Signal Processing: Image Communication 47 (2016) 476–481

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

FaceHunter: A multi-task convolutional neural network based face detector

Dong Wang, Jing Yang, Jiankang Deng, Qingshan Liu *

B-DAT Lab, School of Information and Control, Nanjing University of Information Science and Technology, No. 219, Ningliu Road, Nanjing, China

ARTICLE INFO

Article history:
Received 27 October 2015
Received in revised form
19 April 2016
Accepted 19 April 2016
Available online 23 April 2016

Keywords:
Face detection
Convolutional neural network
Multi-task
Adaptive pooling layer
Region proposal network

ABSTRACT

In this paper, we propose a new multi-task Convolutional Neural Network (CNN) based face detector, which is named *FaceHunter* for simplicity. The main idea is to make the face detector achieve a high detection accuracy and obtain much reliable face boxes. Reliable face boxes output will be much helpful for further face image analysis. To reach this goal, we design a deep CNN network with a multi-task loss, i.e., one is for discriminating face and non-face, and another is for face box regression. An adaptive pooling layer is added before full connection to make the network adaptive to variable candidate proposals, and the truncated SVD is applied to compress the parameters of the fully connected layers. To further speed up the detector, the convolutional feature map is directly used to generate the candidate proposals by using Region Proposal Network (RPN). The proposed *FaceHunter* is evaluated on the AFW dataset, FDDB dataset and Pascal Faces respectively, and extensive experiments demonstrate its powerful performance against several state-of-the-art detectors.

1. Introduction

Face detection is generally the first step in automatic face analysis system, and it has much effect on the performance of face analysis. In the past two decades, a lot of face detection methods have been proposed [1]. In practical real-time applications, the Adaboost-based method and the part-based deformable model are two popular methods. The Adaboost-based method is first proposed by Viloa and Jones [2], in which Adaboost learning is adopted to learn a cascade of weak classifiers with the simple Haar like features for discriminating face and non-face. This idea has been the source of inspiration for countless extension works [3]. The part-based deformable model benefits from the fact that the part of an object often presents less visual variations, and the global variation of the object can be modeled by the flexible configuration of different parts [4]. Although they achieved significant progress in practical applications, detecting face "in the wild" is still to be a challenge, especially when the faces are suffered from large variations in facial appearances, occlusions, and clutters, because they are based on low-level handcraft features, which are insufficient for describing complex variation of face appearance in uncontrolled environments. In recent years, convolutional neural network (CNN) [5–7] based deep learning attracted much attentions in the communities of computer vision and pattern recognition. Its main advantage is that it can learn visual features automatically. CNN has also been successfully applied in face detection [8–10]. Compared with previous works [2,4], the CNN-based detectors are directly to learn the features from the images instead of relying on the hand-crafted features. Hence they can better differentiate faces from uncontrolled environments. In this paper, our proposed face detector is also based on CNN learning.

However, most detectors ignore the issue about the stability and reliability of the bounding box, which has a significant impact on the precision as well as the subsequent analysis. For example, the most common used standard for evaluating a face detector is based on the intersection-over-union (IoU) ratio. The threshold is usually set to be 0.5, and it means all the possible results will be discarded if the IoU is less than 0.5. A few works take this issue into account, such as the R-CNN-based object detectors [11,12], but they regard the bounding box regression as an independent stage in the networks. In [10], a separate CNN network is designed for the bounding box regression and aims at improving another CNN network for face detector.

In this paper, we propose a new face detector named "*FaceHunter*", which is based on a multi-task convolution neural network. The multi-task CNN has been proposed by [13] and it is proved to be very efficient for the general object detection task. Inspired by [13], we consider training our detection network using a multi-task loss in a single training stage by modifying the networks. We make it terminated at two sibling layers (a fully

* Corresponding author.
E-mail address: qsliu@nuist.edu.cn (Q. Liu).
URL: http://bdat.nuist.edu.cn/ (Q. Liu).

http://dx.doi.org/10.1016/j.image.2016.04.004
0923-5965/

connected layer and softmax over 2 categories and bounding-box regressor) sharing convolutional features, so we can classify the proposals and refine the bounding box simultaneously. Considering the high computational expense of the CNN, we introduce an "adaptive pooling layer" to avoid repeatedly computing the convolutional features. Moreover, it removes the fixed-size constraint of the network, as it can generate a fixed-length representation regardless of image size and makes the network more flexible during parameter fine-tuning. To reduce the computation cost of fully connected layers, we use the truncated SVD [13] to compress them. This simple compression method gives a good speedup for detection.

