbor colors.

The color contrast (CC) is a measure of the dissimilarity of a window to its immediate nearby surrounding area and the surroundings; edge density (ED) measures the density of edges near the window borders; superpixels straddling (SS) is a cue to estimate whether a window covers an object. These three cues are specifically described in [10]. With the above-mentioned four cues, the objectness score of test window w is defined by the Naive Bayes posterior probability in [10]

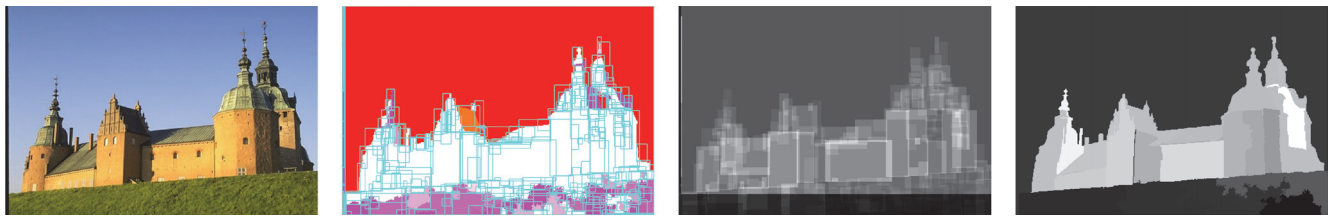
$$P_w\left(\text{obj} \mid \mathscr{C}\right) = \frac{p\left(\mathscr{C} \mid \text{obj}\right) p\left(\text{obj}\right)}{p\left(\mathscr{C}\right)}, \tag{5}$$

where $\mathscr{C}$ is the combined cues set and p(obj) is the priors estimation value of the objectness. All the specific definitions of $p(\mathscr{C}\text{obj})$, p(obj) and $p(\mathscr{C})$ are described in [10].

In the objectness computing methods [10, 11], the sampling windows are selected randomly by scales for calculation of objectness and 100000 windows are sampled for training. Then, they compute the other cues in $\mathscr{C}$ with these windows.

In this paper, we try to sample less windows that most likely contain objects. Thus, presegment results can provide regions for locating object windows with higher probability. We calculate all the circumscribed rectangular windows of the presegmentation regions resulting from Mean-Shift (hs = 10, hr = 10, and minarea = 100) and put all these vertices of circumscribed rectangular windows as the candidates vertex of slide windows. Next, we select all the upper left vertices and lower right corner vertices to compose two sets, respectively: left-up (LU) and right-down (RD).

In order to find the most likely sampling windows including real objects or homogeneous semantic regions, we choose arbitrary vertex V_{LU} from LU to combine with all vertices in RD whose coordinate values are less than the coordinates of V_{LU}. These coupled vertices can provide more reasonable coordinates of sampling windows that contain objects. All the sample windows can be seen in Figure 4. We calculate all the objectness value of sampling windows by (5)



Original image | Mean-Shift+sample windows | Objectness of windows | Objectness

Figure 4: Objectness computing process.

and sum them by pixels locations as pixel-wise objectness with

$$O_p(x,y) = \sum_{p(x,y)\in w}^{w\in W} P(w(x,y)), \tag{6}$$

where W is the sample windows set, w is the single window in W, and p is the score of sample window computed by (5). The pixel-wise objectness cannot represent the similarity of regions, so we compute the average objectness values of Mean-Shift segment regions as new objectness value of all pixels, and the objectness of superpixel is the average value of all pixels in it.

$$\mathrm{Obj}_p(x,y) = \underset{p\in R_i}{\mathrm{AVG}} O_p(x,y). \tag{7}$$

2.2.3. Merging Rules. The regions merging method and stop criteria are crucial for the segmentation results. We established adjacent table and similar matrix for all superpixels to carry out the merging procedure, and the metric of similarity between two regions can be calculated by the following formula:

$$\mathrm{Dis}(X,Y) = \tau * \left(\chi(\mathrm{His}(X), \mathrm{His}(Y))\right) + (1-\tau) * \left|\mathrm{Obj}(X-Y)\right|. \tag{8}$$

The fixed weight assignment is not well adapted to obtain the real object regions by merging process, so we define alterable parameter τ using two types of features to regulate the similarity computing: (1) fixed weight assignment is not well adapted to obtain the real object regions by merging process; (2) when the feature change is greater, the weight should be set smaller in the similarity computing, otherwise it will lead to overmerging. Therefore, we define a balance parameter using two types of features to regulate the similarity computing:

$$\tau = \frac{\Delta(\mathrm{Obj})}{R(\mathrm{Obj})} \times \frac{\Delta(\mathrm{His})}{2\mu + \sigma(\mathrm{His}(\mathrm{Superpixels}))}, \tag{9}$$

where Δ is the difference of given variables, R is the range of Obj, μ is the mean value of His features of superpixels, and σ is the standard deviation of His, Obj of current merged superpixels. Here, we set the threshold based on V to control the merging process. If the similarity between A and B is greater than V, the adjacent merges two regions A and B, updates adjacent relations and labels, and checks next regions until no pair of regions satisfies the merging criterion. V is the combined standard deviation of all combined features (His, Obj) of current merged superpixels, and it is defined in

$$V = \tau * \sigma(\mathrm{His}(\mathrm{Superpixels})) + (1-\tau) * \sigma(\mathrm{Obj}(\mathrm{Superpixels})). \tag{10}$$

The description of the merging strategy is displayed in Algorithm 1. Each time after merging, the combined feature CF(His, Obj) of the new region is updated as follows: firstly, the connection relations of all pixels are unchanged in the new region, then compute the statistics value for each bin of quantitative colors, and update of the histogram feature using formula (3) of the new region. Second, Obj feature of the new region is obtained by computing average objectness value of all the pixels in the new region.

3. Experimental Results

3.1. Dataset. The proposed segmentation model has been tested on Berkeley Segmentation Datasets [21, 22] and compared with human ground truth segmentation results. Specifically, the BSD dataset images were selected from the Corel Image Database in size of 481 × 321. The Berkeley Dataset and Benchmark is widely used by the computer vision for two reasons: (1) it provides a large number of color images with complex scenes and (2), for each image, there is multiple ground truth segmentation which are provided for the numerical evaluation of the testing segmentation algorithms.

3.2. Visual Analysis of Segmentation Results. For a subjective visual comparison, we selected 5 segmentation results of test images and compared them with three classic algorithms (JSEG [23], CTM [24], and SAS [25]). The superpixels number of our method for comparing is set to be $K = 900$; the JSEG [23] algorithm needs three predefined parameters: quantized colors = 10, number of scales = 5, and merging parameter = 0.78; the results for CTM [24] is ($\lambda = 0.1$); the SAS [25] method is implemented with default parameters. From the results shown in Figure 4, it is clear that the proposed method has performed better in detecting complete objects from images (such as flowers, zebra, cheetah, person, and house in the test images). For all the five test images, our method has shown clear and complete contours of the primary targets. The SAS method is more accurate in homogeneous region detection than other methods, but it

Input:
superpixels $G(L)$, adjacent table $\mathrm{Sim}(G)$ of $G(L)$
Output:
The merged result.
(1) For each N node in G, $N(i)$ is neighbor region of N
(2) **if** $\mathrm{Dis}(N, N(i)) < V$, **then**
(3) merged $N(i)$ to N, $L = L - 1$,
(4) change the region label of $N(i)$ to N,
(5) modified the histogram feature of two merged regions
(6) update $G(L)$, $\mathrm{Sim}(G)$ and V;
(7) **end if**
(8) If no regions satisfied $\mathrm{Dis}(N, N(i)) < V$, end the merge process, otherwise, return to (1);
(9) Merge meaningless and small areas to adjacent regions
(10) Return relabeled image.

Algorithm 1

perform less oversegmentation except in complex areas. CTM performs better in texture regions but lost in the completeness of the objects. JSEG is also a texture-oriented algorithm which has the same weakness as CTM. Thus, three other segmentation results have brought more over-segments and meaningless regions from the test results in Figure 5. The region number versus iteration times in the merging process of some test images is shown in Figure 6, and the region number versus parameter V in the merging process of some test images is shown in Figure 7.

In Figure 8, we have displayed fourteen segmentation results from test images. These results include different type of images, such as animals, buildings, person, vehicles, and planes with various texture features. The segmentation results show that the proposed growing-merging framework can segment the images into compact regions and restrain the over- or undersegmentation.

3.3. Quantitative Evaluation on BSD300. The comparison is based on the four quantitative performance measures: probabilistic rand index (PRI) [26], variation of information (VoI) [27], global consistency error (GCE) [21] and Boundary Displacement Error (BDE) [28]. The PRI Index compares the segmented result against a set of ground truth by evaluating the relationships between pairs of pixels as a function of variability in the ground truth set. VoI defines the distance between two types of segmentation as the average conditional entropy of one type of segmentation given the other, and GCE is a region-based evaluation method by computing consistency of segmented regions. BDE measures the average displacement error of boundary pixels between two boundaries images. Many researchers have evaluated these indexes to compare the performance of various methods in image segmentation. The test values of five example images on four algorithms are shown in Table 1, and the average performance on BSD300 [21] are shown in Table 2. With the above comparison strategy, our algorithm provided more approximate boundaries with manual expert segmentation results. The bins number of the combined histograms is critical parameter for segmentation, so we have take some test examples by varying bins in Figure 9 to verify 64 bins is more proper for accurate segmentation.

Table 1: The PRI, VoI, GCE, and BDE value of four test images.

	Index	Proposed	SAS	CTM	JSEG
12084	PRI	0.7848	0.76155	0.7051	0.7026
	VoI	2.4184	2.1261	3.6425	4.3114
	GCE	0.2256	0.1753	0.1658	0.1209
	BDE	7.7028	6.8194	7.4629	9.2061
130066	PRI	0.8209	0.7286	0.7710	0.7457
	VoI	1.4075	2.2986	2.6194	3.6813
	GCE	0.1416	0.3001	0.1117	0.1226
	BDE	9.4618	18.6371	10.2823	13.3104
134052	PRI	0.7630	0.6454	0.6303	0.5833
	VoI	1.2913	2.0665	2.6110	3.9948
	GCE	0.1020	0.2263	0.0833	0.1302
	BDE	6.1061	11.2896	9.4271	15.1976
15062	PRI	0.8882	0.8973	0.8673	0.8705
	VoI	1.5428	1.4820	2.1843	2.6231
	GCE	0.2303	0.1959	0.1386	0.1075
	BDE	8.7102	21.2930	12.2526	13.9926
385028	PRI	0.9152	9.1108	0.9031	0.9036
	VoI	1.5868	3.6813	1.8808	2.2501
	GCE	0.1436	0.2236	0.0888	0.1072
	BDE	9.1108	9.1016	8.6668	8.6987

Table 2: The average values of PRI, VoI, GCE, and BDE for BSD300.

	SAS	JSEG	CTM	Proposed
PRI	0.8246	0.7754	0.7263	0.8377
VOI	1.7724	2.3217	2.1010	1.773
GCE	0.1985	0.2017	0.2033	0.1964
BDE	12.6351	14.4072	11.9245	10.8311

3.4. Boundary Precision Comparison on BSD500. Estrada and Jepson [29] have proposed a thorough-quantitative-evaluation matching strategy to evaluate the segment accuracy for boundary extraction. The algorithms are evaluated using an efficient algorithm for computing precision and recall with regard to human ground truth boundaries. The

Figure 5: Comparison of four segmentation methods (proposed, SAS, CTM, and JSEG).

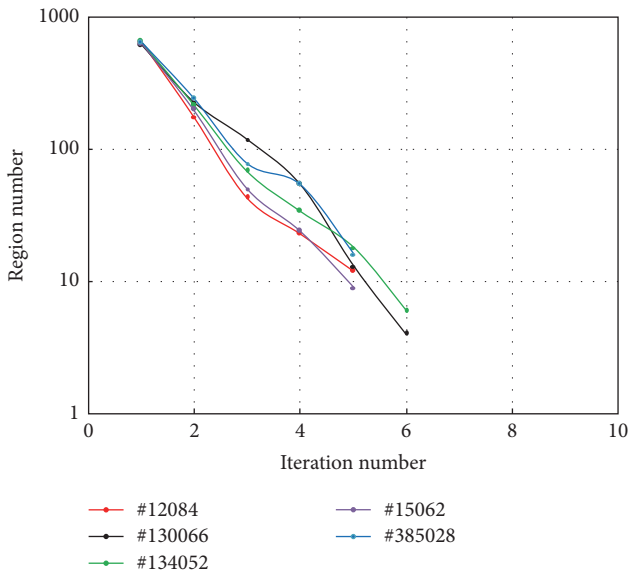


Figure 6: Merging curves of iteration-region numbers.

evaluation strategy of precision and recall is reasonable and equitable as measures of segmentation quality because of not being biased in favor of over- or undersegmentation. An excellent segmentation algorithm should bring low segmented error and high boundary recall. It defines precision and recall to be proportional to the total number of unmatched pixels between two types of segmentation S_{source} and S_{target}, where S_{source} is the boundaries extracted from segmentation results by computer algorithms and S_{target} is the boundaries of human segmentation provided by BSD500 [22]. A boundary pixel is identified as true position when the smallest distance between the extracted and ground truth is less than a threshold ($\varepsilon = 5$ pixels in this paper). The precision, recall, and F-measures are defined as follows:

$$\text{Precision}\left\{S_{\text{source}}, S_{\text{target}}\right\} = \frac{\text{Matched}\left(S_{\text{source}}, S_{\text{target}}\right)}{\left|S_{\text{source}}\right|}.$$
$$\text{Precision}\left\{S_{\text{source}}, S_{\text{target}}\right\} = \frac{\text{Matched}\left(S_{\text{source}}, S_{\text{target}}\right)}{\left|S_{\text{target}}\right|} \quad (11)$$

The precision and recall measures provide a more reliable evaluation strategy based on two reasons: (1) it refined boundaries and avoided duplicate comparison of identical boundary pixels, and (2) it provides dynamic displacement parameters for comprehensive and flexible evaluation of segmentation performance. In this section, we displayed the

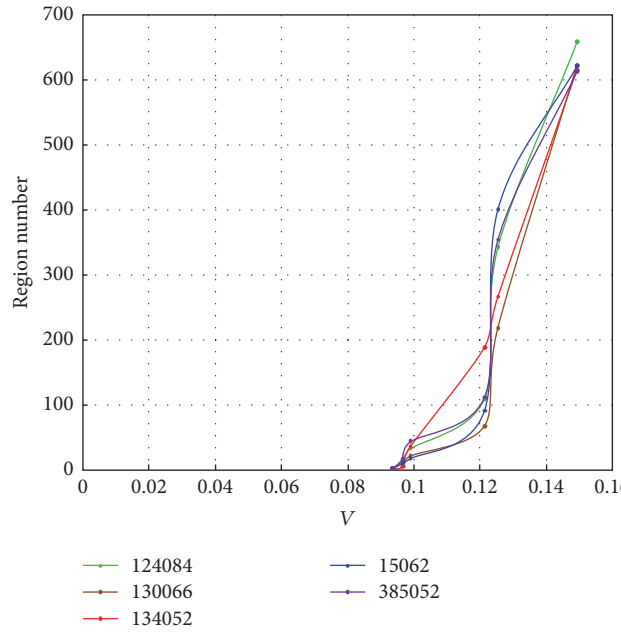


FIGURE 7: Merging curve of region number-V of test images in Figure 3.

FIGURE 8: Some segmentation results and objectness map.

boundaries precision and recall curves with five segmentation algorithms (MS [15], FH [2], JSEG [23], CTM [24], and SAS [25]) in Figure 10. The ROC curves fully characterize the performance in a direct comparison of the segmentations quality provided by different algorithms. Figure 10 shows that our method provides more closely boundaries of segmentations with the ground truth.

4. Conclusion

In this paper, we propose a novel hierarchical color image segmentation model based on region growing and merging. A proper growing procedure is designed to extract compact and complete superpixels with TV diffusion filtered image at first. Second, we integrate color texture and objectness information of superpixels into the merging procedure and

Figure 9: Some segmentation results of varying bins.

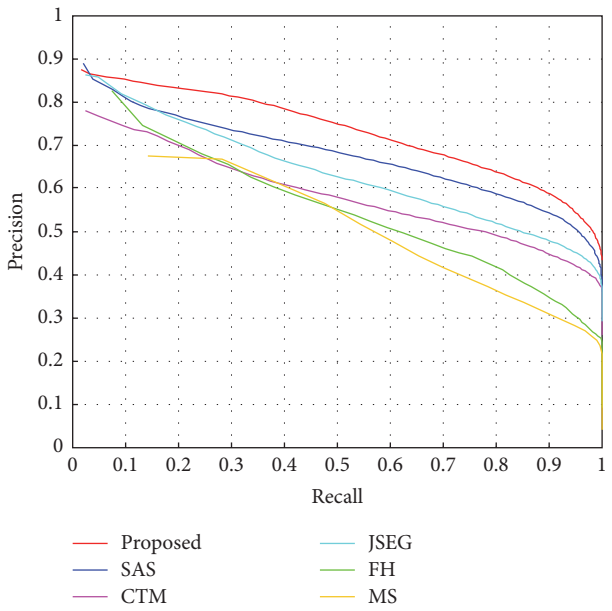


Figure 10: Boundaries of ROC curves on BSD500 when $\varepsilon = 5$.

develop effective merging rules for ideal segmentation. In addition, using a histogram-based color-texture distance metric, the merging strategy can obtain complete objects with smooth boundaries and locate the boundaries accurately. The subjective and objective experimental results for nature color images show a good segmentation performance and demonstrate a better discriminate ability with objectness feature for complex images of the proposed method. In the future, we will focus on the extension of objectness to different kinds of segmentation strategies.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

Haifeng Sima acknowledges the support of the Key Projects of National Natural Foundation, China (U1261206), National Natural Science Foundation of China (61572173 and 61602157), Science and Technology Planning Project of Henan Province, China (162102210062), the Key Scientific Research Funds of Henan Provincial Education Department for Higher School (15A520072), and Doctoral Foundation of Henan Polytechnic University (B2016-37).

References

[1] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1454–1466, 2001.

[2] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[3] M. Mignotte, "Segmentation by fusion of histogram-based K-means clusters indifferent colour spaces," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 780–787, 2008.

[4] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, 2010.

[5] S. Gould and X. He, "Scene understanding by labeling pixels," *Communications of the ACM*, vol. 57, no. 11, pp. 68–77, 2014.

[6] T. Athanasiadis, P. Mylonas, Y. Avrithis, and S. Kollias, "Semantic image segmentation and object labeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 298–311, 2007.

[7] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2217–2224, IEEE, Providence, RI, USA, June 2011.

[8] 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.

[9] W. Tao, Y. Zhou, L. Liu, K. Li, K. Sun, and Z. Zhang, "Spatial adjacent bag of features with multiple superpixels for object segmentation and classification," *Information Sciences*, vol. 281, pp. 373–385, 2014.

[10] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

[11] 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, Sydney, Australia, December 2013.

[12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[14] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[15] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[16] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[17] M. Rousson, T. Brox, and R. Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 699–704, Madison, Wis, USA, June 2003.

[18] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[19] J.-P. Braquelaire and L. Brun, "Comparison and optimization of methods of color image quantization," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 1048–1052, 1997.

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

[21] 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 IEEE International Conference on Computer Vision*, pp. 416–423, Vancouver, Canada, July 2001.

[22] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html.

[23] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[24] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[25] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: a bipartite graph partitioning approach," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 789–796, Providence, RI, USA, June 2012.

[26] C. Pantofaru and M. Hebert, "A comparison of image segmentation algorithms," Tech. Rep. CMU-RI-TR-05-40, Carnegie Mellon University, 2005.

[27] M. Meila, "Comparing clusterings: an axiomatic view," in *Proceedings of the 22nd ACM International Conference on Machine Learning (ICML '05)*, pp. 577–584, New York, NY, USA, 2005.

[28] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cuff, "Yet another survey on image segmentation," in *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III*, vol. 2352 of *Lecture Notes in Computer Science*, pp. 408–422, Springer, Berlin, Germany, 2002.

[29] F. J. Estrada and A. D. Jepson, "Benchmarking image segmentation algorithms," *International Journal of Computer Vision*, vol. 85, no. 2, pp. 167–181, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 7638985, 21 pages
http://dx.doi.org/10.1155/2016/7638985

Research Article

Visibility Video Detection with Dark Channel Prior on Highway

Jiandong Zhao,[1] Mingmin Han,[1] Changcheng Li,[2] and Xin Xin[2]

[1]*School of Mechanical and Electronic Control Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2]*Key Laboratory of Road Safety Technologies, Ministry of Transport, Beijing 100044, China*

Correspondence should be addressed to Jiandong Zhao; zhaojd@bjtu.edu.cn

Received 16 April 2016; Revised 16 June 2016; Accepted 11 July 2016

Academic Editor: Wonjun Kim

Dark channel prior (DCP) has advantages in image enhancement and image haze removal and is explored to detect highway visibility according to the physical relationship between transmittance and extinction coefficient. However, there are three major error sources in calculating transmittance. The first is that sky regions do not satisfy the assumptions of DCP algorithm. So the optimization algorithms combined with region growing and coefficient correction method are proposed. When extracting atmospheric brightness, different values lead to the second error. Therefore, according to different visibility conditions, a multimode classification method is designed. Image blocky effect causes the third error. Then guided image filtering is introduced to obtain accurate transmittance of each pixel of image. Next, according to the definition meteorological optical visual range and the relationship between transmittance and extinction coefficient of Lambert-Beer's Law, accurate visibility value can be calculated. A comparative experimental system including visibility detector and video camera was set up to verify the accuracy of these optimization algorithms. Finally, a large number of highway section videos were selected to test the validity of DCP method in different models. The results indicate that these detection visibility methods are feasible and reliable for the smooth operation of highways.

1. Introduction

According to the road traffic accidents annual reports of China from 2007 to 2014, more than 50 percent of accidents and deaths are caused by low visibility which is below 200 meters [1]. Particularly on highway, severe congestion and major accident can easily occur in low visibility conditions. Therefore, timely and accurate detection of low visibility and early warning response are important to ensure the smoothness of highway system operation.

In the present of expensive test equipment, one often can be used to obtain accurate visibility value nowadays. However, on Chinese highway, the displacement distance between these devices is very sparse, about 30 km, which results in limited detection range and failing to meet the demand in reality. In recent years, with the development of video surveillance technologies, using image processing methods to detect low visibility has attracted a vast amount of attention and obtained numerous extensive achievements. By reviewing other literatures, current video visibility detection algorithms contain four main fields: template matching, camera model calibration, dual differential luminance, and DCP method.

For template matching method, Robert et al. [2, 3] comparatively analyzed the detected image and the images of which the visibility had been known in advance and then obtained the relative visibility. Bäumer et al. [4] used time-series images and detected the farthest object to estimate meteorological visibility. Bronte et al. [5] defined three kinds of weather patterns including mist, fog, and heavy fog; moreover, they used camera projection method to estimate visibility based on the degree of fog.

For camera calibration model based on the contrast method, Steffens [6] photographed black object image and calculated visibility value according to the relative brightness ratio of object and its background. According to the contrast values of multiple targets in an image, Taek [7] fitted the nonlinear visibility curve and got daytime visibility. Babari et al. [8, 9] presented a mapping model between image contrast and atmospheric visibility to estimate visibility base on the assumption that objects in the scene distributed continuously with the changing of distance. By selecting a virtual target

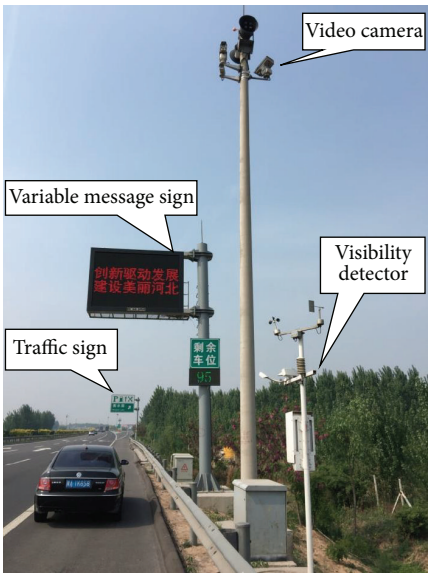
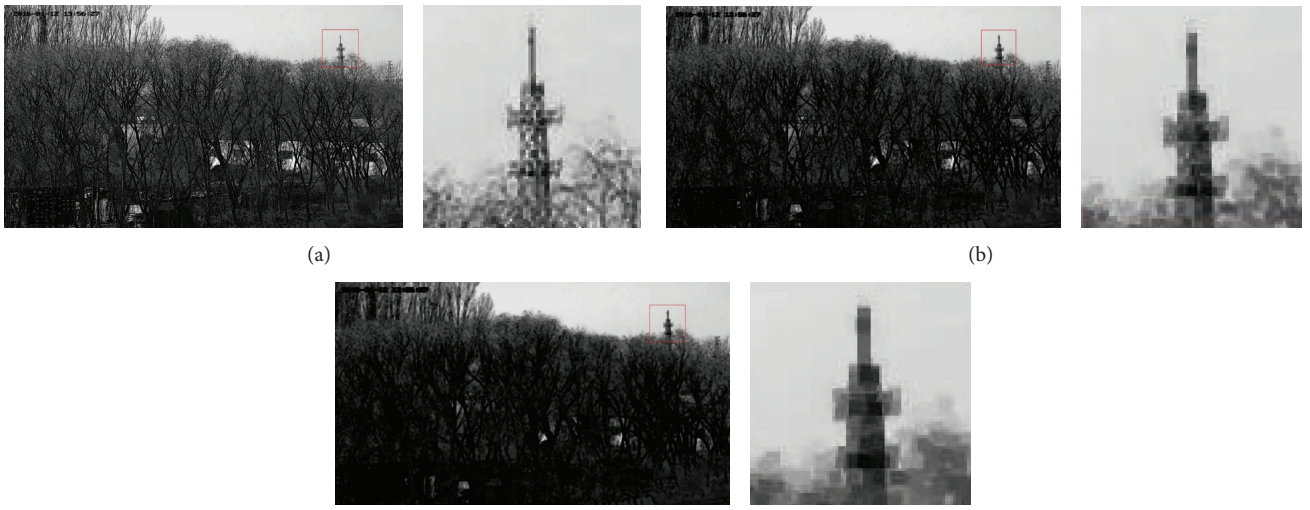


Figure 1: Traffic engineer installations on highway.

