A Convolutional Neural Network Cascade for Face Detection

Haoxiang Li[†], Zhe Lin[‡], Xiaohui Shen[‡], Jonathan Brandt[‡], Gang Hua[†]

[†]Stevens Institute of Technology
Hoboken, NJ 07030
{hli18, ghua}@stevens.edu

[‡]Adobe Research
San Jose, CA 95110
{zlin, xshen, jbrandt}@adobe.com

Abstract

In real-world face detection, large visual variations, such as those due to pose, expression, and lighting, demand an advanced discriminative model to accurately differentiate faces from the backgrounds. Consequently, effective models for the problem tend to be computationally prohibitive. To address these two conflicting challenges, we propose a cascade architecture built on convolutional neural networks (CNNs) with very powerful discriminative capability, while maintaining high performance. The proposed CNN cascade operates at multiple resolutions, quickly rejects the background regions in the fast low resolution stages, and carefully evaluates a small number of challenging candidates in the last high resolution stage. To improve localization effectiveness, and reduce the number of candidates at later stages, we introduce a CNN-based calibration stage after each of the detection stages in the cascade. The output of each calibration stage is used to adjust the detection window position for input to the subsequent stage. The proposed method runs at 14 *FPS on a single CPU core for VGA-resolution images and* 100 *FPS using a GPU, and achieves state-of-the-art detection performance on two public face detection benchmarks.*

1. Introduction

Face detection is a well studied problem in computer vision. Modern face detectors can easily detect near frontal faces. Recent research in this area focuses more on the uncontrolled face detection problem, where a number of factors such as pose changes, exaggerated expressions and extreme illuminations can lead to large visual variations in face appearance, and can severely degrade the robustness of the face detector.

The difficulties in face detection mainly come from two aspects: 1) the large visual variations of human faces in the cluttered backgrounds; 2) the large search space of possible face positions and face sizes. The former one requires the face detector to accurately address a binary classification problem while the latter one further imposes a time efficiency requirement.

Ever since the seminal work of Viola *et al.* [27], the boosted cascade with simple features becomes the most popular and effective design for practical face detection. The simple nature of the features enable fast evaluation and quick early rejection of false positive detections. Meanwhile, the boosted cascade constructs an ensemble of the simple features to achieve accurate face vs. non-face classification. The original Viola-Jones face detector uses the Haar feature which is fast to evaluate yet discriminative enough for frontal faces. However, due to the simple nature of the Haar feature, it is relatively weak in the uncontrolled environment where faces are in varied poses, expressions under unexpected lighting.

A number of improvements to the Viola-Jones face detector have been proposed in the past decade [30]. Most of them follow the boosted cascade framework with more advanced features. The advanced feature helps construct a more accurate binary classifier at the expense of extra computation. However, the number of cascade stages required to achieve the similar detection accuracy can be reduced. Hence the overall computation may remain the same or even reduced because of fewer cascade stages.

This observation suggests that it is possible to apply more advanced features in a practical face detection solution as long as the false positive detections can be rejected quickly in the early stages. In this work, we propose to apply the Convolutional Neural Network (CNN) [13] to face detection. Compared with the previous hand-crafted features, CNN can automatically learn features to capture complex visual variations by leveraging a large amount of training data and its testing phase can be easily parallelized on GPU cores for acceleration.

Considering the relatively high computational expense of the CNNs, exhaustively scanning the full image in multiple scales with a deep CNN is not a practical solution. To achieve fast face detection, we present a CNN cascade,

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

which rejects false detections quickly in the early, low-resolution stages and carefully verify the detections in the later, high-resolution stages. We show that this intuitive solution can outperform the state-of-the-art methods in face detection. For typical VGA size images, our detector runs in 14 FPS on single CPU core and 100 FPS on a GPU card [1].

In this work, our contributions are four-fold:

- we propose a CNN cascade for fast face detection;
- we introduce a CNN-based face bounding box calibration step in the cascade to help accelerate the CNN cascade and obtain high quality localization;
- we present a multi-resolution CNN architecture that can be more discriminative than the single resolution CNN with only a fractional overhead;
- we further improve the state-of-the-art performance on the Face Detection Data Set and Benchmark (FDDB) [7].

2. Related Work

2.1. Neural network based face detection

Early in 1994 Vaillant *et al.* [26] applied neural networks for face detection. In their work, they proposed to train a convolutional neural network to detect the presence or absence of a face in an image window and scan the whole image with the network at all possible locations. In 1996, Rowley *et al.* [22] presented a retinally connected neural network for upright frontal face detection. The method was extended for rotation invariant face detection later in 1998 [23] with a "router" network to estimate the orientation and apply the proper detector network.

In 2002 Garcia *et al.* [5] developed a neural network to detect semi-frontal human faces in complex images; in 2005 Osadchy *et al.* [20] trained a convolutional network for simultaneous face detection and pose estimation.

It is unknown how these detectors perform in today's benchmarks with faces in uncontrolled environments. Nevertheless, given recent break-through results of CNNs [13] for image classification [24] and object detection [3], it is worth to revisit the neural network based face detection.

