Detecting Faces Using Inside Cascaded Contextual CNN

Kaipeng Zhang[1*], Zhanpeng Zhang[2], Hao Wang[1], Zhifeng Li[1], Yu Qiao[3], Wei Liu[1]
[1]Tencent AI Lab [2]SenseTime Group Limited
[3]Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
kp_zhang@foxmail.com, zhzhanp@gmail.com
{hawelwang, michaelzfli}@tencent.com, yu.qiao@siat.ac.cn, wliu@ee.columbia.edu

Abstract

Deep Convolutional Neural Networks (CNNs) achieve substantial improvements in face detection in the wild. Classical CNN-based face detection methods simply stack successive layers of filters where an input sample should pass through all layers before reaching a face/non-face decision. Inspired by the fact that for face detection, filters in deeper layers can discriminate between difficult face/non-face samples while those in shallower layers can efficiently reject simple non-face samples, we propose Inside Cascaded Structure that introduces face/non-face classifiers at different layers within the same CNN. In the training phase, we propose data routing mechanism which enables different layers to be trained by different types of samples, and thus deeper layers can focus on handling more difficult samples compared with traditional architecture. In addition, we introduce a two-stream contextual CNN architecture that leverages body part information adaptively to enhance face detection. Extensive experiments on the challenging FDDB and WIDER FACE benchmarks demonstrate that our method achieves competitive accuracy to the state-of-the-art techniques while keeps real time performance.

1. Introduction

Face detection is essential to many face applications (e.g. face recognition, facial expression analysis). However, the large visual variations of face, such as occlusion, large pose variation, and extreme illumination impose great challenges for face detection in unconstrained environments. Recently, deep convolutional neural networks (DCNNs) achieve remarkable progresses in a variety of computer vision tasks, such as image classification [8], object detection [5], and face recognition [19]. Inspired by this, several studies [13, 14, 25, 27, 29, 23, 26] utilize deep CNNs for face detection and achieve the leading detection performance.

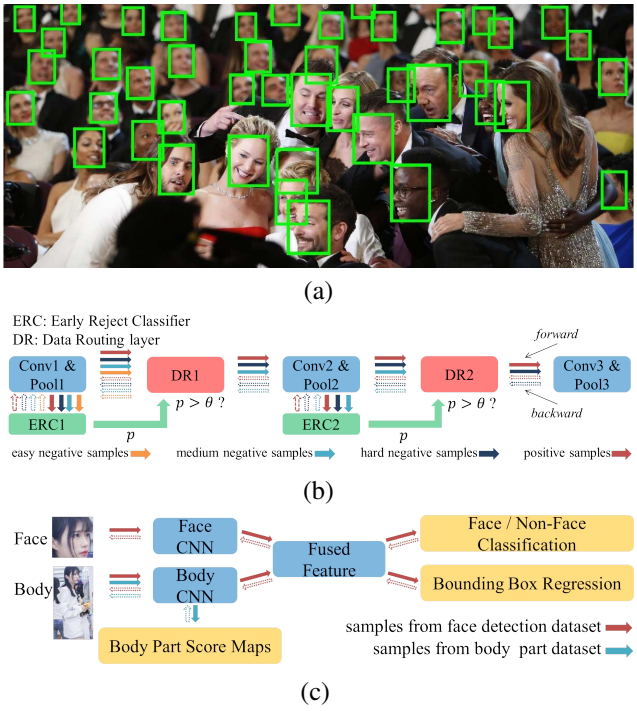


Figure 1. (a) An example of face detection result using our proposed method. It leverages Inside Cascaded Structure (ICS) to encourage the CNN to handle difficult samples at deep layers, and utilizes the two-stream contextual CNN to exploit the body part information adaptively. (b) Illustration of the proposed ICS and Data Routing (DR) training. (c) Illustration of two-stream contextual CNN and Body Part Sensitive Learning (BPSL). Solid arrows denote the samples processed in the forward pass, while the dashed arrows are for backward propagation. Best viewed in color.

The key part of recent CNN-based face detection methods is to train a powerful CNN as a face/non-face classifier. Previous works formulate the feature extractor and classifier in an end-to-end learning framework to obtain high accura-

*Corresponding author

cy. For an arbitrary sample, the feature is extracted through a forward pass of all the layers. However, this is inefficient because the filters in deeper layers should focus on discriminating difficult non-face samples while easy non-face samples can be rejected in shallower layers.

Different from previous works, we notice that different layers of CNN can learn features of different perceptions that are suitable for discriminating face/non-face examples of different difficulties. This insight inspires us to treat CNN as a cascade of layer classifiers and use them to handle samples of various difficulties. In our approach, filters in different layers are optimized for different types of samples in the training process. More specially, we construct cascaded classifiers inside the CNN, and introduce a data routing strategy to guide the data flow for optimizing layer parameters (see Fig. 1(b)). This architecture allows deeper layers to focus on discriminating faces and difficult non-face samples while easy non-face samples are rejected in shallower layers. Experiments show that this method not only reduces the computation cost in testing stage but also increases detection accuracy.