(a) (b)

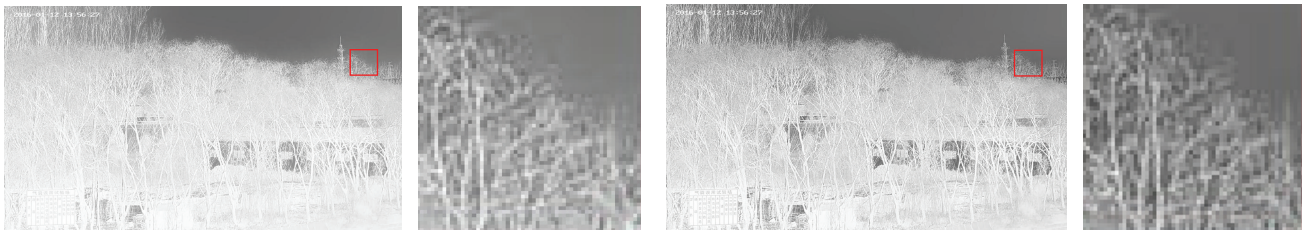
Figure 2: Results comparison after region growing: (a) original image; (b) segmentation result after region growing.



(a) (b)

(c)

Figure 3: Blocks effect of different patch size. (a) Using 3 × 3 patches. (b) Using 6 × 6 patches. (c) Using 9 × 9 patches.



(a) (b)

Figure 4: Comparison result of estimated transmittance. (a) An estimated transmittance maps before optimization. (b) An estimated transmittance map after optimization.

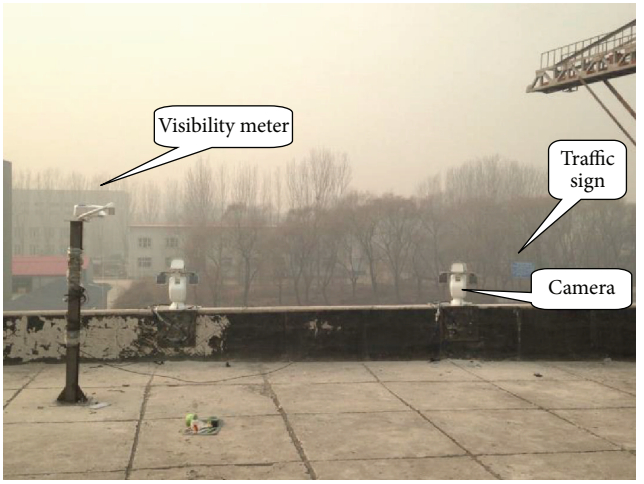
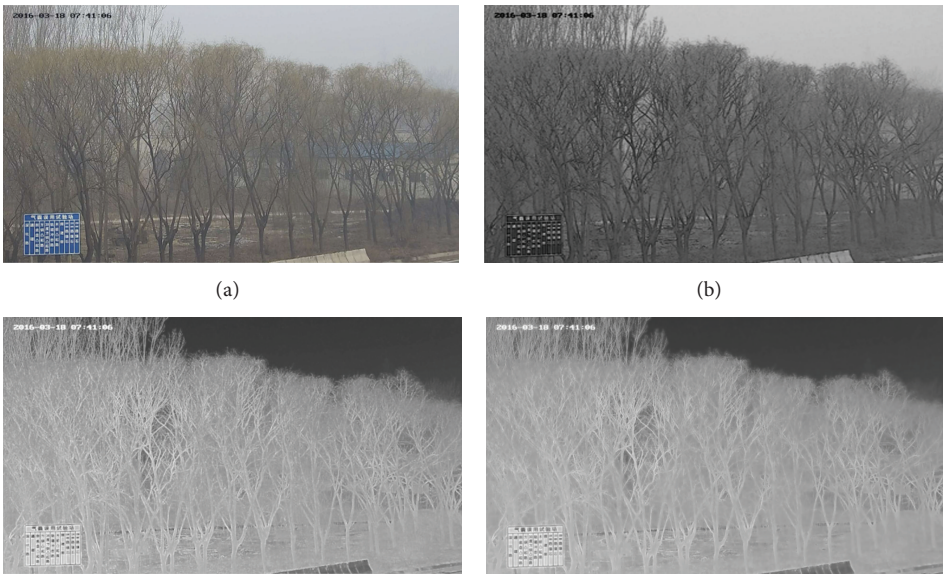


Figure 5: The comparative test system for visibility detection.



(a) (b)

(c) (d)

Figure 6: Daytime visibility detection based on comparative test system. (a) The test system image. (b) Dark channel image of test system. (c) Estimated transmittance maps of test system before optimization. (d) Estimated transmittance maps of test system after optimization.

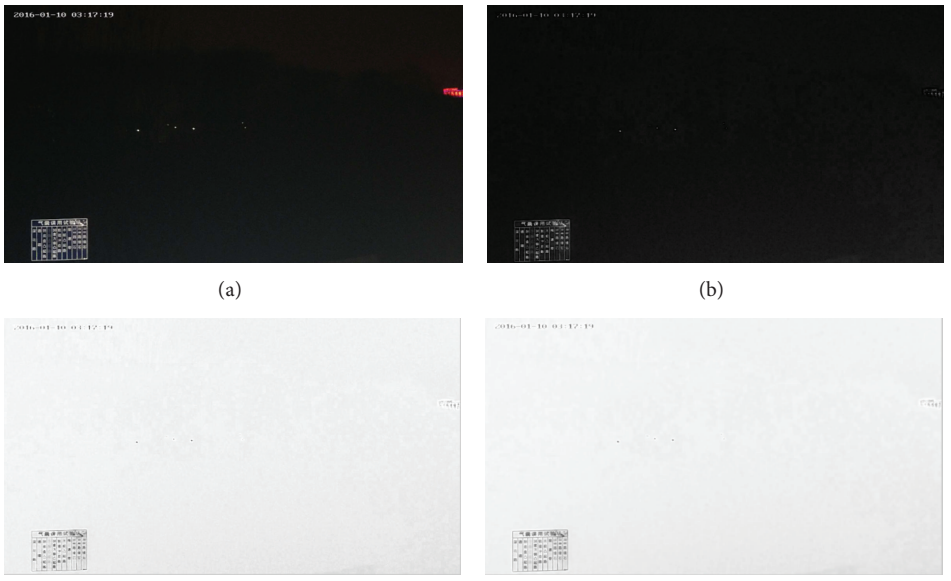


Figure 7: Nighttime visibility detection based on comparative test system. (a) The test system image. (b) Dark channel image of test system. (c) Estimated transmittance maps of test system before optimization. (d) Estimated transmittance maps of test system after optimization.

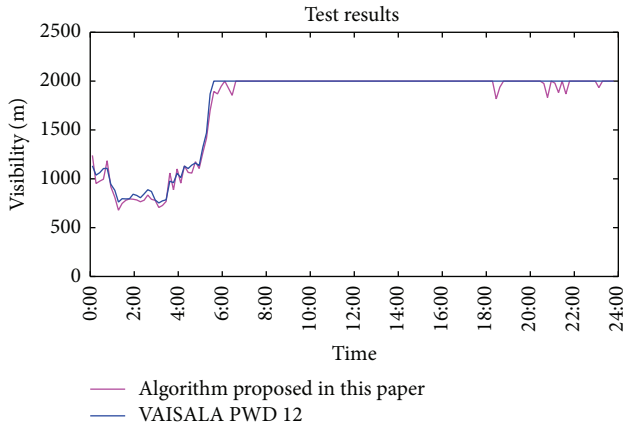


Figure 8: Visibility comparison results of the experimental system.

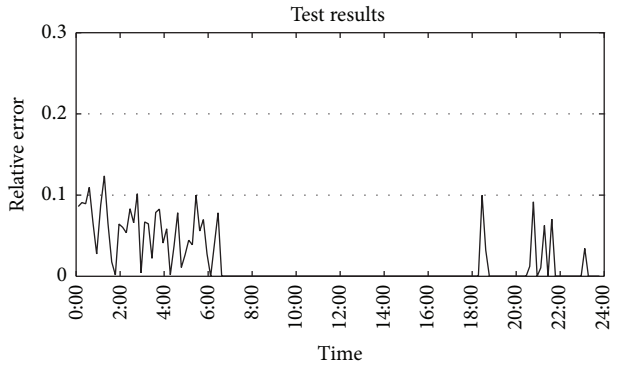
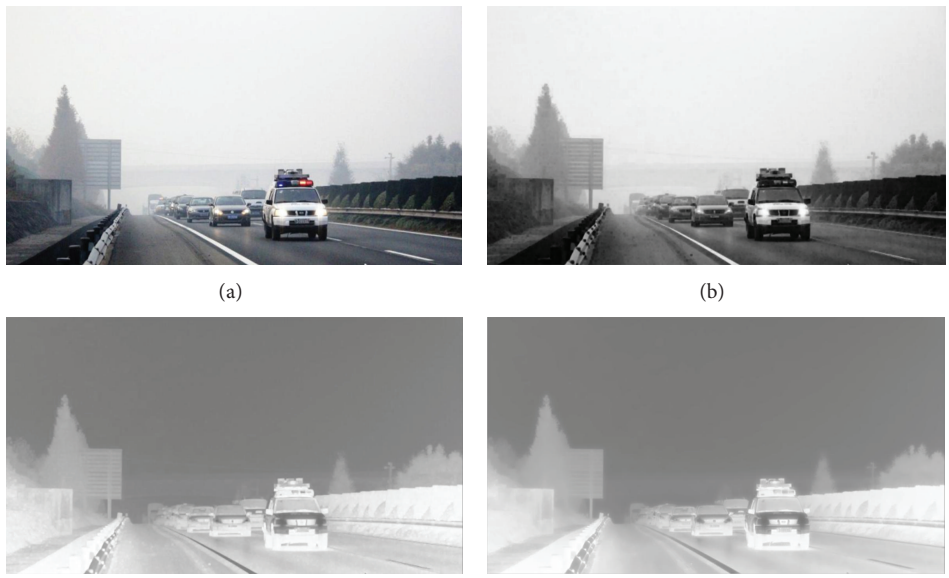
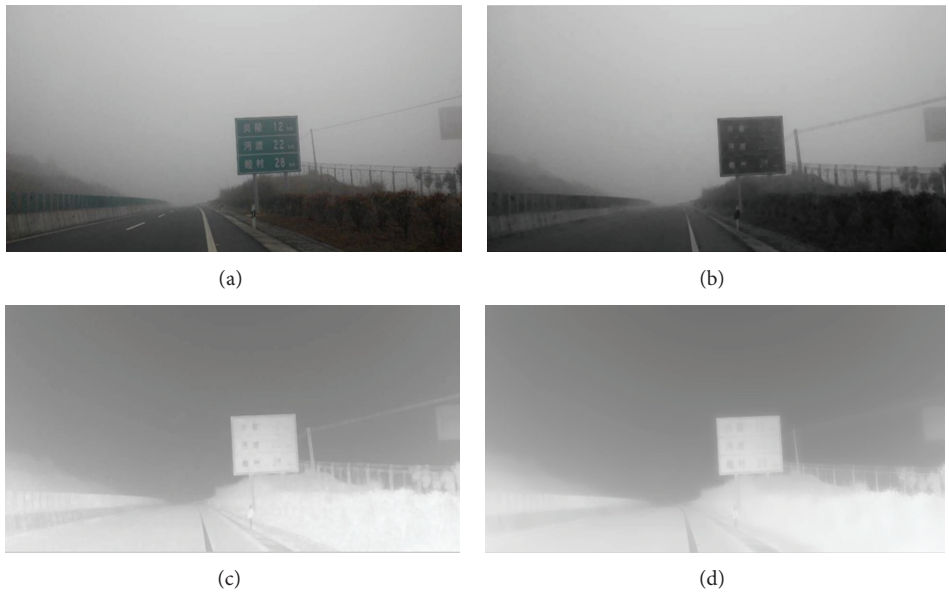


Figure 9: Relative error of the experimental system.



(a) (b) (c) (d)

Figure 10: Verification results under the first model. (a) The original image. (b) Dark channel image of the first model. (c) Estimated transmittance maps of the first model before optimization. (d) Estimated transmittance maps of the first model after optimization.

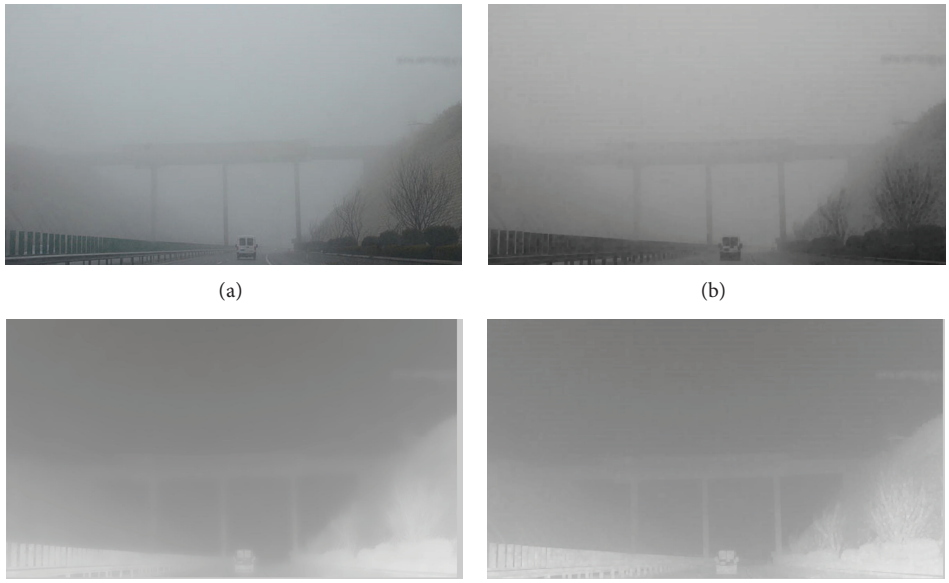


(a) (b) (c) (d)

Figure 11: Verification results under the second model. (a) The original image. (b) Dark channel image of the second model. (c) Estimated transmittance maps of the second model before optimization. (d) Estimated transmittance maps of the second model after optimization.

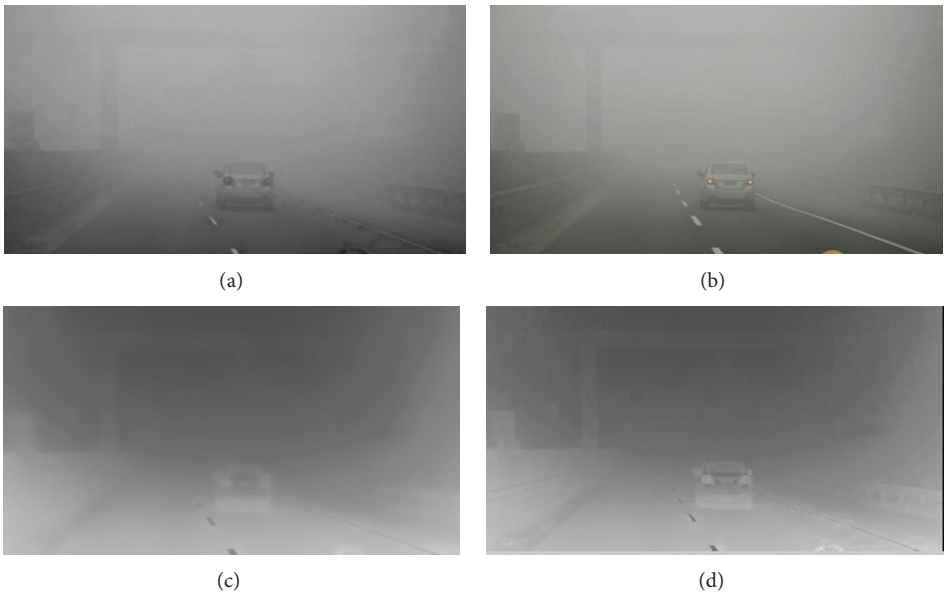
area in an image marked with distance information on the road, Chen et al. [10–12] obtained the corresponding function of spatial brightness contrast and visibility by using image matching and edge feature extraction. By extracting image feature parameters which could reflect visibility changes, An et al. [13] used the least squares method and inverse transform to find a fitting function between image features and distance to determine visibility value.

For camera model calibration method based on vanishing point, by using sky and road as references, Hautiére et al. [14] determined atmospheric extinction coefficient with camera model calibration and real-time graphics processing program and calculated atmospheric visibility through the Koschmieder's law. Li et al. [15] figured out the interested region by detecting lane boundaries and analyzed contrast values of each pixel and its neighbors. When the maximum contrast value exceeded the preset threshold, it would be considered as the distinguishable pixel by human operator. The maximum distinguishable pixel was called vanishing point; the distance between this point and camera was regarded as



(a) (b) (c) (d)

Figure 12: Verification results under the third model. (a) The original image. (b) Dark channel image of the third model. (c) Estimated transmittance maps of the third model before optimization. (d) Estimated transmittance maps of the third model after optimization.

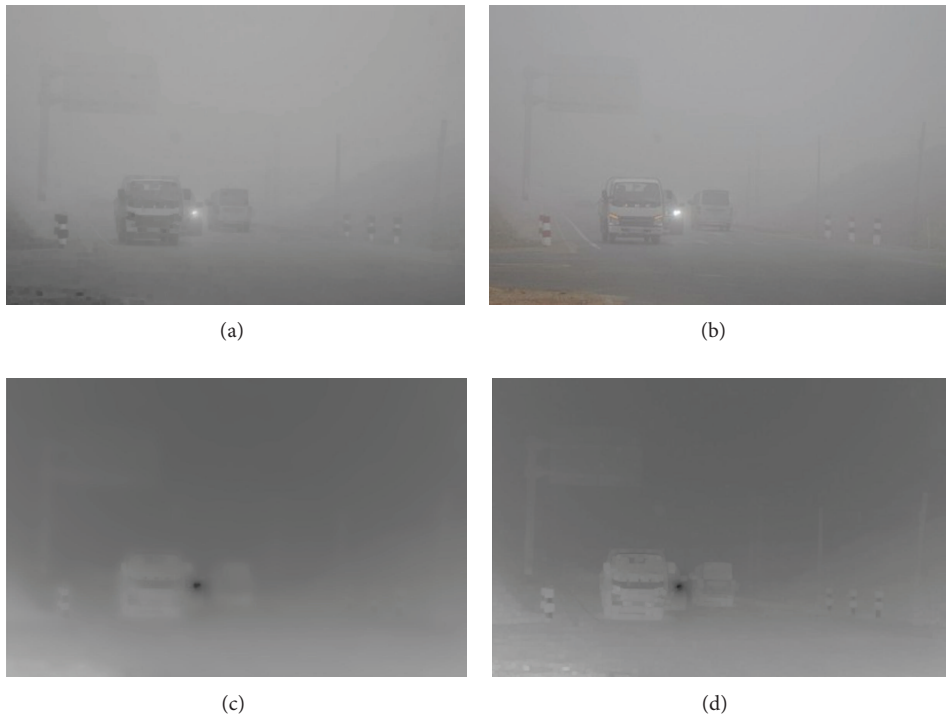


(a) (b) (c) (d)

Figure 13: Verification results under the fourth model. (a) The original image. (b) Dark channel image of the fourth model. (c) Estimated transmittance maps of the fourth model before optimization. (d) Estimated transmittance maps of the fourth model after optimization.

the current visibility. By collecting the distinguishable pixels by human operator when the contrast value is greater than the threshold of luminance contrast, Li and Zhou [16] established a camera parameters model to calculate the distances between these points and camera. The maximum distance is visibility value. Based on region growing algorithm, Zhang et al. [17] figured out the feature points in precise road region. The contrast curve of these points could reflect road luminance variation. Moreover, they found the maximum distinguishable pixels to calculate the maximum visibility distance.

For dual differential luminance algorithm, by setting two targets with different distances from observation station, Lv et al. [18] deduced daytime visibility according to the ratio of two measured luminance differences between targets and their horizon sky background. Chang et al. [19] calculated daylight visibility by dual differential luminance algorithm and gained night time visibility with double light sources. These algorithms could solve the problem that precision was affected by uneven distribution of sky brightness. Zhao et al. [20] proposed optimization dual brightness difference



(a) (b)

(c) (d)

Figure 14: Verification results under the fifth model. (a) The original image. (b) Dark channel image of the fifth model. (c) Estimated transmittance maps of the fifth model before optimization. (d) Estimated transmittance maps of the fifth model after optimization.

algorithm with correction coefficient and multimode division method, and she developed a special visibility detection system.

Based on statistical analysis of outdoor fog-free images, the concept of DCP was put forward by He et al. [21]. In the vast majority of local area of outdoor fog-free images, at least one color channel intensity value of some pixels is very low. When their minimum intensity values are close to zero, these pixels are known as the dark pixels. The DCP algorithm is widely used in image enhancement and image defogging for calculating the transmittance [22]. Guo et al. [23] obtained transmittance from building to camera by using DCP and guided filter. And then, they calculated extinction coefficient based on Lambert-Beer law and got visibility according to the relationship between transmittance and extinction coefficient.

Comparative analysis of the advantages and disadvantages of the method described above is as follows:

(1) The principle of template matching method is simple. However, this method requires a lot of images calibrated with visibility information as a matching database and is not suitable for practical applications.

(2) Using camera model calibration method, one can directly consider road as object to determine the extinction coefficient through camera model calibration and real-time graphics processing procedures. Then, the atmospheric visibility can be calculated with Koschmieder's law. The advantages of this method are low cost, high stability, and high precision for moving vehicles and fixed cameras. However, self-calibration model is complicated and easily disturbed by weather and environment. Also, this method is difficult to work at night and cannot meet the demand for all-weather real-time detection.

(3) Dual brightness difference method has great advantages in night visibility detection and meets the demand for all-weather visibility detection with better reliability, but the detection conditions of this method are high. Therefore, the additional artificial target should be set up. So it is not insufficient for highway application.

(4) DCP method combines the advantages of digital imaging method and the transmittance method. The distinguishing features are simple principle and low cost. However it is still in theoretical stage and requires higher accuracy during the calculation of transmittance.

On the majority of Chinese highways, a video surveillance camera is implemented every 2 Km along the side of the shoulder. These devices are used to monitor traffic flow and traffic environment and can provide a wealth of video resources for the development of video detection of low visibility. Taking into account the advantages of DCP method, we attempt to apply this method to detect highway visibility. According to assumptions that there are some dark pixels in local area of image, DCP calculates the transmittance. However, when this method is applied directly to the highway

Figure 15: Image sequences of the tunnel on 01/23/2014.

Figure 16: Image sequences of the tunnel on 01/31/2014.

visibility detection, large errors are produced due to the complexity of real highway condition. Therefore, an optimized DCP algorithm is presented which combines region growing, correction coefficient and multimode detection method to obtain more accurate transmittance.

2. Visibility Detection Principle of Dark Channel Prior Method

The digital image is defined as follows in computer vision and digital picture processing:

Figure 17: Image sequences of the tunnel on 02/03/2014.