One of the recent CNN based detection method is the R-CNN by Girshick *et al.* [6] which has achieved the state-of-the-art result on VOC 2012. R-CNN follows the "recognition using regions" paradigm. It generates category-independent region proposals and extracts CNN features from the regions. Then it applies class-specific classifiers to recognize the object category of the proposals.

Compared with the general object detection task, uncontrolled face detection presents different challenges that make it impractical to directly apply the R-CNN method to face detection. For example, the general object proposal methods may not be effective for faces due to small-sized faces and complex appearance variations.

[1] Intel Xeon E5-2620 2.00GHz CPU and GeForce GTX TITAN BLACK GPU with Caffe [9].

2.2. Face detection in uncontrolled environments

Previous uncontrolled face detection systems are mostly based on hand-crafted features. Since the seminal Viola-Jones face detector [27], a number of variants are proposed for real-time face detection [10, 17, 29, 30].

Recently within the boosted cascade with simple features framework, Chen *et al.* [2] propose to use the shape indexed features to jointly conduct face detection and face alignment. Similar to this idea, we have alternative stages of calibration and detection in our framework. Considering the success of CNNs in a number of visual tasks including the face alignment [31], our framework is more general in that we can adopt a CNN-based face alignment method to achieve joint face alignment and detection, and we use CNN to learn more robust features for faces.

Zhang *et al.* [32] and Park *et al.* [21] adopt the multi-resolution idea in general object detection. While sharing the similar technique, our method utilizes CNNs as the classifiers and combines the multi-resolution and calibration ideas for face detection.

Additionally, the part-based model has motivated a number of face detection methods. Zhu *et al.* [33] propose the tree structured model for face detection which can simultaneously achieve the pose estimation and facial landmarks localization. Yan *et al.* [28] present a structural model for face detection. Mathias *et al.* [19] show that a carefully trained deformable part-based model [4] achieves state-of-the-art detection accuracy.

Different from these model-based methods, Shen *et al.* [25] propose to detect faces by image retrieval. Li *et al.* [15] further improve it to a boosted exemplar-based face detector with state-of-the-art performance.

Compared with these face detection systems, our work learns the classifier directly from the image instead of relying on hand-crafted features. Hence we benefit from the powerful features learned by the CNN to better differentiate faces from highly cluttered backgrounds. Meanwhile, our detector is many times faster than the model-based and exemplar-based detection systems and has a frame rate comparable to the classical boosted cascade with simple features. Sharing the advantages of the CNN, our detector is easy to be parallelized on GPU for much faster detection.

3. Convolutional Neural Network Cascade

We present a specific design of our detector here for a clear explanation of the proposed method. In practice, the CNN cascade can have varied settings for accuracy-computation trade off.

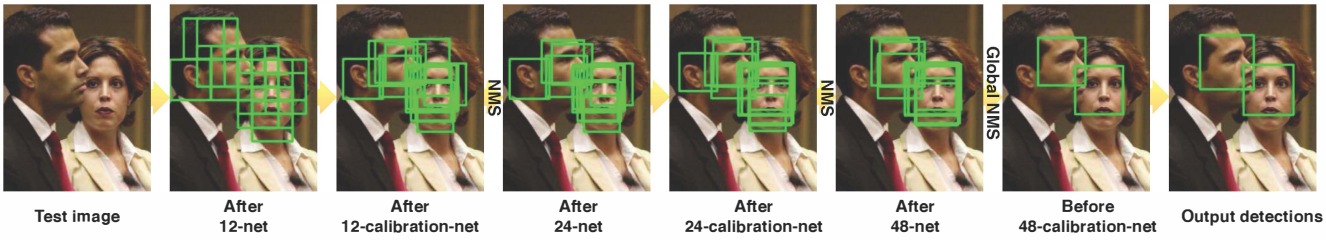


Figure 1: Test pipeline of our detector: from left to right, we show how the detection windows (green squares) are reduced and calibrated from stage to stage in our detector. The detector runs on a single scale for better viewing.

3.1. Overall framework

The overall test pipeline of our face detector is shown in Figure 1. We briefly explain the work-flow and will introduce all the CNNs in detail later.

Given a test image, the 12-*net* scans the whole image densely across different scales to quickly reject more than 90% of the detection windows. The remaining detection windows are processed by the 12-*calibration-net* one by one as 12×12 images to adjust its size and location to approach a potential face nearby.

Non-maximum suppression (NMS) is applied to eliminate highly overlapped detection windows. The remaining detection windows are cropped out and resized into 24×24 as input images for the 24-*net* to further reject nearly 90% of the remaining detection windows. Similar to the previous process, the remaining detection windows are adjusted by the 24-*calibration-net* and we apply NMS to further reduce the number of detection windows.

The last 48-*net* accepts the passed detection windows as 48×48 images to evaluate the detection windows. NMS eliminates overlapped detection windows with an Intersection-Over-Union (IoU) ratio exceeding a pre-set threshold. The 48-*calibration-net* is then applied to calibrate the residual detection bounding boxes as the outputs.