Additionally, generating proposal is another computational bottleneck in detection systems. Selective Search [14] used in Fast-RCNN [13] is one of the most popular proposal methods, which costs about 2 s per image in a CPU implementation. Considering this issue, to further speed up the detector, we directly use the convolutional feature map to generate the candidate windows. We apply the Region Proposal Network (RPN) [15] to generate the proposals based on the convolutional features, so the RPN and multi-task CNN can share the same convolutional features. The RPN is a kind of fully convolutional network (FCN) [16] and can be trained end-to-end specifically for the task of generating detection proposals. It is a small sliding window network, which is fully connected to a $n \times n$ (3×3 in our work) spatial window of the input conv feature maps. Each sliding window is mapped to a low-dimensional vector (256-d in our work). The vector will be fed into two sibling fully connected layers: a box-regression layer and a box-classification layer. RPN is not the same as the multi-task CNN, but it uses the same multi-task training loss with the multi-task CNN. With this strategy of sharing convolutions, the marginal cost for computing the proposals becomes very small (i.e. 4 ms for a typical 1000×600 image) enabling nearly cost-free proposals. Moreover, our experiment results show that the proposal method with RPN can speed up Fast-RCNN and improve the precision of detection.

We test the proposed detector on AFW [17], FDDB [18] and Pascal Faces [19] datasets, and the experimental results show its promising results. Our contributions can be summarized as:

- The multi-task CNN is applied in the face detection task, and it is validated to be very efficient. An adaptive pooling layer is integrated into the network to make it more flexible during parameter fine-tuning, and also the truncated SVD is introduced to compress the fully connected layers for reducing the computation cost of them.
- To further speedup the network, the RPN network is introduced to generate the proposals, which is directly performed on the convolutional feature maps. The RPN and multi-task CNN share the same convolutional features, so the proposal generating cost is very small.

2. Our work

As described above, the proposed *FaceHunter* consists of two networks, as shown in Fig. 1. One is the Region Proposal Network (RPN) [15], which is to generate the candidate proposals. Another is the multi-task CNN for final detection output. Both networks share the same convolution feature maps, which are pre-trained on the ImageNet database [20]. The details of the both networks are presented in the following.

2.1. Feature maps

We consider a variant of the network in [20] to generate the feature maps, which removes the last pooling layer (i.e. $pool_5$). As depicted in Fig. 2, the net contains 5 layers. The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer. The kernels of the third convolutional layer are connected to all the kernel maps in the second layer. The response-normalization layers follow the first and second convolutional layers. The Max-pooling layers follow response-normalization layers. The ReLU non-linearity is applied to the output of every convolutional and fully connected layer. The first convolutional layer filters the input with 96 kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels. The second convolutional layers input is the (response-normalized and pooled) output of the first convolutional layer and filters it with 256 kernels of size $5 \times 5 \times 48$. The third, fourth, and fifth convolutional layers are connected without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size $3 \times 3 \times 256$ connected to the outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size $3 \times 3 \times 192$, and the fifth convolutional layer has 256 kernels of size $3 \times 3 \times 192$.

2.2. Proposal with RPN

The purpose of proposals is to generate the candidate windows for further process, and it is an important component in fast object detection system. In this paper, we directly use the convolution features to generate the candidate proposals by performing RPN on the convolutional feature map. Because the RPN network shares the same convolution features with the multi-task CNN, it can obtain candidate proposals with very low computational cost. The original RPN is a fully convolutional network (FCN) [16]. Same as [15], we add two additional layers in RPN, as shown in Fig. 1. One layer is to encode each *conv* map position into a short (e.g. 256-d) feature vector, and another is the output layer, which has two kinds of outputs: the candidate proposal/non-candidate proposal probability and the proposal coordinates. We slide a small network over the *conv* feature map output by the last shared *conv*