Contextual information yields effective cues for object detection [28]. In this paper, we propose to leverage body information to enhance face detection accuracy. However, roughly cropping body region cannot perform well in practice, since there may exist large visual variations of body regions in real-world, caused by various pose, body occlusions, or even body absence. To relieve this difficulty, we propose a two-stream contextual CNN that joint body parts localization and face detection in an optimal way. This network can automatically predict the existence of the body part and thus exploit the contextual information adaptively. We call this process Body Part Sensitive Learning (BPSL, see Fig. 1(c)).

The main contributions of this paper are as summarized following: (1) We propose a novel deep architecture with a cascade of layer classifies for face detection and introduce data routing strategy to train this architecture in an end-to-end way. This architecture encourages layers to focus on rejecting non-face samples of different types. (2) We propose to jointly optimize body part localization and face detection in a two-stream contextual CNN that exploits body information to assist face detection by learning filters sensitive to the body parts. (3) Extensive experiments show that our method achieves competitive accuracy to the state-of-the-art techniques on the challenging FDDB and WIDER FACE benchmarks while keeps real time performance.

2. Related Works

Face detection attracts extensive research interests and remarkable progresses have been made in the past decade. The cascaded face detector [20] utilizes Haar-Like features and AdaBoost algorithm to train a cascade of face/non-face classifiers which achieves a good accuracy with real-time efficiency. A few works [17, 22, 30] improve this cascaded detector using more advanced features and classifiers. Besides the cascade structure, [21, 31, 16] introduce deformable part models (DPM) for face detection and achieve remarkable performance. However, they are computationally expensive and usually require expensive annotation in the training stage.

Recently, several CNN-based face detection techniques show state-of-the-art performance. Faceness [25] uses some CNNs trained for facial attribute recognition to obtain response map of face regions that further yield candidate face windows. It shows impressive performance on the face with partial occlusion. Zhang et al. [29] propose to jointly solve face detection and alignment using multi-task CNNs. Convnet [14] integrates a CNN and a 3D mean face model in an end-to-end multi-task learning framework. UnitBox [27] introduces a new intersection-over-union loss function.

How to use CNN with cascade structure is widely studied. Cascaded CNN based methods [13, 29, 18] treat CNN as a face/non-face classifier and use hard sample mining scheme to construct a cascade structure outside CNNs. However, filters inside a CNN are stacked layer by layer and these methods ignore the correlation among these cascaded filters. [24] proposes to train cascaded classifiers using AdaBoost algorithm and features from different fixed layers for higher testing speed. However, it separates the CNN optimization and cascaded classifiers optimization. Therefore, the filters from different layers do not specialize in handling in different kinds of data which is adverse for cascaded classifiers performance. In this work, we propose the inside cascade structure to feed different layers with different data. This method can encourage deeper layers to focus on discriminating faces and difficult non-face samples. Therefore, it can produce data-specific features in different layers and also handle different data in different layers properly.

On the other hand, the effectiveness of using contextual information for object detection has been demonstrated in [28]. It crops regions of different sizes from convolutional feature maps using ROI pooling and makes a classification based on these features.

3. Overall Framework

We use a cascaded CNN framework as our basic due to its good performance and runtime efficiency [13, 29, 18]. Different from these works, for the CNN-based face/non-face classifier, we introduce the Inside Cascaded Structure (ICS) and combine contextual CNN for more robust face detection. In general, the framework has three stages as shown in Fig. 2 (a). It contains three successive CNNs: Proposal Net (P-Net), and two Refinement Nets (R-Net-1 and R-Net-2). P-Net is a fully convolutional CNN that quickly produces candidate windows through a sliding scan on the