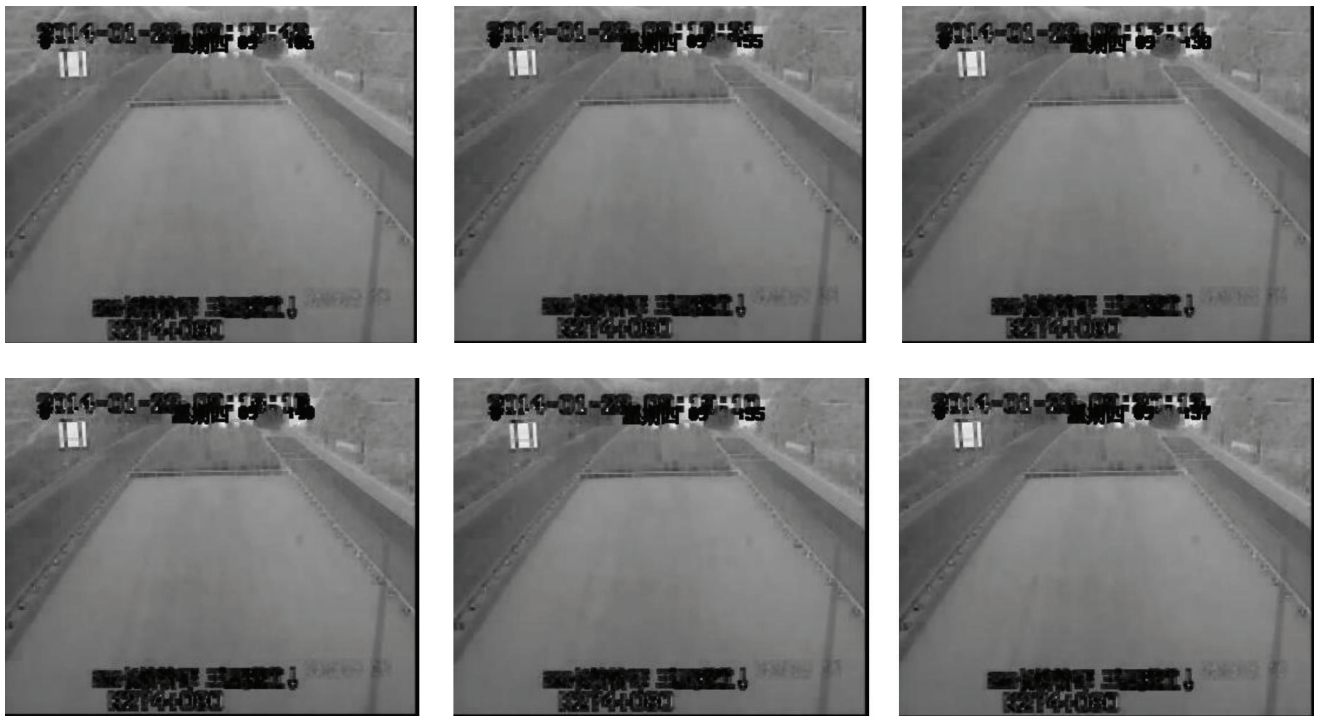


Figure 18: Dark channel images of the tunnel on 01/23/2014.

$$I(x) = J(x)\,t(x) + A\,(1 - t(x)), \tag{1}$$

where I is image intensity, J is actual scene intensity, A is atmosphere light intensity, x is the pixel of digital image, and t is transmittance describing the proportion of the light that is reflected from an object and transmitted directly to camera without scattering.

2.1. Theoretical Basis of Dark Channel Prior. According to the literature [23], the corresponding dark channel $J^{\text{dark}}(x)$ of

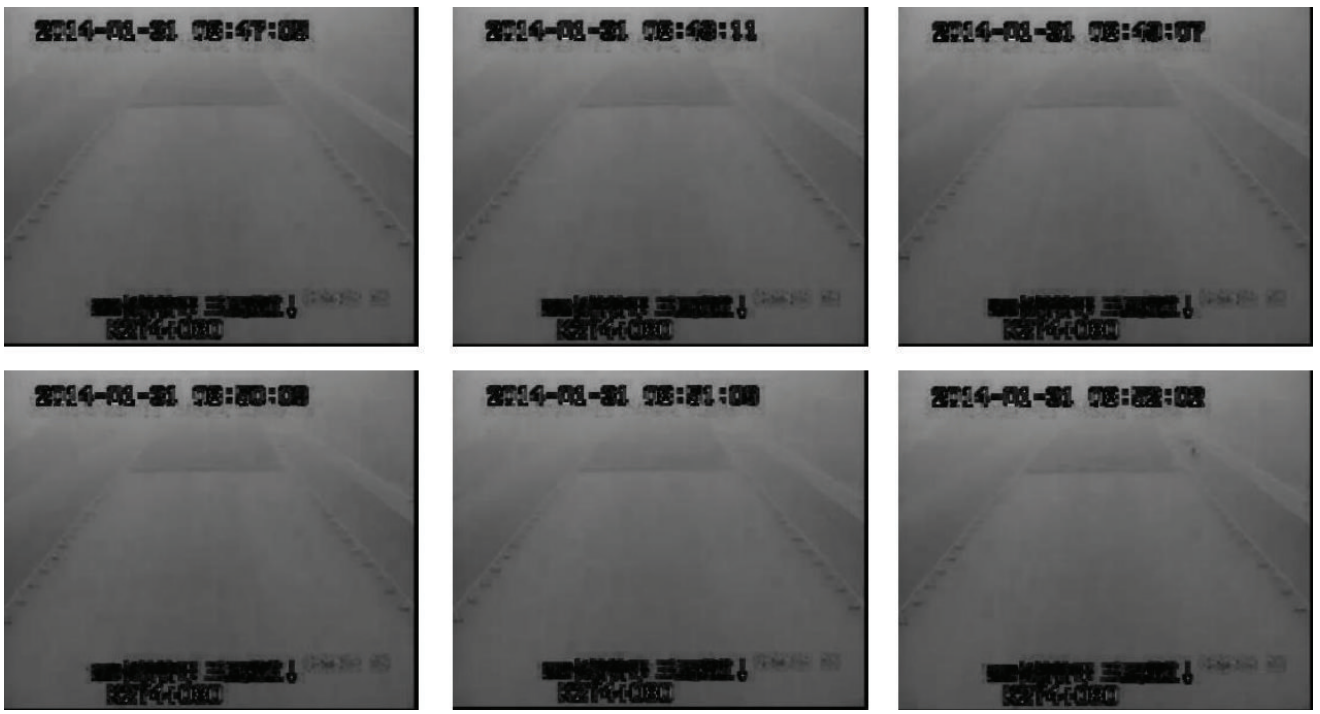


Figure 19: Dark channel images of the tunnel on 01/31/2014.

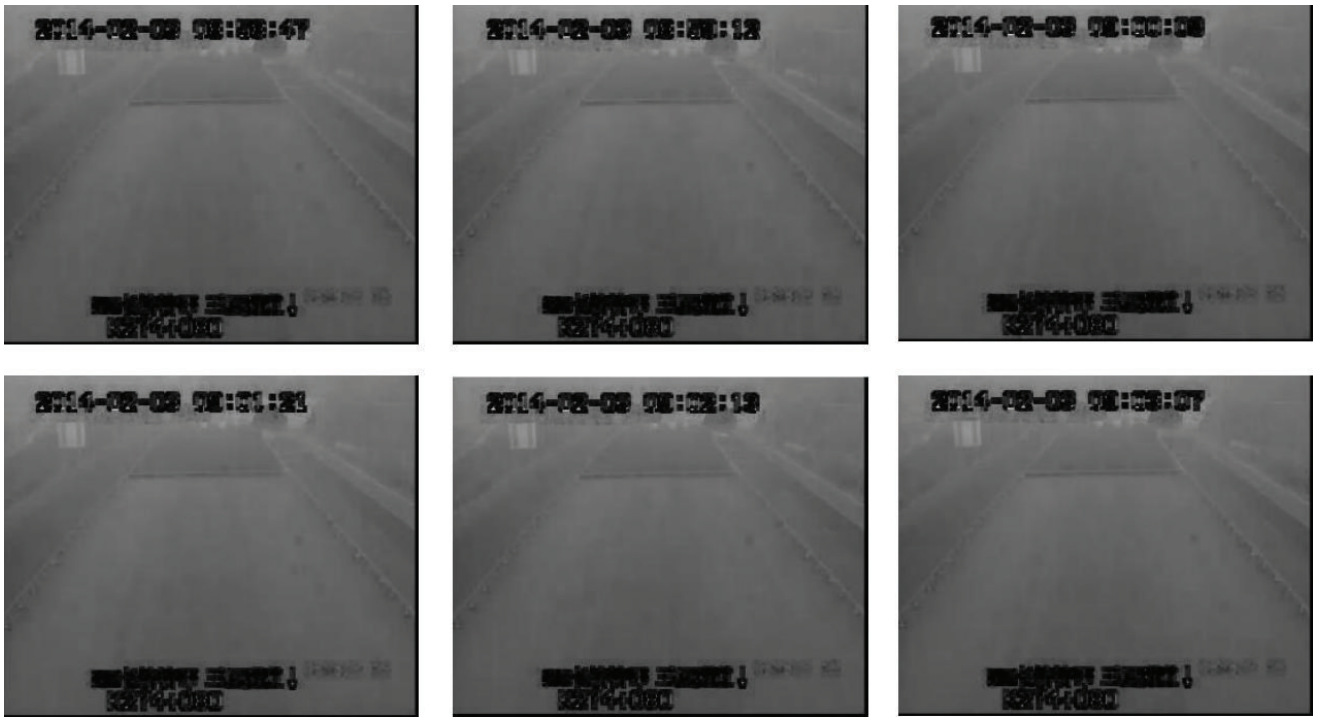


Figure 20: Dark channel images of the tunnel on 02/03/2014.

dark pixel x is given in

$$J^{\text{dark}}(x) = \min_{c\in\{r,g,b\}} \left(\min_{y\in\Omega(x)} \left(J^c(y) \right) \right) \longrightarrow 0, \tag{2}$$

where c is an arbitrary channel in RGB color space, $J^c(y)$ is pixel value of channel c at pixel y, and $\Omega(x)$ is a local patch of which center pixel is x.

Based on DCP, the transmittance of each pixel can be calculated. Main steps are shown as follows.

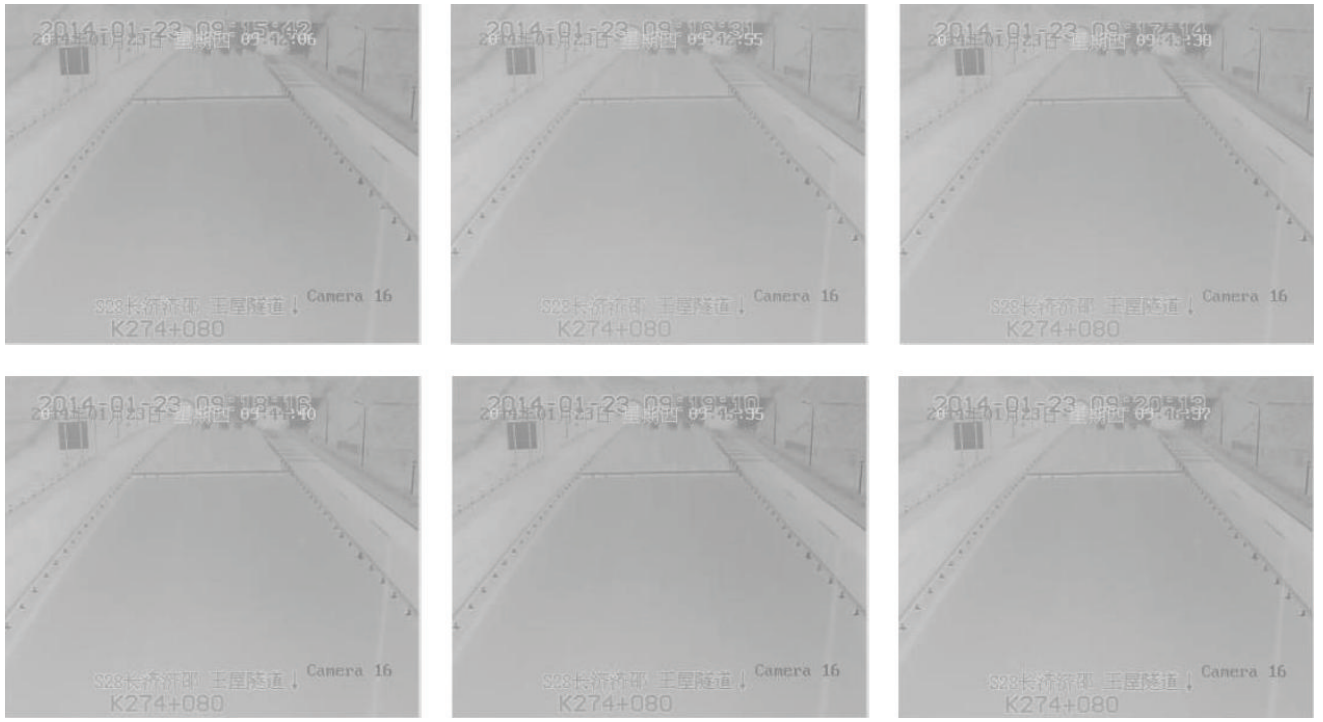


Figure 21: Estimated transmittance maps of the tunnel before optimization on 01/23/2014.

Step 1. For the channel c, assuming that the transmittance of each pixel in the patch $\Omega(x)$ is the same as a constant $\widetilde{t}(x)$, the minimum value of pitch $\Omega(x)$ is taken as formula

$$\min_{y\in\Omega(x)}\left(I^c(y)\right) = \min_{y\in\Omega(x)}\left(J^c(y)\right)\widetilde{t}(x) + A^c\left(1-\widetilde{t}(x)\right). \quad (3)$$

Step 2. Calculate the minimum value of three channels:

$$\min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} I^c(y)\right) = \min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} J^c(y)\right)\widetilde{t}(x) + A^c\left(1-\widetilde{t}(x)\right). \quad (4)$$

Step 3. Without considering DCP, based on the atmosphere light regularity of A^c, $\widetilde{t}(x)$ can be described as formula

$$\widetilde{t}(x) = \frac{1-\min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} I^c(y)/A^c\right)}{1-\min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} J^c(y)/A^c\right)}. \quad (5)$$

Step 4. According to DCP, formula (4) can be presented as follows:

$$\min_{c\in\{r,g,b\}}\left(\frac{\min_{y\in\Omega(x)} I^c(y)}{A^c}\right) = 1-\widetilde{t}(x). \quad (6)$$

Step 5. The estimated transmittance $\widetilde{t}(x)$ of each pixel in an image is obtained as formula

$$\widetilde{t}(x) = 1-\min_{c\in\{r,g,b\}}\left(\frac{\min_{y\in\Omega(x)} I^c(y)}{A^c}\right). \quad (7)$$

2.2. Visibility Detection Principle with Dark Channel Prior Algorithm. The essence of DCP algorithm is to calculate the transmittance of image pixel, and based on the physical relationship between transmittance and extinction coefficient, the latter can be obtained; therefore, the visibility value is carried out easily.

Bouguer-Lambert law describes the relationship between incident light flux Φ_0 and transmittance light flux Φ_τ, under the condition of a beam of light source transmits from a certain distance d. Thus, mathematical relationship between atmospheric transmittance and meteorological optical visual range can be deduced. This law provides a theoretical basis for visibility detection with the method of DCP.

Bouguer-Lambert law is expressed as follows:

$$\Phi_\tau = \Phi_0\tau = \Phi_0 e^{-\sigma d}, \quad (8)$$

where τ is the transmittance with light propagation distance d and σ is the atmospheric extinction coefficient.

The transmittance can be derived as follows:

$$\tau = e^{-\sigma d}. \quad (9)$$

According to the definition of τ in meteorological optical range, select $\tau = 0.05$ for Bouguer-Lambert law; there is

$$0.05 = e^{-\sigma V}. \quad (10)$$

Thus, the visibility V can be obtained:

$$V = \frac{1}{\sigma}\ln\frac{1}{0.05} = \frac{2.996}{\sigma} = -\frac{2.996d}{\ln(\tau)}. \quad (11)$$

It is worth noting that we should choose the objects which contain dark channel when using DCP to detect

Figure 22: Estimated transmittance maps of the tunnel before optimization on 01/31/2014.

visibility. Then the extinction coefficient can be deduced based on the accurate transmittance of this region. When the transmittance is 0.05, we can calculate the meteorological optical range, namely, the atmospheric visibility.

As shown in Figure 1, video camera, variable message sign, and traffic sign are generally installed along the side of Chinese highway. Firstly, a fixed marker, such as signs, is selected as the target in the camera image collection range. Secondly, then based on DCP, camera images are used to detect region visibility. Thirdly, this visibility information will be displayed on the upstream variable message sign. Thus, the drives in the downstream section can be able to perceive the visibility information in advance. The right device in Figure 1 is the visibility detector whose data will be used to compare with the detection results of DCP method in this paper.

2.3. Error Analysis and Optimization of Dark Channel Prior. Because of the unpredictable environment in different highway segments, when DCP method is directly used to detect visibility, great number of errors will be generated. With preliminary analysis, the main errors come from sky area transmittance calculation [24], different atmosphere brightness values [25], and image block effect [21]. Therefore, we need to make the appropriate optimization algorithm for different error sources as our first step.

2.3.1. Error Analysis and Optimization in Sky Region. Based on the assumption that there are dark pixels in local image region, transmittance can be calculated by DCP algorithm. Video camera images can be divided into sky region and nonsky region. However, as the sky region does not conform to this assumption, a greater deviation in the calculation of transmittance will be generated [24].

When dark channel is not considered, the actual transmittance is calculated by formula (5). In the regions without dark channel, such as sky region, there is $\min_{c\in\{r,g,b\}}(\min_{y\in\Omega(x)} J^c(y)/A^c) > 0$. So the actual transmittance is larger than the estimated one calculated by formula (7). In contrast, in the dark channel area, as $\min_{c\in\{r,g,b\}}(\min_{y\in\Omega(x)} J^c(y)/A^c)$ is close to zero, the difference between actual transmittances and estimated one is very small.

Analyzing pixel characteristics of the sky region, the gray value changes are small and different from that of other pixels of nonsky region. By researching literatures [26–28], region growing method gathers the pixels based on similar characteristics in the same pixel area. One region starts from an initial pixel. And these pixels with same characteristics of the initial pixel will be incorporated into with this region. So this region gradually grows until there are no other pixels can be merged. Thus, during the growth process, region growing method requires a group of proper seed pixel in the desired area and precise guidelines for merging adjacent pixels. When seed pixel growth and appropriate guidelines are selected, region growing method has obvious advantages in computation time and segmentation accuracy. Therefore, we use region growing method to divide image into sky region and nonsky region.

(1) Seed Selection. The seed selection directly influences the effect of region growing. In the image, the brightness of sky

Figure 23: Estimated transmittance maps of the tunnel before optimization on 02/03/2014.

region is higher than that of nonsky region. So the maximum brightness pixel in dark channel image is used as seed pixel. Seed pixel is selected as formula

$$\text{Seed} = \max\left(\min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} I^c(y)\right)\right). \tag{12}$$

(2) Growth Rules Based on Region Gray Scale Difference. According to the analysis of normal gray images, sky region has relatively higher brightness than nonsky region. The gray scale difference is set as growth criterion. When the gray scale difference between seed pixel and its neighbor is less than a predetermined threshold value, the latter will be merged into sky region. Then, the merged pixel is selected as a new starting point, comparison and merging will continue pixel by pixel, until the region can no longer expand. The grayscale values of sky region are greater than that of nonsky region and the grayscale changes slowly. When it meets the following constraints, the region is sky; otherwise, it is the nonsky region:

$$\begin{aligned} \text{s.t.}\quad & \left|I^c(y) - \text{Seed}\right| \le \text{th1} \\ & \left|\min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} I^c(y)\right) - \text{Seed}\right| \le \text{th2}, \end{aligned} \tag{13}$$

where th1 is grayscale threshold in the gray image and th2 is grayscale threshold in the dark channel image. Due to the fact that white cloud regions do not meet the growth rules of sky and have no dark channels, th3 is used as threshold to distinguish these regions. Therefore, the pixels of these regions meet formula (14) and will be merged into sky region:

$$\min_{c\in\{r,g,b\}}\left(\min_{y\in\Omega(x)} I^c(y)\right) > \text{th3}. \tag{14}$$

According to literatures [26–28], both ranges of th1 and th2 are from 0 to 0.1. The range of th3 is from 0.5 to 1. In this paper, based on a large number of video image sample tests, there are th1 = th2 = 0.5 and th3 = 0.75. By using the method described above, original image and regional segmented image are shown in Figure 2.

Next, two correction coefficients K and ω_s are introduced to realize accurate transmittance calculation in sky region, where $K = 1/(1 - \min_{c\in\{r,g,b\}}(\min_{y\in\Omega(x)} J^c(y)/A^c))$ and ω_s is the correction factor for atmospheric brightness. In nonsky region, considering the existence of dark channels, we directly use a correction parameter ω_t to revise transmittance. Transmittance corrections are calculated as follows:

$$t(x) = \begin{cases} K\left(1 - \omega_s \min\limits_{c\in\{r,g,b\}}\left(\dfrac{\min_{y\in\Omega(x)} I^c(y)}{A^c}\right)\right) & (\text{sky region}), \\ 1 - \omega_t \min\limits_{c\in\{r,g,b\}}\left(\dfrac{\min_{y\in\Omega(x)} I^c(y)}{A^c}\right) & (\text{nonsky region}). \end{cases} \tag{15}$$

2.3.2. Transmittance Optimization Based on Multimode Detection. When light transmits in the atmosphere, various factors cause energy attenuation with different degrees. In the same scene, different fog concentrations correspond to different atmospheric brightness, if the same correction factor is selected under different conditions that will result in

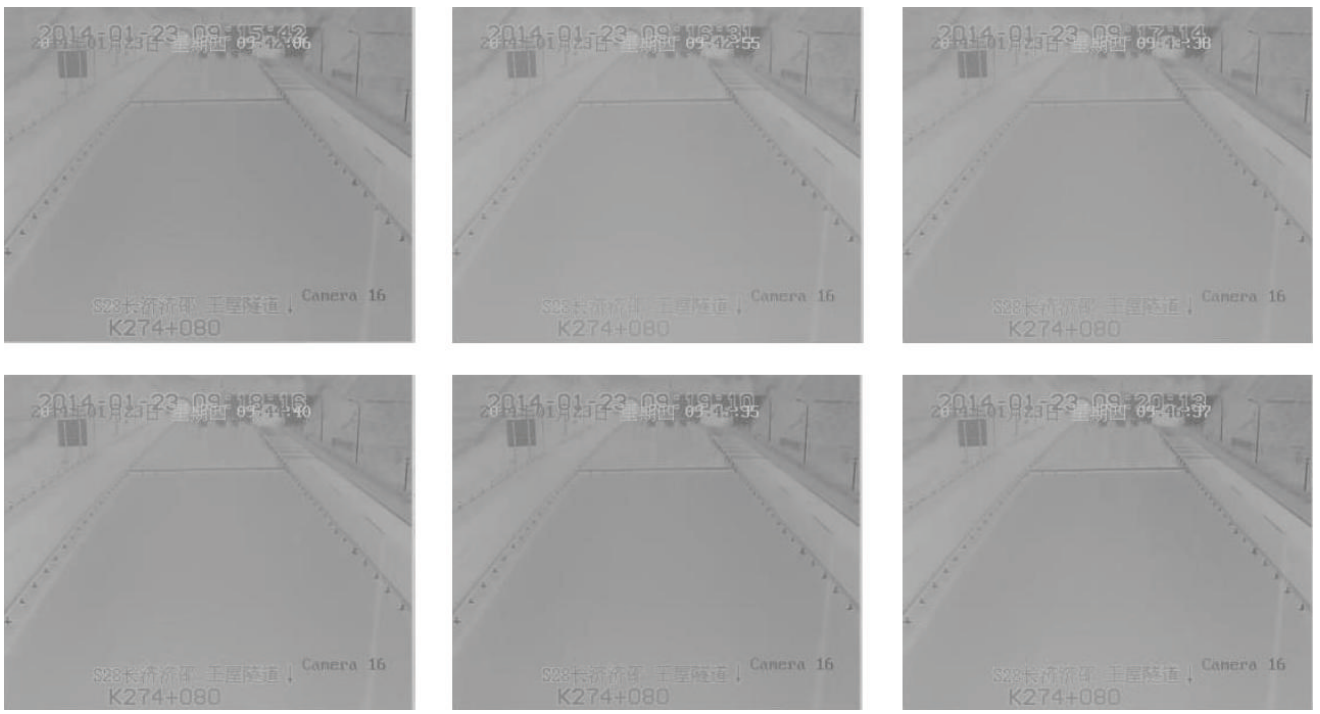


Figure 24: Estimated transmittance maps of tunnel after optimization on 01/23/2014.

