IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 3, 2018, accepted September 6, 2018, date of publication September 10, 2018,
date of current version September 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869465

Simultaneous Face Detection and Pose Estimation Using Convolutional Neural Network Cascade

HAO WU, KE ZHANG, AND GUOHUI TIAN
School of Control Science and Engineering, Shandong University, Jinan 250061, China

Corresponding author: Ke Zhang (sdu_kezhang@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61773239, in part by the Natural Science Foundation of Shandong Province under Grant ZR201702180022 and Grant ZR2015FM007, in part by the Shandong Major Research Plan Project under Grant 2015GGX103034 and Grant 2015ZDXX0101F03, in part by the Taishan Scholars Program of Shandong Province, and in part by the Spring City Industry Leading Talent Support Program of Jinan.

ABSTRACT Recent studies show that convolutional neural networks (CNNs) has made a series of breakthroughs in the two tasks of face detection and pose estimation, respectively. There are two CNN frameworks for solving these two integrated tasks simultaneously. One is to use face detection network to detect faces firstly, and then use pose estimation network to estimate each face's pose; the other is to use region proposal algorithm to generate many candidate regions that may contain faces, and then use a single deep multi-task CNN to process these regions for simultaneous face detection and pose estimation. The former's problem is pose estimation's performance is affected by face detection network because two networks are separate. The latter generates lots of candidate regions, which will bring huge computation cost to CNN and can't achieve real-time. To solve the above existing problems, we propose a multi-task CNN cascade framework that integrates these two tasks. We show that multi-task learning of face detection and head pose estimation helps to extract more representative features. We exploit CNN feature fusion strategy to further improve head pose estimation's performance. We evaluate face detection on FDDB benchmark, and evaluate pose estimation on AFW benchmark. Our method achieves comparative result compared with state-of-the-art in these two tasks and can achieve real-time performance.

INDEX TERMS Face detection, pose estimation, CNN cascade, multi-task learning, feature fusion.

I. INTRODUCTION

Face detection and pose estimation [1] in the field of computer vision refer to detecting all faces in the picture and estimating the orientation of each face which can be expressed by three orientation angles: yaw, pitch and roll. Because head pose provides clues for judging people's motivation, intention and gaze, these two tasks have a wide range of applications such as human behavior analysis and gaze estimation. Although tremendous progress have been made in face detection and pose estimation respectively, it is still a daunting task to achieve a multi-task framework that has good real-time performance and robustness to complex environments.

In the past two decades, there have been many excellent achievements in solving face detection and pose estimation. Convolutional neural networks (CNNs) has achieved remarkable success in a series of complicated computer vision tasks such as image classification [2]–[4], face recognition [5]–[8] and object detection [9]–[11]. Inspired by these facts, methods of face detection and pose estimation based on CNN begin to appear and occupy the dominant position.

To solve face detection and pose estimation simultaneously, there are two kinds of CNN-based framework. One is to use face detection network to detect all faces in the picture, and then use pose estimation network to estimate each face's pose such as [12]. The other is to use region proposal algorithm to generate many candidate regions that maybe contain exactly a face, and then use a deep multi-task CNN to process these regions for simultaneous face detection and pose estimation such as [13] and [14]. The former's problem is because the two networks are separate, the inaccuracy of face detection network will affect the result of pose estimation. In addition, such a framework does not take advantage of the inherent correlation of two tasks to promote each other's performance. The latter generates lots of candidate regions in order to achieve a sufficiently high detection rate, which will bring huge computation cost to CNN and can't achieve real-time.

Face detection and pose estimation are face-related tasks. They all rely on potential facial features. So combining two tasks in a network is feasible. Combining these two tasks

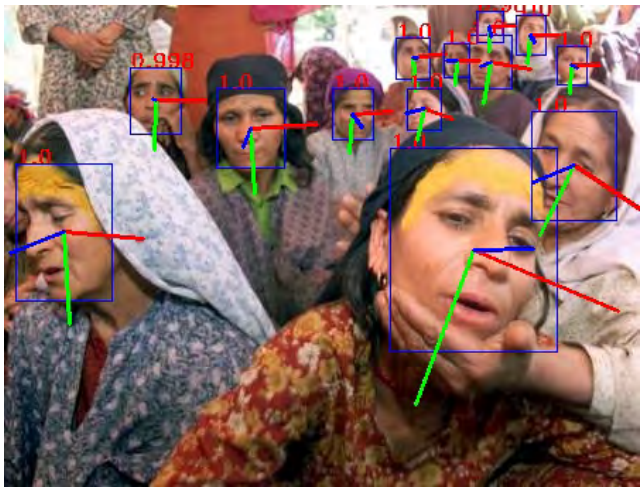


FIGURE 1. Our framework can simultaneously detect the face and estimate the pose, while maintaining real-time. The blue boxes denote detected faces. The three axes are drawn from the three estimated angles(yaw, pitch, roll), with the blue axis pointing to the front of the face, green pointing downward and red pointing to side.

enables the network to take into account the accuracy of the two tasks in the optimization process. The network causes the two tasks to receive the same input, and finally outputs the results of both tasks. Therefore, multi-task network makes use of the correlation between the two tasks and overcomes the influence of using face detection results as input of pose estimation network.

Viola-Jones framework [15] makes face detection truly achieve real-time with very low computational complexity. Cascade structure in this framework connects many weak classifiers with strong classifiers to form a sequence of simple-to-complex classifiers. Weak classifiers quickly reject a large number of false positive detections in the early stage. And strong classifiers carefully evaluate a small number of remaining detections in the later stage. Viola-Jones framework popularized the cascade structure. The cascade structure based on convolutional neural networks emerged later and achieved great success. Reference [16] proposed a CNN-based cascade architecture for face detection, and introduced a calibration stage after each detection stage for improving localization effectiveness. Reference [17] simplified the above architecture and proposed a multi-task cascaded CNN framework for face alignment and detection.

In this paper, we propose a multi-task convolutional neural network cascade framework for real-time face detection and pose estimation simultaneously (see Fig. 1). Our framework consists of three stages. The first two stages filter the candidate regions, and the final stage further rejects some false positives and perform pose estimation on each face. As mentioned earlier, the framework consisting of two single-task networks has the problem that face detection network affects pose estimation network's performance and the multi-task framework has the problem of being unable to achieve real-time. Because multi-task learning can take advantage of the inherent correlation between two tasks and cascade structure helps to reach real-time, our proposed framework solves these two problems at the same time.