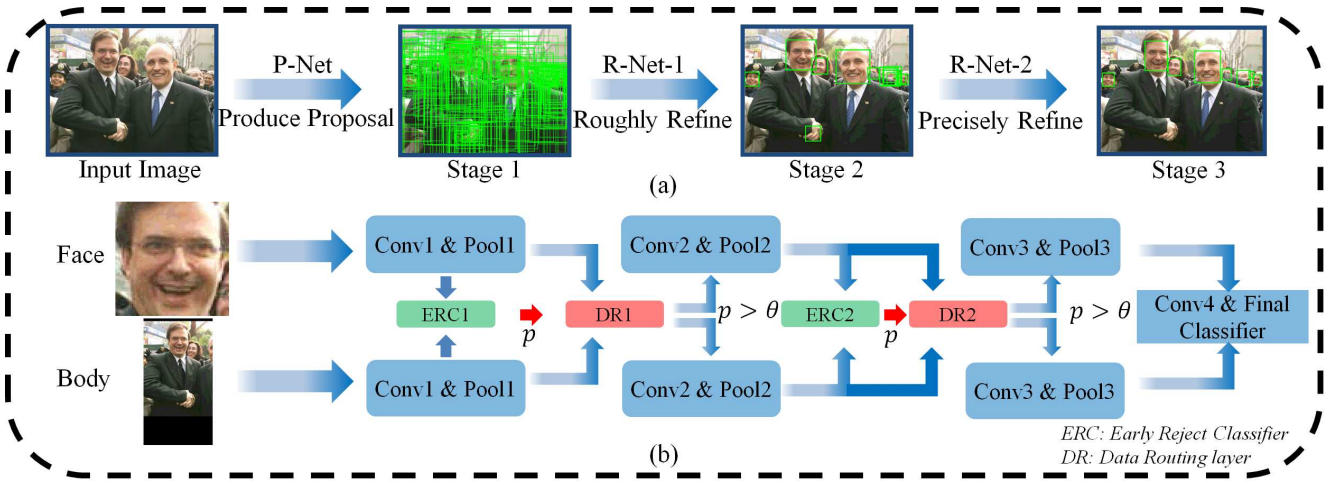


Figure 2. (a) The pipeline of our overall face detection framework, which contains three stages. Face proposals are generated from the input image in the first stage and refined in the next two stages. (b) An example of inside cascaded two-stream contextual CNN structure. It is a combination of Inside Cascaded Structure (ICS) and two-stream contextual CNN.

whole image in different scales (image pyramid). R-Net-1 and R-Net-2 are the inside cascaded two-stream contextual CNN (shown in Fig. 2 (b)), which will be discussed in the following text. These two networks will refine the candidates from P-Net (i.e., patch cropped from input image) by bounding box regression and reject the remaining false alarms.

4. Inside Cascaded Structure

In most CNN-based face detectors, the key part is to train a powerful face and non-face classifier. In this section, we present the Inside Cascaded Structure (ICS) that is capable of learning more effective filters and achieving faster running speed. Compared to traditional CNNs structure, ICS has two extra components, Early Rejection Classifier (ERC) and Data Routing (DR) layer. Illustrations of ICS and its data flow are given in Fig. 1 (b).

Each pooling layer of the CNN is connected to an ERC that predicts the probability of a sample being a face for each sample. These probabilities will be passed to the DR layer to determine what samples should be passed to the following layers. Faces and hard non-face samples will retain in deeper layers while easy non-face samples will be rejected in the shallower layer. This strategy allows deeper layers to focus on discriminating faces and difficult non-face samples while easy negative samples are addressed in shallower layers. Therefore, deeper layers can focus on handling more difficult samples compared to traditional CNN. In addition, easy negative samples are rejected quickly and testing computation cost can be reduced. The ERC and DR layer will be presented in the following text.

4.1. Early Rejection Classifier

The ERC is a small classifier for face and non-face classification. The probability of being a face predicted from ERC will be passed to the next DR layer to determine whether the sample should be passed to the following layers or not. The ERC can be introduced to one or multiple layers of the neural network (a simple example is shown in Fig. 3). In particular, for a sample i in the j-th ERC, we first compute a vector $z_j^i \in \mathbb{R}^2$ by:

$$z_j^i = \phi_j(fea_j^i), \tag{1}$$

where fea_j^i is the features in j-th pooling layer, $\phi_j(\cdot)$ denotes the non-linear transformation of the j-th ERC.

Then we use the softmax function to compute the probability p_j^i for sample i being a face:

$$p_j^i = \frac{e^{z_{j,1}^i}}{e^{z_{j,1}^i} + e^{z_{j,2}^i}}, \tag{2}$$

where $z_{j,1}^i$ is the first element in z_j^i, similar for $z_{j,2}^i$.

We use the cross-entropy loss for training ERC to discriminate face and non-face regions:

$$L_j^i = -(y_i^{det} \log(p_j^i) + (1 - y_i^{det})(\log(1 - p_j^i))), \tag{3}$$

where L_j^i denotes cross-entropy loss for sample i in the j-th ERC and $y_i^{det} \in \{0, 1\}$ denotes the ground-truth label.

4.2. Data Routing Layer

The DR layer receives the probabilities from last ERC for the samples. If the probability of a sample being a face is lower than a preset threshold θ, the sample will be rejected as non-face sample and stop being processed in forward