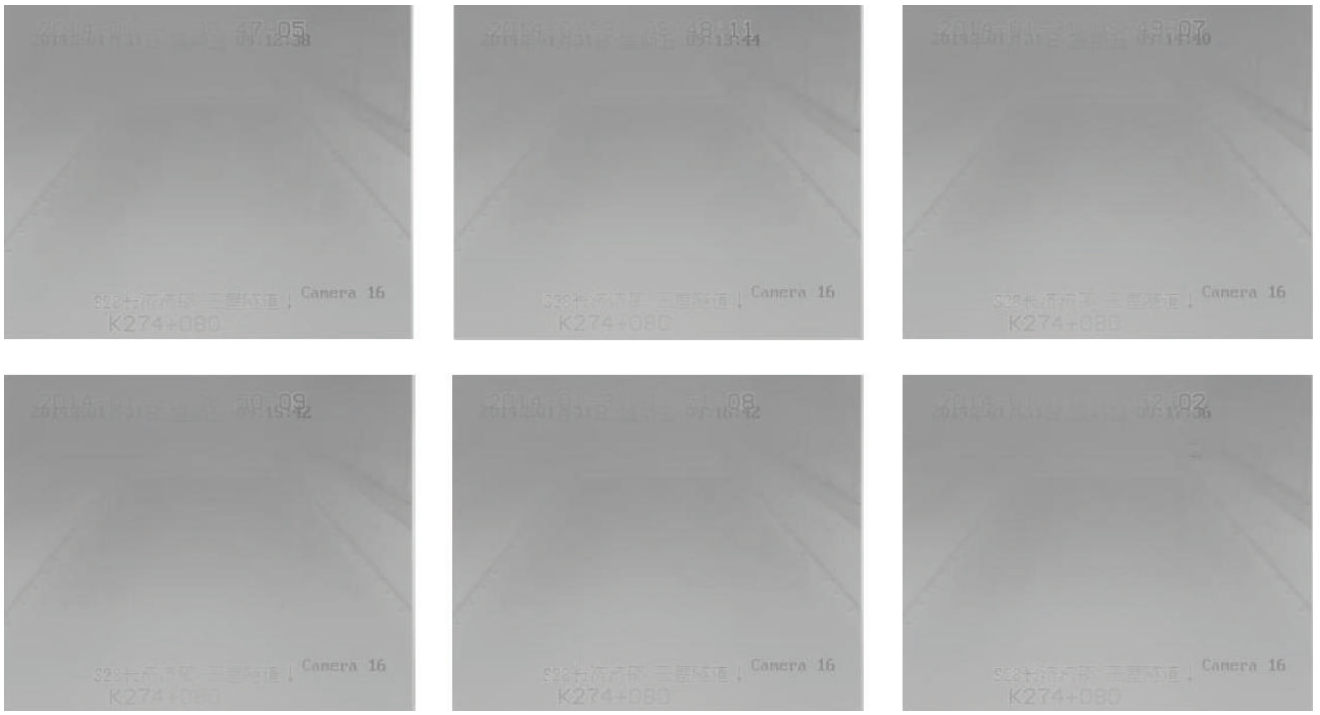


Figure 25: Estimated transmittance maps of the tunnel after optimization on 01/31/2014.

an inaccuracy in transmittance calculation [29]. Therefore, based on the visibility level cited from the standard of "Monitoring of Visibility and Warning of Heavy Fog on Highway" [30], the visibility detection is divided into five modes as shown in Table 1.

For each mode, the difference between estimated visibility and actual visibility is considered as the objective function. By using the least square method, different correction coefficients of transmittance calculation are derived for sky region and nonsky region.

Figure 26: Estimated transmittance maps of the tunnel after optimization on 02/03/2014.

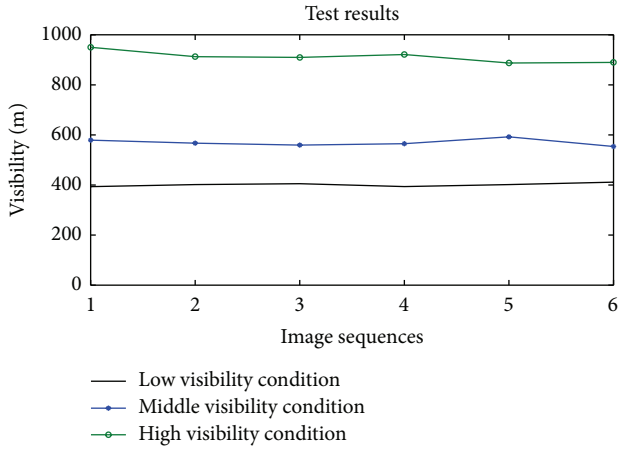


Figure 27: Visibility detection results of the tunnel.

In sky region, the objective function and transmittance correction function in mode i are established as formula

$$\text{s.t.} \quad \varphi\left(K_i, \omega_{s_i}\right) = \min \sum_{i=0}^{n} \left[V\left(t_i\left(x\right)\right) - V_i\right]^2$$
$$t_i\left(x\right) = K_i\left(1 - \omega_{s_i} \min_{c\in\{r,g,b\}}\left(\frac{\min_{y\in\Omega(x)} I^c\left(y\right)}{A^c}\right)\right) \tag{16}$$

in which V_i is actual visibility. $V(t_i(x))$ is estimated visibility when the transmittance is $t_i(x)$. K_i and ω_{s_i} stand for correction parameters of sky region in mode i, respectively.

Table 1: Visibility grade of highway and pattern classification.

Mode	Visibility grade	Visibility range (m)
Mode 1	**0**	$500 < V$
Mode 2	**1**	$200 < V \leq 500$
Mode 3	**2**	$100 < V \leq 200$
Mode 4	**3**	$50 < V \leq 100$
Mode 5	**4**	$V \leq 50$

In nonsky region, the objective function model and transmittance correction function in mode i are set up as formula

$$\text{s.t.} \quad \varphi\left(\omega_{t_i}\right) = \min \sum_{i=0}^{n} \left[V\left(t_i\left(x\right)\right) - V_i\right]^2$$
$$t_i\left(x\right) = 1 - \omega_{t_i} \min_{c\in\{r,g,b\}}\left(\frac{\min_{y\in\Omega(x)} I^c\left(y\right)}{A^c}\right), \tag{17}$$

where ω_{t_i} is the correction parameter of nonsky region in model i.

2.3.3. Transmittance Optimization Based on Guided Filtering. When the sizes of local region $\Omega(x)$ are different, the dark channel images are different. The larger the patch is, the greater the potential dark channel is, and the more obvious the block effect will be. However, significant block effect influences the accurate calculation of transmittance. As shown in Figure 3, in the dark channel image, block effect of 9×9 patches is more obvious than that of 3×3 patches.

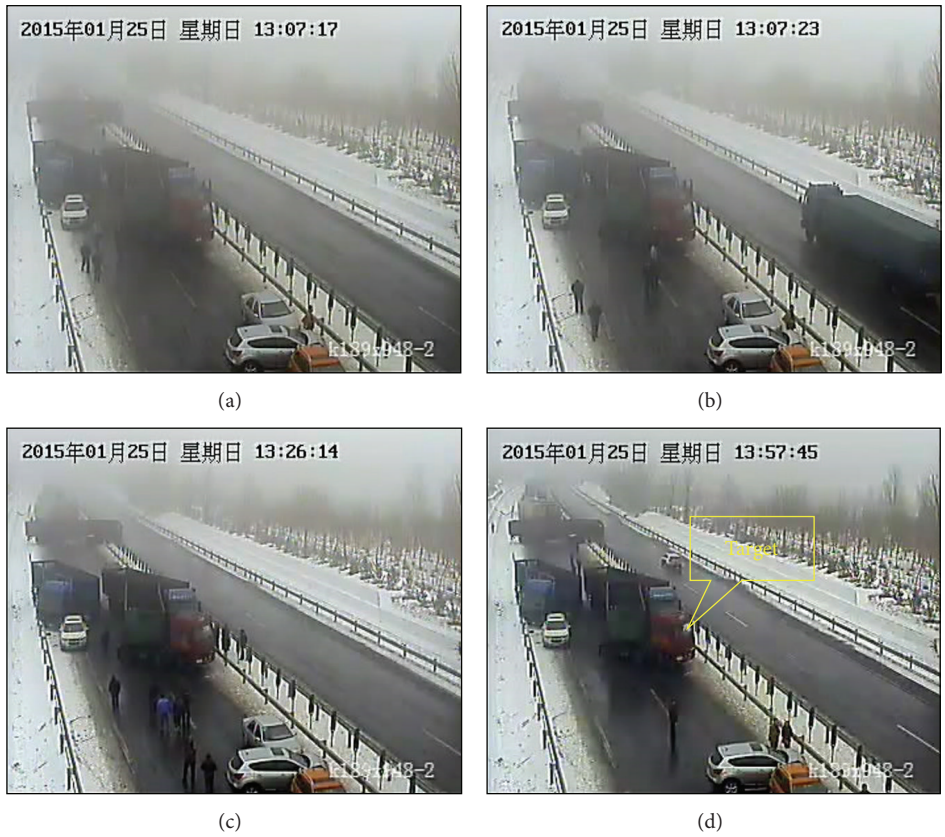


Figure 28: Image sequences of the mountain highway. (a) 13:07:17. (b) 13:07:23. (c) 13:26:14. (d) 13:57:45.

Guided filtering theory was proposed by He et al. [31] and applied to eliminate block effect of transmittance calculation in DCP method. This theory is based on local linear model. It assumes that image is a two-dimensional function, and the output and input in a two-dimensional window have a linear relation as formula

$$q_i = a_k I_i + b_k, \quad \forall i \in \omega_k, \tag{18}$$

where q_i is the value of output pixel, I_i is the value of guidance image, and i and k are pixel indexes. ω_k is a window which includes pixel i and k is the center of ω_k. a_k and b_k are two coefficients of linear function in ω_k.

In order to obtain the linear function coefficients, the following cost function is used:

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left(\left(a_k I_i + b_k - p_i \right)^2 + \varepsilon a_k^2 \right), \tag{19}$$

where ε is a regularization parameter. Linear regression is used to minimize the difference between output value q and actual value p. The linear function coefficients are obtained as follows:

$$a_k = \frac{(1/|\omega|) \sum_{i \in \omega_k} I_i p_i - \mu_k \overline{p}_k}{\sigma_k^2 + \varepsilon}, \qquad b_k = \overline{p}_k - a_k \mu_k, \tag{20}$$

where μ_k and σ_k^2 are the mean and variance of I in ω_k, $|\omega|$ is the number of pixels in ω_k, and $\overline{p}_k$ is the mean of p in ω_k.

A pixel may be contained in many windows; that is, each pixel can be described by a number of linear functions. Therefore, when calculating the linear coefficients of each window, the output value of a certain pixel is the average of all the linear functions which contains this point. There is

$$q_i = \frac{1}{|\omega|} \sum_{k: i \in \omega_k} \left(a_k I_i + b_k \right) = \overline{a}_i I_i + \overline{b}_i. \tag{21}$$

Therefore, when Figure 3(a) is processed according to the transmittance correction functions (16) and (17), Figure 4(a) can be obtained. After Figure 4(a) is processed with guided filtering optimization method, Figure 4(b) can be obtained. Comparisons of results of estimated transmittance are shown in Figure 4. After optimization, the block effect caused by the size of window can be eliminated, and the edge feature information is well-preserved.

2.4. Visibility Detection Processes Based on Dark Channel Prior. In summary, images captured by highway video surveillance systems are colored; the first step is to convert these images to dark channel images, which will be segmented into sky region and nonsky region based on region growing method. Secondly, in accordance with the appropriate detection mode, different factors are selected for

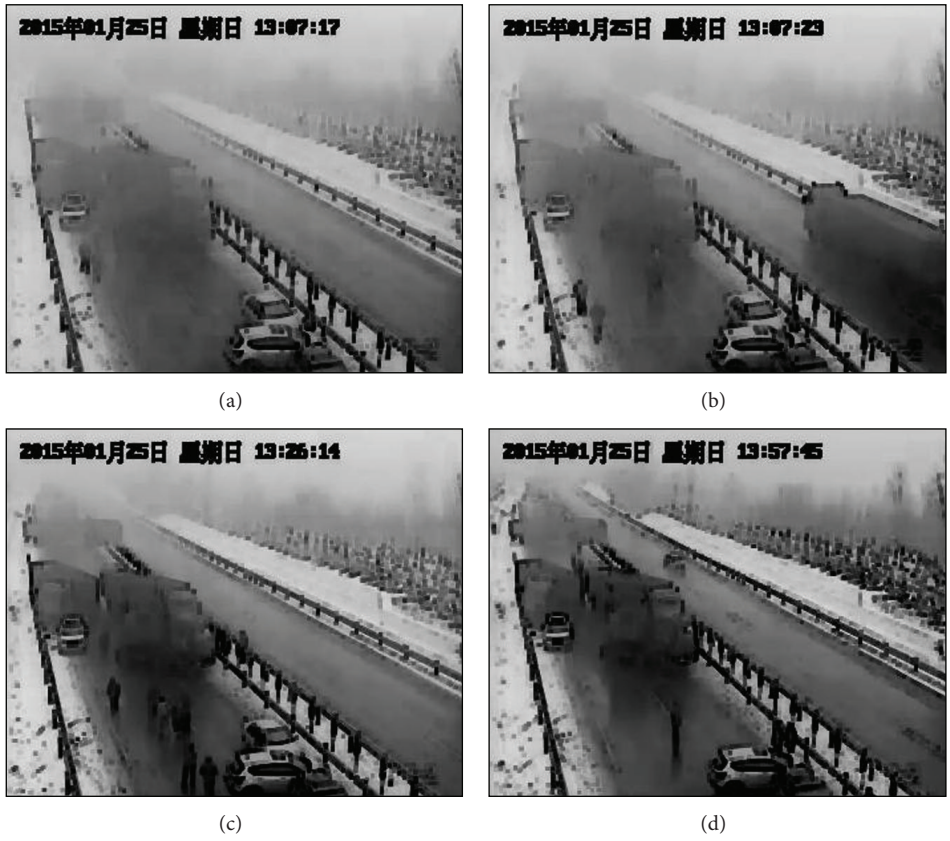


Figure 29: Dark channel images of the mountain highway. (a) 13:07:17. (b) 13:07:23. (c) 13:26:14. (d) 13:57:45.

correction. In order to improve the accuracy in transmittance calculation, guided filtering is used to eliminate block effect. Then, atmospheric brightness and target region transmittance are extracted. Finally, the Lambert law is utilized to calculate extinction coefficient and visibility value.

3. Precision Analysis of Visibility Detection Based on Dark Channel Prior

3.1. Contrast Test Analysis of Visibility Detection. To verify the accuracy of the aforementioned detection algorithms, as shown in Figure 5, a comparative test system is built based on the traffic test site in Beijing, China [32]. This test system includes a visibility detector, a video camera, and a traffic sign of which distance is 105 meters away from camera. The visibility detector model is VAISALA PWD 12 of which range is from 10 to 2000 meters. And this detector can not only measure accurate visibility but also display the reason for visibility decreasing. Due to the fact that traffic sign regions contain dark channel, this region is selected as the target. In addition, due to lack of light, the current highway camera images cannot be used to detect visibility at night. A light source is installed near by the traffic sign.

Based on the video image database captured from 01/09/2016 to 05/30/2016, an image was intercepted for analysis every ten minutes, and all the parameters of DCP algorithm were carefully verified and validated.

By using the daytime image captured on 03/18/2016, the visibility detection process of DCP is shown as Figure 6. The test system image is displayed in Figure 6(a). When the local patch size was set to 3×3, the dark channel image is shown in Figure 6(b). Estimated transmittance map before optimization is shown in Figure 6(c). Estimated transmittance map after optimization is shown in Figure 6(d). The visibility detection value is 956 meters. Simultaneously, the visibility detector value is 974 meters. The relative error is 1.85%.

By using the nighttime image captured on 03/10/2016, the visibility detection process of DCP is shown as Figure 7. The test system image is displayed in Figure 7(a). When the local patch size was set to 3×3, the dark channel image is shown in Figure 7(b). Estimated transmittance map before optimization is shown in Figure 7(c). Estimated transmittance map after optimization is shown in Figure 7(d). The visibility detection value is 732 meters. Simultaneously, the visibility detector value is 765 meters. The relative error is 4.31%.

One-day video image sequence was selected on 01/10/2016 as a validation. The visibility comparison results and relative errors of test system are shown in Figures 8 and 9. From two figures, the visibility results of optimization algorithm are consistent with that of the visibility detector in both trend and the numerical value. The relative error range is from 0% to 12.3% which meets the demands of highway visibility detection.

Figure 30: Estimated transmittance maps before optimization. (a) 13:07:17. (b) 13:07:23. (c) 13:26:14. (d) 13:57:45.

3.2. Verification of Dark Channel Prior Algorithm under Five Models. To further verify the effectiveness of the DCP algorithm, a large number of video images, captured by the camera in different locations, were selected to verify the validation of dark channel optimization algorithm under five detection models.

(1) The Verification Results under the First Model. The original image came from Jinliwen Highway of Zhejiang province, China. The time is 5:00 p.m. on 12/05/2013. The traffic sign was selected as target which is 72 meters away from the camera. The visibility detection process of the first model is shown in Figure 10. The visibility detection value is 769.05 meters which are consistent with the day weather record of traffic monitoring center.

(2) The Verification Results under the Second Model. The original image came from Jinggangao Highway of Hunan province, China. The time is 9:36 a.m. on 01/17/2015. The traffic sign was selected as target which is 31.6 meters away from the camera. The visibility detection process of the second model is shown in Figure 11. The visibility detection value is 236.82 meters which is consistent with the day weather record of traffic monitoring center.

(3) The Verification Results under the Third Model. The original image came from Rongwu Highway of Shandong province, China. The time is 10:15 a.m. on 02/01/2014. The bridge was selected as target which is 109 meters away from the camera. The visibility detection process of the third model is shown in Figure 12. The visibility detection value is 149.96 meters which are consistent with the day weather record of traffic monitoring center.

(4) The Verification Results under the Fourth Model. The original image came from Chengyu Highway of Sichuan province, China. The time is 9:40 a.m. on 10/21/2015. The bridge was selected as target which is 61 meters away from the camera. The visibility detection process of the fourth model is shown in Figure 13. The visibility detection value is 81.51 meters which are consistent with the day weather record of traffic monitoring center.

(5) The Verification Results under the Fifth Model. The original image came from Tuwu Highway of Jilin province, China. The time is 8:00 a.m. on 10/21/2013. The traffic sign was selected as target which is 29.8 meters away from the camera. The visibility detection process of the fifth model is shown in Figure 14. The visibility detection value is 46.17 meters which is consistent with the day weather record of traffic monitoring center.

3.3. Verification of Dark Channel Prior Algorithm under the Typical Tunnel. The original images were selected from Wangwu Tunnel of Henan province, China. Images were

Figure 31: Estimated transmittance maps after optimization. (a) 13:07:17. (b) 13:07:23. (c) 13:26:14. (d) 13:57:45.

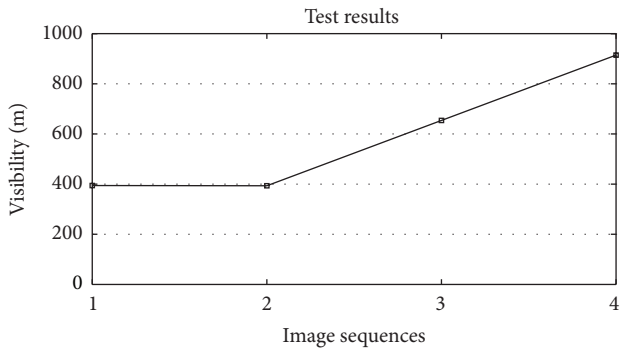


Figure 32: Detection visibility results of the mountain highway.

sampled every minute from continuous video streams of three days.

As shown in Figure 15, the sample times of six images are 9:15, 9:16, 9:17, 9:18, 9:19, and 9:20 on 01/23/2014. And the traffic sign was selected as target which was 88 meters away from the camera. As shown in Figure 16, the sample time of six images is 8:47, 8:48, 8:49, 8:50, 8:51, and 8:52 on 01/31/2014. As shown in Figure 17, the sample time of six images is 8:58, 8:59, 9:00, 9:01, 9:02, and 9:03 on 02/03/2014.

When the local patch size was set to 3×3, the dark channel images in three conditions are shown in Figures 18, 19, and 20.

Before optimization, the estimated transmittance maps of the tunnel are shown as Figures 21, 22, and 23.

The estimated transmittance maps of tunnel after optimization of three days are shown in Figures 24, 25, and 26.

By calculating the transmittance, the visibility values in three days are obtained, as shown in Figure 27. According to these figures, low visibility condition occurred on 01/31/2014; middle visibility condition occurred on 02/03/2014; high visibility condition occurred on 01/23/2014.

In a certain visibility condition, the actual visibility varied slowly in a short time interval. From Figure 27, visibility is relatively stable in six minutes which is consistent with visibility change rule. Corresponding to the original images, visual visibilities of three cases are from low to high sequentially. Detection visibility by DCP optimization algorithm has the same trend. The test results show that this detecting method and system were feasible.

3.4. Verification of Dark Channel Prior Algorithm under a Mountain Highway. The video image sequences were selected from Xuanda Highway in Hebei province, China. This is a mountain highway. As shown in Figure 28, the times of images are 13:07:17, 13:07:23, 13:26:14, and 13:57:45 on 01/25/2015. Because of the low visibility and snowy weather, an accident occurred during this time. The cabin area of the red truck is selected as target area which is 42 meters away from the camera. The dark channel images are shown

in Figure 29. The estimated transmittance maps before optimization and after optimization are separately shown in Figures 30 and 31.

The eliminating degree of block effect in Figure 31 is better than the one in Figure 30. From the original images of Figure 28, fog gradually diminished over time, and the visual visibility changes from low to high. As shown in Figure 32, the detection visibility values based on DCP optimization algorithm are gradually increasing and fully agree with the actual situation.

4. Conclusions

Low visibility is a major factor of causing of traffic accidents, which often result in a heavy loss of human life and financial properties. The accurate detection of visibility and timely warning play important roles in assuring the highway operation smoothness. In order to make full use of the camera image resources on highways, after summarizing the advantages and disadvantages of template matching method, camera calibration method, and dual differential luminance algorithm, a new method is elucidated to detect highway visibility by using the method of DCP.

The region growing method is used to divide the sky and the nonsky region. Due to the fact that sky regions do not meet the assumptions of DCP method, different correction factors are adopted to decrease the deviations derived from transmittance calculation. According to different visibility conditions, the multimode division method is designed, which can reduce the errors from extracting atmospheric brightness values. Guided filtering method is used to eliminate the block effect and to obtain the accurate transmittance of pixels in those images. Thus, based on the theoretical relationship between transmittance and extinction coefficient in Lambert-Beer law, the accurate visibility values can be calculated.

For verifying the feasibility, a comparative system was developed to test the visibility detection accuracy of optimization algorithm. Also, a large number of highway video images were used to verify the validation of this proposed optimization algorithm under five models. The verification results of video image sequences of tunnel and mountain highway indicate that the detection values are reliable and consistent with the actual situation. Based on a number of experiments, these optimization algorithms can well satisfy the general requirements of low visibility detection on the highway. Meanwhile, we found from those researches that it still needs further analysis to select the target area precisely and automatically in complex highway environments.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Authors' Contributions

Jiandong Zhao and Mingmin Han presented the algorithms, analyzed the data, and cowrote the paper; Changcheng Li and Xin Xin installed the experiment system and performed the experiments.

Acknowledgments

This work is supported by "the Fundamental Research Funds for the Central Universities (2016JBM053)."

References

[1] S. Xin, *Research on Risk Coupling of Highway Traffic Safety Based on Catastrophe Theory*, Beijing Jiaotong University, Beijing, China, 2015.

[2] R. G. Hallowell, M. P. Matthews, and P. A. Pisano, "Automated extraction of weather variables from camera imagery," in *Proceedings of the Mid-Continent Transportation Research Symposium*, Ames, Iowa, USA, 2005.

[3] G. H. Robert and P. M. Michael, "Clarus Research: Visibility Estimation from Camera Imagery," For FHWA CLARUS Meeting, 2006.

[4] D. Bäumer, S. Versick, and B. Vogel, "Determination of the visibility using a digital panorama camera," *Atmospheric Environment*, vol. 42, no. 11, pp. 2593–2602, 2008.

[5] S. Bronte, L. M. Bergasa, and P. F. Alcantarilla, "Fog detection system based on computer vision techniques," in *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC '09)*, pp. 30–35, St. Louis, Mo, USA, October 2009.

[6] C. Steffens, "Measurement of visibility by photographic photometry," *Industrial & Engineering Chemistry*, vol. 41, no. 11, pp. 2396–2399, 1949.

[7] M. K. Taek, "Atmospheric visibility measurements using video cameras: relative visibility," Minnesota Department of Transportation Technique Report CTS-0403, 2004.

[8] R. Babari, N. Hautière, É. Dumont, R. Brémond, and N. Paparoditis, "A model-driven approach to estimate atmospheric visibility with ordinary cameras," *Atmospheric Environment*, vol. 45, no. 30, pp. 5316–5324, 2011.

[9] R. Babari, N. Hautière, É. Dumont, N. Paparoditis, and J. Misener, "Visibility monitoring using conventional roadside cameras—emerging applications," *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 17–28, 2012.

[10] Q. K. Zhou, Z. Z. Chen, and Q. M. Chen, "Visibility detection system based on road monitoring camera," *Electronic Measurement Technology*, vol. 32, no. 6, pp. 72–76, 2009.

[11] Z. Z. Chen, Q. K. Zhou, and Q. M. Chen, "Video visibility detection algorithm based on wavelet transformation," *Chinese Journal of Scientific Instrument*, vol. 31, no. 1, pp. 92–98, 2010.

[12] Z.-Z. Chen and Q.-M. Chen, "Video contrast visibility detection algorithm and its implementation based on camera self-calibration," *Journal of Electronics & Information Technology*, vol. 32, no. 12, pp. 2907–2912, 2010.

[13] M. W. An, Q. M. Chen, and Z. L. Guo, "Visibility detection method and system design based on traffic video," *Chinese Journal of Scientific Instrument*, vol. 31, no. 5, pp. 1148–1153, 2010.

[14] N. Hautiére, J.-P. Tarel, J. Lavenant, and D. Aubert, "Automatic fog detection and estimation of visibility distance through use of an onboard camera," *Machine Vision and Applications*, vol. 17, no. 1, pp. 8–20, 2006.