3.2. CNN structure

There are 6 CNNs in the cascade including 3 CNNs for face vs. non-face binary classification and 3 CNNs for bounding box calibration, which is formulated as multi-class classification of discretized displacement pattern. In these CNNs, without specific explanation we follow AlexNet [12] to apply ReLU nonlinearity function after the pooling layer and fully-connected layer.

3.2.1 12-*net*

12-*net* refers to the first CNN in the test pipeline. The structure of this CNN is shown in Figure 2. 12-*net* is a very shallow binary classification CNN to quickly scan the testing image. Densely scanning an image of size $W \times H$ with 4-pixel spacing for 12×12 detection windows is equivalent to apply the 12-*net* to the whole image to obtain a $(\lfloor (W-12)/4 \rfloor + 1) \times (\lfloor (H-12)/4 \rfloor + 1)$ map of confidence scores. Each point on the confidence map refers to a 12×12 detection window on the testing image.

In practice, if the acceptable minimum face size is F, the test image is first built into image pyramid to cover faces at different scales and each level in the image pyramid is resized by $\frac{12}{F}$ as the input image for the 12-*net*. On a single CPU core, it takes 12-*net* less than 36 ms to densely scan an image of size 800×600 for 40×40 faces with 4-pixel spacing, which generates $2,494$ detection windows. The time reduces to 10 ms on a GPU card, most of which is overhead in data preparation.

3.2.2 12-*calibration-net*

12-*calibration-net* refers to the CNN after 12-*net* for bounding box calibration. The structure is shown in Figure 4. 12-*calibration-net* is a shallow CNN. N calibration patterns are pre-defined as a set of 3-dimensional scale changes and offset vectors $\{[s_n, x_n, y_n]\}_{n=1}^{N}$. Given a detection window (x, y, w, h) with top-left corner at (x, y) of size (w, h), the calibration pattern adjusts the window to be

$$(x - \frac{x_n w}{s_n}, y - \frac{y_n h}{s_n}, \frac{w}{s_n}, \frac{h}{s_n}). \quad (1)$$

In this work, we have $N = 45$ patterns, formed by all combinations of

$$\begin{aligned} s_n &\in \{0.83, 0.91, 1.0, 1.10, 1.21\} \\ x_n &\in \{-0.17, 0, 0.17\} \\ y_n &\in \{-0.17, 0, 0.17\}. \end{aligned}$$

Given a detection window, the region is cropped out and resized to 12×12 as the input image for the 12-*calibration-net*. The calibration net outputs a vector of confidence scores $[c_1, c_2, \cdots, c_N]$. Since the calibration patterns are not orthogonal to each other, we take the average results of the patterns of high confidence score as the adjustment

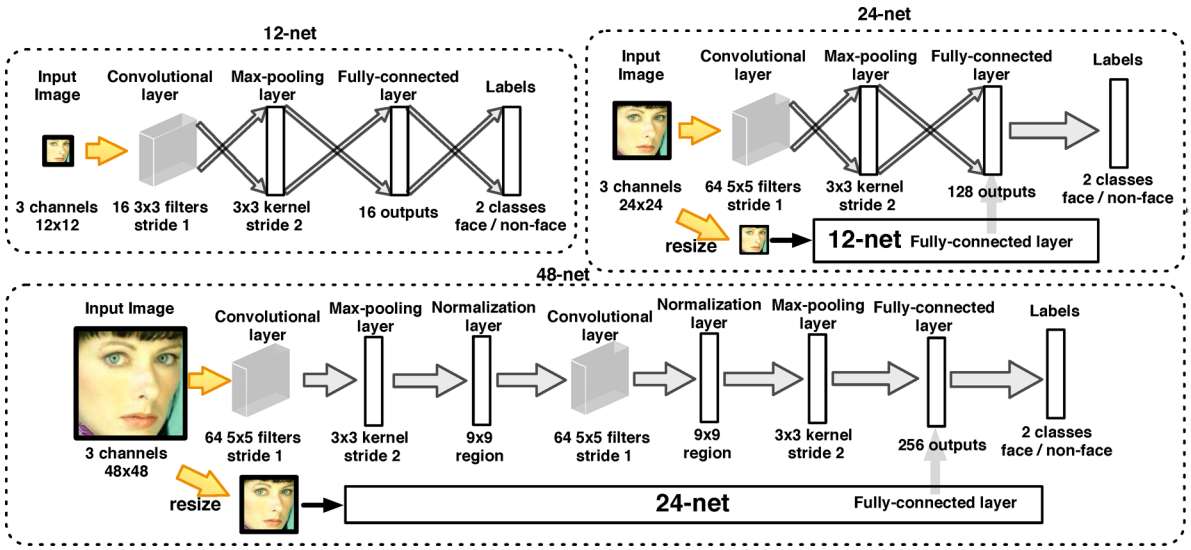


Figure 2: CNN structures of the 12-*net*, 24-*net* and 48-*net*.