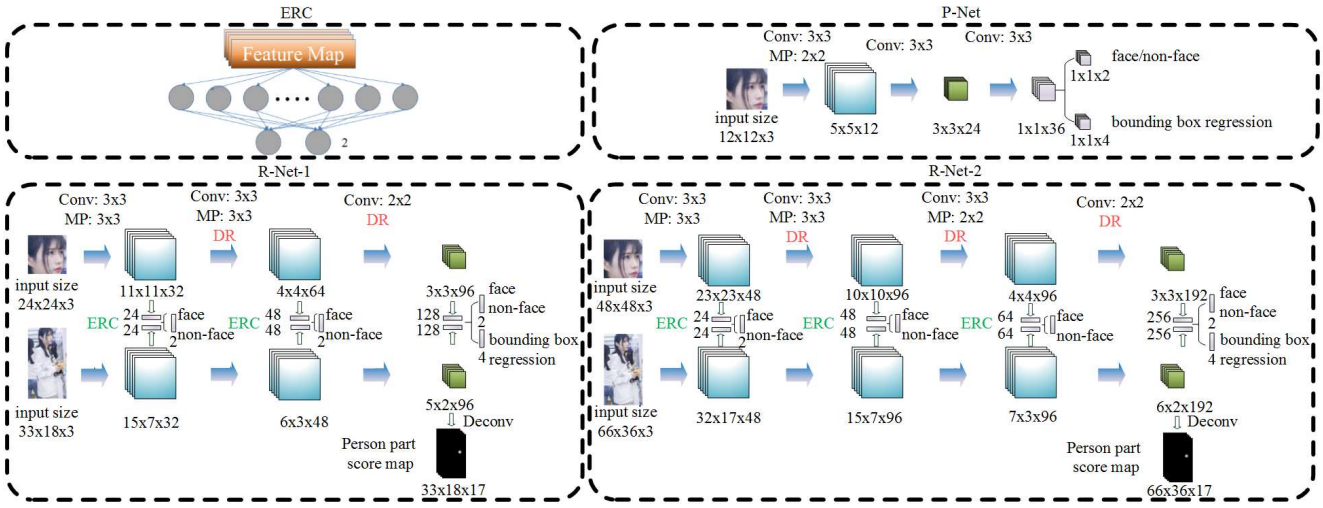


Figure 3. An example of neural network in ERC and CNN architectures of P-Net, R-Net-1 and R-Net-2. ERC denotes Early Reject Classifier. DR denotes data routing layer. MP denotes max pooling. PReLU [6] is used as activation function.

pass. The remaining samples will continue in the following layers. In other words, DR layer will change the sample set for the following network components. Let Ω_j be the set of samples retained in j-th DR layer (Ω_0 is the whole training set), we have:

$$\Omega_j = \Omega_{j-1} - \Omega_j^R, \tag{4}$$

where Ω_j^R denotes the set of samples rejected in the j-th DR layer. We have a sample $i \in \Omega_j^R$ if $p_j^i < \theta$. The experiment and evaluation on θ's sensitiveness are presented in Sec. 6.2.

4.3. Training Process

In addition to ERC classifiers, there is a final face and non-face classifier and a bounding box regressor after the last convolutional layer. The CNN with ICS can be optimized using regular stochastic gradient descent [10] and the optimization of different layers are different due to the different training samples sets selected by the DR layers. In this way, deeper layers' optimization is guided by difficult samples.

4.4. Testing Process

In the testing process, each sample will go through the forward pass of the network until it is rejected by one of the DR layers. Easy non-face samples will be rejected in shallower layers while faces and difficult non-faces samples will be discriminated in the deeper layers or the final classifier with bounding box regression. This strategy actually accelerates the detection process since the easy non-face samples (huge numbers in practice) can be rejected in early layers.

5. Two-stream Contextual CNN

In this section, we will introduce the proposed two-stream contextual CNN and Body Part Sensitive Learning (BPSL) that jointly optimizes body parts localization and face detection to help the CNN to exploit body information adaptively in large visual variations.

5.1. Network Architectures

The network architectures of R-Ne1 and R-Net2 are shown in Fig. 3. In the two-stream contextual CNN, we use two images (face and body regions) as input. The body region is roughly cropped according to the face location predicted in the previous stage. These two inputs are fed to face CNN and body CNN separately. Then we concatenate the features from the last fully-connected layers in these two CNNs and pass them to a classifier to make face/non-face classification and a regressor for bounding box regression. In this way, CNN can exploit not only the face but also body information.

5.2. Body Part Sensitive Learning

As above, the body region is roughly cropped according to face location predicted in last stage. However, there may exist large visual variations of this additional region, such as occlusions for the body, large human pose change, or even the absence of the body. Hence, we propose to use a body CNN to model the appearance of the body parts. In particular, we aim to learn the CNN filters that are sensitive to the body parts and showing discriminative appearance in convolutional features. Such that the extracted features can assist face detection adaptively. This is different from the existing method [28] that simply uses a larger exterior

region for classification.

For body part localization, using CNN to generate body part score map is very prevalent [3, 1, 2] and thus we use the body part score map as supervision signal in our methods. It will encourage CNN to learn visual body appearance related filters and naturally formulates the cases where the body parts are occluded or even whole body region is absent. Specifically, in training processing, after the last convolutional layer in body CNN, there is a deconvolutional layer that generates multiple body part score maps (each score map indicates a kind of body part, see Fig. 3). The score maps are defined as Gaussian distributions around the annotated body joint location. For the predicted score maps and ground truths, we use Euclidean loss as the loss function

$$E = \sum_{i=1}^{n} \sum_{j=1}^{m} \left|\left|\hat{y}_j^i - y_j^i\right|\right|_2^2, \tag{5}$$

where n denotes the number of score maps (i.e., body joints), m denotes the number of pixels in each score map, and $\hat{y}_j^i$ and y_j^i denote the predicted score and ground-truth of j-th pixel in i-th map being a body part, respectively.

