Nested Shallow CNN-Cascade for Face Detection in the Wild

Jingjing Deng Xianghua Xie
Department of Computer Science, Swansea University, United Kingdom
http://csvision.swansea.ac.uk

Abstract— **Face detection in the wild is a challenging vision problem due to large variations and unpredictable ambiguities commonly existed in real world images. Whilst introducing powerful but complex models is often computationally inefficient, using hand-crafted features is hence problematic. In this paper, we propose a nested CNN-cascade learning algorithm that adopts shallow neural network architectures that allow efficient and progressive elimination of negative hypothesis from easy to hard via self-learning discriminative representations from coarse to fine scales. The face detection problem is considered as solving three sub-problems: eliminating easy background with a simple but fast model, then localising the face region with a soft-cascade, followed by precise detection and localisation by verifying retained regions with a deeper and stronger model. The face detector is trained on the AFLW dataset following the standard evaluation procedure, and the method is tested on four other public datasets, i.e. FDDB, AFW, CMU-MIT and GENKI. Both quantitative and qualitative results on FDDB and AFW are reported, which show promising performances on detecting faces in unconstrained environment.**

I. Introduction

View-specific face detection under controlled environment is largely considered a solved problem due to recent advances in object detection, in particular the work by Viola-Jones (VJ) [35]. As a typical detection problem, the class distribution between face and background is extremely unbalanced and heavily biased towards the background. The traditional VJ framework uses a multi-stage cascade detector, where individual stage is a binary classifier separating the face from a subset of background hypotheses. For efficiency, the traditional methods use simple visual features or weak classifiers at multiple stages (typically over 15). However, they perform poorly on the so-called *Face in the Wild* problem, where faces are captured with large pose and facial expression variations, severe occlusions and clutters, and poor lighting scenarios. Built upon those classical detection frameworks, several works have been recently reported in developing discriminative visual features [29], [6], [38] and strong classifiers [11], [40], [24] to improve face detection performances in the wild. Deep learning methods [22], especially Convolutional Neural Networks (CNNs), have shown outstanding successes in representative feature learning and supervised classification for various computer vision problems. Our work leverages recent advances in deep learning for efficient face detection. More in-depth discussions to these related work are presented in the next section.

From image retrieval perspective, face detection can be considered as a visual matching problem, where a window candidate is determined as face by successfully finding reliable correspondences in a pre-built exemplar database. In [33], [23], exemplar database is constructed using localised visual words, and detection is obtained by finding the high confidence regions on the voting map provided by matched exemplars. The performance of those non-parametric searching methods can be severely compromised by the quality of exemplar database, such as the discriminative power of visual features, and the variation in coverage of different poses, illuminations, occlusions and so on. In addition, using a large exemplar database also slows down the detection speed as exploring large search space is a time consuming task. Deformable Part Model (DPM) was originally proposed for object recognition, and can be considered as an alternative searching based method for detecting faces [12], [10]. It considers that the target object is consisted of several deformable parts. The part candidates are proposed by individual part detectors, and then the entire object can be found by searching for a most plausible configuration of displaced parts. DPM helps to overcome the difficulties introduced by severe occlusion and clutter, provided reliable performance of part detectors. However, assembling individual parts into objects is equivalent to solving a combinational optimisation problem which could also be computationally expensive even with approximation algorithms.

Computational efficiency is one of the main concerns for practical detection system, especially when dealing with large number of hypothesis, complex visual feature, and strong classifier. For example, to precisely locate faces in the image, exhaustive search methods, such as sliding window, are commonly used to generate candidates. However, examining all hypotheses is computationally expensive, thus relatively simple features and weak classifiers are typically used to reduce the complexity [35], [6]. It is worth noting by taking this approach the detection problem is divided into a set of sub-problems first and then solved by combining individual sub-problem solver into a multi-stage detector [35], [3]. For example, Koestinger [19] trained a 20-stage VJ face detector using Local Binary Patterns (LBP) features. Object region-proposal methods are popular for image recognition and object localisation, such as objectness [1], selective search [34], category independent object proposals [8], combinatorial grouping [2], and segmentation based methods [4]. However, generating object candidates generally involves region segmentation, classification, and grouping, which slow down the detection speed drastically. Furthermore, the recall rate of region proposal is generally lower than exhaustive search, such as sliding window.

In this paper, we present a multi-resolution face detector, which embeds 5 shallow CNN classifiers into a nested cascade framework. Detecting faces in image is carried out in three phases: (1) A large amount of easy background patches from the whole hypothesis set generated by sliding window are eliminated at the very early stage using a shallow but fast net at a coarse scale. (2) A nested soft cascade with 3 nets is used to further reject hard false positive hypotheses while keeping a high recall rate. (3) To precisely locate the face region, all retained hypotheses from previous stages are verified by a deeper net using higher resolution.

The rest of paper is organised as follows: Sec. II reviews related works on CNN-based face detection methods. Sec. III provides detailed descriptions of proposed method, including network architectures for individual stage. Training strategies, parameter settings, and experimental results on three public datasets are presented in Sec. IV, and concluding remarks are provided in Sec. V.