[15] B. Li, R. Dong, and Q. Chen, "Visibility detection based on traffic video contrast analysis without artificial markers,"

Journal of Computer-Aided Design and Computer Graphics, vol. 21, no. 11, pp. 1575–1582, 2009.

[16] Y. Li and W. T. Zhou, "Visibility detection algorithm based on date camera technology for road monitoring," *Modern Electronics Technique*, vol. 35, no. 20, pp. 95–97, 2012.

[17] X. Zhang, B. Li, and Q. M. Chen, "PTZ visibility detection algorithm based on luminance characteristic and its implementation," *Chinese Journal of Scientific Instrument*, vol. 32, no. 2, pp. 381–387, 2011.

[18] W. T. Lv, S. C. Tao, Y. F. Liu, Y. B. Tang, and B. G. Wang, "Measuring meteorological visibility based on digital Photography-Dual differential luminance method and experimental study," *Chinese Journal of Atmospheric Sciences*, vol. 28, no. 4, pp. 559–568, 2004.

[19] F. Chang, X. T. Chen, M. X. Xiao, and W. W. Jiang, "Visibility algorithm design and implementation of digital camera visibility instrument," *Microcomputer & Its Applications*, vol. 32, no. 9, pp. 35–41, 2013.

[20] J. D. Zhao, M. M. Han, X. Xin, and C. C. Li, "Multi-mode detection techniques of video visibility based on improved dual differential luminance algorithm," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 1, pp. 147–158, 2016.

[21] K. M. He, J. Sun, and X. O. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.

[22] Y. C. Wang, *Research on Haze Removal Using Dark Channel Prior*, Dalian University of Technology, 2011.

[23] S. S. Guo, W. X. Qi, and Y. Qi, "Video visibility measurement method based on dark channel prior," *Computer & Digital Engineering*, vol. 42, no. 4, pp. 694–697, 2014.

[24] N. Graves and S. Newsam, "Using visibility cameras to estimate atmospheric light extinction," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '11)*, pp. 577–584, January 2011.

[25] Y. W. Zhou, Q. S. Sun, and B. P. Hu, "Remote sensing image enhancement based on dark channel prior and bilateral filtering," *Journal of Image and Graphics*, vol. 19, no. 2, pp. 313–321, 2014.

[26] S. Duan and Y. F. Wang, "Color images feature extraction based on region growing," *Journal of South-Central University for Nationalities (Natural Science Edition)*, vol. 31, no. 2, pp. 104–108, 2012.

[27] Z. J. Liu, R. N. Ma, G. P. Zou, B. J. Zhong, and J. D. Ding, "An algorithm for color image segmentation based on region growth," *Journal of Shandong University (Natural Science)*, vol. 45, no. 7, pp. 76–80, 2010.

[28] J. H. Yang, J. Liu, J. C. Zhong, and M. Z. He, "Image segmentation by integrating watershed with automatic seeded region growing," *Journal of Image and Graphics*, vol. 15, no. 1, pp. 63–68, 2010.

[29] H. G. Sun, C. Fang, H. J. Zhang, L. H. Liu, and J. Z. Wang, "An adaptive haze removal based on dark channel prior," *Journal of Jilin University*, vol. 50, no. 5, pp. 987–992, 2012.

[30] QX/T76-2007, *Monitoring of Visibility and Warning of Heavy Fog on Highway*, Meteorology Press, Beijing, China, 2007.

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

[32] M. M. Han, *Highway Visibility Detecting Technology Based on Video Images*, Beijing Jiaotong University, Beijing, China, 2016.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8740593, 11 pages
http://dx.doi.org/10.1155/2016/8740593

Research Article

Graph-Based Salient Region Detection through Linear Neighborhoods

Lijuan Xu,[1] Fan Wang,[1] Yan Yang,[2] Xiaopeng Hu,[1] and Yuanyuan Sun[1]

[1]*School of Computer Science and Technology, Dalian University of Technology, No. 2 Linggong Road, Dalian 116024, China*
[2]*School of Computer and Information Technology, Liaoning Normal University, No. 850 Huanghe Road, Dalian 116024, China*

Correspondence should be addressed to Xiaopeng Hu; xphu@dlut.edu.cn

Received 17 March 2016; Accepted 9 May 2016

Academic Editor: Chanho Jung

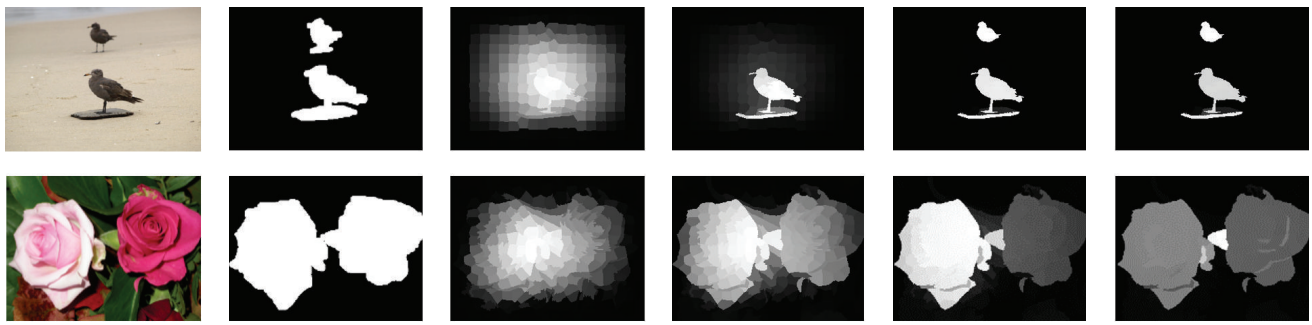
Pairwise neighboring relationships estimated by Gaussian weight function have been extensively adopted in the graph-based salient region detection methods recently. However, the learning of the parameters remains a problem as nonoptimal models will affect the detection results significantly. To tackle this challenge, we first apply the adjacent information provided by all neighbors of each node to construct the undirected weight graph, based on the assumption that every node can be optimally reconstructed by a linear combination of its neighbors. Then, the saliency detection is modeled as the process of graph labelling by learning from partially selected seeds (labeled data) in the graph. The promising experimental results presented on some datasets demonstrate the effectiveness and reliability of our proposed graph-based saliency detection method through linear neighborhoods.

1. Introduction

The goal of saliency detection is to identify and locate the most interesting and important region that pops out from the rest in an image, which has been widely used for applications in computer vision, including object detection and recognition [1, 2], image compression [3], image segmentation [4], content based image retrieval [5], image cropping [6], and photo collage [7].

Numerous researches have been conducted to design various algorithms for salient region detection. Among these works, graph-based saliency detection models have aroused considerable interest in recent years. Previous works on detecting salient regions from images represented as graphs include [8–15]. These models describe the input image as an undirected weight graph, in which vertices represent the image elements (pixels/regions) and edges represent the pairwise dissimilarity between vertices, and the salient object detection problem is formulated as random walks [8–10], binary segmentation [11, 12], labelling (ranking) task [13, 14], or distance metric [15] on the graph, which aims at finding the pop-out vertices at some local or global locations.

In methods of the random walks on graphs [8–10], the identification of salient regions is determined by the frequency of visits to each node at equilibrium. In [8], while some results are presented on only two synthetic images, there is no evaluation of how the method will work on real images. In [9], Harel et al. constructed the full-connected directed graph to represent the image in which the weight of the edge between two vertices is proportional to their dissimilarity, as well as their closeness in the spatial domain. Nonsalient regions are defined as the most frequently visited vertices in a local context. Wang et al. [10] analyzed multiple cues in a unified energy minimization framework and used the model in [9] to detect salient objects. A major problem is that cluttered backgrounds usually yield higher saliencies for possessing high local contrasts. Lu et al. [11] and Liu et al. [12] regarded the saliency detection problem as binary segmentation on a graph. In [11], Lu et al. developed a hierarchical graph model and utilize concavity context to compute weights between nodes, from which the graph is bipartitioned for salient object detection. Gopalakrishnan et al. [13] and Yang et al. [14] defined the saliency as the labelling or ranking task on a graph and applied the semisupervised learning technique to infer the binary labels of the unlabeled vertices with the salient seeds. However, it is difficult to determine the number and location of salient seeds that the semisupervised method requires, which is a known problem



(a) Original image (b) Ground truth (c) $\sigma = 1$ (d) $\sigma = 5$ (e) $\sigma = 10$ (f) $\sigma = 20$

Figure 1: The results of different variances in graph-based saliency models. This figure shows the saliency maps obtained by the manifold ranking method proposed by Yang et al. [14] under different variances σ of the edge weight defined with Gaussian function in the graph. It is obvious that perturbation of the variance could make the detection results dramatically different.

with graph labelling. In addition, the geodesic distance metric was applied to measure the feature contrast along paths on the graph in [15].

The reason why the graph model can be associated with the saliency detection is that the prior consistency or cluster assumption [16, 17] observed in semisupervised learning or manifold learning problem, which have been demonstrated effectively to preserve the intrinsic data structure hidden in the dataset, is also appropriate for uncovering the relationships between pixels in the image. The prior consistency mainly consists of two aspects: (1) nearby pixels are likely to have the same saliency; (2) pixels on the same structure (such as an object or a homogeneous region) are likely to have the same saliency. Note that the first assumption is local, while the second one is global. The cluster assumption advises us to consider both local and global information contained in the image during learning. It is straightforward to apply cluster assumption to the graph-based saliency detection models developed in recent years, since the central idea of these methods is to find the pop-out or salient nodes while preserving the global structure hidden in the image.

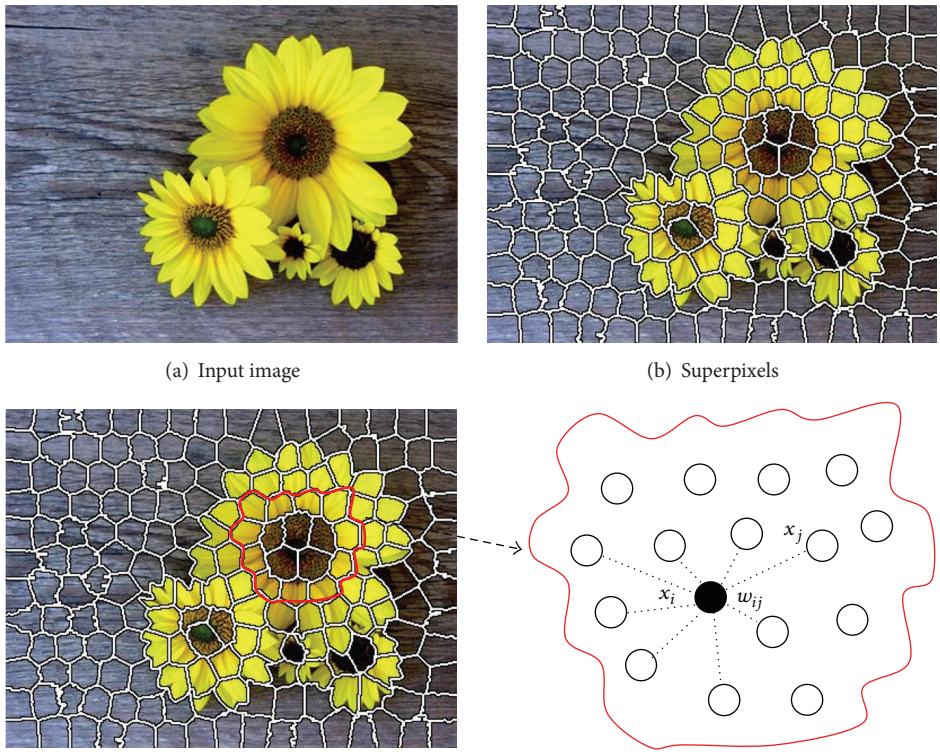
Although there has been some success with the graph-based saliency detection approaches, identifying salient objects in natural scenes remains a challenge because factors such as the local or global structure information are not fully described. The graph-based semisupervised learning or manifold learning methods model the whole dataset as a graph. Similarly, the graph-based saliency detection method models the input image in the same way. In most graph-based models, the superpixels are extracted and denoted as the basic graph nodes in consideration of the computation efficiency and perception meaning. In addition, the complete graph [9, 13], k nearest neighboring graph or k-regular graph [13, 15], or the close-loop graph [14] is applied to simulate the local graph structure in different saliency models. However, how to estimate the weight of each edge has not been fully studied. More concretely, most of methods adopted a Gaussian function to calculate the edge weights of the graph [9, 13–15]. But the variance σ of the Gaussian function will affect the detection results significantly. This problem has been demonstrated in the semisupervised learning methods [18], which occurs in the graph-based saliency models (illustrated in Figure 1) as well. However, there is no reliable approach for model selection if only very few labeled seeds are available; that is, it is hard to determine optimal σ, as pointed out by Zhou et al. [16].

To address the above issues, we propose a more reliable and stable graph-based saliency detection model in this paper. Firstly, the nodes of the graph are made up of a series of neighboring image superpixels, and the edges represent the neighborhood relationships between different image superpixels. Instead of considering pairwise neighborhood relationships adopted in current graph-based saliency detection methods, we apply the adjacent information provided by all neighbors of each image node to estimate the edge weighs in the graph based on the locally linear assumption that nearby superpixels are likely to have the same saliency. Then the edge weights of all nodes are assigned to the edges for constructing the undirected weight graph. Finally, we model the saliency detection as the process of labelling by learning from partially selected seeds (labeled data) in the graph. The experiments on some datasets demonstrate the effectiveness and higher parameter stability of our proposed graph-based saliency method.

The remainder of the paper is organized as follows. Section 2 will describe our proposed model in detail. In Section 3, the experiments on some popular datasets are presented, followed by the conclusions and future works in Section 4.

2. Proposed Method

In the proposed method, a spatially neighboring graph, where superpixels are extracted and considered as the basic graph nodes and the linear relationships for all neighbors of each node are applied to estimate the weights of edges, is constructed to represent the local structure in the image. The problem of saliency detection can be tackled by modeling the task of labelling by using the selected seeds in the whole graph. The emphasis of this section and the major contribution of this paper are the construction of the graph by all neighbors of each node with the assumption of the prior



(a) Input image (b) Superpixels

(c) The weighted graph

FIGURE 2: The illustration of the undirected weight graph. (a) An input image. (b) The image superpixels generated by SLIC segmentation. The construction of the weighted graph is shown in (c). If x_j belongs to the set of neighbors for node x_i, w_{ij} describes the contribution of node x_j to reconstruct node x_i.

consistency and the following graph labelling for saliency detection.

2.1. Graph Construction. As shown in Figure 2, an undirected weighted graph $G = (V, E)$ is constructed to represent the input image, where V is a set of nodes and E is a set of undirected edges with weight w_{ij}. In this paper, the nodes are visually homogeneous superpixels, which are computationally efficient and perceptually meaningful compared to regular image patches, and generated by the Simple Linear Iterative Clustering (SLIC) algorithm proposed by Achanta et al. [19]. The reason why we choose the SLIC is that the resulting superpixels are almost regular and compact image patches with better boundary adherence, which facilitate the preservation of the object edges in the saliency map [14, 15, 20]. In addition, the spatially neighboring superpixels are connected to simulate the local neighborhood relationships for all the nodes in the graph. Each edge is assigned a weight to represent the relationship between the two nodes.

Conventionally [9, 13–15], the weight w_{ij} of edge that links nodes x_i and x_j in E is computed as

$$w_{ij} = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right). \quad (1)$$

This Gaussian weight function mainly describes the pairwise dissimilarity between two neighboring superpixels. However, it is parameter dependent and sensitive as most natural images are more cluttered and complicated. Moreover, the locally linear information is ignored. Thus a more reliable and stable way to construct the graph based on all neighbors of each node is derived, which is capable of recovering the global nonlinear structure from the locally linear fits in the graph.

Based on the cluster assumption that each superpixel and its neighbors lie on or close to a locally linear region of the same manifold in the image, we characterize this local structure of these superpixels by linear coefficients that reconstruct each node from its neighbors in the corresponding graph. We measure the reconstruction errors by the following cost function:

$$\varepsilon = \sum_i \left\| x_i - \sum_{x_j \in N(x_i)} w_{ij} x_j \right\|^2, \quad (2)$$

where $N(x_i)$ denotes all the neighbors of x_i and w_{ij} summarizes the contribution of the node x_j to the node x_ith reconstruction. To estimate the weight w_{ij}, our objective is to minimize the cost function subject to three constraints: (1) each node x_i is reconstructed only from its neighbors $N(x_i)$; if x_j does not belong to the set of neighbors for node

x_i, w_{ij} is set to 0; (2) the sum of the contribution from all neighbors to node x_i equals 1; that is; $\sum_{x_j \in N(x_i)} w_{ij} = 1$; (3) we only consider the nonnegative edge weight, $w_{ij} \geq 0$, in our saliency detection model. Clearly, when x_j is more similar to the node x_i, w_{ij} will be larger, which means that x_j plays a more important role in reconstructing the node x_i. Thus the reconstruction weights of each node can be solved by the least-square algorithm with the three constraints:

$$\begin{aligned} \min_{w_{ij}} \quad & \left(\sum_i \left\| x_i - \sum_{x_j \in N(x_i)} w_{ij} x_j \right\|^2 \right) \\ \text{s.t.} \quad & \sum_j w_{ij} = 1, \\ & w_{ij} \geq 0. \end{aligned} \tag{3}$$

One question that should be noticed is that usually $w_{ij} \neq w_{ji}$; here we introduce the algorithm proposed in [18] to get a symmetric matrix. Once the reconstruction weights for all nodes in the graph are computed, we will obtain a sparse weight matrix $\mathbf{W} = w_{ij}$ for graph G. This weight graph describes the locally neighboring relationships through synthesizing the linear neighborhood around each node, which facilitates the discovery of the globally hidden structure and the following pop-out salient nodes in the image. Furthermore, our graph structure is capable of solving the problem of determining an optimal parameter that occurred in conventional Gaussian function method.

2.2. Saliency Model

2.2.1. Graph Labelling. In this paper, the problem of saliency detection is modeled as the task of graph labelling, which aims to propagate the labels of the selected seeds to the unlabeled nodes using the graph constructed in Section 2.1. Given a dataset $\{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\}$, the first l data belong to the set of labeled X_l and the remainders belong to the set of unlabeled X_u which need to be labeled according to the relevance to the labeled ones. Let $\mathbf{f} = (f_1, \ldots, f_n)^T$ denote some classifying functions defined on X, which can assign a real value f_i to each data point x_i. Let $\mathbf{y} = (y_1, \ldots, y_n)^T$ denote the label indicator for each data point. If $i \leq l$, $y_i = L_i$; otherwise, $y_i = 0$. Thus, the problem of labelling the unlabeled data can be solved by an iterative procedure.

In each iteration, each data point can "absorb" a fraction of label information from its neighbors and retain some label value of its present state. In other words, all data spread their label scores to their neighbors via the weighted graph. Therefore, the label of data point x_i at time $t+1$ becomes

$$f_i^{t+1} = \alpha \sum_{x_j \in N(x_i)} w_{ij} f_j^t + (1 - \alpha)\, y_i, \tag{4}$$

where $\mathbf{f}^t = (f_1^t, \ldots, f_n^t)^T$ is the classifying function learned at iteration t and $f^0 = \mathbf{y}$. The parameter α specifies the relative contributions to the labelling scores from neighbors and the initial labels, which is in $[0, 1)$. Consider

$$\mathbf{f}^{t+1} = \alpha \mathbf{W} \mathbf{f}^t + (1 - \alpha)\, \mathbf{y}. \tag{5}$$

Iterating (5) to update the label scores of all the data until convergence, let $\mathbf{f}^*$ be the limit of the sequence $\mathbf{f}^t$

$$\mathbf{f}^* = (1 - \alpha)\, (\mathbf{I} - \alpha \mathbf{W})^{-1}\, \mathbf{y}, \tag{6}$$

where $\mathbf{I}$ is the n-dimensional identity matrix. In (6), the constant term $1 - \alpha \geq 0$, will not affect the labelling results of $\mathbf{f}^*$. Here, $\mathbf{f}^*$ is equivalent to

$$\mathbf{f}^* = (\mathbf{I} - \alpha \mathbf{W})^{-1}\, \mathbf{y}. \tag{7}$$

The resulting label function $\mathbf{f}^*$ provides a general framework for semisupervised learning. Here, multiple labels are predefined for applications including clustering, segmentation, and classification. In [14], Yang et al. applied $\mathbf{f}^*$ to learning of the ranking scores for saliency detection according to the relations with the single label, which they call the manifold ranking [21] based algorithm.

2.2.2. Saliency Measure. For saliency detection, how to determine the number and location of salient labels that (7) requires remains a big problem. Reference [13] applied the most salient node and some background nodes together for label extraction of the salient region; however, the results are sensitive to the choosing of the seeds. Observing that background often presents local or global appearance consistent with the image boundary [15, 20, 22–24], Yang et al. [14] used the nodes on the image boundary to label most of the background nodes, which aims to produce some salient labels. Similar to the scheme proposed in [14], the salient regions are estimated by the process of two-phase graph labelling on the constructed weight graph (shown in Figure 3). In other words, the image boundary and the salient nodes are separated to generate the final saliency maps.

(1) Labelling with the Background Nodes. Based on the assumption that the image boundary is more likely to be background, we apply the superpixels on the image boundary as the seeds to remove some background clutters and in turn lead to better salient seeds for labelling. Considering the basic rule of the photographic composition that the four sides on the image boundary often show different appearances of the background, specially, they are marked with different labels for strong learning. If all of these boundary superpixels are characterized with the same label in the procedure of learning, the learned classifying function is usually less optimal as these nodes are dissimilar. Therefore, the first-phase saliency maps are generated using the four sides on the image boundary superpixels as the labeled nodes, respectively, and the rest as the unlabeled ones.

In this paper, we first obtain the four label indicator vectors $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\}$ of the top, bottom, left, and right image boundary. Then all the nodes on the graph are labelled based on $\mathbf{f}^*$ in (7) with the four label indicator vectors separately, and the results are four n-dimensional vectors, in which each

(a) Input image (b) Ground truth

(c) Saliency map for first phase (d) Saliency map for second phase

Figure 3: The framework of the two-phase saliency detection model. (a) Input image and (b) the ground truth labelled by humans. ((c) and (d)) The saliency maps generated by the first phase and second phase of graph labelling based saliency model, respectively.

element denotes the relevance to the labelled image boundary superpixels. We then normalize them to the range between 0 and 1. If the image elements are similar to the boundary, the labelling values obtained by $\mathbf{f}^*$ are closer to 1. However, the saliency values are characterized with labelling with salient seeds in saliency detection models. So the final label score for each superpixel is modified as

$$F(i) = 1 - \mathbf{f}^*(i), \tag{8}$$

where i is a node on the constructed graph. In order to obtain some candidate salient superpixels in the image, (8) is applied to integrate the four label scores generated by the four sides of image boundary, respectively. Thus, the first-phase saliency is defined as

$$S_{\text{first}}(i) = F_1(i) * F_2(i) * F_3(i) * F_4(i). \tag{9}$$

Saliency maps generated by (9) can suppress most of the background superpixels, thus highlighting some candidate salient ones. To tackle this problem, we use the label function in (7) with the candidate salient nodes to further improve the performance of detection salient regions.

(2) Labelling with the Candidate Salient Nodes. The second phase aims to eliminate some background superpixels and highlight salient regions using the labelling function in (7) with the possible salient nodes produced in the first-phase labelling, since the possible salient regions inferred from the image boundary are weak and prior dependent, in order to get stronger label seeds which are salient or belong to the actual salient regions, "Otsu's method" [25] is applied to partition the saliency maps generated in the first phase adaptively, and the remaining foreground superpixels are treated as strong salient seed for labelling. Then the label indicator vector $\mathbf{y}$ is established to compute the relevance function $\mathbf{f}^*$ in (7). Here, the saliency is defined as the normalized labelling scores between 0 and 1. Consider

$$S_{\text{second}}(i) = \mathbf{f}^*(i). \tag{10}$$

Despite some imprecise foreground labels, the salient objects can be well detected in the final saliency maps. The reason is that the salient regions are usually compact and not dispersed in terms of spatial distribution compared to background regions and homogeneous and consistent when considering the appearance in the aspect of feature distribution, such as color and texture [14]. All of these priors affect the detection results greatly.

In conventional semisupervised learning methods for graph labelling, the diagonal elements of matrix $\mathbf{A}$ ($\mathbf{A} = (\mathbf{I} - \alpha\mathbf{W})^{-1}$) play an important role in computing the relevance to the labelled data. Nevertheless, this diagonal matrix means that the relevance to each superpixel itself is considered in saliency detection if it does not equal 0, which can weaken the contributions of other labelled nodes as the self-relevance is usually to be large. In order to get better detection results, we set the diagonal elements of $\mathbf{A}$ to 0.