On the other hand, [3] proposed Deconvnet algorithm to visualize the feature map of each layer and showed the hierarchical nature of the features in the CNN. Exploiting this fact, [13] fused intermediate layer features of a deep CNN and improved the performance of their face-related tasks. Inspired by this, we fuse low-level, middle-level and high-level features from the CNN of last stage in the cascade-based framework to provide richer information for pose estimation. Compared with the CNN architecture without feature fusion, pose estimation accuracy is improved while maintaining real-time. This paper makes the following contributions.

- We propose a CNN cascade framework for simultaneous face detection and pose estimation, while maintaining real-time.
- We show that multi-task learning of face detection and pose estimation helps to extract more representative features.
- We study the performance of multi-task learning with and without feature fusion.
- We achieve comparative performance with state-of-art methods on challenging unconstrained datasets for these two tasks.

II. RELATED WORK

In this section, we present an overview of related work on face detection, pose estimation, multi-task learning and CNN feature fusion. Among them, there is a rich history in face detection and pose estimation. Space does not allow us for a very detailed review. Refer to these surveys [18]–[21] for more details. We focus on most relevant methods to ours.

A. FACE DETECTION

In the past few decades, face detection has attracted a large amount of research. Early methods are only suitable for up-right faces without occlusion. With the in-depth research, a large number of methods for faces in extreme environments emerge and achieve remarkable results. Most recent work can be divided into three categories [22].

1) EARLY CNN-BASED METHODS

Convolutional neural networks(CNNs) have been used for face detection in 1990s. In 1994, Vaillant *et al.* [23] scanned the whole image using a sliding window and then used a CNN to judge whether each image window contains a face. In 1996, Rowley *et al.* [24] trained a CNN for upright frontal face detection and showed good result. Reference [25] extended the above method and reached certain degree of rotation invariance detection. In 2002, Garcia and Delakis [26] designed a convolutional neural network architecture to detect more complex semi-frontal faces on images. These methods shows the feasibility of CNN in face detection. However, due to hardware limitations, these CNNs are very time consuming and just work well on some simple datasets.

2) HAND-CRAFTED FEATURE BASED METHODS

Previous face detection methods mostly adopt hand-crafted features. To achieve real-time detection, Viola and Jones [15] proposed a cascade face detector which utilizes Haar-Like features and Adaboost-based learning. Since then, a large number of variants based on the Viola-Jones framework have been proposed for real-time face detection. These variants improved the original performance through more complex features [27], [28], new boosting algorithms [29], [30] and new cascade structures [31], [32]. Besides Viola-Jones framework, [33]–[35] introduced deformable part model(DPM) [36] which uses HOG feature. These methods define a face as a collection of its parts and are robust to partial occlusion.

3) MODERN CNN-BASED METHODS

Recently, CNN-based methods have occupied the dominant position in object detection [9]–[11]. Inspired by this, [37] introduced Faster-RCNN [9] for face detection. Some other CNN-based methods [16], [17], [38]–[40] are proposed for face detection. In particular, [39] proposed a Scale-aware Face Detection framework which can estimate which scales have faces to save computation. Reference [40] used face's contextual information and achieves state-of-art in finding tiny face.

B. POSE ESTIMATION

Individual pose estimation task assumes the face detection problem is solved and only concerns about pose estimation's accuracy. References [41], [42] used random forest to solve head pose estimation problem and achieve real-time. Manifold embedding methods [43], [44] modeled the continuous variation in head pose by seeking low-dimensional manifolds. Ahn *et al.* [45] built the mapping function between three dimensional head orientation angles and visual appearance using CNN. Ghiass *et al.* [46] fit a 3D morphable model which contains pose parameters to perform pose estimation.

Head pose can also be estimated by establishing correspondence between the landmarks and a standard 3D head model. Recently, facial landmark detection [47]–[50] has reached a high degree of precision with the rapid development of deep learning. Hence, the progress of facial landmark detection popularizes the landmark-based methods. But when sufficient landmarks can't be detected, pose estimation fails to perform. So directly predicting the head pose from unconstrained images has the potential to be more robust. References [12], [16], and [50] regard pose estimation as a regression problem and train CNN to directly predict head pose.

C. MULTI-TASK LEARNING

Multi-task learning refers to using the inherent correlation of the relevant tasks to promote the result of each individual task. Caruana [51] first analyzed Multi-task learning (MTL) in detail. Inspired by this, there have been some approaches of adopting MTL to solve problems in the field of computer vision. Zhu and Ramanan [33] first proposed a framework that uses global mixtures of every facial landmark to capture viewpoint changes for face detection, pose estimation, and landmark localization in real-world images. Chen *et al.* [52] proposed an approach that jointly learns face detection and alignment in a cascade framework and observed that face classification obtains better features from aligned face shapes.

Recently, MTL framework with deep CNNs yield unusually brilliant results in face-related tasks. Ranjan *et al.* [14] proposed a multi-task learning framework for simultaneous face detection, pose estimation, face alignment, age estimation, smile detection, gender recognition and face recognition using a single convolutional neural network (CNN). Zhang *et al.* [17] designed a deep cascaded multi-task framework that jointly predict face and localize landmark. Reference [13] proposed a model called HyperFace that fuses the features of intermediate layers from a CNN with very deep depth for multi-task learning including face detection, pose estimation, landmarks localization and gender recognition. Zhang *et al.* [53] optimized facial landmark location together with correlated tasks such as facial attribute inference and head pose estimation. Levi and Hassner [54] proposed a convolutional network architecture for simultaneous age and gender estimation.

D. CNN FEATURE FUSION

Zeiler and Fergus [3] proposed features throughout the Convolutional Neural Network is hierarchically distributed. Quite a few works fuse intermediate layers from CNN to combine the features of each layer together and get good results. Hariharan *et al.* [55] proposed Hypercolumn that represents a pixel by fusing the activations of different layers for image segmentation. Farabet *et al.* [56] concatenated CNN feature maps at multiple scales for semantic segmentation. Sermanet *et al.* [57] combined the subsampled 1st stage output of CNN with 2st stage output as the input of classifier stage for Pedestrian Detection. Yang and Ramanan [58] proposed DAG-CNNs which extract multi-scale features from multiple layers for image classification. Reference [13] fused the max1, conv3 and pool5 layers of Alexnet [2] for face detection, pose estimation, landmark localization, gender recognition and boosted up these tasks's performance.

III. GENERAL FRAMEWORK FOR SIMULTANEOUS FACE DETECTION AND POSE ESTIMATION

In this section, we make a simple analysis of the two general frameworks for simultaneous face detection and pose estimation.

