Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1376726, 9 pages
https://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.

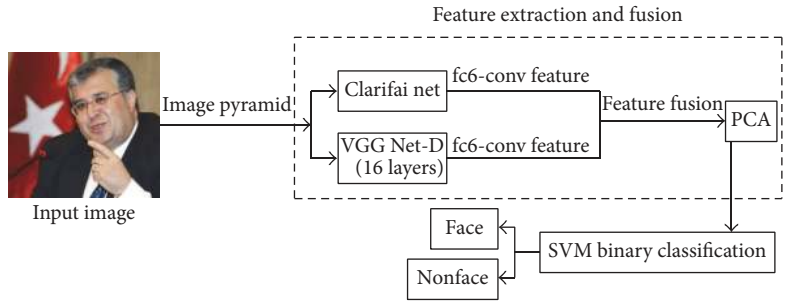


Figure 1: The framework of the proposed method.

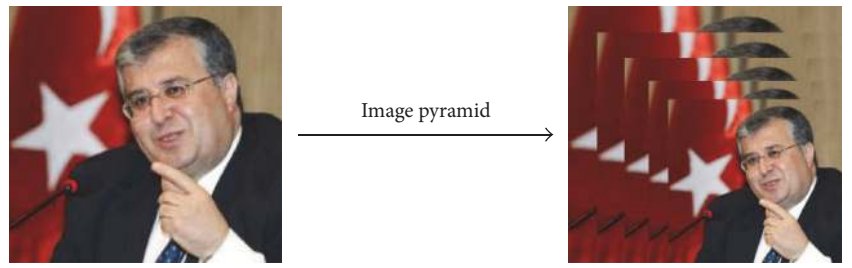


Figure 2: Sketch map of image pyramid.

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

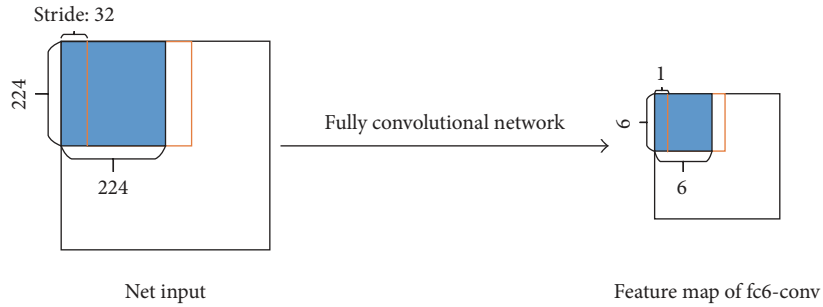


Figure 3: Sketch map of feature map in fully convolutional network.

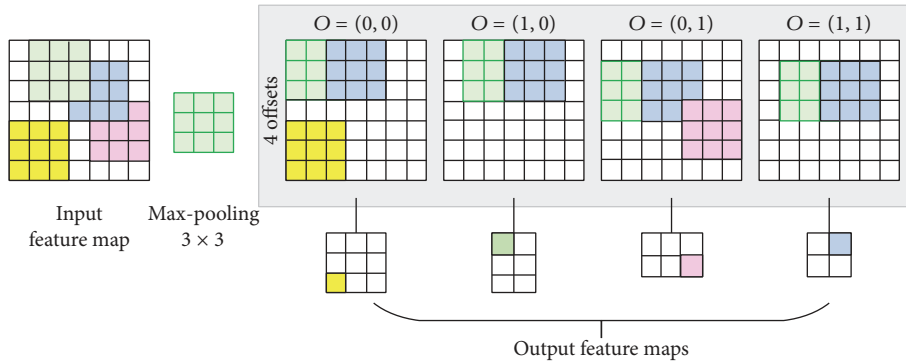


Figure 4: Image forward propagation techniques for max-pooling layers.

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.