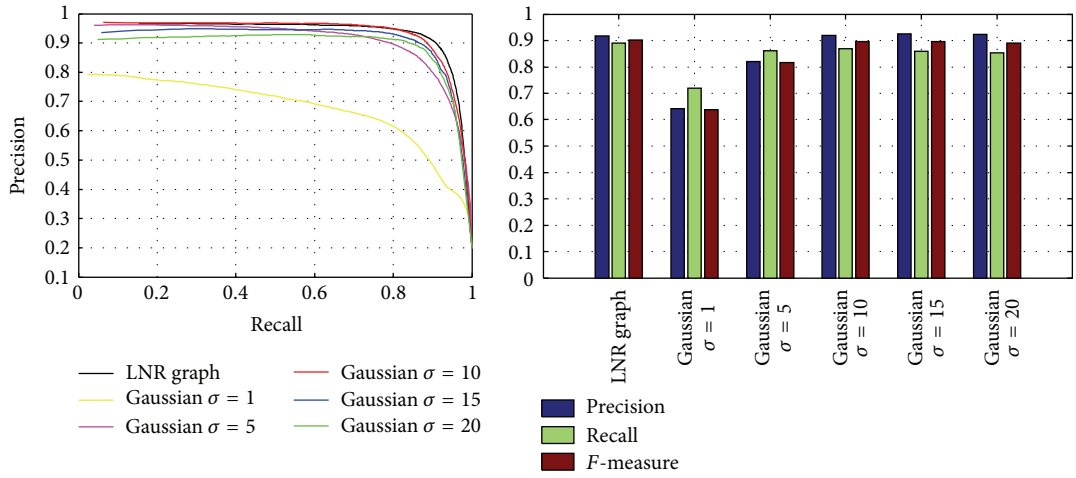


Figure 4: The comparison of different graphs. ((a) and (b)) The PR curves, as well as the precision, recall, and F-measure values for the comparison of LNR graph and graph weighted by the Gaussian function with different variances $\sigma = 1, 5, 10, 15, 20$, respectively. We note that the Gaussian weight function with the variance $\sigma = 10$ achieves the best performance. In fact, choosing an appropriate parameter is difficult.

3. Experiments

Dataset. Two standard benchmark datasets ASD and SOD are adopted to evaluate the proposed graph-based saliency model in this paper.

(1) ASD. It contains 1000 images with accurate human labeled segmentation masks for salient objects provided in [26], which has been used for testing almost all saliency models.

(2) SOD. The dataset comes from the well-known Berkeley segmentation dataset of 300 images [27], in which the images are cluttered and complex in terms of the scales, appearances, and positions of the foreground objects, as well as the appearance of the background regions. The pixel-labeled ground truth is obtained from [15].

Evaluation Metrics. For average performance evaluation, we use the standard PR (precision-recall) curve and F-measure. For each detected saliency map and the corresponding ground truth, precision rate corresponds to the ratio of salient pixels which are correctly detected in the saliency map, while recall rate is the percentage of all detected salient pixels belonging to salient objects in ground truth. To generate a PR curve, the saliency map is normalized into $[0, 255]$ first. A series of binary masks are then produced by segmenting the saliency map with a threshold varying from 0 to 255. We compare these binary masks with the ground truth to obtain the PR curve for each saliency map. The curves obtained from all images on each dataset are averaged to generate an overall PR curve.

Although commonly used, PR curve is limited in that it only considers whether the saliency of the object saliency is higher than that of the background. Since high precision can be achieved at the cost of decreasing the recall and vice versa, the F-measure is used to trade off the overall performance of the precision rate and recall rate:

$$F\text{-measure}_{\beta} = \frac{\left(1 + \beta^2\right) \text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}}. \tag{11}$$

And β^2 is set to 0.3 as in [26]. As different images have optimal binary threshold dissimilarly, a constant threshold adopted in the PR curve ignores it. Then an adaptive threshold value is proposed and determined as the mean saliency of all pixels in each saliency map as in [26] to measure the average precision, recall, and F-measure values on each dataset:

$$T = \frac{2}{W * H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} S\left(x, y\right), \tag{12}$$

where W and H are the width and height of the saliency map in pixels, respectively, and $S(x, y)$ is the saliency value of the pixel (x, y).

3.1. Validation of the Graph Constructed through Linear Neighborhood. In this section, the proposed LNR (linear neighborhood relationship) graph is compared with the conventional pairwise relationship based graph weighted by the Gaussian function with different variances to generate saliency maps for all images on the ASD dataset in our framework of saliency detection. Figure 4(a) illustrates the results of average PR curves, and Figure 4(b) shows the precision, recall, and F-measure values generated by the adaptive threshold, respectively. The experimental results indicate the reliability and stability of our LNR graph. The reason is that our LNR graph is more capable of representing the local information through the linear neighborhood around each superpixel,

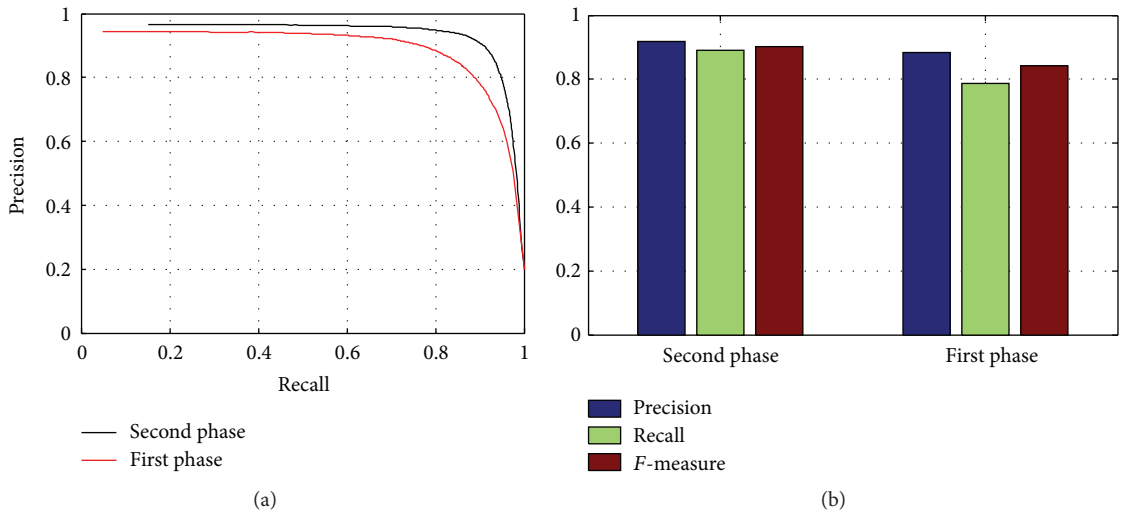


Figure 5: The comparison of two-phase saliency maps. ((a) and (b)) The PR curves and F-measure values for the comparison of the image boundary and some candidate salient nodes based graph labelling for saliency detection, respectively. We note that the second phase achieves better performance for more strong labels generated with the first phase and Otsu's method.

and discovering the global manifold structure through the semisupervised learning and labelling.

3.2. Validation of the Two-Phase Saliency. This paper applies the image boundary and some candidate salient nodes, respectively, to generate the label indicator vectors for graph labelling based saliency detection. We then compare the performance of the proposed approach for each phase. Figure 5 demonstrates that the second phase using the strong candidate foreground superpixels generated by Otsu's method further enhances the performance of the first phase just with the image boundary ones.

3.3. Comparison with the State of the Art on Some Datasets. To verify the effectiveness of proposed saliency detection method through linear neighbors on some datasets, we compare with the most recently state-of-the-art approaches (PCA [28], SF [29], FT [26], RC [30], and HS [31]), some graph-based methods (GS_SP [15], MR [14], and GB [9]), and two traditional ones (IT [32] and MZ [33]). We select these methods for their varieties in design of the saliency models or the description of saliency. In the experiment, we use the implementation from Achanta et al. [26] for FT, IT, GB, and MZ. For RC, PCA, HS, and MR, we run the authors' public codes. For SF and GS_SP, we directly use the provided saliency maps by authors. Figure 6 shows the comparison results of various saliency methods on the ASD datasets and demonstrates the performance of our method.

For the SOD dataset, we compare with the PCA, HS, GS_SP, RC, and MR methods. Example results of PR curves and the precision, recall, and F-measure values achieved from the adaptive threshold are illustrated in Figure 7, respectively. We note that when measured with the fixed threshold method, our PR curve cannot perform well compared to some models at high recall rate (lower threshold). But as the threshold rises, the proposed method works best when compared with the methods mentioned in Figure 7(a), which means our method can highlight the whole object region uniformly. The reason is that the calculated pixel values of the saliency maps in our method are distributed within a fixed interval, which occupies a small range in the gray space. When the threshold is smaller than certain value, the segmented saliency maps cannot represent all the salient objects in the images, which results in the lower precision rate.

Results of visual comparison on the two datasets are shown in Figure 8.

3.4. Experimental Settings and Run Time. The proposed method is applied on the ASD and SOD datasets based on a machine with Intel Core i5-4590 3.30 GHz CPU and 8 GB RAM, and we implement the saliency model by using the Matlab language. Specifically, our method spends 0.201 s, 0.515 s, and 0.342 s on superpixel generation, graph construction, and saliency map computation, respectively, for each image on the ASD dataset, and the run time of superpixel generation is estimated by segmenting each image into 200 superpixels.

4. Conclusion

Recently, graph-based salient regions detection methods have applied the Gaussian weight function to measure the pairwise neighboring relationships between different image elements, which is parameter dependent and sensitive. To solve the problem, we propose the graph-based saliency detection model through linear neighbors, which means that each node is measured by a linear combination of

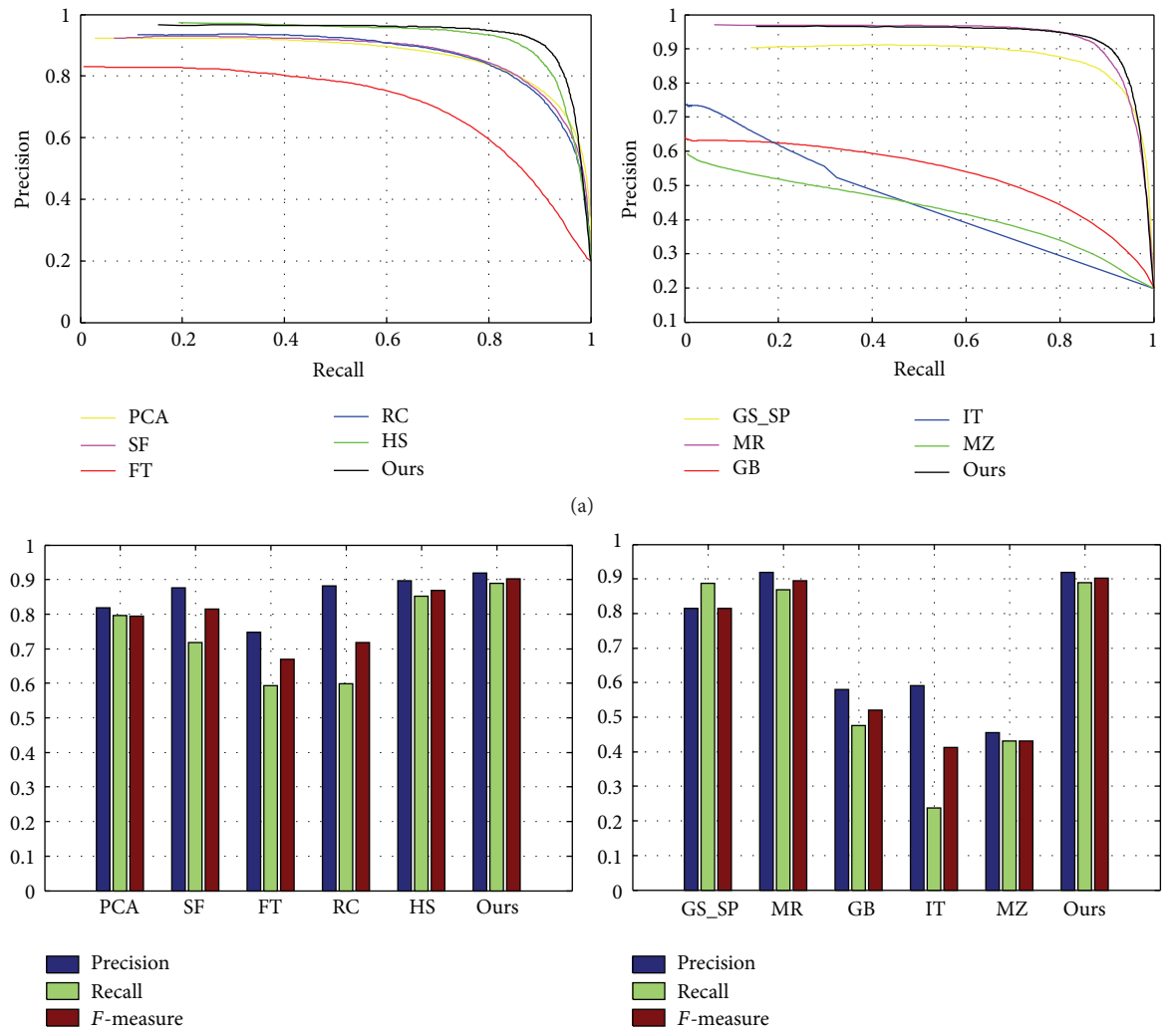


Figure 6: The comparison results on the ASD dataset. ((a) and (b)) The PR curves and F-measure values for the comprehensive evaluation with the state-of-the-art saliency models on the ASD dataset. We note that the proposed model shows the advantage of detecting salient regions when compared with other methods.

all its neighborhoods around to further represent the local information in the image. The saliency is defined as the process of semisupervised learning to label other nodes from partially selected seeds (labeled nodes) in the whole graph, which considers the global data structure hidden in the image for labelling. As a result, both the local grouping information and the global intrinsic structure of the cluster assumption are fully captured with the graph construction and labelling to learn the pop-out nodes in our linear neighborhood relationship based saliency detection method. In addition, our LNR graph has presented the reliability and stability of detecting salient regions, which does not need the predefined parameter for optimal model selection. We then evaluate the proposed model on some benchmark datasets in this paper. The experimental results indicate the effectiveness of our graph-based saliency detection method through linear neighborhoods when compared with some other state-of-the-art approaches.

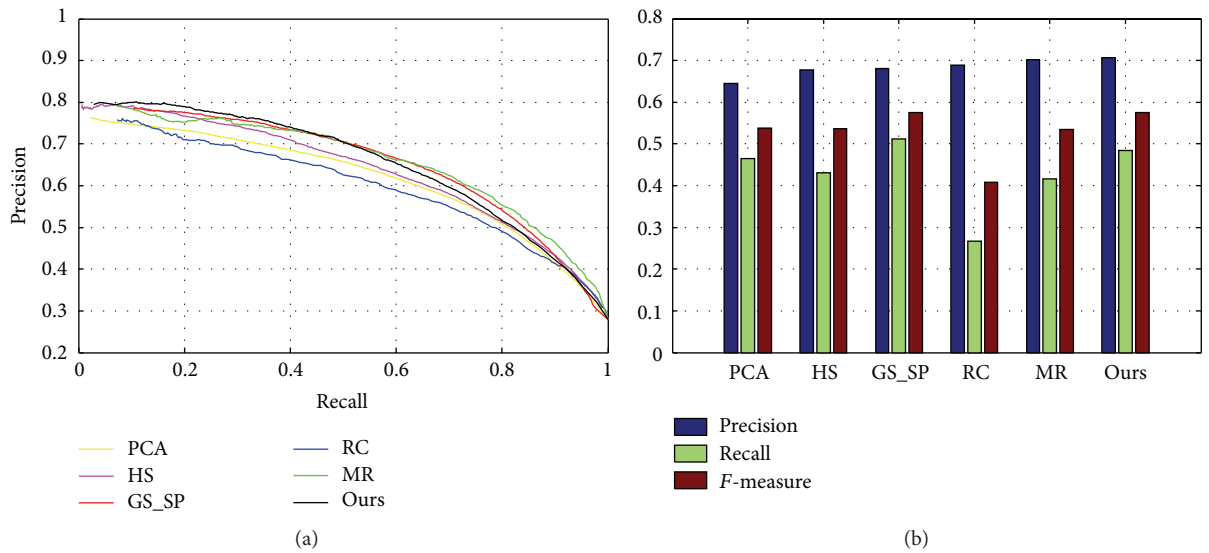
It should be noted that the proposed method can be further improved if more strong labels are provided. Our future work will focus on the choosing of the seeds.

Competing Interests

The authors declare that they have no competing interests.

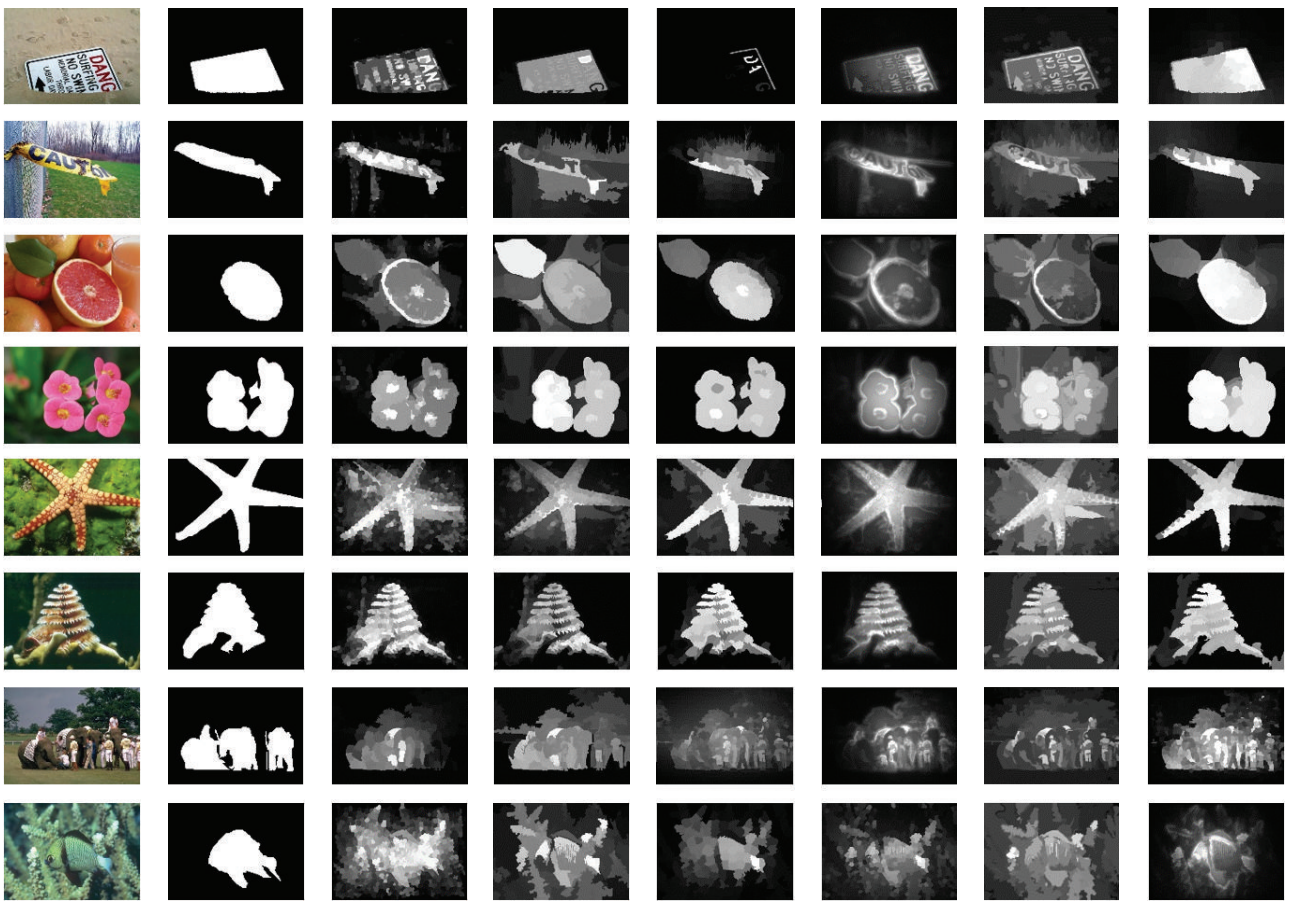
Acknowledgments

This research was supported by "National Natural Science Foundation of China" (no. 61272523, no. 61572103), "the National Key Project of Science and Technology of China"



(a) (b)

FIGURE 7: The comparison results on the SOD dataset. ((a) and (b)) The PR curves and F-measure values, respectively, for the comprehensive evaluation with the state-of-the-art saliency models on the SOD dataset. We note that our model performs better than other methods on this challenging dataset.



(a) Image (b) Ground truth (c) GS_SP (d) HS (e) MR (f) PCA (g) RC (h) Ours

FIGURE 8: Visual results of the saliency maps generated by different methods on the two datasets.

(no. 2011ZX05039-003-4), and "the Fundamental Research Funds for the Central Universities" (no. DUT15QY33).

References

[1] C. Kanan and G. Cottrell, "Robust classification of objects, faces, and flowers using natural image statistics," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2472–2479, IEEE, San Francisco, Calif, USA, June 2010.

[2] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II37–II44, July 2004.

[3] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1304–1318, 2004.

[4] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915–1926, 2012.

[5] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2Photo: internet image montage," *ACM Transactions on Graphics*, vol. 28, no. 5, p. 124, 2009.

[6] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2232–2239, IEEE, Kyoto, Japan, October 2009.

[7] J. Wang, J. Sun, L. Quan, X. Tang, and H.-Y. Shum, "Picture collage," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 347–354, IEEE, June 2006.

[8] L. da Fontoura Costa, "Visual saliency and attention as random walks on complex networks," http://arxiv.org/abs/physics/0603025.

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

[10] L. Wang, J. Xue, N. Zheng, and G. Hua, "Automatic salient object extraction with contextual cue," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 105–112, Barcelona, Spain, November 2011.

[11] Y. Lu, W. Zhang, H. Lu, and X. Xue, "Salient object detection using concavity context," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 233–240, IEEE, Barcelona, Spain, November 2011.

[12] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.

[13] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs for salient object detection in images," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3232–3242, 2010.

[14] C. Yang, L. Zhang, H. 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, IEEE, Portland, Ore, USA, June 2013.

[15] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Computer Vision—ECCV 2012*, pp. 29–42, Springer, Berlin, Germany, 2012.

[16] D. Zhou, O. Bousquet, T. N. Lal et al., "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, no. 16, pp. 321–328, 2004.

[17] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *Proceedings of the 16th Annual Neural Information Processing Systems Conference (NIPS '02)*, pp. 585–592, Vancouver, Canada, December 2002.

[18] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.

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

[20] 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.

[21] D. Zhou, J. Weston, A. Gretton et al., "Ranking on data manifolds," in *Advances in Neural Information Processing Systems 16*, pp. 169–176, MIT Press, 2004.

[22] 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 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2083–2090, IEEE, Portland, Ore, USA, June 2013.

[23] M. Xu and H. Zhang, "Saliency detection with color contrast based on boundary information and neighbors," *The Visual Computer*, vol. 31, no. 3, pp. 355–364, 2015.

[24] J. Zhang and S. Sclaroff, "Saliency detection: a boolean map approach," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 153–160, IEEE, Sydney, Australia, December 2013.

[25] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.

[26] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1597–1604, Miami, Fla, USA, June 2009.

[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] R. Margolin, A. Tal, and L. Zelnik-Manor, "What makes a patch distinct?" in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 1139–1146, Portland, Ore, USA, June 2013.

[29] 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, IEEE, Providence, RI, USA, June 2012.

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

[31] 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.

[32] 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.

[33] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proceedings of the 11th ACM International Conference on Multimedia (MM '03)*, pp. 374–381, November 2003.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 3460281, 10 pages
http://dx.doi.org/10.1155/2016/3460281

Research Article

Sparse Representation Based Binary Hypothesis Model for Hyperspectral Image Classification

Yidong Tang, Shucai Huang, and Aijun Xue

School of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China

Correspondence should be addressed to Yidong Tang; 18109267859@163.com

Received 9 March 2016; Revised 9 May 2016; Accepted 17 May 2016

Academic Editor: Wonjun Kim

The sparse representation based classifier (SRC) and its kernel version (KSRC) have been employed for hyperspectral image (HSI) classification. However, the state-of-the-art SRC often aims at extended surface objects with linear mixture in smooth scene and assumes that the number of classes is given. Considering the small target with complex background, a sparse representation based binary hypothesis (SRBBH) model is established in this paper. In this model, a query pixel is represented in two ways, which are, respectively, by background dictionary and by union dictionary. The background dictionary is composed of samples selected from the local dual concentric window centered at the query pixel. Thus, for each pixel the classification issue becomes an adaptive multiclass classification problem, where only the number of desired classes is required. Furthermore, the kernel method is employed to improve the interclass separability. In kernel space, the coding vector is obtained by using kernel-based orthogonal matching pursuit (KOMP) algorithm. Then the query pixel can be labeled by the characteristics of the coding vectors. Instead of directly using the reconstruction residuals, the different impacts the background dictionary and union dictionary have on reconstruction are used for validation and classification. It enhances the discrimination and hence improves the performance.

1. Introduction

The technology for artificial target recognition in nature background is important to military science and civil autocontrol. Hyperspectral remote sensor captures digital images in hundreds of narrow spectral bands, which span the visible to infrared spectrum. The high spectral resolution of the data provides an invaluable source of information regarding the physical nature of the different materials and strengthens the capability to identify structures and objects in the image scene. As a result, it makes detecting and classifying the target at the same time possible, that is, integrated detection and classification. However, such a large number of spectral channels imply the high dimensionality of the data and bring challenge to image analysis. Most of the common technologies designed for the analysis of grey level, color, or multispectral images are not applicable to hyperspectral images.