II. Related Work

Applying Neural Networks (NNs) to face detection dates back to at least early 1990s [17], [39], [13]. Back then, training a multi-layer neural networks was difficult as the number of parameters increases exponentially with the number of layers. However, Deep Neural Network (DNN) is becoming more and more mainstream [22], as it has been shown superior over many other methods, especially for visual recognition tasks. The following can be considered as three of the key reasons that contributed to the success of DNNs. First, training a multi-layer neural network involves finding a local minimum of a highly non-linear function. In order to obtain a reasonable local minimum, gradient descent based methods require a good initialisation. Layer-wise unsupervised pre-training methods [16] were developed and have been proved to be more efficient compared with random initialisation. Second, a large amount of labelled datasets [32], [9], [26] are vitally important to the advance in supervised training. For example, Microsoft COCO dataset [26] contains more than 300,000 images, and over 2,000,000 instances from 80 object categories, where each image has 5 caption labels. Moreover, advances in hardware makes both forward pass and backward propagation computationally efficient. Especially, with dedicated high speed memory module and Single Instruction Multiple Data (SIMD) architecture, General-Purpose Graphics Processing Unit (GPGPU) are particularly well placed for learning deep neural network structures [5].

As for face detection, Farfade *et al.* [11] proposed a multi-view face detection method, so-called Deep Dense Face Detector (DDFD), which uses a fine-tuned 8-layer AlexNet [21] that was initially designed for object recognition. It has 5 convolutional layers and 3 fully connected layers. A pre-trained AlexNet was fine-tuned for face detection on 200,000 face patches and 20,000,000 background patches, which were all resized to 227×227 pixels in order to match the input size of AlexNet. During the testing stage, the sliding window approach was used to generate hypotheses. DDFD classifies each candidates into face or background, and decision confidence scores is obtained. Non-Maximal Suppression (NMS) was followed to remove redundant bounding boxes. In [40], the authors introduced a deep CNN based deformable part model for face detection. The whole face is decomposed into 5 facial regions: hair, eye, nose, mouth and beard. The part detectors are constructed using 5 binary CNN classifiers that shared the same deep layers for computational efficiency. The window candidates are generated using object proposal methods, such as selective search [34]. The confidence scores of each candidate can then be inferred via examining the spatial configurations of part detector responses. Finally, to further refine the detection results, a CNN with similar architecture to AlexNet is trained for face-background classification and bounding box regression.

Very recently, several works have shown that Regions with CNN features (R-CNN) [14], [30] and Spatial Pyramid Pooling CNNs (SPPnet) [15] are effective in simultaneous object localisation and recognition. These methods contain four main components: convolutional feature extraction, obtaining region proposal, region of interest (ROI) classification, and bounding box refinement. In [14], the authors showed that the representation feature learnt with CNN using deep structure can be effectively used for visual classification and ROI regression. By introducing spatial pyramidal pooling layer to generate a fixed length output feature regardless the size of input image, [15] overcame the limitation of [14] without cropping or wrapping the images that are problematic as they result in information loss and distortion. The work in [30] improved the computational efficiency further by sharing the deep convolutional layers with region proposal, classification and regression networks. However, for small objects, R-CNNs have difficulty to detect them in small scale due to low resolution and the lack of visual context.

Although deeper models generally outperforms shallow ones, training complex models is not a trivial task, especially for binary detection problems where the distribution of target object and background is extremely unbalanced. Given millions of parameters to optimise using back-propagation, deep nets have the tendency to overfit the data, even with strong regularisations such as dropout and batch normalisation. Due to the smaller amount of parameters, training shallow nets is significantly faster. Embedding shallow nets into traditional cascade framework can also significantly reduce the number of stages and drastically increase the discriminative power of the model [7]. One of limitation of shallow nets is that the recall rate drops quickly with the increase of the number of stages. In this paper, we introduce a nest soft cascade to compensate the loss of recall while adding multiple stages to remove false positives.

The most relevant work to ours is [24], where 3 face-nonface classification CNNs are used for separating face regions from background and 3 calibration CNNs are used to refine the location of detected bounding box. Sliding window method is used to generate region candidates. These hypotheses pass through 3 classification-refinement components with different image resolutions, from coarse to fine, and the retained ones are considered as object regions. However, cascade based method has to make a compromise between the number of stages, accuracy and efficiency. For example, in a hard cascade setting, adding more stages helps to reduce false positives, while it decreases the detection rate and speed, especially when a computationally intensive model is used such as CNN. In addition, refining the detected windows between stages introduces re-sampling the patches from the original image, which is non-trivial during the testing phase.

III. Method

Fig. 1 shows the basic flowchart of the proposed nested cascade face detector. It consists three main phases as follows: fast elimination, nested soft cascade, and precise detection. Window patches are firstly generated by densely scanning the input image at multiple scales using sliding windows. Majority of those window patches are quickly eliminated as background by an *ElmNet* using a patch resolution of 12×12. A soft-cascade is built by combining 3 *LocNets* in a weighted fashion, which is used to further reject the hard false positives with a patch resolution of 24×24. Then, all retained candidates from the previous stages are verified by *DetNet* using a patch resolution of 48×48. The final detections is obtained via removing redundant detections with Non-Maximum Suppression (NMS).

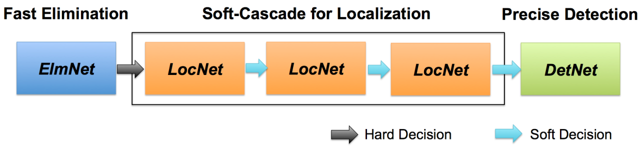

Fig. 1. The pipeline of the proposed nested cascade face detector.