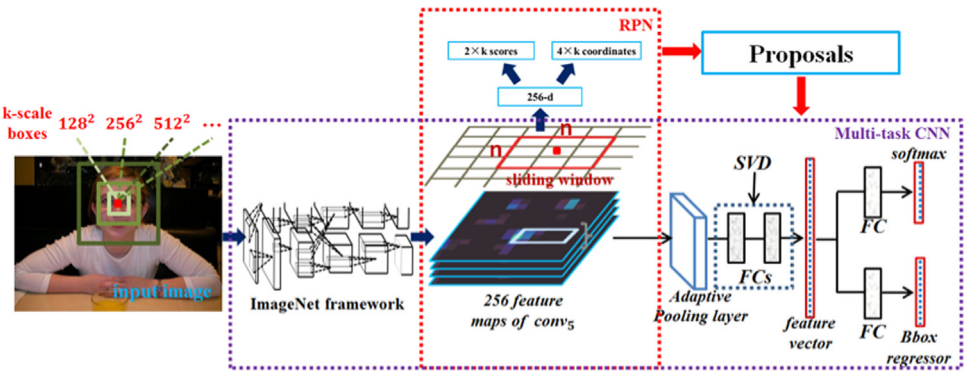

Fig. 1. The overview of our multi-task CNN based face detector *FaceHunter*.

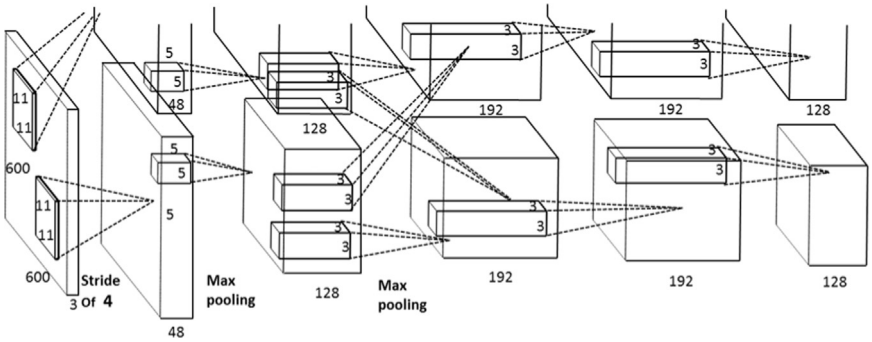

Fig. 2. The deep structure which is used for generating feature maps. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers.

layer. At each $n \times n$ sliding-window location, we can predict k proposal regions associated with k scales simultaneously.

2.3. Detection with multi-task CNN

In an automatic face analysis system, face detection can be regarded as the pre-processing and initialization step, so a robust face detector should provide good face bounding boxes for further analysis besides having a high detection accuracy. To reach this goal, we modify the conventional CNN with two sibling output layers. The first layer outputs the probabilities p of face and non-face. The second layer outputs the bounding box offsets $t = (t_x, t_y, t_w, t_h)$. Then, we use a multi-task loss L to train the network jointly for classification and bounding-box regression:

$$L(p, p^*, t, t^*) = L_c(p, p^*) + \lambda L_r(t, t^*) \tag{1}$$

where p^* is a true class label and $L_c(p, p^*) = -\log pp^*$ is the standard cross-entropy/log loss. For the second task loss, the true bounding-box regression target is $t^* = (t_x^*, t_y^*, t_w^*, t_h^*)$ and the prediction is $t = (t_x, t_y, t_w, t_h)$. We use the loss

$$L_r(t, t^*) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i, t_i^*) \tag{2}$$

where

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{3}$$

is a smoothed L_1 loss. The hyper-parameter λ in Eq. (1) is used to control the balance between the two task losses.

To make the network more flexible, we adopt an adaptive pooling layer after $conv_5$, which is inspired by the spatial pyramid pooling layer used in [10]. Different from the spatial pyramid pooling layer in [10], the adaptive pooling layer has only a single level, which allows all the network layers to be updated during fine-tuning. Using the adaptive pooling layer, the network only needs to scan the convolutional layers one time on the entire image. It is applied to the regions on the $conv_5$ feature maps corresponding to each candidate window and output max-pooled feature maps with $L \times L$ spatial bins, and its bin's size is proportional to the feature map size.