A. FACE DETECTION NETWORK PLUS POSE ESTIMATION NETWORK

This framework is to first scale an image to different scales. Face detection network processes these multi-scale images to find faces of different sizes and cut out all face regions. Then, each face region image is scaled to a fixed size and fed

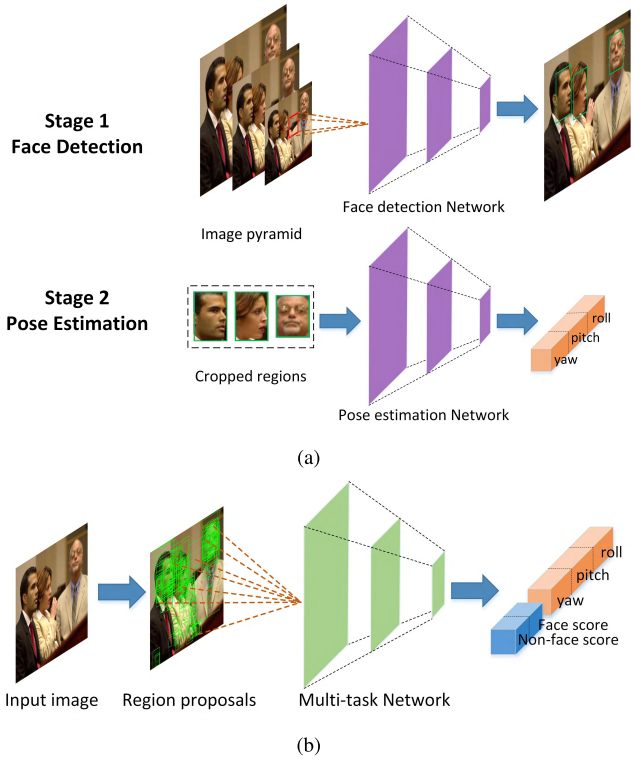


FIGURE 2. **(a) Overview of the framework which first use face detection network to detect faces and then use pose estimation network to estimate the pose of each face, (b) Overview of the framework which first generates a large number of regional proposals and then feeds these proposals through a multi-task network for face detection and pose estimation.**

into the pose estimation network to get the pose of each face (see Fig. 2(a)).

However, face detection network and pose estimation network are completely independent. The training of these two networks is conducted separately. Public datasets used by face detection have a large number of pictures of various sizes containing more or less faces. The annotation contains only the position information of the face on the image but lacks the pose information. Face position annotation is expressed by (x, y, w, h), where (x, y) denotes top-left coordinate of the face region in the picture, w and h denote region width and height respectively. Public datasets used by pose estimation network contain face region images with detailed pose annotation, but the number of images is small. Two tasks use their own datasets to train. The training set adopted by face detection network and pose estimation network has different definition of face region. If face regions obtained by face detection network can't fit well with the face regions needed by the pose estimation network, pose estimation network can't accurately estimate face pose.

B. REGION PROPOSAL ALGORITHM PLUS MULTI-TASK NETWORK

This framework first runs a robust region proposal algorithm on a single image, such as selective search [59] and Edge boxes [60]. The image contains a variety of different levels of semantics and the information is very rich, including texture, shape, color and so on. Based on these information, the region proposal algorithm obtain a large number of candidate regions with different sizes and categories. Multi-task network performs face vs. non-face classification and pose estimation for each candidate region (see Fig. 2(b)).

Firstly, it is very time-consuming to run region proposal algorithm on an image and usually takes 1 to 2 seconds. In addition, in order to achieve a higher detection rate, the number of generated regions is large, which brings huge computational overhead to the CNN. These two factors make this framework can not achieve real-time. Most of the regions don't contain any face, and it's a waste of time to judge these regions.

IV. PROPOSED METHOD

In this section, we present the proposed multi-task framework for real-time face detection and pose estimation. In addition, we improve the CNN architecture with feature fusion. The key ideas are 1) utilize cascaded networks for fast and accurate face detection and in last stage, perform pose estimation on final fine face detections and 2) introduce multi-task learning in each individual CNN architecture to take advantage of the inherent correlation of these tasks and 3) fuse the simple features from the lower layers and the complex features from the deeper layers to fully exploit features for different tasks.

A. OVERALL FRAMEWORK

The overall pipeline of our multi-task framework is shown in Fig. 3. The framework consists of three cascaded CNN components called Coarse-detection Network, Optimized-detection Network and Fine-detection & Pose-estimation Network. We first explain the work-flow and introduce the CNN architectures in detail later.

Given an image, build an image pyramid by scaling image in a fixed proportion in order to detect faces of all sizes as much as possible. Sliding window with fixed-size scans a set of whole images across multiple scales to generate a large number of candidate windows. Coarse-detection Network is a very shallow convolutional neural network, which can quickly process these windows and produce a classification probability and bounding box regression vector corresponding to each window. Keep windows with probabilities above a preset threshold and then use the estimated bounding box regression vector to calibrate remaining windows's position. Finally use non-maximum suppression(NMS) to merge highly overlapping windows.

Optimized-detection Network further rejects a large number of false detections, calibrates the position of the remaining windows with the bounding box regression vectors and performs NMS. The first two networks also output pose regression vectors for each window, but because of the large number of false positives in the initial stages, these vectors are not used as the final pose estimation result. In the final stage, face detection result from the Fine-detection & Pose-estimation Network is very fine. At this moment, we use the

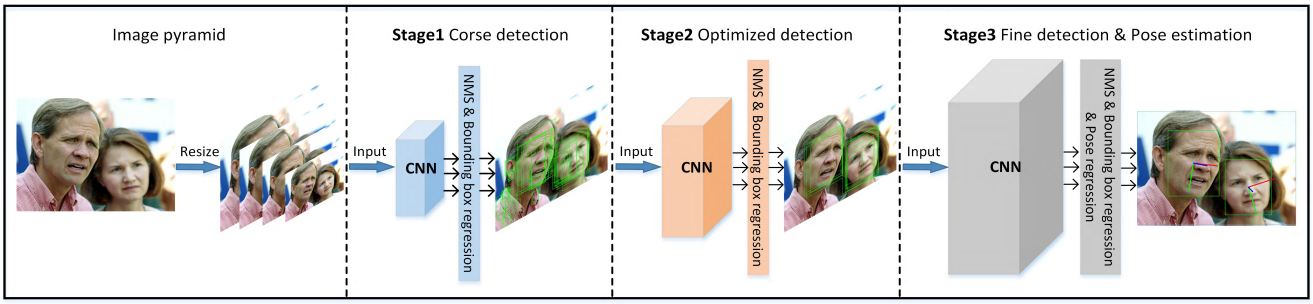


FIGURE 3. Pipeline of our multi-task framework for simultaneous face detection and pose estimation. The framework consists of three cascaded deep convolutional neural networks. From left to right, the following CNN optimizes the face detections of the front CNN. In the last stage, CNN performs pose estimation on fine face detections.