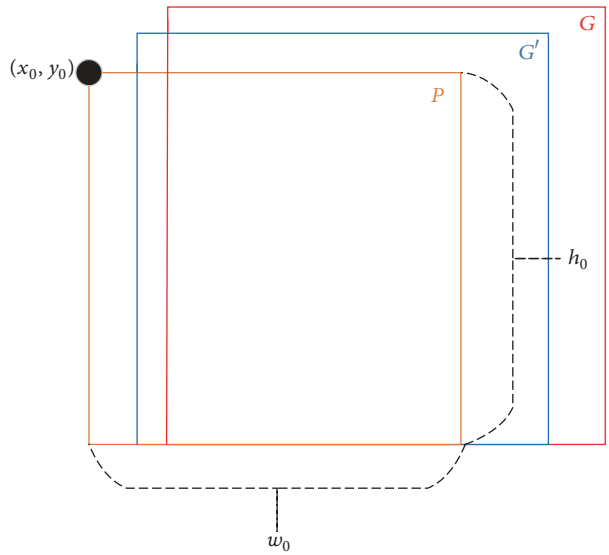
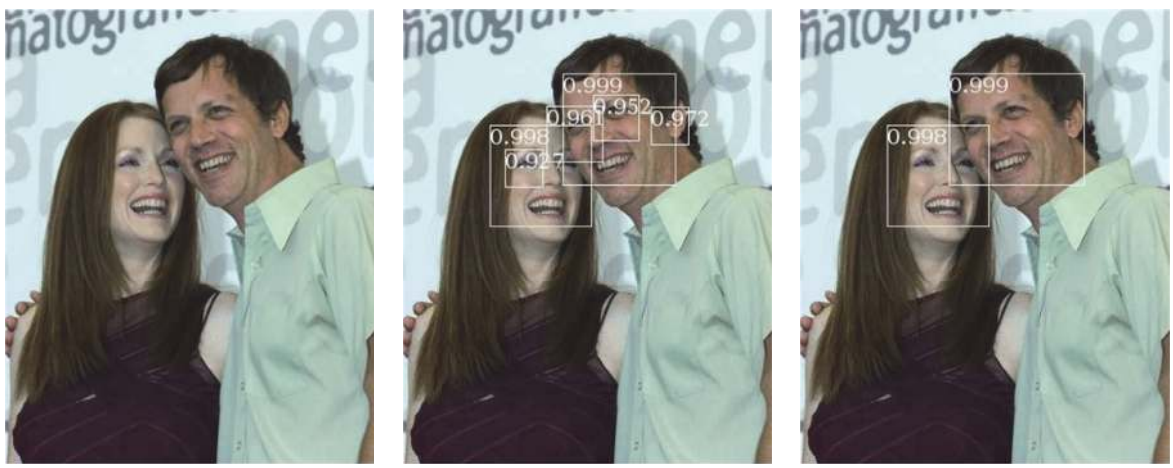


Figure 5: Sketch map of bounding box regression.



(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

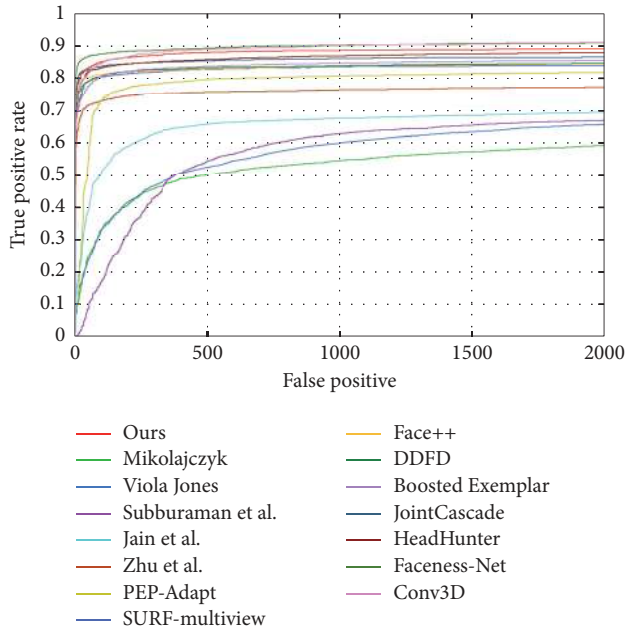


Figure 7: Comparisons of our method with other face detectors on FDDB dataset.

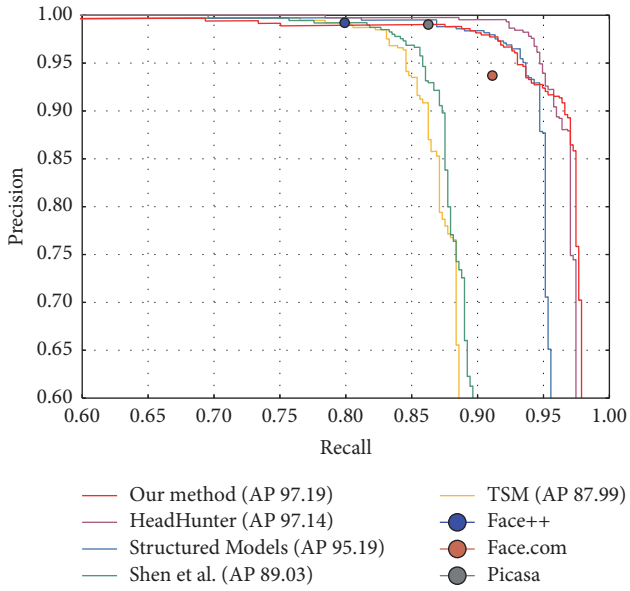


Figure 8: Comparisons of our method with other face detectors on AFW dataset.

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.