Additionally, the detector needs to scan all the candidate proposals, and the fully connected layers often have high computation costs. To deal with this issue, we use the truncated SVD to compress fully connected layers as in [13]. As shown in Fig. 3, the fully connected layer can be parameterized by the $u \times v$ weight matrix $W \approx U \sum_t V^T$ according to the theory of SVD. U is a $u \times t$ matrix comprising the first t left-singular vectors of W, $\sum_t$ is a $t \times t$ diagonal matrix containing the top t singular values of W, and V is $v \times t$ matrix comprising the first t right-singular vectors of W. So it reduces the parameters from uv to $t(u + v)$. Using SVD, the fully connected layer with weight matrix W is replaced by two fully connected layers using the weight matrix $\sum_t V^T$ and U respectively.

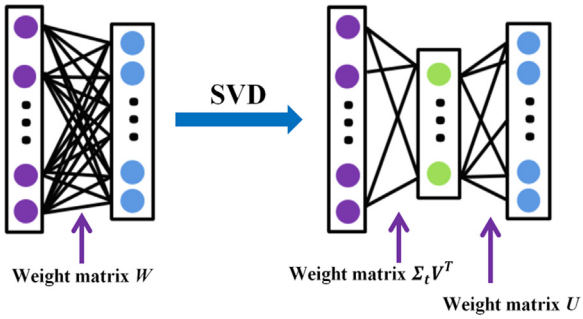

Fig. 3. Use truncated SVD to compress fully connected layers.

2.4. Algorithm

As shown in Fig. 1, the proposed face detection framework includes two networks, i.e., the RPN for generating proposals and the multi-task CNN for final detection. The both networks share the same convolutional layers. Thus, we develop a 4-step training algorithm to learn the shared features via alternating optimization. Firstly, we train the RPN, which is initialized by an ImageNet pre-trained model. Secondly, based on the proposals generated by the trained RPN, we train the multi-task CNN . Thirdly, we use the trained CNN network to initialize the RPN. We keep the shared *conv* layers fixed and only fine-tune the layers of the RPN. Finally, keeping the shared *conv* layers fixed, we fine-tune the *fc* layers of the multi-task CNN. Algorithm 1 summarizes the whole algorithm.

Algorithm 1. The whole optimization algorithm.

Input: Face image, face rect.
Initialization: ImageNet pre-trained model.
1. Train the RPN network.
2. Use the proposals generated by the RPN to train the multi-task CNN.
3. Use the CNN network to initialize the RPN.
4. Fine-tune the *fc* layers of the multi-task CNN.

Output: Multi-task CNN based face detection model.

3. Experiments

3.1. Experimental setting

We conduct the experiments on five datasets: AFW [17] (205 images with bounding boxes), FDDB [18] (2845 images with ellipses annotations), Pascal Faces dataset [19] (851 Pascal VOC

images with bounding boxes), AFLW [21] (26,000 annotated faces), and the dataset collected by ourselves, which has a total of 980,354 annotated faces. All the face images are captured in uncontrolled conditions with cluttered backgrounds and large variations in both face viewpoint and appearance.

In the training phase, we select 183,200 face images from our own dataset to train *FaceHunter*, and we use the AFLW dataset for validation. The positive samples are cropped according to the labeled ground truth windows, and the negative samples are cropped around the labeled ground truth windows, but the overlaps between positive window and negative windows are less than 25% (measured by the intersection-over-union (IoU) ratio). To reduce redundant information resulting from overlapping negative samples, the overlaps between two negative windows are more less than 75%. Our implementation is based on the publicly available code of cuda-convnet [20] and Caffe [22]. Multi-scale features may improve accuracy, but single-scale processing offers the best tradeoff between speed and accuracy [13]. In our experiments, we consider a brute-force learning (single scale) to achieve scale invariance. We rescale the images such that their shorter side is $s=600$ pixels. The ground-truth regression targets t^* is normalized with zero mean and unit variance. The hyper-parameter λ is set to 1 empirically. We set the size of the adaptive pooling layer L to be compatible with the first fully connected layer (i.e. $L=6$). We use dropout, as [20], in the two fully connected layers after the adaptive pooling layer. To compress the fully connected layers, we use the top 1024 singular values from the 9216×4096 matrix in the fc_6 layer and the top 256 singular values from the 4096×4096 fc_7 layer. For training the RPN, we use the same loss function with the hyper-parameter λ changing from 1 to 10. That means we bias towards better box locations. We use $n=3$ for the size of the sliding window in the RPN. For each sliding position, we define $k=3$ proposal regions associated with 3 scales (i.e. 128^2, 256^2 and 512^2 pixels).