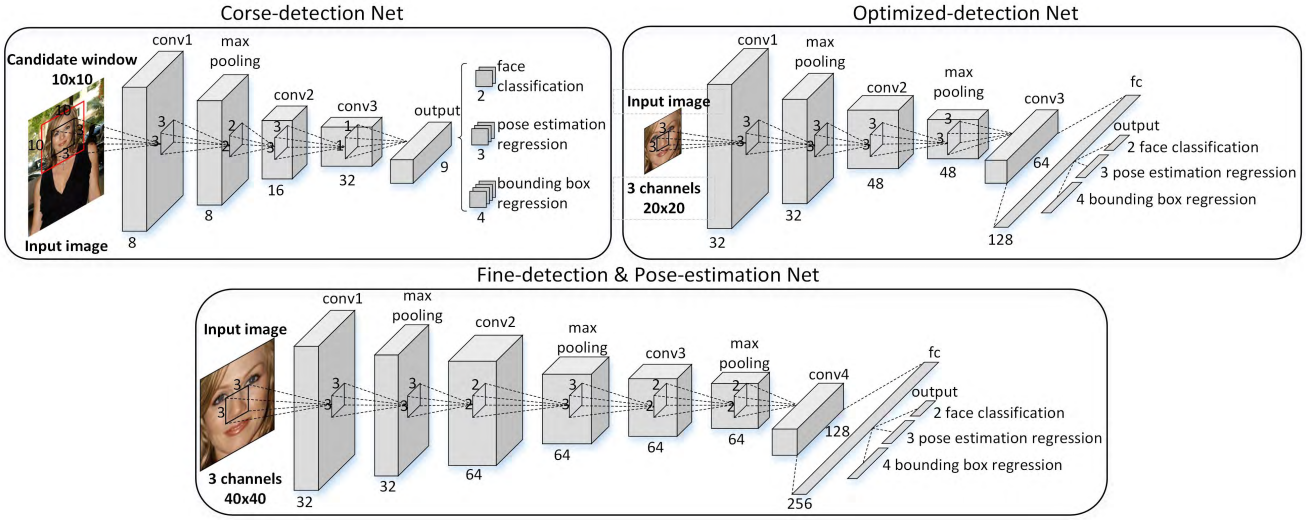


FIGURE 4. CNN architectures of the Coarse-detection Network, Optimized-detection Network and Fine-detection & Pose-estimation Network.

pose regression vectors from this network to estimate the pose for each detected face.

B. CNN ARCHITECTURES

1) INSPIRATION AND CONSIDERATIONS

Cascaded convolutional neural networks have achieved very good performance in the field of face detection. This structure is inspired by the Viola-Jones framework, which not only has high detection rate, but also has strong real-time performance. Reference [16] first designed a three-stage cascaded convolutional neural network architecture for face detection and designed three calibration networks to calibrate the position of the face boxes. Reference [17] removed calibration networks above and added bounding box regression to detection networks to correct face boxes's position and joined the face alignment task to promote face detection's performance. In view of our face detection and pose estimation tasks, we refer to these excellent achievements to design our CNN architectures.

Considering reducing the size of the input image helps to detect smaller faces, we respectively adjust the input image size of the three networks to $10 \times 10, 20 \times 20, 40 \times 40$. As for the specific network width and depth, we mainly consider our training data's scale. We observe that if dataset size is too small and the network is too complex, the designed model will be overfitting and waste a lot of computation resources. We determine the final network structure based on the classification effect of the model to the dataset we have. These network structures are flexible, and when a larger dataset is used for training, the network complexity can be increased, but more computing time will be brought. We can determine the final model by making a trade-off between accuracy and calculation time.

2) COARSE-DETECTION NETWORK

Coarse-detection Network refers to the first CNN in the framework. The structure of this CNN is shown in Fig. 4. Coarse-detection Network is a very shallow fully convolutional network which is used to quickly find regions that contain exactly one face on the images. The input size of the network is 10×10. Given an input image, scale the image with a fixed proportion to obtain the multi-scale images, and then fed all the images at different scales into the network to get all feature maps corresponding to each scale. The size of

the final feature map corresponding to the input image of size W × H is:

$$\lceil (W-10)/2 \rceil \times \lceil (H-10)/2 \rceil \quad (1)$$

where $\lceil \cdot \rceil$ denotes ceiling operations.

Each point on the feature map corresponds to a window of size 10 × 10 on the image, which is represented by a vector of size 1 × 9. It gives a face vs. non-face classification probability distribution $p = (p_0, p_1)$, bounding box regression offsets $t = (t_{x_1}, t_{y_1}, t_{x_2}, t_{y_2})$ and pose regression vector $h = (yaw, pitch, roll)$ for each window. We keep the windows whose scores are higher than the pre-set threshold T_1 and then apply Non-maximum suppression (NMS) to merge the highly overlapped windows. Finally, we employ the bounding box regression offsets to refine the location for each remaining window:

$$Loc^*_{x_1} = Loc_{x_1} + |Loc_{x_1} - Loc_{x_2}| * t_{x_1} \quad (2)$$
$$Loc^*_{y_1} = Loc_{y_1} + |Loc_{y_1} - Loc_{y_2}| * t_{y_1} \quad (3)$$
$$Loc^*_{x_2} = Loc_{x_2} + |Loc_{x_1} - Loc_{x_2}| * t_{x_2} \quad (4)$$
$$Loc^*_{y_2} = Loc_{y_2} + |Loc_{y_1} - Loc_{y_2}| * t_{y_2} \quad (5)$$

where (Loc_{x_1}, Loc_{y_1}) and (Loc_{x_2}, Loc_{y_2}) denote the top-left and lower-right coordinates of the window. The variables $t_{x_1}, t_{y_1}, t_{x_2}, t_{y_2}$ denote bounding box regression offsets of the four coordinates.

Vector $(yaw, pitch, roll)$ gives the predicted pose estimation result for each window. The three variables are used as optimization target in the training stage. However, in the testing stage, the detection windows contains a lot of false positives and we discard these values.