One of the most important applications of HSI is classification. Different materials usually reflect electromagnetic energy differently at specific wavelengths. This enables discrimination of materials based on the spectral characteristics. Various techniques have been developed for HSI classification. SVM [1–3] is a powerful tool to solve supervised classification problem in remote sensing image scene and performs pretty well. Variations of SVM-based algorithms have also been proposed to improve the classification accuracy [4, 5]. In the context of supervised classification, such as SVM, an additional problem is the so-called Hughes phenomenon that occurs when the training set does not have enough samples to ensure a reliable estimation of the classifier parameters. It is hard to find the separating hyperplane between two classes with very limited reference data while the number of spectral channels is usually very large in hyperspectral image scene.

The sparse representation [6] has recently been applied to hyperspectral detection and classification [7–9] relying on the observation that the pixels belonging to same class approximately lie in same low-dimensional subspace. Thus, a query pixel can be sparsely represented by a few training samples (atoms) from dictionary, and the associated sparse

representation vector will implicitly encode the class information. While SVM is a binary classifier (multiclass SVM requires one-against-one or one-against-all strategy [10]), the SRC is a multiclass classifier, which is from a reconstruction point of view. The SRC can be regarded as generalized model and often results in good performance. However, the number of classes present in hyperspectral image scene has to be known to structure the dictionary and calculate the residuals when employing the sparsity model as [11, 12]. In addition, the sparsity model may not be suitable for background pixels due to the interaction between the target and background sparse vectors.

In this paper, a sparse representation based binary hypothesis (SRBBH) model, which can be regarded as a discriminative and semisupervised model, is proposed to strengthen the performance of sparsity model. Only the number of desired classes, which is accessible in most practical cases, is required for SRBBH model. Thus, a target pixel is approximately represented by union dictionary consisting of both corresponding target training samples and background training samples while a background pixel can be approximately represented just by background dictionary. Different from SRC, different impacts the background dictionary and union dictionary have on reconstruction, instead of residuals themselves, are used for validation and classification in SRBBH model. This scheme enhances the discriminative power of different subspaces and then improves the classification performance. However, when the data structure is complex and the problem becomes nonlinear, the SRBBH model based classifier (SRBBHC) may not be competent any more. With implicitly exploiting the higher order structure of the given data, the kernel algorithm obtains significant performance improvement. Therefore, the kernel SRBBHC (KSRBBHC) is developed to project the data into high-dimensional feature space in which the data becomes linearly separable. Taking the projected data into consideration, KSRBBHC intends to separately represent the desire class and undesired class in corresponding high-dimensional feature space and makes classification performance better.

The rest of the paper is organized as follows. Section 2 briefly reviews the conventional SRC and its kernel version. Section 3 proposes the SRBBHC and its kernel version for HSI. The effectiveness of proposed SRBBHC and KSRBBHC is demonstrated by experimental results in Section 4. Finally, conclusions are drawn in Section 5.

2. Sparse Representation Based Classification

2.1. SRC. In sparsity model, it is assumed that the spectral signatures of pixels belonging to same class approximately lie in same low-dimensional subspace. A query pixel is given $\mathbf{y} \in \mathbb{R}^B$, where B is the number of bands. Then the linear representation of $\mathbf{y}$ can be written in terms of all training samples as

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_M] \in \mathbb{R}^{B\times N}$ is a structured dictionary whose columns are N training samples of all M classes and $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is the sparse coefficient vector. $\mathbf{x}$ can be recovered by solving

$$\begin{aligned} \widehat{\mathbf{x}} = \arg\min \quad & \left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2 \\ \text{s.t} \quad & \|\mathbf{x}\|_0 \le K_0, \end{aligned} \tag{2}$$

where $\|\cdot\|_0$ denotes the l_0-norm, which is defined as the number of nonzero entries in the vector, and K_0 is a preset upper bound on sparsity level. The problem in (2) is a NP-hard problem, which can be approximately solved by greedy algorithms, such as orthogonal matching pursuit (OMP) [13] and subspace pursuit (SP) [14], or relaxed to convex programming [15]. In this paper, the OMP algorithm is exploited to generate sparse coefficient vector. The OMP algorithm augments the support set by one index per iteration until K_0 atoms are selected or the approximation error is within a preset threshold.

Once the sparse coefficient vector is obtained, the class label of $\mathbf{y}$ is determined by the minimal residual between $\mathbf{y}$ and its approximation from each class of subdictionary:

$$\text{Class}\left(\mathbf{y}\right) = \arg\min_{i=1,\ldots,M} \left\|\mathbf{y} - \mathbf{A}_i\widehat{\mathbf{x}}_i\right\|_2, \tag{3}$$

where $\mathbf{A}_i = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, \ldots, \mathbf{a}_{i,n_i}] \in \mathbb{R}^{B\times n_i}$ is the dataset of training sample from class i and $\widehat{\mathbf{x}}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}]^T$ is the coefficient vector associated with $\mathbf{A}_i$.

2.2. KSRC. Kernel methods outperform the classical linear algorithms by implicitly exploiting the nonlinear information of given data [16]. It relies on the observation that a pixel in kernel-induced feature space can be linearly represented in terms of the training samples in same space [9]. Let $\mathbf{y} \in \mathbb{R}^B$ be the data point of interest and let $\boldsymbol{\phi}(\mathbf{y})$ be its representation in kernel-induced feature space. Similar to the SRC, the linear representation of $\boldsymbol{\phi}(\mathbf{y})$ in terms of training samples in kernel-induced feature space can be formulated as

$$\begin{aligned} \boldsymbol{\phi}\left(\mathbf{y}\right) &= \underbrace{\left[\boldsymbol{\phi}\left(\mathbf{A}_1\right), \ldots, \boldsymbol{\phi}\left(\mathbf{A}_M\right)\right]}_{\mathbf{A}^{\phi}} \underbrace{\left[\boldsymbol{\phi}\left(\mathbf{x}_1\right)^T, \ldots, \boldsymbol{\phi}\left(\mathbf{x}_M\right)^T\right]^T}_{\mathbf{x}^{\phi}} \\ &= \mathbf{A}^{\phi}\mathbf{x}^{\phi}, \end{aligned} \tag{4}$$

where $\mathbf{A}^{\phi}$ is the training dictionary in kernel-induced feature space and $\mathbf{x}^{\phi}$ is the coefficient vector. The vector $\mathbf{x}^{\phi}$ can be recovered by solving

$$\begin{aligned} \widehat{\mathbf{x}}^{\phi} = \arg\min \quad & \left\|\mathbf{y} - \mathbf{A}^{\phi}\mathbf{x}^{\phi}\right\|_2 \\ \text{s.t} \quad & \left\|\mathbf{x}^{\phi}\right\|_0 \le K_0. \end{aligned} \tag{5}$$

Problem (5) can be approximately solved by kernelized sparse recovery algorithms, such as the kernelized orthogonal matching pursuit (KOMP) and kernelized subspace pursuit (KSP). Implementation details of the KOMP and KSP can be found in [17]. In this paper, the KOMP is used to solve the problem with RBF kernel function. To avoid directly evaluating the inner product in high-dimensional feature

space, the kernel-based learning algorithm uses an effective kernel trick to implement dot products in the feature space without knowing the exact mapping function ϕ.

Once the sparse vector $\hat{\mathbf{x}}^{\phi}$ is obtained, the residual associated with ith subject in the feature space is then computed by

$$\begin{aligned} r_i(\mathbf{y}) &= \left\| \boldsymbol{\phi}(\mathbf{y}) - \mathbf{A}_i^{\phi} \hat{\mathbf{x}}_i^{\phi} \right\| \\ &= \sqrt{\left(k(\mathbf{y},\mathbf{y}) - 2\left(\hat{\mathbf{x}}_i^{\phi}\right)^T k\left(\mathbf{A}_i^{\phi}, \mathbf{y}\right) + \left(\hat{\mathbf{x}}_i^{\phi}\right)^T k\left(\mathbf{A}_i^{\phi}, \mathbf{A}_i^{\phi}\right) \hat{\mathbf{x}}_i^{\phi} \right)} \\ &= \sqrt{\left(k(\mathbf{y},\mathbf{y}) - 2\left(\hat{\mathbf{x}}_i^{\phi}\right)^T \left(\mathbf{k}_{\mathbf{A},\mathbf{y}}\right)_i + \left(\hat{\mathbf{x}}_i^{\phi}\right)^T \left(\mathbf{K}_{\mathbf{A}}\right)_{i,i} \hat{\mathbf{x}}_i^{\phi} \right)}, \end{aligned} \tag{6}$$

where $\mathbf{k}_{\mathbf{A},\mathbf{y}}$ and $\mathbf{K}_{\mathbf{A}}$ are, respectively, kernel tricks of target dictionary with the query pixel and itself. The class label of query pixel $\mathbf{y}$ is then determined as

$$\text{Class}(\mathbf{y}) = \arg \min_{i=1,\ldots,M} r_i(\mathbf{y}). \tag{7}$$

Though SRC and KSRC have been proved to be powerful approaches as shown in [8], the main idea of the SRC and KSRC is only appropriate for extended surface target classification such as plantation and geology. In this case, the pixels in a large neighborhood are likely to consist of similar materials, and different subjects are next to each other without undesired subject (background) existing between them. As a result, the spectral mixture only occurs along the boundary, leading to the fact that most of the target pixels are observed without corruption brought by background and most of the training samples selected from dataset for dictionary are pure. However, this probably would never happen to small size target whose spectrum is almost mixed with background. Thus, it may be unreliable for conventional SRC and KSRC to use the residual information for validation and classification. On the other hand, it is assumed that we have hold all class of subjects present in the hyperspectral image scene, including the number of classes and corresponding training samples. In other words, the SRC and KSRC cannot distinguish between small targets in hyperspectral image scene for generally lacking training samples of undesired class.

3. SRBBHC and the Kernelized Version

In this section, we introduce the proposed SRBBH model based classification algorithm for HSI, which utilizes the binary hypothesis for quality validation as well as the reconstruction residuals by the two hypotheses for classification. Moreover, a kernelized version of the proposed classifier is also introduced for nonlinear classification in a high-dimensional feature space.

3.1. SRBBHC. When employing SRC and KSRC for HSI, it is assumed that the number of classes present in the image scene M is known. However, it is actually difficult to know this information due to scene complexity. In many practical situations, only the number of desired classes is available. Fortunately, considering the regions of interest, such as artificial target in nature background, what we wonder is the class label of desired subjects, but not the class label of all kinds of subjects. In other words, we need to detect and then reject the undesired query sample before classification. On the other hand, different from extended surface target, the pixels of target with small size are almost mixed with background spectrum. Detecting and classifying desired subjects from the mixed pixel are generally difficult, especially when the background spectrum has a close or even larger abundance than target. In addition, although the background and target training samples have distinct spectral signatures and lie in two different subspaces, the two subspaces are usually not orthogonal, due to spectral variation [18]. In such case, the reconstruction residual via corresponding target training samples may be on the contrary larger than that via background training samples, which will lead to mistake target for background. Thus, it is no longer sufficient to directly use the reconstruction residuals for validation and classification. The SRBBHC solves these problems by utilizing a binary hypothesis model with more reasonable dictionaries, where the query pixel is, respectively, modeled with background dictionary under the null hypothesis and with union dictionary under the alternative hypothesis. And then the binary hypothesis is used for validation. In SRBBHC and its kernelized version, the different impacts the background dictionary and union dictionary have on reconstruction, instead of residuals themselves, are used for validation and classification. In a sense, the SRBBHC can be viewed as a joint target detection and classification scheme, which firstly detects the valid samples and then classifies them.

In detail, denote the union dictionary consisting of both target training samples from class i and background training samples as $\mathbf{A}_i^u = [\mathbf{A}_i^t, \mathbf{A}^b]$, $i = 1, \ldots, M$, where $\mathbf{A}_i^t$ is target subdictionary associated with class i and $\mathbf{A}^b$ is background subdictionary. If $\mathbf{y}$ belongs to undesired class, the spectrum lies in a low-dimensional subspace spanned by the background training samples. As a result, the residuals of different union dictionary $\mathbf{A}_i^u$ are similar to the residual of background subdictionary $\mathbf{A}^b$, that is, hypothesis H_0. On the other hand, if $\mathbf{y}$ belongs to class i, the union dictionary $\mathbf{A}_i^u$ will give better representation, leading to smaller residual than background subdictionary, that is, hypothesis H_1. The binary hypothesis for quality validation is modeled as follows:

$$\begin{aligned} H_0 &: \left\| \mathbf{y} - \mathbf{A}^b \boldsymbol{\gamma} \right\| \\ &\quad \approx \left\| \mathbf{y} - \mathbf{A}_i^u \boldsymbol{\beta}_i \right\|_{i=1,2,\ldots,M}, \text{ undesired class} \\ H_1 &: \left\| \mathbf{y} - \mathbf{A}^b \boldsymbol{\gamma} \right\| > \left\| \mathbf{y} - \mathbf{A}_i^u \boldsymbol{\beta}_i \right\|_{i=1,2,\ldots,M}, \text{ desired class}, \end{aligned} \tag{8}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}_i$ are coefficient vectors associated with $\mathbf{A}^b$ and $\mathbf{A}_i^u$, respectively. In other words, the problem is reformed into local binary classification problem, where the binary hypothesis is used to decide if the test pixel is a valid sample from one of the classes we desire.

According to the sparse coding theory [15], the coefficient vectors can be recovered by solving following minimization problem with the same sparsity level:

$$\begin{aligned} \widehat{\boldsymbol{\gamma}} &= \arg\min \quad \left\|\mathbf{x} - \mathbf{A}^b\boldsymbol{\gamma}\right\|_2 \\ &\text{s.t} \quad \left\|\boldsymbol{\gamma}\right\|_0 \le K_0, \\ \widehat{\boldsymbol{\beta}}_m &= \arg\min \quad \left\|\mathbf{x} - \mathbf{A}_i^u\boldsymbol{\beta}_i\right\|_2 \\ &\text{s.t} \quad \left\|\boldsymbol{\beta}_i\right\|_0 \le K_0. \end{aligned} \tag{9}$$

Once the sparse coefficient vectors are obtained, the semantic information can be directly extracted from the coefficient vectors. The residuals of background subdictionary and different union dictionaries are calculated as

$$\begin{aligned} r_0 &= \left\|\mathbf{y} - \mathbf{A}^b\widehat{\boldsymbol{\gamma}}\right\|, \\ r_i &= \left\|\mathbf{y} - \mathbf{A}_i^u\widehat{\boldsymbol{\beta}}_i\right\|_{i=1,2,\ldots,M}. \end{aligned} \tag{10}$$

If we decide the given $\mathbf{y}$ as a valid sample belongs to class i, the union dictionary $\mathbf{A}_i^u$ will also give much better representation than the other union dictionaries $\mathbf{A}_{j,j\neq i}^u$, leading to larger difference between r_i and r_0. Then, $\mathbf{y}$ will be labeled to the class with greatest difference between r_i and r_0. Defining a vector $\mathbf{R} = (r_0 - r_1, r_0 - r_2, \ldots, r_0 - r_M)$, the outputs of integrated detection and classification decision are then made by

$$\begin{aligned} \text{Detector}\left(\mathbf{y}\right) &= \max \quad (\mathbf{R}) \\ \text{Class}\left(\mathbf{y}\right) &= \arg\max_{i=1,\ldots,M} \quad (r_0 - r_i) \\ &\text{s.t} \quad \text{Detector}\left(\mathbf{y}\right) \ge \delta, \end{aligned} \tag{11}$$

where δ is a threshold used for validation. When $\text{Detector}\left(\mathbf{y}\right) < \delta$, the query pixel will be labeled as undesired class, that is, background. The threshold δ makes important effects on validation and hence classification. However, in this study, the threshold δ is determined experimentally due to lack of parameter analysis theory. In our future work, we will investigate how to automatically choose appropriate δ for different test datasets.

Considering the size of the desired subjects, the background dictionary is generated locally for each query pixel through a dual concentric window centered at query pixel. Only the samples in the outer region are involved in $\mathbf{A}^b$. As a result, the background dictionary is constructed adaptively for each pixel and captures the background spectral signature of the query pixel better. It is important to note that same sparsity level must be adopted for each union dictionary $\mathbf{A}_i^u$ to make sure of the comparability of residuals of two hypotheses in SRBBHC.

3.2. KSRBBHC. For a hyperspectral image scene, the spectral mixing may be nonlinear due to the complex imaging condition in many practical situations [19]. As a result, the data structure may become complex and the problem becomes no longer linearly separable. In such case, the linear SRBBHC is not competent any more. Fortunately, kernel methods can project the linearly nonseparable data into a high-dimensional feature space in which those data become more separable. Here we extend the proposed SRBBHC into a kernel vision, referred to as KSRBBHC.

Similar to the KSRC, suppose that $\boldsymbol{\phi}(\mathbf{y})$ is the representation of query pixel in the high-dimensional feature space; the SRBBH model becomes

$$\begin{aligned} H_0 &: \left\|\boldsymbol{\phi}\left(\mathbf{y}\right) - \mathbf{A}^{b\phi}\boldsymbol{\gamma}^{\phi}\right\| \\ &\quad \approx \left\|\boldsymbol{\phi}\left(\mathbf{y}\right) - \mathbf{A}_i^{u\phi}\boldsymbol{\beta}_i^{\phi}\right\|_{i=1,2,\ldots,M}, \ \text{undesired class} \\ H_1 &: \left\|\boldsymbol{\phi}\left(\mathbf{y}\right) - \mathbf{A}^{b\phi}\boldsymbol{\gamma}^{\phi}\right\| \\ &\quad > \left\|\boldsymbol{\phi}\left(\mathbf{y}\right) - \mathbf{A}_i^{u\phi}\boldsymbol{\beta}_i^{\phi}\right\|_{i=1,2,\ldots,M}, \ \text{desired class}, \end{aligned} \tag{12}$$

where $\mathbf{A}^{b\phi}$ and $\mathbf{A}_i^{u\phi}$, respectively, result from mapping $\mathbf{A}^b$ and $\mathbf{A}_i^{\phi}$ into kernel-induced feature space by mapping function $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}^{\phi}$ and $\boldsymbol{\beta}_i^{\phi}$ are coefficient vectors, respectively, associated with $\mathbf{A}^{b\phi}$ and $\mathbf{A}_i^{u\phi}$.

Employing the RBF kernel function, the residuals of the kernel SRBBHC in terms of background dictionary and different union dictionaries are, respectively, computed by

$$\begin{aligned} r_0^{\phi} &= \left\|\boldsymbol{\phi}\left(\mathbf{y}\right) - \mathbf{A}^{b\phi}\widehat{\boldsymbol{\gamma}}^{\phi}\right\| = \sqrt{\left(k\left(\mathbf{y},\mathbf{y}\right) - 2\left(\widehat{\boldsymbol{\gamma}}^{\phi}\right)^T k\left(\mathbf{A}^{b\phi},\mathbf{y}\right) + \left(\widehat{\boldsymbol{\gamma}}^{\phi}\right)^T k\left(\mathbf{A}^{b\phi},\mathbf{A}^{b\phi}\right)\widehat{\boldsymbol{\gamma}}^{\phi}\right)}, \\ r_i^{\phi} &= \left\|\boldsymbol{\phi}\left(\mathbf{y}\right) - \mathbf{A}_i^{u\phi}\widehat{\boldsymbol{\beta}}_i^{\phi}\right\| = \sqrt{\left(k\left(\mathbf{y},\mathbf{y}\right) - 2\left(\widehat{\boldsymbol{\beta}}_i^{\phi}\right)^T k\left(\mathbf{A}_i^{u\phi},\mathbf{y}\right) + \left(\widehat{\boldsymbol{\beta}}_i^{\phi}\right)^T k\left(\mathbf{A}_i^{u\phi},\mathbf{A}_i^{u\phi}\right)\widehat{\boldsymbol{\beta}}_i^{\phi}\right)}_{i=1,2,\ldots,M}, \end{aligned} \tag{13}$$

where $\widehat{\boldsymbol{\gamma}}^{\phi}$ and $\widehat{\boldsymbol{\beta}}_i^{\phi}$ are the estimation for $\boldsymbol{\gamma}^{\phi}$ and $\boldsymbol{\beta}_i^{\phi}$ by KOMP.

Similarly, $\mathbf{y}$ will be labeled to the class with greatest difference between r_i^{ϕ} and r_0^{ϕ}. Defining a new vector $\mathbf{R}^{\phi} = (r_0^{\phi} - r_1^{\phi}, r_0^{\phi} - r_2^{\phi}, \ldots, r_0^{\phi} - r_M^{\phi})$, the outputs of integrated detection and classification decision are then made by

$$\begin{aligned} \text{Detector}\left(\mathbf{y}\right) &= \max \quad \left(\mathbf{R}^{\phi}\right) \\ \text{Class}\left(\mathbf{y}\right) &= \arg\max_{i=1,\ldots,M} \quad \left(r_0^{\phi} - r_i^{\phi}\right) \\ &\text{s.t} \quad \text{Detector}\left(\mathbf{y}\right) \ge \delta^{\phi}, \end{aligned} \tag{14}$$

where δ^{ϕ} is also a threshold used for validation. When Detector $(\mathbf{y}) < \delta^{\phi}$, the query pixel will be labeled as undesired class, that is, background. Similar to SRBBHC, although δ^{ϕ} is important for validation and classification, it is also determined experimentally.

4. Experimental Results

In this section, the classification performance of KSRBBHC is evaluated and compared to the other four classifiers (SVMC, SRC, KSRC, and SRBBHC), and RBF kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$ is used for KSRC and KSRBBHC. The average recognition rate (ARR) and overall recognition rate (ORR) are suggested as performance parameters. The effectiveness of the proposed algorithms is evaluated with two datasets: a synthetic dataset ROI-I and a real dataset ROI-II, as shown in Figures 4 and 5.

The ROI-I is constructed by implanting five classes of targets, which are, respectively, *artificiality*, *clay*, *tree*, *plane*, and *grass* with a background scene size of 100 × 100 pixels collected by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) from San Diego, CA, USA. The image has 224 spectral channels (189 available) in wavelengths ranging from 370 to 2510 nm. In detail, each class of target is linearly mixed with background by varying abundance from 0.3 to 0.7 with step 0.1 in five small neighborhoods of size 2 × 2 pixels. In other words, the image contains 25 desired subjects occupying 100 pixels. 4 unmixed samples per class are randomly chosen for training, and the dual window sizes ω_{in}, ω_{out} are set as 3 × 3 and 9 × 9 according to the size of desired target. We compute the ARRs and ORRs of KSRC and KSRBBHC for ROI-I with varying kernel parameter σ and sparsity level K_0, as shown in Figure 1. One can see from Figure 1 that the KSRC and KSRBBHC both are sensitive to σ, while K_0 plays a nearly negligible role for the two kernel vision algorithms when σ is fixed, especially when σ is fixed as 0.001, 0.01, and 0.1. When σ is fixed as 10, the ARRs and ORRs of KSRC and KSRBBHC both remain at a relatively high level and change smoothly with K_0. As a result, the kernel parameter σ is set as 10 for the KSRC and KSRBBHC.

The ROI-II with the size of 100 × 100 pixels is a region directly taken from the real AVIRIS image, San Diego. It contains 3 classes of desired subjects occupying 317 pixels and undesired subjects occupying 9683 pixels. For each class, around 10% of the labeled samples are chosen randomly for training. The dual window sizes ω_{in}, ω_{out} are set as 7 × 7 and 11 × 11 manually according to our previous work [20]. The ARRs and ORRs of KSRC and KSRBBHC with varying σ and K_0 for ROI-II are shown in Figure 2. In detail, as shown in Figure 2(a), when σ is very small $(0.001, 0.01)$ the KSRC always has a poor performance. When σ is relatively large, in general, the performance of KSRC increases with K_0 smoothly, until the parameter reaches a certain level. Only when σ is too large ($\sigma = 100$), the performance of KSRC fluctuates with the value of K_0. For KSRBBHC, as shown in Figure 2(b), the performance remains at high level at all sparsity level when σ is small (0.001, 0.01, and 0.1). And the performance increases with K_0 smoothly when σ is fixed as 1, until the parameter reaches a certain level. However, the performance decreases with K_0 when σ is large $(10, 100)$. In a word, the experiment results for ROI-II also show that the classification performance is more sensitive to kernel parameter rather than sparsity level. Unlike the same optimal kernel parameter setting in ROI-I, the kernel parameter σ is, respectively, set as 10 for the KSRC and 0.01 for KSRBBHC in ROI-II.