In the testing phase, we keep using single-scale testing with $s=600$ pixels for *FaceHunter*. Specially, there exist big overlaps among some proposals. To reduce redundancy, we perform non-maximum suppression (NMS) [4] on the proposal regions based on their *cls* scores. We fix the IoU threshold for NMS at 0.7. After NMS, we use the top-300 ranked proposal regions for detection. Before outputting the final results, non-maximum suppression is performed again for the detectors. Besides, we use all the neurons but multiply their outputs by 0.5, which is a reasonable approximation to taking the geometric mean of the predictive distributions produced by the exponentially many dropout networks.

3.2. Experimental results

We evaluate our face detector on the AFW [17], FDDB [18] and Pascal Faces [19] datasets respectively. As in the previous work [10], we use the intersection-over-union (IoU) ratio to measure the detection results (i.e. $S = \frac{\| r_d \cap r_g \|}{\| r_d \cup r_g \|}$, where r_d is the detecting output box and r_g is the ground truth bounding box), and IoU ≥ 0.5 is regarded as an efficient detection.

For further investigating the proposed *FaceHunter* detector, we particularly evaluate it without using RPN, i.e. "Ours multi-task network without RPN" (expressed using the green curve in Figs. 4–6). "Ours multi-task network without RPN" is similar to a Fast-RCNN [13], but it is not the same as Fast-RCNN. In the general object detection task, such as RCNN, SPP-net and Fast-RCNN, they use the fast mode of selective search to generate 2000 candidate windows per image. In our experiments, we observe that such method is not suitable due to the fact that faces in datasets suffer from large variations in facial appearances and very small faces. To fit the face detection task, as in hierarchical deep detector [10], a low threshold is set to achieve 99% recall of the ACF [23] detector and each image outputs 500 windows on average. In addition, we also analyze the virtues of the multi-task learning style over single-task. We compare the performance of "Ours multi-task network without RPN" with hierarchical deep detector [10](expressed using the blue curve in Figs. 4–6), which trains a separate CNN network for the bounding box regression to refine another CNN network for face detector.

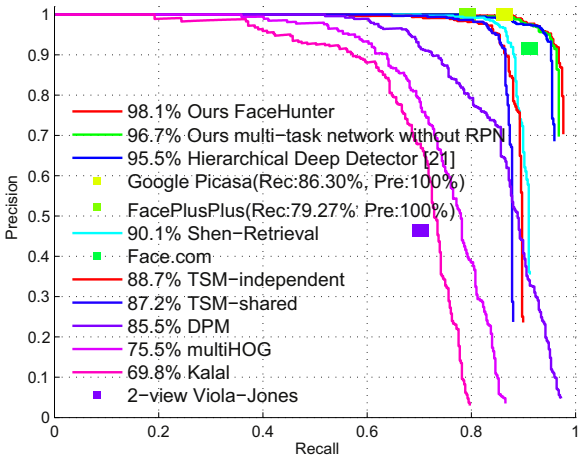

Fig. 4. PR curve on AFW. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

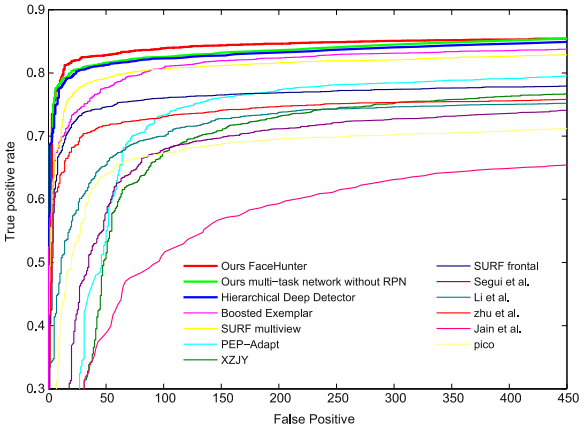

Fig. 5. Discrete ROC on FDDB. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