3) OPTIMIZED-DETECTION NETWORK

Optimized-detection Network is the CNN after Coarse-detection Network, which further rejects false detection windows from last CNN. Its CNN structure is shown in Fig. 4. The input size is 20 × 20. Because we use the bounding box regression offsets to calibrate the position of the remaining window in the Coarse-detection stage, the window is not square in this stage. Given a detection window $(x_{min}, y_{min}, x_{max}, y_{max})$ with top-left corner at (x_{min}, y_{min}) and lower-right corner at (x_{max}, y_{max}), we pad pixels near the window center to form a square box:

$$x'_{min} = x_{center} - \frac{1}{2} max\,(w, h) \quad (6)$$
$$y'_{min} = y_{center} - \frac{1}{2} max\,(w, h) \quad (7)$$
$$x'_{max} = x_{center} + \frac{1}{2} max\,(w, h) \quad (8)$$
$$y'_{max} = x_{center} + \frac{1}{2} max\,(w, h) \quad (9)$$

where (x_{center}, y_{center}) denotes the coordinate of window center and $max\,(w, h)$ denotes the length of longer side. We crop out each window region from the original image and resize them to 20 × 20 as CNN input.

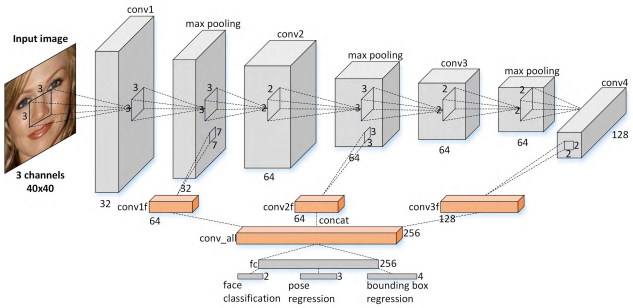


FIGURE 5. The architecture of improved CNN in the last stage.

The final layer outputs a vector (1 × 9) containing face vs. non-face binary classification score (1 × 2), bounding box regression offsets (1 × 4) and pose regression vector (1 × 3). Like before, we remain the windows whose scores are higher than the pre-set threshold T_2, perform NMS merge and calibration with bounding box regression. Similarly, the remaining windows in this stage still contain some false positives, and we discard yaw, pitch and roll values.

4) FINE-DETECTION & POSE-ESTIMATION NETWORK

Fine-detection & Pose-estimation Network is the last CNN in the framework. After filtering in the first two stages, the number of remaining windows is very few at this time. We do not need to think about the time efficiency and the network structure is more complicated to achieve better discrimination. Its network structure is shown in Fig. 4 and the input size is 40 × 40. We remain the windows whose scores are higher than the pre-set threshold T_3. Unlike the previous two stages, the detection windows in this stage is very fine and contain only a very small number of false positives. And pose estimation is more accurate because of the more complicated network structure. We take the remaining windows of this network as face detection results, and take three orientation angles: $(yaw, pitch, roll)$ of each detection window as pose estimation results.

C. CNN FEATURE FUSION STRATEGY

The information contained in features throughout the CNN is hierarchically distributed. Lower layers in the network contain better localization properties because they respond to various edges and corners and hence are more suitable for structure dependent tasks such as pose estimation. On the other hand, deeper layers contain more complicated features and are class-specific. As a result, they are more suitable for learning face detection task.

In our baseline framework, we construct the network structure layer by layer, which only takes advantage of the complex features of deeper layers and ignores the simple features of lower layers. To emphasize the importance of CNN feature fusion, we reconstruct a new CNN by fusing the low, mid and high-level layers of the last CNN in the baseline framework. Its CNN structure is shown in Fig. 5. The effect of feature fusion is demonstrated in Section VI.

V. TRAINING THE NETWORK

In this section, we will introduce the training details of our framework. The training pipeline contains three separate CNNs. We crop image patches from two public datasets to collect four kinds of samples and resize them to 10×10, 20×20, 40×40 as input. The learning target is to minimize a multi-task loss including face classification, bounding box regression and pose regression.

A. TRAINING DATASETS

1) WIDER FACE

WIDER FACE [61] is a very challenging face detection dataset, of which faces vary largely in facial expression, scale, pose, occlusion and background clutters. Based on the detection rate of Edge Box [60], the dataset is divided into three partitions: Easy, Medium and Hard. It consists of 32,203 images and 393,703 labeled faces with rich annotations including poses, occlusions, event categories and face bounding boxes. WIDER FACE contains 60 event classes covering a large number of real world scenes. For each event class, 40%/10%/50%/ data is for training, validation and testing. We use the training set which contains 12,880 images as training dataset for face detection task.

2) ANNOTATED FACIAL LANDMARKS IN THE WILD

Annotated Facial Landmarks in the Wild (AFLW) dataset [62] provides a large-scale collection of images which vary largely in face appearance such as age, gender, pose, expression and ethnicity. The dataset is designed to meet the needs of a multi-view, large-scale, real-world face dataset for facial feature localization. It contains 25,993 faces in 21,997 real-world images with realistic scenes and provides annotations for 21 landmark points per face together with the face bounding boxes, gender information and face pose (yaw, pitch, roll). We use all images as training dataset for pose estimation task.

B. DATA PREPROCESSING

In training our multi-task CNNs in the cascade, we use the 12,880 images from Wider Face training dataset and 21,997 images from AFLW dataset. No other data is used. We use these images to generate four kinds of training data: (i)Positive samples: Regions that have Intersection-over-Union (IoU) overlap with any ground-truth faces of larger than 0.7; (ii)Negative samples: IoU ratio with a ground-truth face in the interval [0, 0.3); (iii)Part samples: IoU ratio with a ground-truth face in the interval [0.4, 0.7); (iv)Pose samples: Regions that have IoU ratio above 0.7 to any ground-truth face and have face pose annotations.

To make CNNs converge easily, the pixel values of all input patches are normalized to (-1, 1). No other pre-processing is used.

C. OPTIMIZATION

We employ a multi-task loss including face vs. non-face classification, bounding box regression and pose regression to jointly optimize three tasks.

1) FACE CLASSIFICATION

Positive samples and Negative samples are used for face vs. non-face classification task. For each sample x_i, a cross-entropy loss L_i^{cls} is used for training:

$$L_i^{cls} = -\left[y_i^{cls} log(p_i) + (1 - y_i^{cls}) log(1 - p_i)\right] \tag{10}$$

where p_i is the probability that indicates a sample being a face. $y_i^{cls} = 1$ if the sample belongs to positive samples, otherwise 0.

2) BOUNDING BOX REGRESSION

Positive samples and Part samples are used for bounding box regression task. For each sample, we calculate the bounding box offsets between it and the nearest ground truth:

$$t_{x_1}^* = (x_1^* - x_1)/w_p \tag{11}$$
$$t_{y_1}^* = (y_1^* - y_1)/h_p \tag{12}$$
$$t_{x_2}^* = (x_2^* - x_2)/w_p \tag{13}$$
$$t_{y_2}^* = (y_2^* - y_2)/h_p \tag{14}$$

where (x_1, y_1) and (x_2, y_2) denote the top-left and lower-right coordinates of the sample, (x_1^*, y_1^*) and (x_2^*, y_2^*) denote the top-left and lower-right coordinates of ground truth, w_p and h_p denote box width and height of sample respectively.