Figure 3 shows the classification performances of the four sparsity-based classifiers with varying K_0 when kernel parameter σ is, respectively, optimized for corresponding algorithm. For the ROI-I, as shown in Figure 3(a), in general, the experiment result shows that the classification performances of four algorithms increase with K_0, until the parameter reaches a certain level, and then the performances hold the line. As a result, the parameter K_0 is set as 8 to balance the performance and complexity for all sparsity-based classifiers in ROI-I. For the ROI-II, the classification performances of SRC, KSRC, and SRBBHC increase with K_0 when the parameter is smaller than 6; then the performances of SRC and KSRC keep increasing smoothly with K_0, whereas the performance of SRBBHC reduces gradually. The reason for this may be that only proper K_0 can lead a good discriminative performance of sparsity-based classifiers. In detail, for a very small K_0, the residuals of background dictionary and union dictionary are both large, which finally weakens the classification performance. When K_0 is too large, along with the nonorthogonality between desired target and background, the solution may be dense and result in a degraded discriminative power. Exactly as expected, for both ROI-I and ROI-II the performance of KSRBBHC reaches a good result with relatively small K_0 and remains at high level when K_0 increases (i.e., insensitive to K_0). The parameter K_0 is set as 6 to balance the performance and complexity for all sparsity-based classifiers in ROI-II.

After getting the optimal values of K_0 and σ, the experimental results of the five classifiers, averaged over five independent realizations, are determined via ORR and ARR, as shown in Tables 1 and 2. Because only the number of desired classes is known, it should be noted that the SVMC is designed as a local $M + 1$-class ($M = 5$ for ROI-I; $M = 3$ for ROI-II) classifier with one-against-all strategy, where the training samples from class $M + 1$ (background) are adaptively collected for each query pixel in local dual concentric window. One can observe from Tables 1 and 2 that both SRBBHC and KSRBBHC have significantly improved the classification accuracy. One can also see that the relationship of ORR and that of ARR between two classifiers are sometimes mismatched. The reason for this phenomenon may be that the total number of desired pixels is so small that even modest increase of desired target classification accuracy can lead to enormous increase on ARR but only small change on ORR. For example, for the KSRBBHC in Table 1, the classification accuracy for every single class is greater than or equal to that of SRBBHC, leading to a greater ARR. However, due to the small ratio but big absolute amount growth of classification on background, the ORR of SRBBHC is turning to greater than KSRBBHC. Considering the mismatch, the best one of the five independent results for each classifier

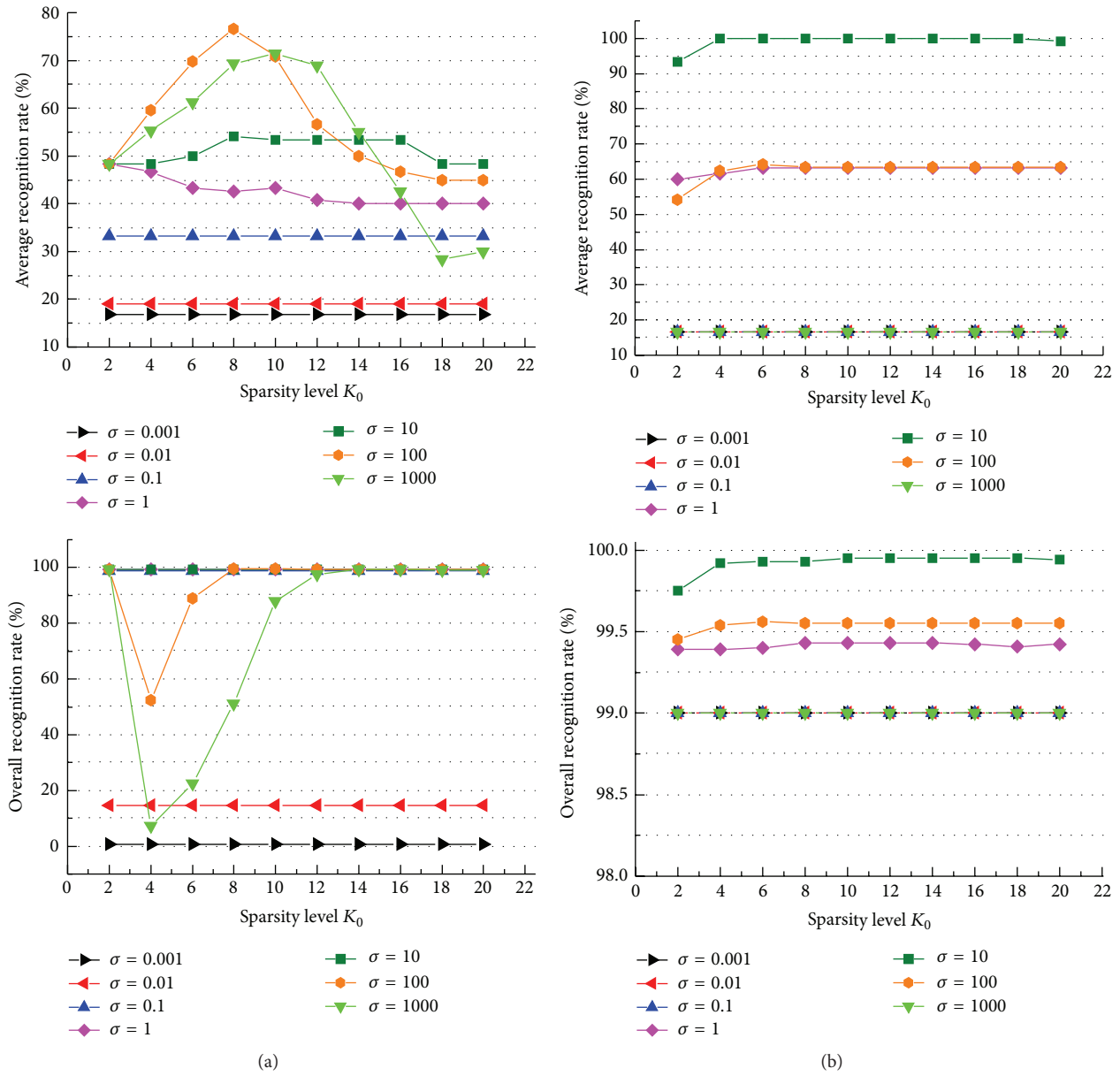


FIGURE 1: Effect of sparsity level K_0 and RBF kernel parameter σ on performance of two kernel-based classifiers: (a) KSRC; (b) KSRBBHC for ROI-I.

is chosen to estimate the performance intuitively, as shown in Figures 4 and 5. The corresponding algorithm and ORR are presented on the bottom of each map. One can clearly see that both binary hypothesis model based classifiers have significantly improved classification performance. The SRBBHC not only classified the desired target exactly but also avoid mistaking background to desired target effectively. Furthermore, the KSRBBHC outperforms all other classifiers. In detail, for ROI-I, because of the mixture of desired target and background, the SVMC, SRC, and KSRC mistake desired target to background to a great extent. Severely, the SRC mistakes abundant background pixels to desired target, while the KSRBBHC and SRBBHC yield a superior performance. For ROI-II, the SVMC barely classify the 2nd class of target correctly because of lack of enough training sample and the similarity between 2nd class of target and background. And the conventional sparsity-based classifiers (SRC and KSRC) lead to quite a number of misclassifications due to their weak discriminative power. In a word, one can observe from the experimental results that the SRBBH model based classifier (SRBBHC) offers better performance than some traditional techniques. Moreover, the kernelization of SRBBHC further improves the performance.

5. Conclusion

In this paper, a sparse representation based binary hypothesis model is proposed for target classification in HSI. And the

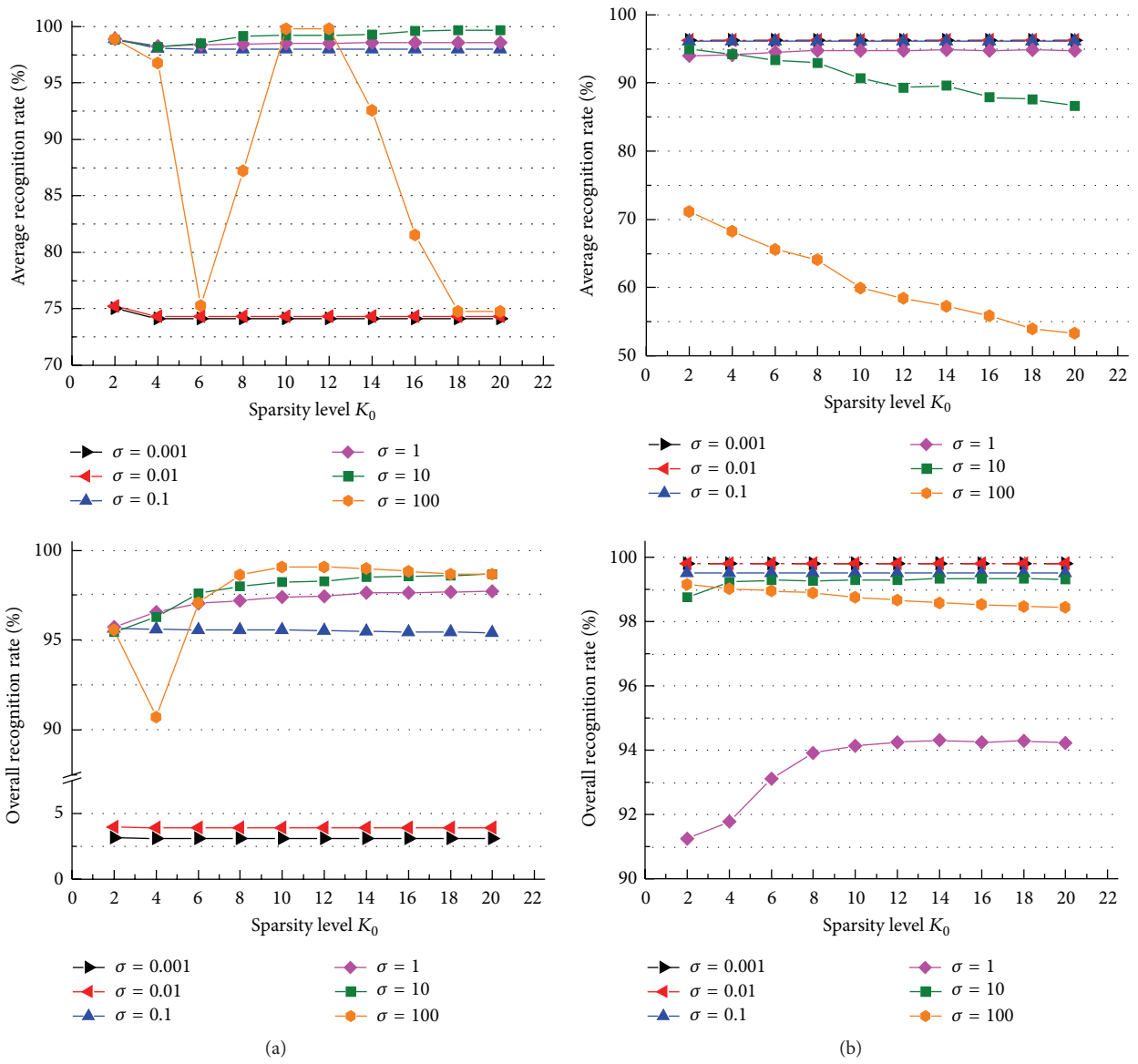


FIGURE 2: Effect of sparsity level K_0 and kernel parameter σ on classification performance of two kernel classifiers: (a) KSRC; (b) KSRBBHC for ROI-II.

TABLE 1: Classification accuracy for the ROI-I using 4 unmixed samples as training set.

Class	SVMC	SRC	KSRC	SRBBHC	KSRBBHC
1	19	73	59	**100**	**100**
2	0	55	36	**100**	**100**
3	64	3	36	**100**	**100**
4	35	54	48	**100**	**100**
5	80	56	61	**100**	**100**
Background	100	91.42	**100**	99.73	99.92
ORR	99.40	86.54	99.49	99.73	**99.92**
ARR	49.67	59.01	57.33	99.95	**99.98**
Time	172.96	**11.35**	15.28	33.03	72.71

TABLE 2: Classification accuracy for the ROI-II using around 10% labeled samples as training set.

Class	SVMC	SRC	KSRC	SRBBHC	KSRBBHC
1	99.64	99.82	**100**	89.01	91.62
2	5.38	**100**	**100**	92.31	94.13
3	**100**	98.65	98.60	93.24	93.22
Background	98.52	98.16	98.15	99.81	99.69
ORR	97.66	96.81	97.55	98.65	**99.07**
ARR	75.38	97.20	**98.05**	91.46	94.11
Time	471.45	23.91	20.15	**14.69**	30.91

corresponding algorithm (SRBBHC) based on this model is proposed for HSI classification. Furthermore, taking the

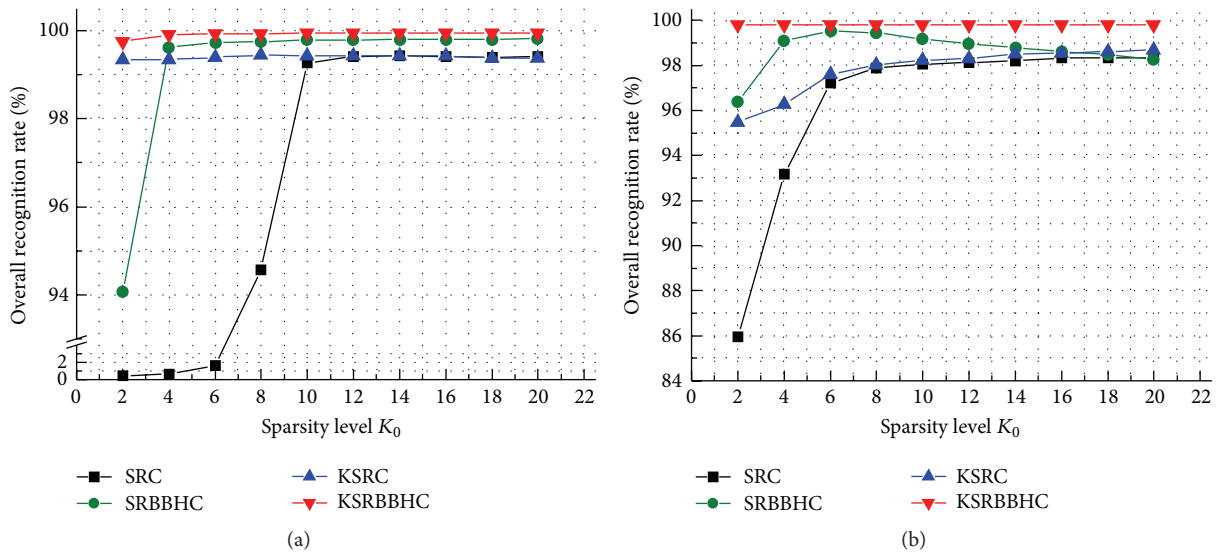


FIGURE 3: Classification accuracies of four sparse-based classifiers with different K_0 for two datasets: (a) ROI-I; (b) ROI-II when σ is optimized.

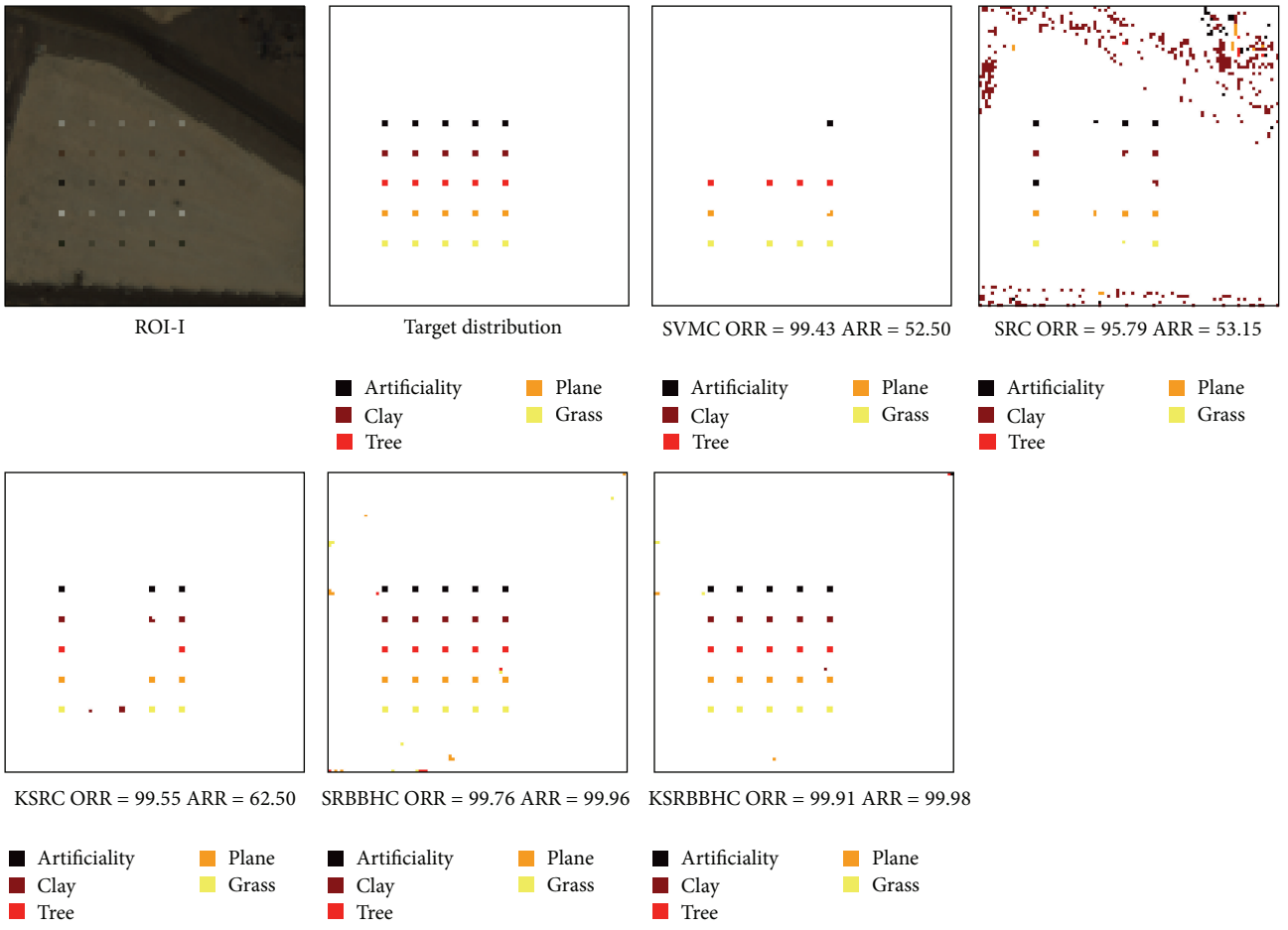


FIGURE 4: Classification maps of classifiers for ROI-I.

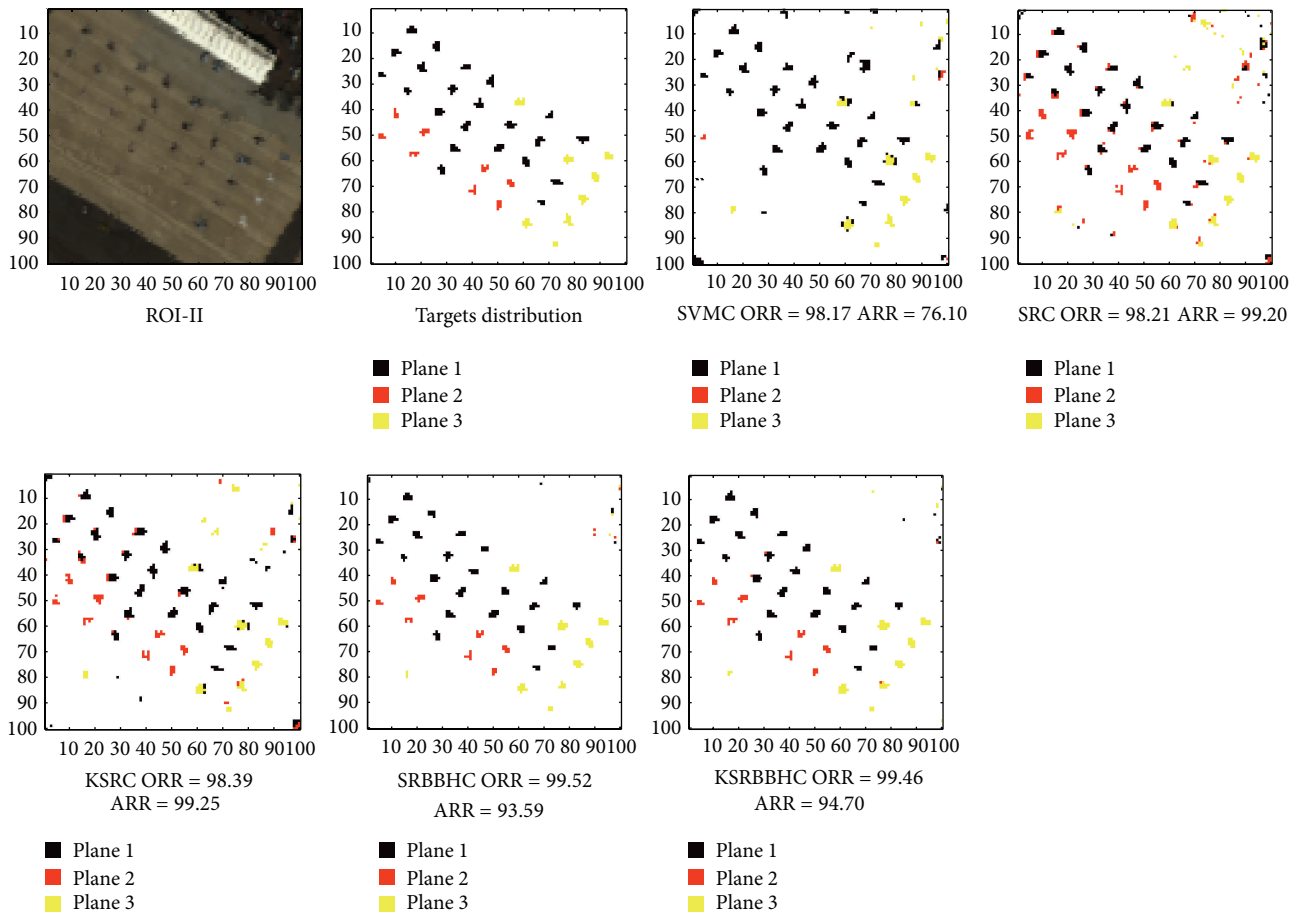


Figure 5: Classification maps of classifiers for ROI-II.

nonlinearly separable and complex data structure into consideration, the kernel version of SRBBHC (KSRBBHC) is then proposed to solve the complicated classification problem.

The proposed algorithms are tested on two HSI datasets, and the experimental results confirm the effectiveness of the proposed SRBBHC and KSRBBHC. From the experiment results, it can be concluded that (1) by collecting desired background training samples adaptively, the proposed local classifier solves the small target classification problem without requiring the number of classes present in image scene; (2) the SRBBH model enhances the discriminative power, which results in a better classification performance; (3) the kernelization further improves the classification performance to a certain extent.

All bands are involved in the proposed method, while the HSI may have more discriminative power at specific spectral channels. And the performance of SRBBHC and KSRBBHC is heavily influenced by background dictionary. Thereby, the performance can be further improved by some preprocess such as band selection and dictionary learning. The effect of spatial information [21] on the performance of SRBBHC and KSRBBHC will be investigated in our future work. In addition, we will also investigate the selection of appropriate validation threshold. By the way, backanalyzing the distribution of vector $\mathbf{R}$ of desired pixels and undesired pixels is a potential direction.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank the Wuhan University, Wuhan, China, for providing the San Diego airport data used in the experiments. This work is supported by the National Natural Science Foundation of China under Grant 61273275 and Aeronautical Science Foundation of China under Grant 20130196004.

References

[1] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual*

ACM Workshop on Computational Learning Theory (COLT '92), pp. 144–152, Pittsburgh, Pa, USA, July 1992.

[3] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, vol. 3584 of *Proceedings of SPIE*, pp. 221–232, Washington, DC, USA, October 1998.

[4] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[5] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3372, 2006.

[6] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[7] Z. Yuan, H. Sun, K. Ji, Z. Li, and H. Zou, "Local sparsity divergence for hyperspectral anomaly detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1697–1701, 2014.

[8] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 217–231, 2013.

[9] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 629–640, 2011.

[10] W. Liguo and Z. Chunhui, *Processing Techniques of Hyperspectral Imagery*, National Defense Industry Press, Beijing, China, 2013 (Chinese).

[11] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[12] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Simultaneous joint sparsity model for target detection in hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 676–680, 2011.

[13] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[14] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[15] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, NY, USA, 2010.

[16] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: a nonlinear anomaly detector for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4099–4109, 2010.

[17] V. M. Patel and R. Chellappa, *Sparse Representations and Compressive Sensing for Imaging and Vision*, Springer Briefs in Electrical and Computer Engineering, Springer, New York, NY, USA, 2013.

[18] Y. Zhang, L. Zhang, and S. Wang, "A Nonlinear sparse representation-based binary hypothesis model for hyperspectral target detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2513–2522, 2014.

[19] T. Wang, B. Du, and L. Zhang, "A kernel-based target-constrained interference-minimized filter for hyperspectral sub-pixel target detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 626–637, 2013.

[20] Y. Tang, S. Huang, Q. Ling, and Y. Zhong, "Adaptive kernel collaborative representation anomaly detection for hyperspectral imagery," *High Power Laser and Particle Beams*, vol. 27, no. 9, Article ID 091008, 2015.

[21] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Multiple spectral-spatial classification approach for hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4122–4132, 2010.