In the training process, only examples annotated with body part location will be passed to deconvolutional layer for the prediction of body part score maps. The face CNN and body CNN are trained jointly.

6. Experiments

In the experiments, we will first present the implementation details (Sec. 6.1) and discuss the impact of threshold θ (Sec. 6.2) and the loss weight λ (Sec. 6.3) of body part localization (Eq. (5)). Then, we evaluate the effectiveness of body part sensitive learning in variant body poses or closed up faces without body region in Sec. 6.4. Furthermore, we evaluate the effectiveness of jointly using inside cascaded structure and body part sensitive learning in Sec. 6.5. In Sec. 6.6 and 6.7, extensive experiments are conducted on two challenging face detection benchmarks (FDDB [7] and WIDER FACE [26]) to verify the effectiveness of the proposed approach over the state-of-the-art methods. In Sec. 6.8, we compare the runtime efficiency of our method and other state-of-the-art methods.

Dataset statistics. FDDB contains the annotations for 5,171 faces in a set of 2,845 images. WIDER FACE dataset consists of 393,703 labeled face in 32,203 images. In WIDER FACE, 50% of the images are used for testing, 40% for training and the remaining for validation. The validation and testing set is divided into three subsets according to their detection rates on EdgeBox [32]. COCO [15] contains 105,968 person instances labeled with 17 kinds keypoints (e.g. eyes, knees, elbows, and ankles).

6.1. Implementation details

The architectures of the three CNNs are shown in Fig. 3. P-Net, R-Net-1, and R-net-2 are trained with the batch size of 6000, 1000, and 500 respectively. For P-Net and R-Net-1, the learning rate starts from 0.1, and divided by 5 at the 20K, 40K, and 60K iterations. A complete training is finished at 70K iterations. For R-net-2, the learning rate starts 0.01, and divided by 5 at the 25K, 40K, 50K, and 60K iterations. A complete training is finished at 70K iterations.

For face/non-face classification and bounding box regression, we construct training and validation data set from WIDER FACE in our experiments. For the P-Net, we randomly collect positive samples with Intersection-over-Union (IoU) ratio above 0.65 to a ground-truth face and negative samples with IoU ratio less than 0.3 to any ground-truth faces. In particular, there are 4,000,000 training images and 1,000,000 validation images collected from the WIDER FACE training and validation images, respectively. And the negative/positive ratio is 3:1. For R-Net-1, we use stage1 in our detection framework as an initial face detector to collect training images from WIDER FACE. For R-Net-2, we use a similar way with stage1 and stage2 to collect training images from WIDER FACE.

For body part sensitive learning, we first use MTCNN [29] to detect faces in COCO [15]. Then we generate the body part score maps from all person instances labeled with keypoints as training data. In each mini batch, the number of images for body part localization is equal to 25% numbers of images for face/none-face classification.

6.2. Experiments on the threshold θ

Parameter θ denotes the threshold probability of being a face used in DR layer. If the probability is lower than θ, the sample will be rejected as a negative sample and stop being processed in forward pass. As discussed above, ICS helps to train a more powerful face/non-face classifier. Therefore we evaluate the classification accuracy on the validation set (for details about the validation set see Sec. 6.1).

In this experiment, to remove the effect of body part sensitive learning, we fix the loss weight λ to 0 and vary θ from 0 to 0.02 to learn different R-Net-2 models. The accuracies of these models on constructed validation set are shown in Fig. 4. It is clear that the accuracy first increases and then decreases along with θ raising. It is a trade-off to set proper θ to keep high recall in DR layer and utilize ICS to reject negatives as early as possible. In addition, please be noted that if we set θ as 0, it is equivalent to deeply supervised net [11] that gets lower accuracy.

Finally, we set θ as 0.01 for both R-Net-1 and R-Net-2. Though 0.01 seems small, it can help DR layer to reject nearly 70% negative samples before the last classifier.

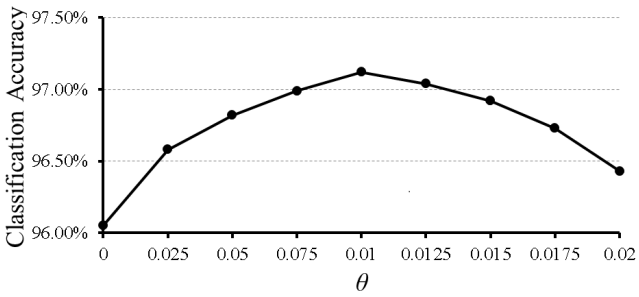


Figure 4. Comparison of face/non-face classification accuracy of models ($\lambda = 0$) trained with different θ on validation set. Note that when $\theta = 0$, it is equivalent to deeply supervise [11].