For each sample x_i, we use Euclidean loss L_i^{loc} for training:

$$L_i^{loc} = \left\| \hat{y}_i^{loc} - y_i^{loc} \right\|^2 \tag{15}$$

where $\hat{y}_i^{loc}$ denotes the four ground-truth offsets $[t_{x_1}^*, t_{y_1}^*, t_{x_2}^*, t_{y_2}^*]$ and y_i^{loc} denotes the predicted offsets $[t_{x_1}, t_{y_1}, t_{x_2}, t_{y_2}]$.

3) POSE REGRESSION

Pose samples are used for pose regression task. The learning objective is to regress three angles (yaw,pitch,roll) describing face pose. Among three angles, $\forall\,(yaw, pitch, roll) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Similar to the bounding box regression task, we use Euclidean loss L_i^{pose} for training:

$$L_i^{pose} = \left\| \hat{y}_i^{pose} - y_i^{pose} \right\|^2 \tag{16}$$

where $\hat{y}_i^{pose}$ denotes the three ground-truth pose angles and y_i^{pose} denotes the predicted pose angles.

4) MULTI-TASK LOSS

Since we jointly perform face vs. non-face classification, bounding box regression and pose regression in each CNN, multi-task loss is the weighted sum of the three individual losses. The overall learning objective is formulated as:

$$argmin \Sigma_{x_i} \Sigma_{t \in \{cls, loc, pose\}} \lambda_t \left[u_i^t \geq 1\right] L_i^t + \alpha \sum_w w^2 \tag{17}$$

where x_i is each training sample. The loss weight parameter λ_t is decided based on the task importance in the total loss. We choose ($\lambda_{cls} = 1, \lambda_{loc} = 1, \lambda_{pose} = 0.5$) in first two stages, while ($\lambda_{cls} = 1$, $\lambda_{loc} = 0.5$, $\lambda_{pose} = 1$) in last stage. Higher weights are assigned to pose estimation task in last stage

because they need more accuracy. The indicator function $\left[u_i^t \geq 1\right]$ indicates whether sample x_i is used to compute loss of task t. A sample is only used for one task. It evaluates to 1 when sample x_i is used for task t, otherwise 0. To reduce overfitting, we add L2 Regularization in total loss. α denotes the regularization parameter.

D. TRAINING PROCEDURE

We train three networks one by one. During the training process, each network needs to optimize three subtasks simultaneously, including face classification, bounding box regression and pose regression. The front network generates a lot of hard negatives for the back network as a supplement to their training set. The training procedure is as follows.

We randomly crop patches on the images from Wider Face dataset to collect positive, part and negative samples. Then we crop patches near the ground-truth faces from AFLW dataset to collect pose samples. All samples are cropped and resized to 10×10, 20×20, 40×40. We mark these samples of different sizes as x10-Subset-I, x20-Subset-I and x40-Subset-I, respectively. We train the Coarse-detection Network using the samples from x10-Subset-I. After the training is completed, we then apply Coarse-detection Network on images from Wider Face to collect patches whose confidence score are higher than a pre-set threshold γ_1. The patches are divided into positive, part or negative samples based on their IoU ratio with any ground-truth face. All samples are cropped and resized to 20×20, 40×40. We mark these samples of different sizes as x20-Subset-II and x40-Subset-II, respectively.

Optimized-detection Network is trained with the samples from x20-Subset-I and x20-Subset-II. After that, we follow the same procedure and apply a 2-stage cascade consisting of the Coarse-detection Network and Optimized-detection Network to collect patches with confidence score larger than γ_2. We crop out all patches and resize them to 40×40. These samples are marked as x40-Subset-III.

Following the same procedure, we train the Fine-detection & Pose-estimation Network with the samples from x40-Subset-I, x40-Subset-II and x40-Subset-III.

During training, samples are horizontally flipped with probability 0.5 to apply data augmentation. For the sample ratio, we set the ratio of positive, part, pose and negative samples as 1:1:1:3 in the first two stages and adjust it to 1:1:2:2 in last stage. We set γ_1 and γ_2 empirically. The principle is to generate a sufficient number of hard samples while keeping high recall.

E. SGD HYPER-PARAMETERS

We employ stochastic gradient descent to train the CNNs. Each CNN has three sibling output layers (face classification, bounding box regression, pose regression). Except that the classification output-layer uses softmax activation function, we apply ReLU nonlinearity activation function after the remaining layers. Weights in each layer are initialized from zero-mean Gaussian distributions with standard deviations 0.001 and biases are initialized to 0. We choose iteration number based on the number of training samples. We train the fist two CNNs for 15 epochs and train the last CNN for 30 epochs. In each epoch, we guarantee to feed all positive samples into the CNN to train the network. The number of Mini-batch is related to GPU memory. Setting it to 256 is suitable for our model. We set initial learning rate 0.01 for all layers and adjust it throughout training. We divide the learning rate by 10 every 1/3 epochs. A momentum of 0.9 and L2 regularization parameter of 0.0005 are used.

VI. EXPERIMENTS

In this section, we conduct an empirical study of evaluating our face detector on the challenging Face Detection Data Set and Benchmark (FDDB) dataset [63] and evaluating our pose estimation on the Annotated Faces in the Wild (AFW) dataset [33]. On the two datasets, our method are comparable with state-of-the-art methods. We then evaluate the effect of joint training of face detection and pose estimation. Meanwhile, we show that CNN feature fusion strategy can further improve the performance of pose estimation. Finally, we evaluate the runtime efficiency of our method and make a comparison with other methods.

A. FACE DETECTION RESULTS

We compare our face detection result against the state-of-the-art methods on FDDB dataset. FDDB dataset consists of a set of 2,845 images containing 5,171 faces with a wide range of visual difference including occlusions, extreme poses, out-of-focus and low-resolution faces. For each detection region, it defines two kinds of evaluation score: Discrete score (DS) and Continuous score (CS). Discrete score evaluates to 1 when the detection region has an IoU ratio above 0.5 to its corresponding matching ground-truth face, otherwise 0. Continuous score evaluates the matching metric of the detection region according to the continuous IoU ratio.