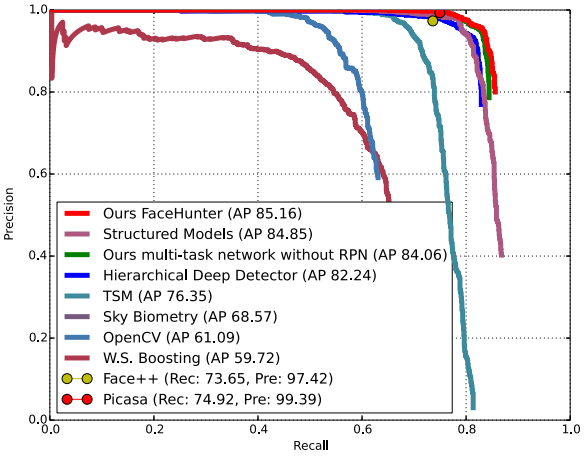

Fig. 6. PR curve on Pascal Faces. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

Fig. 4 shows the experimental results on the AFW dataset, where we compare our *FaceHunter* with hierarchical deep detector [10], Shen-Retrieval [24], TSM-independent/TSM-shared [17], DPM [4], OpenCV implementation of Viola-Jones [2], multiHOG and three commercial systems face.com, FacePP-v2 and Picasa. Our *FaceHunter* outperforms the baseline TSM by 10%, and outperforms the Shen-Retrieval by 8%. It even outperforms the commercial systems face.com, FacePP and Google Picasa.

Fig. 5 shows the experimental results on the FDDB dataset. We compare our detector with the hierarchical deep detector [10], Boosted Exemplar [25], SURF frontal/multiview [3], PEP-Adapt [26], XZJY [24], pico [27], Zhu et al. [17], Jain et al. [28], etc. In the dataset, we can find that our *FaceHunter*, our "multi-task network without RPN" and hierarchical deep detector are among the leading methods, which largely outperform the others.

On the Pascal Faces dataset, as in [19], we compare our *FaceHunter* with the Structured Models [19], hierarchical deep detector [10], TSM [17], OpenCV implementation of Viola-Jones [2], Sky Biometry, W.S. Boosting and two commercial algorithms FacePP-v2 and Picasa. As the results shown in Fig. 6, our *FaceHunter* beats the remarkable Structured Models [19] detector. Our "multi-task network without RPN" can improve the average precision of hierarchical deep detector by about 1%, and *FaceHunter* even can improve it by about 3%.

On all the three datasets, we can see the proposed *FaceHunter* detector achieves a good performance against the state of the art methods. It also indicates that, RPN can not only speed up the detector, but also increase the detecting precision. In addition, the

Fig. 7. Examples of our detection results.

comparison between our "multi-task network without RPN" and hierarchical deep detector [10] shows that the multi-task mode have stronger learning ability than the single-task mode. Fig. 7 shows some examples of our detection results. It can be seen that our detector can detect faces with different poses, in severe occlusions and cluttered background, as well as blurred face images.

3.3. Computational complexity analysis

The complexity of the convolutional feature computation in *FaceHunter* is $O(r \cdot s^2)$ at a scale s (i.e. $s=600$ in this work), where r is the aspect ratio, while the complexity of R-CNN [11] is $O(n \cdot 227^2)$ with the window number n (2000). Thus, our method is much faster than R-CNN. We evaluate the average time of 800 Pascal Faces [19] images using an Nvidia K40 GPU. Our detector takes only 0.041 s per image for convolutions. The GPU time of computing NMS, *fc*, softmax and others is about 0.034 s per image. It takes only 0.004 s per image for proposal on top of the *conv* map.