$[s, x, y]$, *i.e.*,

$$[s, x, y] = \frac{1}{Z}\sum_{n=1}^{N}[s_n, x_n, y_n]I(c_n > t), \quad (2)$$

$$Z = \sum_{n=1}^{N} I(c_n > t), \quad (3)$$

$$I(c_n > t) = \begin{cases} 1, & \text{if } c_n > t \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Here t is a threshold to filter out low confident patterns.

In our experiment, we observe that the 12-*net* and 12-*calibration-net* reject 92.7% detection windows while keeping 94.8% recall on FDDB (see Table 1).

3.2.3 24-*net*

24-*net* is an intermediate binary classification CNN to further reduce the number of detection windows. Remaining detection windows from the 12-*calibration-net* are cropped out and resized into 24×24 images and evaluated by the 24-*net*. The CNN structure is shown in Figure 2.

A similar shallow structure is chosen for time efficiency. Besides, we adopt a multi-resolution structure in the 24-*net*. In additional to the 24×24 input, we also feed the input in 12×12 resolution to a sub-structure same as the 12-*net* in 24-*net*. The fully-connected layer from the 12-*net* sub-structure is concatenated to the 128-output fully-connected layer for classification as shown in Figure 2. With this multi-resolution structure, the 24-*net* is supplemented by the information at 12×12 resolution which helps detect the small faces. The overall CNN becomes more discriminative and the overhead from the 12-*net* sub-structure is only a fraction of the overall computation.

In Figure 3, we compare the detection performance with and without the multi-resolution design in the 24-*net*. We observe that at the same recall rate, the one with the multi-resolution structure can achieve the same recall level with less false detection windows. The gap is more obvious at the high recall level.

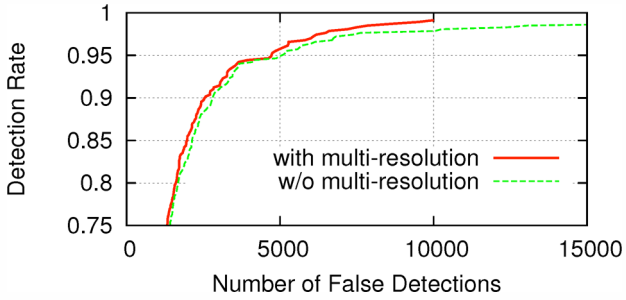


Figure 3: On the Annotated Faces in the Wild dataset, the detection performance of 24-*net* with and without the multi-resolution structure.

3.2.4 24-*calibration-net*

Similar to the 12-*calibration-net*, 24-*calibration-net* is another calibration net with N calibration patterns. The structure is shown in Figure 4. Except for the input size to the 24-*calibration-net* is 24×24, the pre-defined patterns and the calibration process is same as in the 12-*calibration-net*.

In our experiment, we observe that the 24-*net* and 24-*calibration-net* can further reject 86.2% detection windows retained after 24-*calibration-net* while keeping 89.0% recall on FDDB (see Table 1).

3.2.5 48-*net*

48-*net* is the last binary classification CNN. At this stage of the cascade, it is feasible to apply a more powerful but

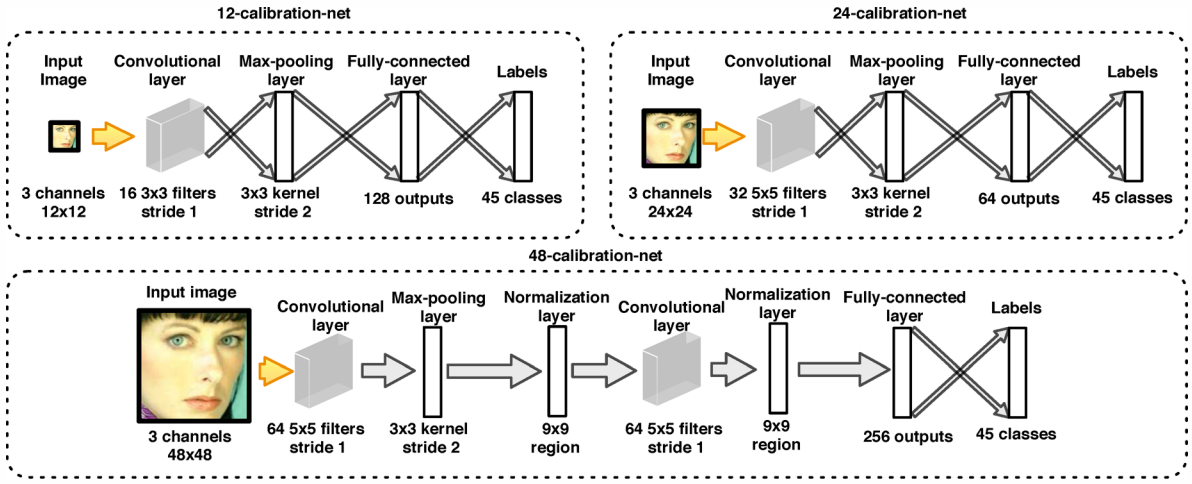


Figure 4: CNN structures of the 12-*calibration-net*, 24-*calibration-net* and 48-*calibration-net*.