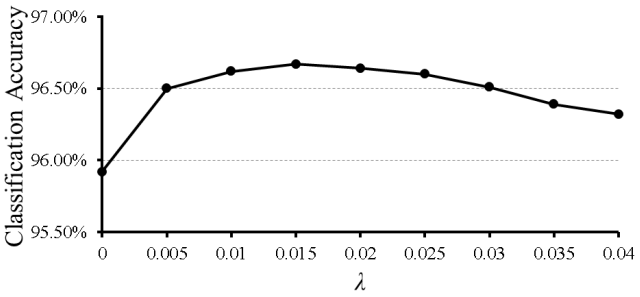


Figure 6. Comparison of face/non-face classification accuracy of models (without ICS) trained with different λ on validation set.

6.3. Experiments on the loss weight λ

Parameter λ is the loss weight of body part localization (Eq. (5)). It is used to balance the body part localization loss, face/non-face classification loss, and bounding box regression loss. In body part localization, the loss is the sum of all Euclidean loss computed in each pixel of the score maps (Eq. (5)) and thus its scale is much larger than that of face/non-face classification and bounding box regression, Hence we have to set a relatively small λ to normalize such a large-scale loss. The contribution of BPSL is also to train a more powerful face/non-face classifier. So, we use the same experiment setting as Sec. 6.2.

In this experiment, we do not use ICS (i.e., ERC and DR layer) and vary λ from 0 to 0.04 to learn different R-Net-2 models. The accuracies of these models on validation set are shown in Fig. 6. The accuracy first increases and then decreases. This is because the body CNN will focus more on localizing body part and less on exploiting contextual information for face detection. Therefore, we fix λ to 0.015 for both R-Net-1 and R-Net-2 in other experiments.

6.4. Effectiveness of body part sensitive learning in variant body poses or without body region

Our method learned both the body parts locations and whether the body parts are presented or not to adaptively exploit the contextual body information. Thus, our method also performs well for variant body poses and faces without body region. To verify this, we select 400 faces with variant body poses (e.g, lying, doing sport) and another 400 faces without body region (i.e. absent or occluded) from the FDDB dataset for evaluation. The evaluation results of only using face CNN and using two-stream CNNs with/without BPSL (i.e., localize body part in training) are shown in Fig. 5. These results indicate that using BPSL can achieve significant performance improvement in large body pose variation and is robust to faces without body region.

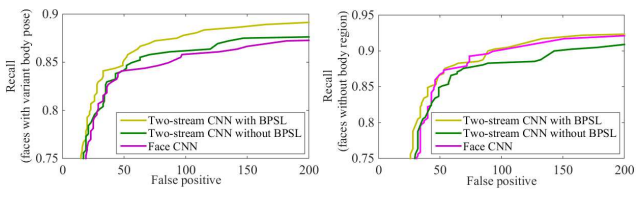


Figure 5. Evaluation of BPSL on face detection with large variant body poses (left) and faces without body region (right). Best viewed in color.

6.5. Effectiveness of jointly using inside cascaded structure and body part sensitive learning

To evaluate the contribution of jointly using the inside cascaded structure (ICS) and body part sensitive learning (BPSL), we train four R-Net-2 networks with and without ICS (i.e., ERC and DR layer) and BPSL (i.e., localize body part in training). We use the same experiment setting as Sec. 6.2 and 6.3. Table 1 shows the accuracy of four different R-Net-2 networks on the validation set ('Baseline' denotes neither use ICS nor BPSL). It is obvious that jointly using ICS and BPSL significantly improve the accuracy. In particular, ICS significantly improves positives accuracy. It demonstrates that the last classifier can handle more difficult faces since most faces and only a few very difficult non-face samples are passed to the last classifier.

We also evaluate the overall detection performance improvement of using ICS and BPSL. We first train four R-Net-1 networks and four R-Net-2 networks with and without ICS and BPSL. Then we compare the overall performance of our framework on FDDB shown in Fig. 7. It is obvious that jointly using ICS and BPSL can significantly improve overall detection performance.

6.6. Evaluation on FDDB

We evaluate the performance of our face detection method on FDDB against the state-of-the-art methods [14, 18, 29, 25, 16, 9, 23, 13, 4, 21, 12, 27]. The results of performance comparison are shown in Fig. 7, which demonstrate the state-of-the-art performance of the proposed method.