A. ElmNet: Fast Elimination

A large amount of patch candidates are generated by the sliding window method. The *ElmNet* is designed to quickly eliminate negative patches to reduce the computational cost for the following phases. Table I and Fig. 2 provide the details of the architecture for *ElmNet*, where only one convolutional layer and one fully connected layer are used. Adopting such simply CNN structure is motivated by the following two reasons. Firstly, *ElmNet* has a small input size of 12×12, a small kernel size of 3×3, and a small number of filters of 16. Compared to other nets, *ElmNet* has significantly smaller number of parameters, which enables a lower memory consumption and a much lower computational cost. Secondly, at this fast elimination stage, low frequency image features extracted from coarse spatial resolution is more effective in rejecting easy negative hypothesis. Since there is no hierarchical feature extraction within *ElmNet*, the discriminative power is limited. In order to retain most positive windows for the following stage, a high recall rate can be achieved by shifting the decision boundary of Softmax layer towards zero. For example, using a minimal face size of 48×48, 87.16% recall can be achieved by shifting the decision boundary to 0.01, whereas 72.62% recall is achieved with 0.50.

TABLE I
The network architecture of *ElmNet* for fast elimination.

No	Layer Type	Parameter Setting
1	Image Input	12x12x3 images scaled to the range [0,1]
2	Convolution	16 3x3 filters with stride 1
3	ReLU	Rectified linear unit
4	Max Pooling	3x3 filter with stride 2
5	Fully connected	Fully connected with 16 outputs
6	ReLU	Rectified linear unit
7	Fully connected	Fully connected with 2 outputs
8	Softmax	Softmax regression for binary classes
9	Classification	Classification output

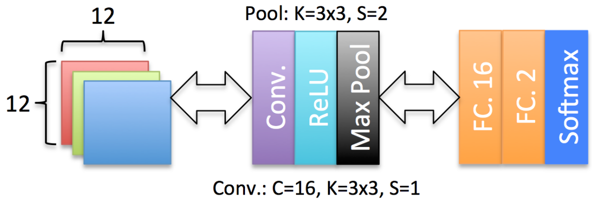

Fig. 2. Network architecture of *ElmNet*.

B. LocNets: Nested Soft-Cascade

Each stage classifier in cascade is trained using the full set of true positives and the false positives passed through previous stages. Although over 90% of negative patches are eliminated by *ElmNet* at the first stage, the number of retained false positives for training following stage is still considerably large, especially when large negative image set is used. In our case, 18,089 negative images are used. In order to retain high recall and remove hard non-face hypotheses further, multiple *LocNets* are trained on different subsets of negative images and then assembled in a soft-cascading fashion, where the final decision confidence is a weighted sum of all Softmax outputs of *LocNets*. Within individual *LocNet*, see Table II and Fig. 3, there are two levels of feature abstraction using convolution layers, each of which is followed by a non-linear mapping and a spatial down-sampling. Such hierarchical network enables more discriminative descriptors being learnt through back-propagation, and lifting up from low-level features to high-level representations. The weights of each *LocNet* are estimated using linear regression by solving an over-conditioned least square problem without the interception term. This linear regression problem can be formally defined as

$$\arg\min_{W} \sum_{n=1}^{N} \|L_n - \sum_{s=1}^{S} W_s \times C_{sn}\|^2, \tag{1}$$

where W, C, L, S and N denote the weights, probability confidences of face category given by Softmax layers, ground truth labels, the number of LocNet stages, and the number of training samples, respectively. The decision boundary of nested soft-cascade is also shifted to 0.01 in order to achieve a high recall rate.

TABLE II
THE NETWORK ARCHITECTURE OF *LocNet* FOR PRECISE LOCALISATION.

No	Layer Type	Parameter Setting
1	Image Input	24x24x3 images scaled to the range [0,1]
2	Convolution	16 5x5 filters with stride 1
3	ReLU	Rectified linear unit
4	Max Pooling	3x3 filter with stride 2
5	Convolution	16 5x5 filters with stride 1
6	ReLU	Rectified linear unit
7	Max Pooling	3x3 filter with stride 2
8	Fully connected	Fully connected with 32 outputs
9	ReLU	Rectified linear unit
10	Fully connected	Fully connected with 2 outputs
11	Softmax	Softmax regression for binary classes
12	Classification	Classification output

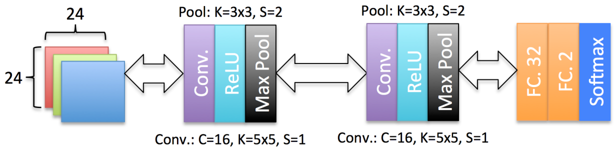

Fig. 3. Network architecture of *LocNet*.

TABLE III
THE NETWORK ARCHITECTURE OF *DetNet* FOR FACE DETECTOR.