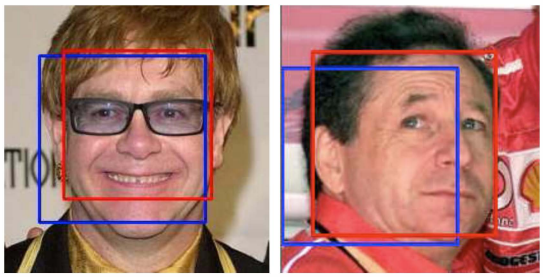


Figure 5: Calibrated bounding boxes are better aligned to the faces: the blue rectangles are the most confident detection bounding boxes of 12-*net*; the red rectangles are the adjusted bounding boxes with the 12-*calibration-net*.

slower CNN. As shown in Figure 2, the 48-*net* is relatively more complicated. Similar to the 24-*net*, we adopt the multi-resolution design in 48-*net* with additional input copy in 24×24 and a sub-structure the same as the 24-*net*.

3.2.6 48-*calibration-net*

48-*calibration-net* is the last stage in the cascade. The CNN structure is shown in Figure 4. The same $N = 45$ calibration patterns are pre-defined for the 48-*calibration-net* as in Section 3.2.2. We use only one pooling layer in this CNN to have more accurate calibration.

3.2.7 Non-maximum suppression (NMS)

We adopt an efficient implementation of the NMS in this work. We iteratively select the detection window with the highest confidence score and eliminate the detection windows with an IoU ratio higher than a pre-set threshold to the selected detection window.

In the 12-*net* and 24-*net*, the shallow CNNs may not be discriminative enough to address challenging false positives. After the 12-*calibration-net* and 24-*calibration-net*, challenging false positives may have a higher confidence score compared with the true positives. Hence after 12-*calibration-net* and 24-*calibration-net*, we conservatively apply NMS separately for the detection windows at the same scale (of the same size) to avoid degrading the recall rate. NMS After 48-*net* is applied globally to all detection windows at different scales to make most accurate detection window at the correct scale stand out and avoid redundant evaluation in the 48-*calibration-net*.

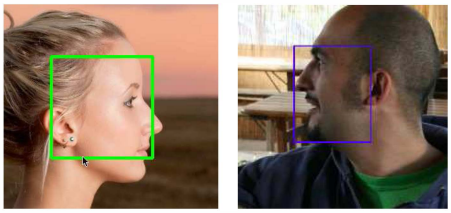
3.3. CNN for calibration

We explain how the calibration nets help in the cascade for face detection. The motivation of applying the calibration is shown in Figure 5. The most confident detection window may not be well aligned to the face. As a result, without the calibration step, the next CNN in the cascade will have to evaluate more regions to maintain a good recall. The overall detection runtime increases significantly.

This problem generally exists in object detection. We explicitly address this problem with CNNs in this work. Instead of training a CNN for bounding boxes regression as in R-CNN, we train a multi-class classification CNN for calibration. We observe that a multi-class calibration CNN can be easily trained from limited amount of training data while a regression CNN for calibration requires more training data. We believe that the discretization decreases the difficulty of the calibration problem so that we can achieve good calibration accuracy with simpler CNN structures. As shown in Figure 5, after calibration the detection bounding box is better aligned to the real face center. As a result, the calibration nets enable more accurate face localization using coarser scanning windows across less scales.

3.4. Training process

In training the CNNs in the cascade, we collect $5,800$ background images to obtain negative training samples and



(a) AFLW (b) AFW

Figure 6: Mismatched face annotations in AFLW and AFW.

use the faces in the Annotated Facial Landmarks in the Wild (AFLW) [11] dataset as positive training samples.

For both the binary and multi-class classification CNNs in the cascade, we use the multinomial logistic regression objective function for optimization in training [2].

3.4.1 Calibration nets

In collecting training data for the calibration nets, we perturb the face annotations with the $N = 45$ calibration patterns in Section 3.2.2. Specifically, for the n-th pattern $[s_n, x_n, y_n]$, we apply $[1/s_n, -x_n, -y_n]$ (following Equation 1) to adjust the face annotation bounding box, crop and resize into proper input sizes (12×12, 24×24 and 48×48).

3.4.2 Detection nets

The detection nets 12-*net*, 24-*net* and 48-*net* are trained following the cascade structure. We resize all training faces into 12×12 and randomly sample $200,000$ non-face patches from the background images to train the 12-*net*. We then apply a 2-stage cascade consists of the 12-*net* and 12-*calibration-net* on a subset of the AFLW images to choose a threshold T_1 at 99% recall rate.

Then we densely scan all background images with the 2-stage cascade. All detection windows with confidence score larger than T_1 become the negative training samples for the 24-*net*. 24-*net* is trained with the mined negative training samples and all training faces in 24×24. After that, we follow the same process for the 4-stage cascade consists of the 12-*net*, 12-*calibration-net*, 24-*net* and 24-*calibration-net*. We set the threshold T_2 to keep 97% recall rate.

Following the same procedure, we mine negative training samples for the 48-*net* with the 4-stage cascade on all the background images. The 48-*net* is trained with positive and negative training samples in 48×48.