Fig. 6 compares the performance of different detectors using two kinds of Receiver Operating Characteristic (ROC) curves corresponding to each kind of score on the FDDB dataset. Compared with the other results of the published methods including ACF-multiscale [28], Zhu and Ramanan [33], HeadHunter [34], Yan *et al.* [35], Joint Cascade [52], HyperFace [13], CascadeCNN [16], MTCNN [17], Faceness [38], Faster RCNN [37], HR [40], our method performs better than most of the reported algorithms. We compare our result with two other cascade-based methods CascadeCNN [16], MTCNN [17]. For the discrete ROC score, our method outperforms [16] by a big margin and is comparable with [17]. But for continuous ROC score, our method performs poorer than the two methods. The reason for the performance drop on the continuous scores can be attributed to the difference of face bounding box results from different detectors. Reference [16] transformed their square detection boxes to be rectangles by vertical extension for better comparison with ellipse annotation on FDDB. We do not do any extra processing on our detection output.

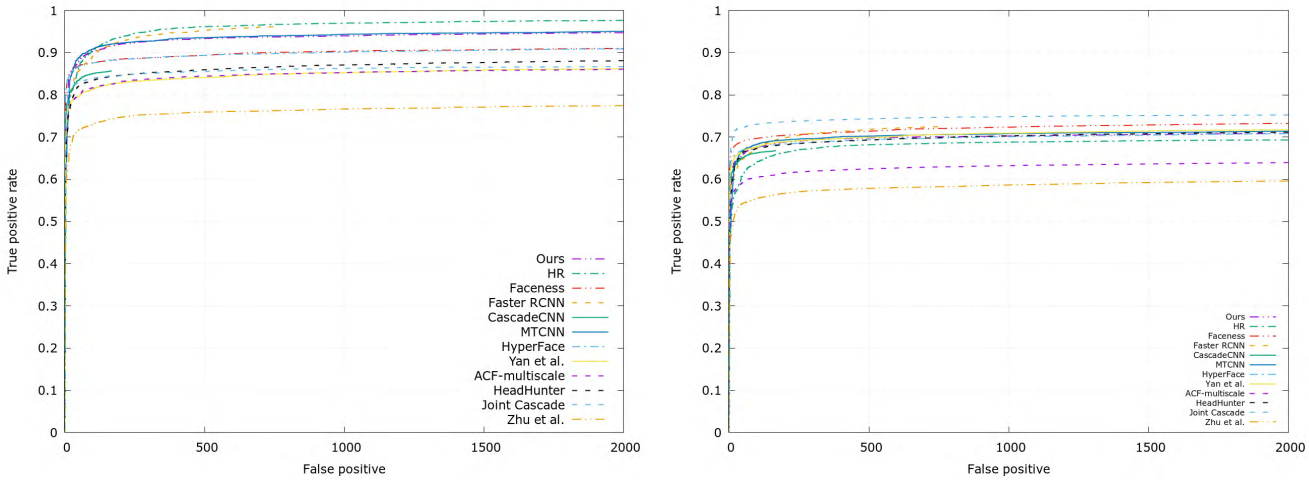


FIGURE 6. Comparisons of face detection with state-of-the-art methods on (a) ROC curves on FDDB with discrete scores, (b) ROC curves on FDDB with continuous scores.

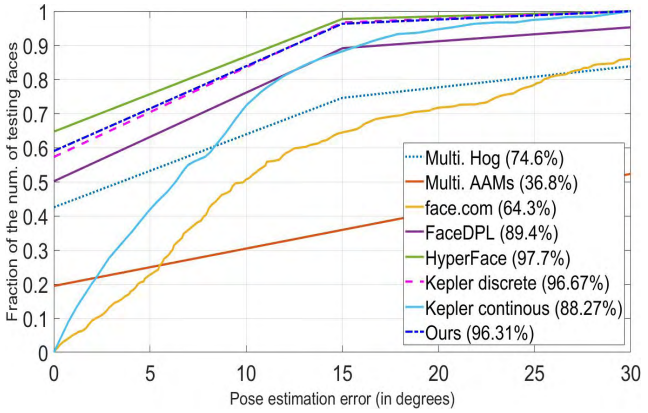


FIGURE 7. Pose Estimation cumulative error distribution curves on AFW dataset. The numbers in the legend are the percentage of test faces with absolute yaw error less than or equal to 15°.

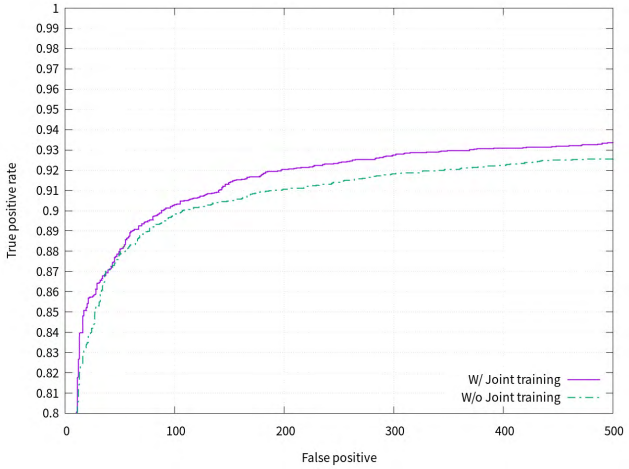


FIGURE 8. Comparisons of ROC curves on FDDB with and without joint training.

Reference [17] jointly train their face detector with face alignment while we jointly train face detection and pose estimation. Different face-related tasks have different promotion on face detection. Other reasons include the scale of the training data, the structure of the network, and so on.

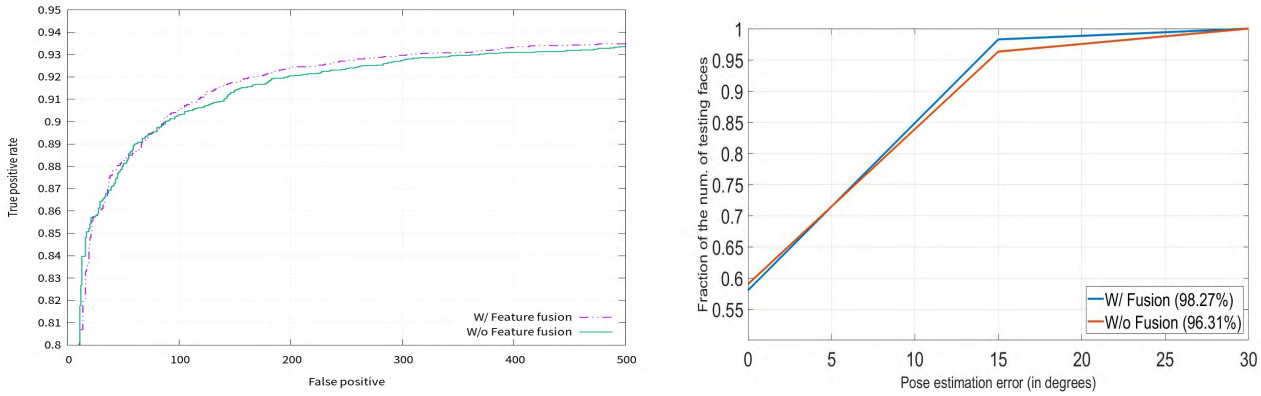
In addition to the quantitative evaluation results, we further randomly choose some qualitative results in Fig. 10. It can be observed that our method is able to deal with challenging cases such as non-frontal faces, heavily occluded faces, small faces, faces with low resolution, faces under weak light and faces with extreme poses and scales.