No	Layer Type	Parameter Setting
1	Image Input	48x48x3 images scaled to the range [0,1]
2	Convolution	32 5x5 filters with stride 1
3	ReLU	Rectified linear unit
4	Normalisation	Cross channel (9) normalisation
5	Max Pooling	3x3 filter with stride 2
6	Convolution	32 5x5 filters with stride 1
7	ReLU	Rectified linear unit
8	Normalisation	Cross channel (9) normalisation
9	Max Pooling	3x3 filter with stride 2
10	Fully connected	Fully connected with 128 outputs
11	ReLU	Rectified linear unit
12	Fully connected	Fully connected with 2 outputs
13	Softmax	Softmax regression for binary-classes
14	Classification	Classification output

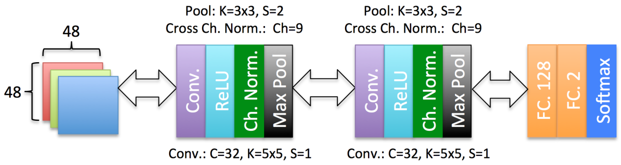

Fig. 4. Network architecture of *DetNet*.

C. DetNet: Precise Detection

DetNet is designed to precisely locate face regions by verifying retained face candidates at a higher image resolution. In order to capture features in detail, the resolution of input, and the number of filters are doubled compared to *LocNet*, while the size of convolutional kernel and the level of feature abstraction are kept as the same (See Table III and Fig. 4) for computational efficiency. Local response normalisation layers are added between the non-linear mapping layer and the maximum spatial pooling layers. Such inhibition scheme is only applied across channels to enforce regularisation to the networks. Since *DetNet* is the last phase of cascade, binary classification is carried out without shifting the decision boundary, and detected square bounding boxes are then refined using a 2-step NMS to remove redundancies. For the detections at the same scale, we iteratively select the detection with highest confidence score and remove the detections that has the intersection over union (IoU) ratio larger than 0.50 with selected window. Then, for the detections at different scales, the redundancies can be found by measuring the intersection over minimum (IoM) ratio, where the threshold is set to 0.90. The first step removes the redundant detections that are spatially offset to the correct location, and the second step enables removing redundancies in scale.

IV. EXPERIMENT AND DISCUSSION

A. Detector Training

The AFLW (Annotated Facial Landmarks in the Wild [20]) dataset was used to train the face detector. The dataset contains 22,712 labelled faces out of 21,123 images. The positive face windows were further augmented by horizontal flipping. In total, 45,424 faces were used in the training procedure, and examples of face images are shown in Fig. 5 (a). The negative images contain no face. To bootstrap non-face images, labelled face windows were replaced with non-face patches which were randomly sampled from PASCAL VOC dataset [9] (the person subset was excluded). In total, 19,458 negative images were generated using this bootstrapping approach. However, there are considerable amount of unannotated faces in AFLW dataset, we thus further applied Koestinger's VJ-LBP detector [19] on the negative images. After those ones which have true positive response were removed, the negative set contains 18,089 images.

To train *ElmNet*, 904,450 non-face samples were cropped randomly from all negative images (50 patches per image), and then resized to 12×12. With cascading set-up, the negative samples for training the next stages were the residuals (false positives) generated by densely scanning the negative image set using all previous stages. It is useful to set a maximum negative-positive ratio (MNP) for LocNets and DetNet. For example, *ElmNet* would generate over 50 million false positives from 18,089 images, where MNP can thus avoid training with extremely imbalanced data. In our case, we used 48×48 scanning window with the stride of 16 pixels, scale factor of 1.18, and MNP of 10. All networks were trained using back-propagation with batch stochastic gradient descent.

B. Evaluation on FDDB Dataset

The proposed face detector was quantitatively evaluated on the Face Detection Dataset and Benchmark (FDDB) [18] dataset that contains 5,171 annotated faces in 2,845 images. The quantitative results were generated following the stan-

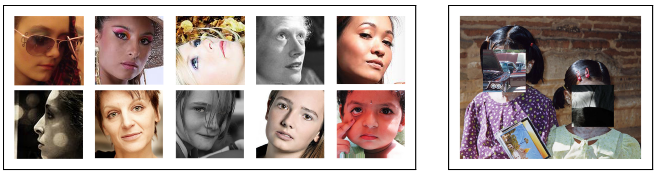
(a) (b)

Fig. 5. Examples training images. (a) Positive images are cropped face from AFLW dataset; (b) negative images are generated by replacing the face region with non-face patches sampled from PASCAL VOC datasets.

dard evaluation procedure with the software provided by the authors. For discrete score evaluation, the detections that has over 0.50 IoU with annotations are counted as true positive. Since the groundtruth faces are labelled using ellipses, we also fitted ellipses to our bounding boxes for fair comparison.