At each stage in the cascade, the CNN is trained to address a sub-problem which is easier than addressing the face vs. non-face classification globally. Compared with the design to have one single CNN to scan the full image for faces, the cascade makes it possible to have simpler CNNs achieve the same or even better accuracy.

[2] We use the cuda-convnet2 [1] CNN implementation.

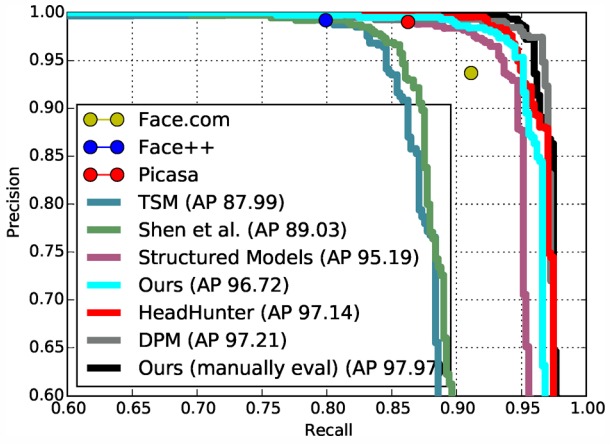


Figure 7: On the AFW dataset we compare our performance with the state-of-the-art methods including TSM [33], Shen *et al.* [25], Structured Models [28], HeadHunter [19], DPM [4, 19], Face.com, Face++ and Picasa.

4. Experiments

We verify the proposed detector on two public face detection benchmarks. On the Annotated Faces in the Wild (AFW) [33] test set, our detector is comparable to the state-of-the-art. This small scale test set is almost saturated and we observe that the evaluation is biased due to the mismatched face annotations. On the challenging Face Detection Data Set and Benchmark (FDDB) dataset [7], our detector outperforms the state-of-the-art methods in the discontinuous score evaluation. Meanwhile, we show that our detector can be easily tuned to be a faster version with minor performance decrease.

4.1. Annotated Faces in the Wild

Annotated Faces in the Wild (AFW) is a 205 images dataset created by Zhu *et al.* [33]. We evaluate our detector on AFW and the precision-recall curves are shown in Figure 7. Our performance is comparable to the state-the-arts on this dataset.

As pointed out by Mathias *et al.* [19], one important problem in the evaluation of face detection methods is the mismatch of the face annotations in the training and testing stages. In our training stage, we generate square annotations to approach the ellipse face annotations on AFLW in preparing the positive training samples. As shown in Figure 6, the annotations in AFLW and AFW are mismatched.

In the evaluation on AFW, we follow the remedial method proposed by Mathias *et al.* to search for a global rigid transformation of the detection outputs to maximize the overlapping ratio with the ground-truth annotations with the shared evaluation tool [19]. However our annotations cannot be simply linearly mapped to AFW annotations, after the global transformation step the mismatches still exist. Therefore we manually evaluate the output detections and get better results. All the curated detections are shown

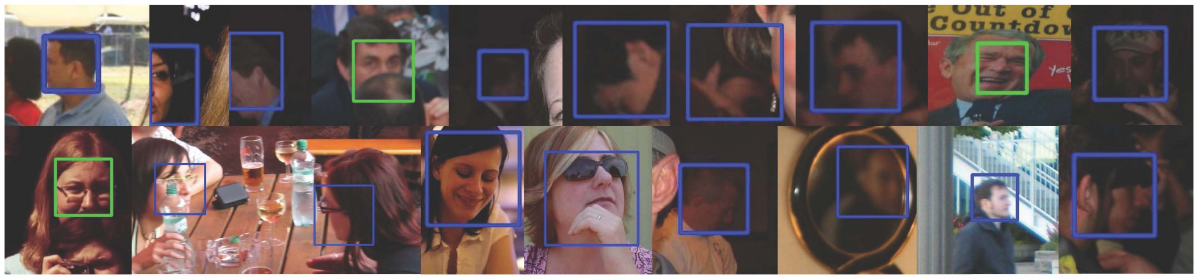


Figure 8: Manually curated detection bounding boxes on AFW: blue boxes are faces mis-evaluated to be false alarms; green boxes are unannotated faces. These are all detections our approach generated but miss-classified by the evaluation. However, with our annotation standards, these detections are examined to be true detections.

Table 1: Performance statistics of the cascade on FDDB: we show the average number of detection windows per image after each stage and the overall recall rate.

Stage	# windows	Recall
sliding window	5341.8	95.9%
12-*net*	426.9	93.9%
12-*calibration-net*	388.7	94.8%
24-*net*	60.5	88.8%
24-*calibration-net*	53.6	89.0%
48-*net*	33.3	85.8%
global NMS	3.6	82.1%
48-*calibration-net*	3.6	85.1%

in Figure 8 including unannotated faces and mis-evaluated faces.

4.2. Face Detection Data Set and Benchmark

The Face Detection Data Set and Benchmark (FDDB) dataset [7] contains $5,171$ annotated faces in $2,845$ images. This is a large-scale face detection benchmark with standardized evaluation process. We follow the required evaluation procedure to report our detection performance with the toolbox provided by the authors.