4. Conclusion

In this paper, we propose a multi-task convolutional neural network (CNN) based face detector named *FaceHunter*. We focus on obtaining stable bounding boxes and achieving high detection accuracies. To simplify the training process, we design a multi-task loss to train the network. The adaptive pooling layer is applied in the network to speed up the detector both in training and testing. To further make the detector faster, we perform the Region Proposal Network (RPN) directly on the convolutional feature maps to generate the proposal candidates. Another improvement is made by the use of truncated SVD to compress the parameters of the fully connected layers, which is effective for face detection. We evaluate our detector on three public datasets and achieve state-of-the-art performance.

Acknowledgments

This work was supported in part by NSFC Grant no. 61532009, 61272223 and in part by the Graduate Education Innovation Project of Jiangsu under Grant KYLX15_0881.

References

[1] S. Zafeiriou, C. Zhang, Z. Zhang, A survey on face detection in the wild: past, present and future, Computer. Vis. and Image Underst 138 (2015) 1–24.
[2] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[3] J. Li, Y. Zhang, Learning surf cascade for fast and accurate object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Portland, OR, USA, 2013, pp. 3468–3475.
[4] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intel. 32 (9) (2010) 1627–1645.
[5] K. Fukushima, Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern. 36 (4) (1980) 193–202.
[6] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[7] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, Technical Report, DTIC Document, 1985.
[8] R. Vaillant, C. Monrocq, Y. Le Cun, Original approach for the localisation of objects in images, IEE Proc. – Vis. Image Signal Process. 141 (4) (1994) 245–250.
[9] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.
[10] D. Wang, J. Yang, J. Deng, Q. Liu, Hierarchical convolutional neural network for face detection, in: Image and Graphics, Springer, Tianjin, China, 2015, pp. 373–384.
[11] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, OH, USA, 2014, pp. 580–587.
[12] K. Grauman, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features. in: IEEE International Conference on ICCV, vol. 2, IEEE, Beijing, China, 2005, pp. 1458–1465.
[13] R. Girshick, Fast r-cnn, arXiv preprint arXiv:1504.08083.
[14] J.R. Uijlings, K.E. van de Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[15] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, arXiv preprint arXiv:1506.01497.
[16] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, arXiv preprint arXiv:1411.4038.
[17] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Rhode Island, US, 2012, pp. 2879–2886.
[18] V. Jain, E.G. Learned-Miller, Fddb: A Benchmark for Face Detection in Unconstrained Settings, UMass Amherst Technical Report.
[19] J. Yan, X. Zhang, Z. Lei, S.Z. Li, Face detection by structural models, Image Vis. Comput. 32 (10) (2014) 790–799.
[20] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[21] M. Kostinger, P. Wohlhart, P.M. Roth, H. Bischof, Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, Barcelona, Spain, 2011, pp. 2144–2151.
[22] Y. Jia, Caffe: an open source convolutional architecture for fast feature embedding ⟨http://caffe.berkeleyvision.org⟩.
[23] B. Yang, J. Yan, Z. Lei, S.Z. Li, Aggregate channel features for multi-view face detection, in: IEEE International Joint Conference on Biometrics (IJCB), IEEE, Tampa, US, 2014, pp. 1–8.
[24] X. Shen, Z. Lin, J. Brandt, Y. Wu, Detecting and aligning faces by image retrieval, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Portland, OR, USA, 2013, pp. 3460–3467.
[25] H. Li, Z. Lin, J. Brandt, X. Shen, G. Hua, Efficient boosted exemplar-based face detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, 2014, pp. 1843–1850.
[26] H. Li, G. Hua, Z. Lin, J. Brandt, J. Yang, Probabilistic elastic part model for unsupervised face detector adaptation, in: IEEE International Conference on Computer Vision (ICCV), IEEE, Sydney, NSW, Australia, 2013, pp. 793–800.
[27] N. Markuš, M. Frljak, I.S. Pandžić, J. Ahlberg, R. Forchheimer, A method for object detection based on pixel intensity comparisons organized in decision trees, arXiv preprint arXiv:1305.4537.
[28] V. Jain, E. Learned-Miller, Online domain adaptation of a pre-trained cascade of classifiers, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Colorado Springs, US, 2011, pp. 577–584.