Table IV shows the discrete metrics of individual stages of the proposed method using minimal face sizes of 36×36 and 48×48. Over 96% of hypotheses were eliminated, but reasonable recall rate was achieved by ElmNet at the first stage, which ensures deeper network can be computed effectively in the following cascade without overwhelming computational cost. The results of different stages shows that higher resolution and hierarchical feature abstraction are the key to build discriminative models. We also compared proposed method with state-of-the-art methods which are trained on the same dataset. The discrete ROC curves are shown in Fig. 6. DPM based methods, such as Yan et al. [36] and HeadHunter [27] are leading the performance, mainly because the variations of facial parts are relatively small, thus detecting facial parts are more robust than detecting face as a whole. Especially, HeadHunter [27] reports the optimal results that obtained through comprehensive studies on training strategies and parameter settings. However, DPM methods require training part detectors, and searching optimal configuration, which make building the detector a laborious, time-consuming task, and are known to be much slower than cascade based methods. ACF-Multiscale [38] method aggregates multiple features, such as colour, gradient, local histogram, into a rich representation, and then trains multiple soft cascade with depth-2 decision tree for different views. It shows that combining multiple models and features outperforms a single model. The computational cost of aggregating feature channels is considerably more. Significantly, Koestinger [19] shows that without rich features, the performance of multi-view based method drops by a significant margin. In addition, sophisticated post-processing is required to combine the multiple detection outputs given by detectors of different views. The proposed method requires no model aggregation. The features are self-learnt through training, and it outperforms the traditional methods which use the cascade framework such as NPDFace [25]. Also image retrieval based methods suffer from efficiency issue much more severely. For example, to process an image of size 1480×986 with minimal face size 80×80, Boosted Exemplar [23], and XZJY [33] take 900ms and 33000ms receptively, whereas our methods only takes 153ms using a non-optimised Matlab implementation.

Qualitative results on the FDDB dataset are shown in Figs. 7, 8, and 9. Red and blue ellipses represent groundtruth and true positives, whereas yellow and green ellipses represent false positives and false negatives respectively. Fig. 7 illustrates some examples of typical detection results with large pose and facial expression variations, blurring, and severe occlusion and clutter. Fig. 8 shows some examples of false positives and false negatives. The false positives are usually observed at the region that contains partial face, and false negatives are mainly caused by severe blurring and faces in small scale. Fig. 9 shows some interesting detections in yellow, which are counted as false positives since there are no annotations to match. However, they are in fact correct detections.

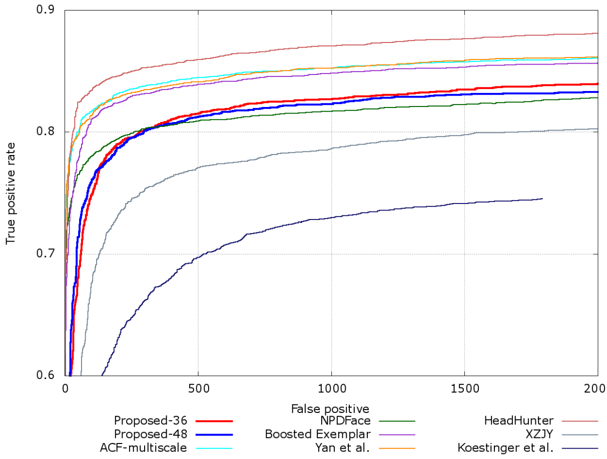

Fig. 6. ROC curves of the proposed detector and recent methods on FDDB database with the discrete score metric.)

TABLE IV
RECALL RATE AND NUMBER OF FALSE POSITIVES OF INDIVIDUAL DETECTION STAGE OF THE PROPOSED METHOD ON FDDB DATASET.

Stages	36×36 minimal face		48×48 minimal face	
	Dis. Recall	#FP	Dis. Recall	#FP
Hypothesis	95.16%	17843K	91.94%	16033K
ElmNet	90.17%	471K	87.16%	314K
S1-LocNet	88.05%	114K	83.52%	64K
S2-LocNet	85.48%	42K	81.63%	28K
S3-LocNet	83.10%	23K	79.37%	16K
Soft-LocNets	88.74%	117K	85.84%	78K
DetNet	82.38%	723	80.89%	450

C. Evaluation on AFW Dataset

We quantitatively evaluated our face detector on another face detection benchmark, namely Annotated Face in the Wild (AFW) [41] that contains 205 images, and 468 annotated faces. 97.43% recall rate was achieved by our face detector, which is slightly lower than CNN-Cascade [24] (97.97%, +0.54%), but outperforms other state of the art methods, such as DPM [12] (97.21%, -0.22%,), HeadHunter [27] (97.14%, -0.29%), Structured Models [37] (95.19%, -2.24%), Shen et al. [33] (89.03%, -8.4%), and