FDDB uses ellipse face annotations and defines two types of evaluations: the discontinuous score and continuous score. In the discontinuous score evaluation, it counts the number of detected faces versus the number of false alarms. The detection bounding boxes (or ellipses) are regarded as true positive only if it has an Intersection-over-Union (IoU) ratio above 0.5 to a ground-truth face. In the continuous score evaluation, it evaluates how well the faces are located by considering the IoU ratio as the matching metric of the detection bounding box.

We uniformly extend our square detection bounding boxes vertically by 20% to be upright rectangles on FDDB to better approach their ellipse annotation. As show in Figure 9, our detector outperforms the best performance in the discontinuous score evaluation.

Our detector outputs rectangle outputs while the HeadHunter [19] and JointCascade [2] generate ellipse outputs. For a more fair comparison under the continuous score evaluation, we uniformly fit upright ellipses for our rectangle bounding boxes. For a rectangle in size (w, h), we fit an upright ellipse at the same center with axis sizes $1.18h$ and $1.13w$, which most overlapped with the rectangle. As shown in Figure 9, with the naively fitted ellipses our detector outperforms the HeadHunter and approaches the JointCascade under the continuous score evaluation. The performance drops a little in the discontinuous score evaluation due to the inaccurate simple fitting strategy.

The performance of the cascade from stage to stage is shown Table 1. We observe the number of detection windows decreases quickly and the calibration nets help further reduce the detection windows and improve the recall.

4.3. Runtime efficiency

One of the important advantages of this work is its runtime efficiency. In this work, the CNN cascade can achieve very fast face detection. Furthermore, by simply varying the thresholds $T_{1,2}$, one can find a task specific accuracy-computation trade off. In Figure 11, we show that the performance of a faster version of our detector is comparable to Picasa on AFW.

In the faster version, we set the thresholds to be aggressively high to reject a large portion of the detection windows in the early stages. The calibration nets help more in the faster version by adjusting the bounding boxes back to the face center to improve recall in the later stage. On average, only 2.36% detection windows passed the 12-*net* and 12-*calibration-net*; 14.3% of the retained detection windows passed 24-*net* and 24-*calibration-net* to feed into the most computationally expensive 48-*net*.

With the same thresholds, we evaluate our detector in a typical surveillance scenario to detect faces from 640×480 VGA images. We only scan for 80×80 faces in the 12-*net* stage but with the calibration nets we can detect faces smaller or larger than 80×80 within the calibration range. In this scenario, our detector processes one image in 71 ms on average on a 2.0 GHz CPU core [3]. Under the same setting, it takes 770 ms to scan the whole image with the 48-

[3] Over an image pyramid with scaling factor 1.414, the average detection time is 110 ms.

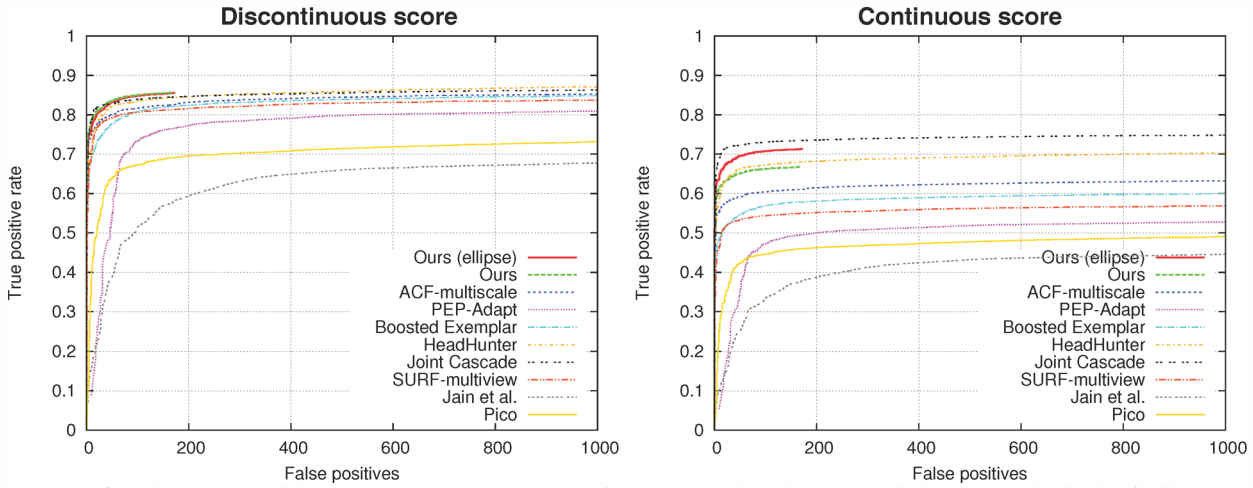


Figure 9: On the FDDB dataset we compare our performance with the state-of-the-art methods including: ACF-multiscale [29], PEP-Adapt [14], Boosted Exemplar [15], HeadHunter [19], Jain *et al.* [8], SURF-multiview [16], Pico [18], and Joint Cascade [2].