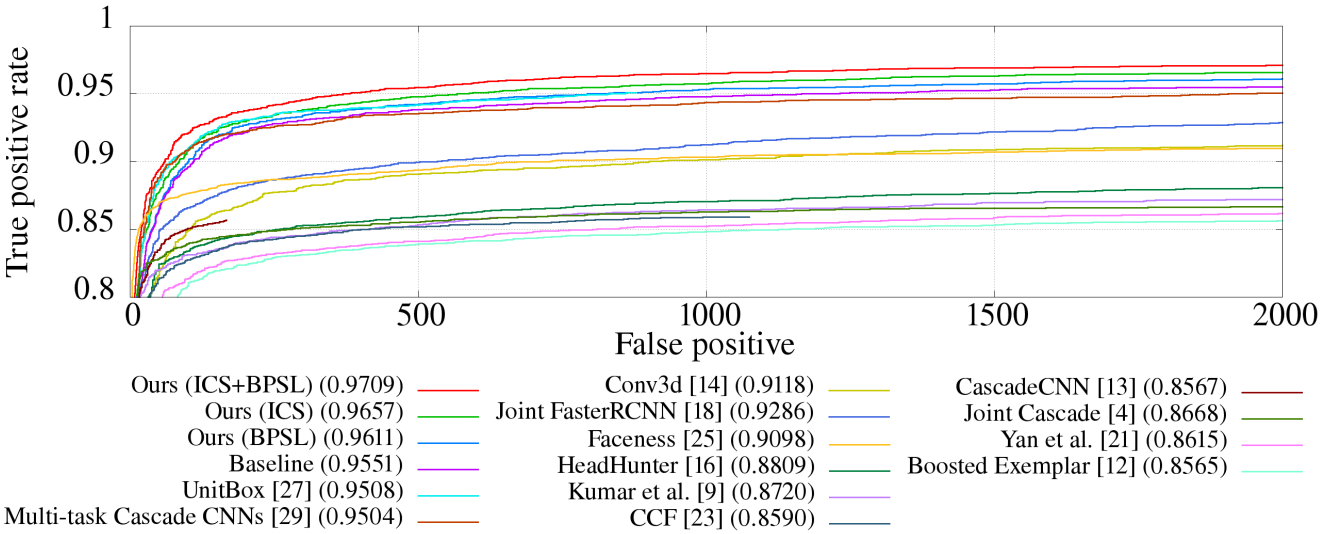


Figure 7. Receiver Operating Characteristic curves (ROC) obtained by our proposed method (with different proposed components) and other techniques on FDDB. ICS denotes inside the cascaded structure (i.e., ERC and DR layer). BPSL denotes body part sensitive learning (i.e., localize body part in training). 'Baseline' denotes neither use ICS nor BPSL. Best viewed in color.

Method	Overall	Positives	Negatives
Baseline	95.92%	90.78%	97.63%
BPSL	96.67%	91.23%	98.48%
ICS	97.12%	92.18%	98.76%
ICS+BPSL	**97.43%**	**92.52%**	**99.06%**

Table 1. Comparison of face/non-face classification accuracy of different proposed components on validation set. "Baseline" denotes neither use ICS nor BPSL.

Some examples of face detection results are shown in Fig. 8 (a).

6.7. Evaluation on WIDER FACE

WIDER FACE is a more challenging benchmark than FDDB in face detection. It is divided into three subsets (Easy set, Medium set, and Hard set) based on their detection rates with EdgeBox [32]. We compare our proposed method against the state-of-the-art methods [26, 25, 29] on the three subsets. Fig. 8 (a) shows some examples of face detection results and Fig. 9 shows the comparison result. It is very encouraging to see that our model consistently achieves the competitive performance across the three subsets. Especially on the hard set, our method can achieve a significant performance improvement over the state-of-the-art. Interestingly, our method gets significant improvement on hard set but is just comparable to the best-performing one on easy and medium sets. A major reason is that our method successfully detects many very hard faces, but some of which are miss-labeled ones in the three sets (i.e, the annotators miss these faces). These miss-labeled faces with high detection scores (some examples are shown in Fig. 8) will decrease the recall in high precision areas of the precision-recall curves.

6.8. Runtime Efficiency

Given the inside cascade structure, our method can achieve high speed by rejecting many negative samples in early stages. We compare our method with several state-of-the-art techniques for typical 640×480 VGA images with 20×20 minimum face size and the results are shown in Table 2. We achieve about 40 FPS on GPU and 12 FPS on CPU. Such computation speed is quite fast among the state-of-the-art. It is noted that our current implementation is based on un-optimized MATLAB codes.

Method	GPU Speed	CPU Speed
UnitBox [27]	12 FPS (Tesla K40)	-
Faceness [25]	20 FPS (Titan Black)	-
MTCNN [29]	99 FPS (Titan Black)	16 FPS
Ours	40 FPS (Titan Black)	12 FPS

Table 2. Speed comparison with other state-of-the-art methods. CPU speed is based on Intel-4770K.