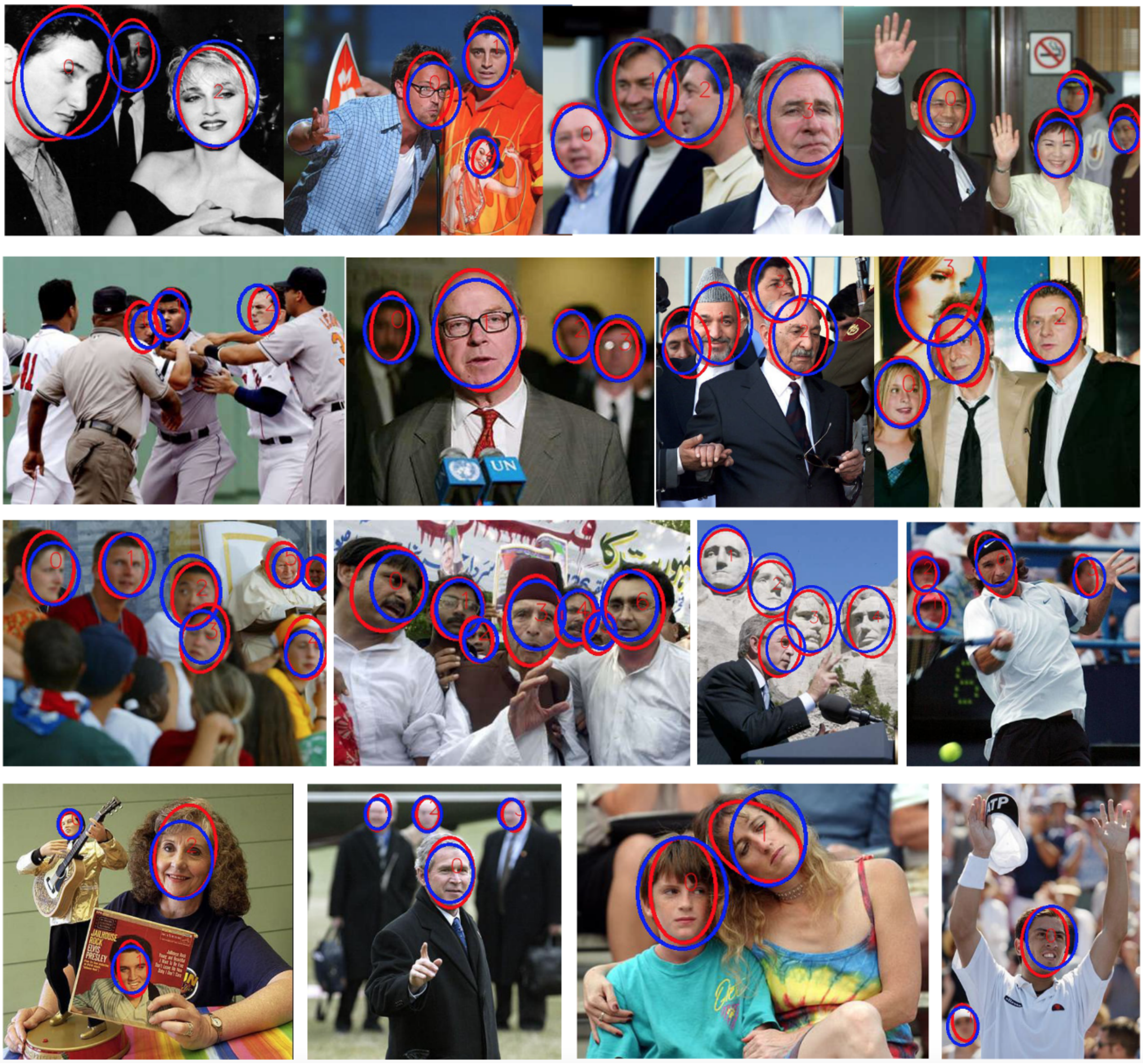

Fig. 7. Typical detection results on FDDB dataset (red: ground truth, blue: true positive).

TSM [41] (87.99%, -9.44%). Qualitative results are shown in Fig. 10, where square detection bounding boxes were used to match the original annotations.

D. Evaluation on CMU-MIT & GENKI Datasets

The proposed method was also evaluated on two early face detection benchmarks, CMU-MIT face dataset [31], and GENKI database [28]. Several examples of typical detection results are presented in Figs. 11 and 12. CMU-MIT dataset contains a total of 511 faces from 130 grey-scale images. The top right image in Fig. 11 shows that our method is able to tolerate rotation variance, and there are only one false negative and two false positives in the top right image. Current release of GENKI database contains two subsets, where GENKI-4K subset contains 4,000 images, and GENKI-SZSL subset contains 3,500 images. Some detection examples with different poses and facial expressions are shown in Fig. 12.

E. Detection Speed

The proposed detector was implemented and evaluated on *Matlab 2016* using two different GPUs, GeForce GTX TITAN X (Maxwell) and Quadro K2000, which have 3,072 CUDA cores with 12GiB memory and 384 CUDA cores with 2GiB memory respectively. Table V shows the running speed of individual stages. It can be observed that TITAN X outperforms K2000 as more CUDA cores and GPU memory are available. The computation time increases as the complexity of the model increases. To processes one 640×480 VGA image with the size of minimum face of 80×80, our method takes 40.1ms using CPU only, whereas [24] takes 71ms on average.