Figure 10: Qualitative results of our detector on FDDB.

net for detection, which demonstrates the time efficiency of our cascade design. On the GPU card the detection time of cascade is further reduced to 10 ms per image without code optimization. This detection speed is very competitive compared with other state-of-the-art methods.

Among the top performers on the FDDB and AFW that reported their detection speed, the runtime for the ACF-multiscale [29] is 20 FPS for full yaw pose face detection and 34 FPS for frontal faces on a single thread of Intel Core i7-3770 CPU in VGA image; for the Boosted Exemplar [15] it is 900 ms for a 1480×986 pixels image; for the Joint Cascade [2] it is 28.6 ms for VGA images on a 2.93 GHz CPU; SURF-multiview [16] runs in real-time for VGA video on a personal workstation with 3.2 GHz Core-i7 CPU (4 cores 8 threads); TSM *et al.* [33] processes a VGA image in 33.8 seconds [2]; Shen *et al.* [25] processes a 1280-pixel dimension image in less than 10 seconds.

5. Conclusion

In this work, we present a CNN cascade for fast face detection. Our detector evaluates the input image at low resolution to quickly reject non-face regions and carefully process the challenging regions at higher resolution for accurate detection. Calibration nets are introduced in the cascade to accelerate detection and improve bounding box quality. Sharing the advantages of CNN, the proposed face detector is robust to large visual variations. On the public face detection benchmark FDDB, the proposed detector outperforms the state-of-the-art methods. The proposed detector is very fast, achieving 14 FPS for typical VGA images on CPU and can be accelerated to 100 FPS on GPU.

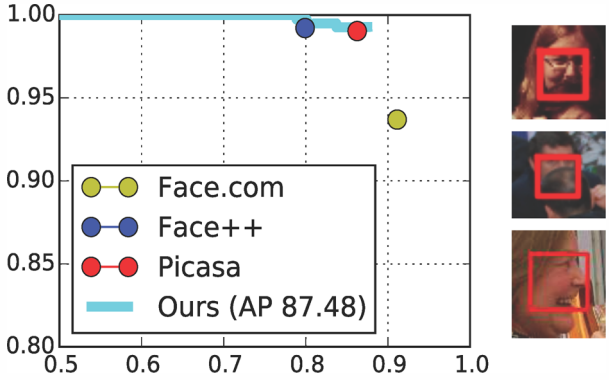


Figure 11: Precision-recall curve of the faster version of our detector the AFW dataset. All three false alarms are mis-evaluated or unannotated faces as shown on the right.

Acknowledgment

This work is partially done when the first author was an intern at Adobe Research. Research reported in this publication was partly supported by the National Institute Of Nursing Research of the National Institutes of Health under Award Number R01NR015371. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This work is also partly supported by US National Science Foundation Grant IIS 1350763 and GH's start-up funds from Stevens Institute of Technology.

References

[1] code.google.com/p/cuda-convnet2. 6

[2] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *Computer Vision–ECCV 2014*. 2014. 2, 7, 8

[3] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge, 2009. 2

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010. 2, 6

[5] C. Garcia and M. Delakis. A neural architecture for fast and robust face detection. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2002. 2

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 2

[7] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010. 2, 6, 7

[8] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011. 8

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 2

[10] M. Jones and P. Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 2003. 2

[11] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011. 6

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 3

[13] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 1995. 1, 2

[14] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic part model for unsupervised face detector adaptation. In *Proc. IEEE International Conference on Computer Vision*, 2013. 8

[15] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014. 2, 8

[16] J. Li, T. Wang, and Y. Zhang. Face detection using surf cascade. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011. 8

[17] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002. 2

[18] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer. A method for object detection based on pixel intensity comparisons organized in decision trees. *arXiv preprint arXiv:1305.4537*, 2013. 8

[19] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *Computer Vision–ECCV 2014*. 2014. 2, 6, 7, 8

[20] M. Osadchy, Y. L. Cun, M. L. Miller, and P. Perona. Synergistic face detection and pose estimation with energy-based model. In *In Advances in Neural Information Processing Systems (NIPS)*, 2005. 2

[21] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *Computer Vision ECCV 2010*. 2010. 2

[22] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Computer Vision and Pattern Recognition*, 1996. 2

[23] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Computer Vision and Pattern Recognition*, 1998. 2

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 2

[25] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2, 6, 8

[26] R. Vaillant, C. Monrocq, and Y. Le Cun. Original approach for the localisation of objects in images. *IEE Proceedings-Vision, Image and Signal Processing*, 1994. 2

[27] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 1, 2

[28] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *Image and Vision Computing*, 2013. 2, 6

[29] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. *arXiv preprint arXiv:1407.4023*, 2014. 2, 8

[30] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, 2010. 1, 2

[31] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Computer Vision–ECCV 2014*. 2014. 2

[32] W. Zhang, G. Zelinsky, and D. Samaras. Real-time accurate object detection using multiple resolutions. In *Proc. IEEE International Conference on Computer Vision*, 2007. 2

[33] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2, 6, 8