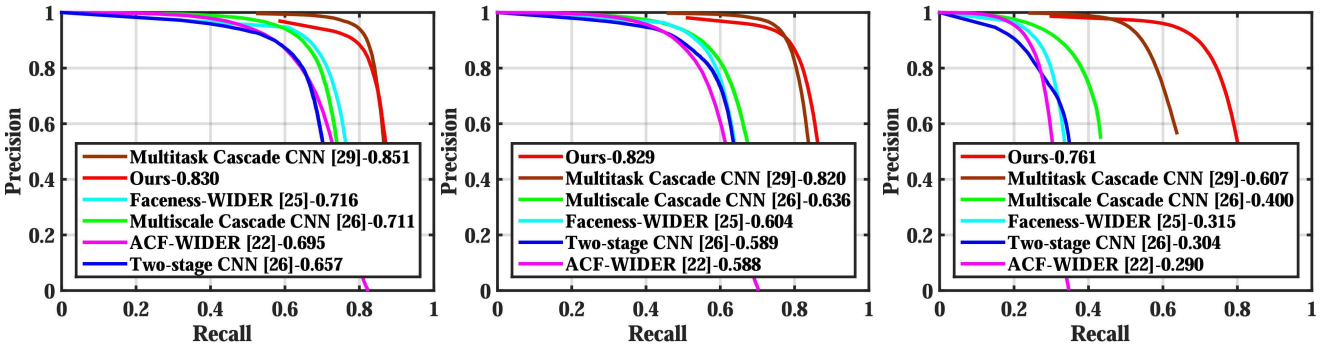
7. Conclusion

In this paper, we develop two new strategies to improve the performance of cascaded CNN for face detection. First, we propose the inside cascaded structure (ICS) that constructs cascaded layer classifies inside a CNN to rejects negative samples layer wise. It encourages deeper layers

(a)

(b)

Figure 8. (a) Examples of face detection results on FDDB (first row) and WIDER FACE (second row). (b) Examples of some false positives (green) obtained by our proposed method and ground-truths (red) on WIDER FACE validation set. The red number is the probability of being a face obtained by R-Net-2. The cases in the first row fail because of miss-labeling and the cases in the second row fail due to the large variations of bounding box annotation.



(a) (b) (c)

Figure 9. Precision-Recall curves obtained by our proposed method and the other strong baselines on WIDER FACE. (a) Easy set, (b) Medium set and (c) Hard set. Best viewed in color. All methods above use the same training and testing protocol. Our method achieves the state-of-the-art results on hard set by a large margin and competitive result on other two sets. Best viewed in color.

to focus on handling difficult samples, while utilizes shallower layers to reject easy non-faces quickly. In particular, we propose the data routing training approach to end-to-end train ICS. In addition to ICS, we propose to jointly optimize body part localization and face detection in a two-stream contextual CNN to improve the robustness of our model. Finally, we develop a unified framework to combine these two components which achieve the competitive performance on the challenging FDDB and WIDER FACE face detection benchmarks while keeps real time performance.

Acknowledgement. This work is mainly conducted when the first author interned in Shenzhen Institutes of Advanced Technology. This work was supported in part by National Natural Science Foundation of China (U1613211,61472410), Guangdong Research Program (2015B010129013,2014A030313688) and External Cooperation Program of BIC Chinese Academy of Sciences (172644KYSB20150019,172644KYSB20160033).

References

[1] N. Alejandro, Y. Kaiyu, and D. Jia. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[2] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[3] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *CVPR*, June 2016.

[4] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *ECCV*, pages 109–122. Springer, 2014.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[7] V. Jain and E. G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*, 2010.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[9] V. Kumar, A. Namboodiri, and C. Jawahar. Visual phrases for exemplar face detection. In *ICCV*, pages 1994–2002, 2015.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[11] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, volume 2, page 6, 2015.

[12] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. In *CVPR*, pages 1843–1850, 2014.

[13] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *CVPR*, pages 5325–5334, 2015.

[14] Y. Li, B. Sun, T. Wu, Y. Wang, and W. Gao. Face detection with end-to-end integration of a convnet and a 3d model. In *ECCV*.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[16] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, pages 720–735. Springer, 2014.

[17] M.-T. Pham, Y. Gao, V.-D. D. Hoang, and T.-J. Cham. Fast polygonal integration and its application in extending haar-like features to improve object detection. In *CVPR*, pages 942–949. IEEE, 2010.

[18] H. Qin, J. Yan, X. Li, and X. Hu. Joint training of cascaded cnn for face detection. In *CVPR*, pages 3456–3465, 2016.

[19] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996, 2014.

[20] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[21] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *CVPR*, pages 2497–2504, 2014.

[22] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *IJCB*, pages 1–8. IEEE, 2014.

[23] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *ICCV*, pages 82–90, 2015.

[24] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, pages 2129–2137, 2016.

[25] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, pages 3676–3684, 2015.

[26] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *CVPR*, 2016.

[27] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *ACM MM*, pages 516–520, 2016.

[28] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. In *BMVC*.

[29] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[30] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, volume 2, pages 1491–1498. IEEE, 2006.

[31] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886. IEEE, 2012.

[32] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405. Springer, 2014.