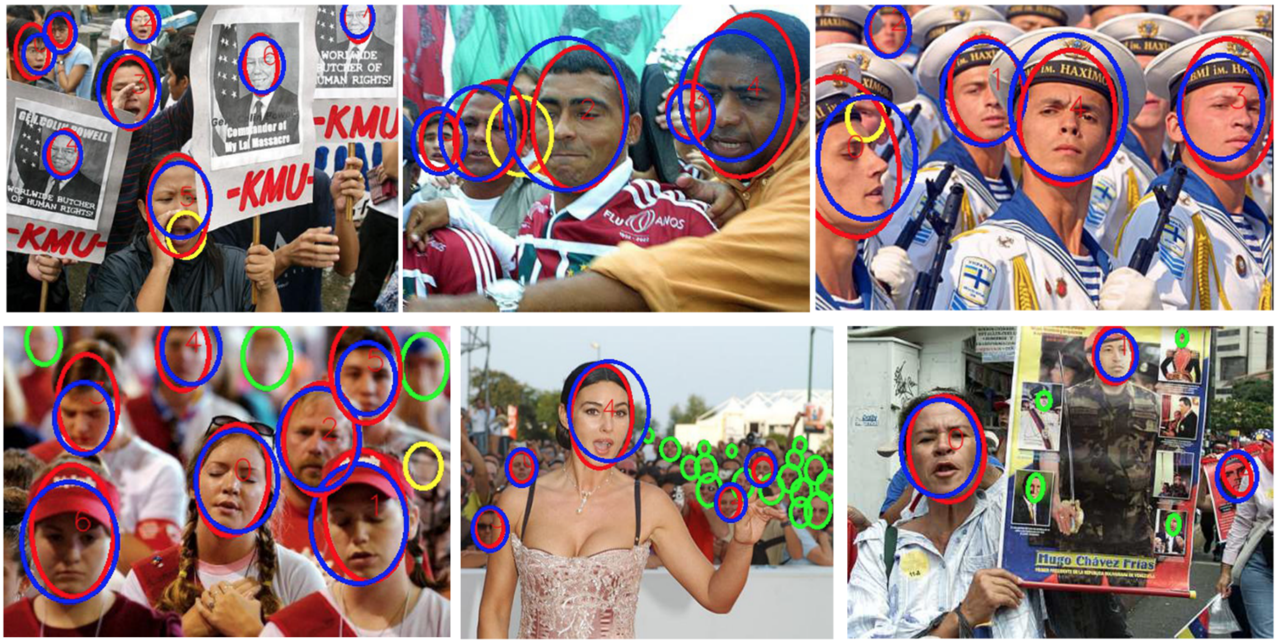

Fig. 8. Examples of false positives and false negatives on FDDB dataset (red: ground truth, blue: true positive, yellow: false positive, green: false negatives).

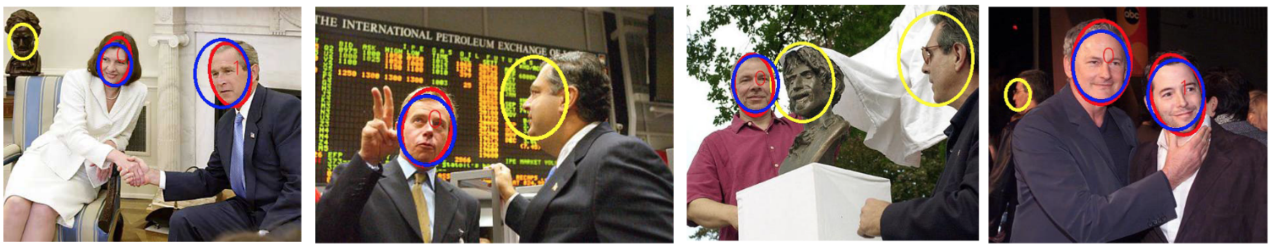

Fig. 9. Examples of correct detections but counted as false positives (red: ground truth, blue: true positive, yellow: false positive).

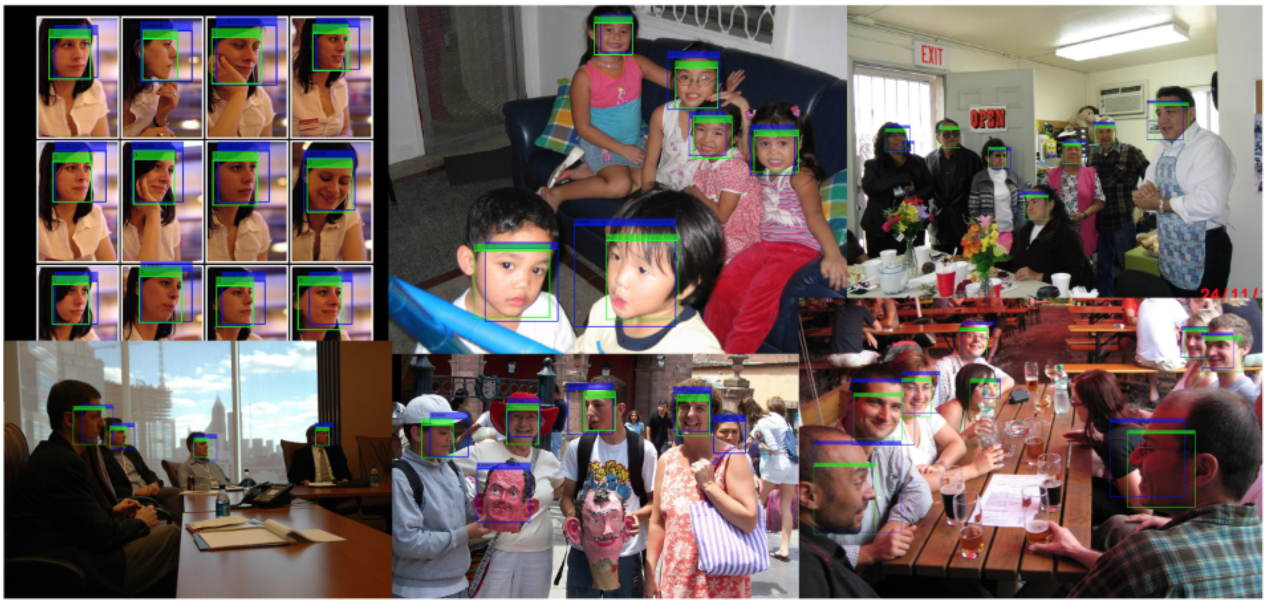

Fig. 10. Examples of qualitative results on AFW dataset. (green: ground truth, blue: detection results of the proposed method).

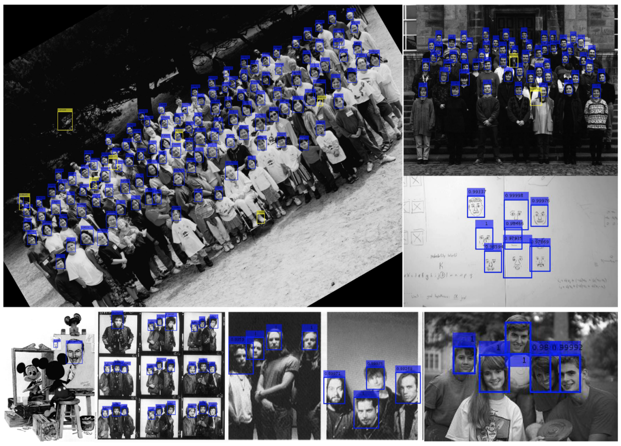

Fig. 11. Examples of qualitative results on CMU-MIT dataset.

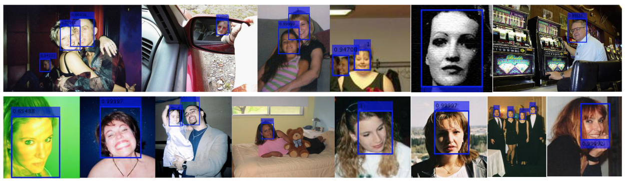

Fig. 12. Examples of qualitative results on GENKI database.

TABLE V
SPEED OF INDIVIDUAL STAGE (HYPOTHESES/SECOND)

	ElmNet	LocNet	DetNet
TITAN Maxwell	102,380 ± 4,964	71,844 ± 574	17,599 ± 99
Quadro K2000	61,112 ± 2799	22,988 ± 415	2,383 ± 9

V. Conclusion

We proposed an efficient multi-stage cascade method that is well suited for binary detection problems, where the number of positive samples is significantly smaller than negative samples. Instead of resorting to deep structures that are time consuming and laborious to train, the proposed nested shallow CNN-cascade overcomes these difficulties by solving three sub-problems from easy to hard using models from weak to strong. In addition, a nested soft cascade is introduced to compensate the loss of recall when multiple classifiers are used to reject a large amount of negatives. The proposed method was evaluated on three datasets including FDDB and AFW. Quantitative and qualitative results show promising performances on detecting face in unconstrained environment with much improved efficiency compared to state of the art.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 34(11):2189–2202, 2012.

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014.

[3] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *CVPR*, volume 2, pages 236–243, June 2005.

[4] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 34(7):1312–1328, 2012.

[5] S. Chetlur et al. CUDNN: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *ICCV*, volume 1, pages 886–893. IEEE, 2005.

[7] J. Deng, X. Xie, and M. Edwards. Combining stacked denoising autoencoders and random forests for face detection. In *ACIVS*, pages 349–360, 2016.

[8] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588. Springer, 2010.

[9] M. Everingham et al. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan 2015.

[10] H. Fang, J. Deng, X. Xie, and P. W. Grant. From clamped local shape models to global shape model. In *ICIP*, pages 3513–3517, Sep 2013.

[11] S. Farfade, M. Saberian, and L. Li. Multi-view face detection using deep CNN. In *ACM ICMB*, pages 643–650, 2015.

[12] P. Felzenszwalb et al. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[13] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *TPAMI*, 26(11):1408–1423, 2004.

[14] R. Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] K. He et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[17] E. Hjelmås and B. K. Low. Face detection: A survey. *CVIU*, 83(3):236–274, 2001.

[18] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, 2010.

[19] M. Koestinger. *Efficient Metric Learning for Real-World Face Recognition*. PhD thesis, 2013.

[20] M. Koestinger et al. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. 2011.

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

[22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

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

[24] 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.

[25] S. Liao, A. K. Jain, and S. Z. Li. A fast and accurate unconstrained face detector. *TPAMI*, 38(2):211–223, Feb 2016.

[26] T.-Y. Lin et al. Microsoft coco: Common objects in context. In *ECCV*. 2014.

[27] M. Mathias et al. Face detection without bells and whistles. In *ECCV*. Springer, 2014.

[28] S. D. MPLab, University of California. The MPLab GENKI Database, GENKI-SZSL Subset, 2009. Accessed: 2016-05-12.

[29] M.-T. Pham et al. Fast polygonal integration and its application in extending haar-like features to improve object detection. In *CVPR*, 2010.

[30] S. Ren et al. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[31] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *CVPR*, pages 38–44, 1998.

[32] O. Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[33] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *CVPR*, pages 3460–3467, 2013.

[34] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

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

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

[37] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *IVC*, 32(10):790–799, 2014.

[38] B. Yang et al. Aggregate channel features for multi-view face detection. In *IEEE IJCB*, 2014.

[39] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *TPAMI*, 24(1):34–58, 2002.

[40] S. Yang et al. From facial parts responses to face detection: A deep learning approach. In *ICCV*, pages 3676–3684, 2015.

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