B. POSE ESTIMATION RESULTS

AFW is a very popular dataset, which is commonly used for the evaluation of face alignment algorithms. It consists of 205 images collected from Flickr containing 468 in-the-wild faces with absolute yaw degree up to 90°. The images in this dataset has a huge difference in face pose. Hence, AFW is widely used for evaluating pose estimation. Many methods have been evaluated on this dataset. In order to make a convenient comparison with these methods, we also present the pose estimation performance on the AFW dataset.

Following the protocol defined in [33], we compute the mean average error only for the yaw angles. Because the dataset provides the ground-truth yaw angles in multiples of 15°, we round-off our predicted yaw to the closest 15° multiple for evaluation like other methods.

We compare our method with Multi. AAM [33], Multiview HoG [33], FaceDPL [64], face.com and deep-learning approaches including HyperFace [13], Kepler [49]. Fig. 7 shows the cumulative error distribution curves on AFW dataset. As can be observed from the figure, our method outperforms the first four methods by a large margin and is comparable with state-of-the-art methods using deep-learning.



(a) (b)

FIGURE 9. (a) Comparisons of ROC curves on FDDB with and without feature fusion, (b) Comparisons of cumulative error distribution curves on AFW with and without feature fusion.

FIGURE 10. Qualitative results of our method on FDDB.

However, HyperFace [13] takes 3s to process an image and All-in-One [14] takes 3.5s per image. The major bottleneck for speed roots in the process of generating region proposals. Not like the first two methods to accept an entire image as input, Kepler [49] only takes face boxes on each image as input and processes 3-4 frames per second. Our method processes 30 entire images per second and the speed is very competitive compared with other methods. We also show some qualitative results in Fig. 11.

C. ABLATION EXPERIMENTS

To explore the correlation between the two tasks and gain the deep insights of the improvement obtained by CNN feature fusion, we conduct more additional experiments for ablation studies.

1) JOINT TRAINING OF FACE DETECTION AND POSE ESTIMATION

To evaluate the contribution of joint training of face detection and pose estimation, we remove pose regression loss from total loss and retrain our model. For fair comparison, all training data and parameter setting are same. Fig. 8 gives the performance of two frameworks(joint pose estimation task and no joint it) on FDDB dataset. It can be observed that joint training promotes the performance of face detection.

2) CNN FEATURE FUSION

To examine the impact of CNN feature fusion strategy, we reconstruct the Fine-detection & Pose-estimation Network by fusing the low, mid and high-level layers of the last CNN in the baseline framework. To simplify the experiment, we just change the structure of the last CNN. Fig. 9 gives the performance of two CNNs(with feature fusion and without feature fusion). As can be seen from the figure, face detection task does not gain much by fusing intermediate layers, but pose estimation task achieves a boosted performance. It can be attributed to the fact that fusing the layers brings hidden information from lower layers which can boost the performance of structure dependent tasks such as pose estimation.

FIGURE 11. Qualitative results of our method on AFW.

D. RUNTIME EFFICIENCY

Cascade-based methods has the advantage of runtime efficiency. Compared with the design to have only one single CNN with very large depth to scan the entire image for faces, cascade structure combines several simple CNN for faster face detection and achieves even better accuracy.

We evaluate our method on VGA images with resolution 640×480. We set scaling factor as 0.8 and minimum face size as 40×40. We implement our experiment on NVIDIA GTX 1080Ti GPU and our method achieve about 30fps for the two tasks. Compared with the methods for resolving multiple face-related tasks including face detection and pose estimation, our method is quite fast in computation speed among them. HyperFace [13] takes 3s per image on a GTX TITAN X GPU and All-in-one [14] takes an average of 3.5s to process an image. Considering GTX Titan X GPU is 62% slower than the 1080Ti GPU, our method is equivalent to reaching 18fps with the previous one. Compared with these methods, our method achieves the real-time performance for simultaneous face detection and pose estimation.

VII. CONCLUSION

In this paper, we propose a multi-task convolutional neural network cascade framework for simultaneous face detection and pose estimation, while keeping real-time performance. Through multi-task learning, face detection task gain a boosted performance from pose estimation task. Through cascade structure, our method achieve real-time performance for resolving these two tasks. We further promote the performance of pose estimation through feature fusion. Extensive experiments on available unconstrained datasets demonstrate that our method outperforms most of the state-of-the-art methods on two tasks. In the future, we will extend our method for other applications such as simultaneous human detection and human pose estimation.

REFERENCES

[1] M. Osadchy, Y. Le Cun, and M. L. Miller, "Synergistic face detection and pose estimation with energy-based models," *J. Mach. Learn. Res.*, vol. 8, pp. 1197–1215, Jan. 2007.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[3] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland:Springer, 2014, pp. 818–833.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[5] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[7] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[11] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[12] N. Ruiz, E. Chong, and J. M. Rehg. (Oct. 2017). "Fine-grained head pose estimation without keypoints." [Online]. Available: https://arxiv.org/abs/1710.00925

[13] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[14] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May/Jun. 2017, pp. 17–24.
[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Dec. 2001, p. 1.
[16] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5325–5334.
[17] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[18] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, Jan. 2002.
[19] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Tech. Rep. MSR-TR-2010-66, 2010.
[20] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," *Comput. Vis. Image Understand.*, vol. 138, pp. 1–24, Sep. 2015.
[21] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, Apr. 2009.
[22] H. Qin, J. Yan, X. Li, and X. Hu, "Joint training of cascaded CNN for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3456–3465.
[23] R. Vaillant, C. Monrocq, and Y. L. Cun, "Original approach for the localisation of objects in images," *IEE Proc.-Vis., Image Signal Process.*, vol. 141, no. 4, pp. 245–250, Aug. 1994.
[24] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.
[25] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1998, pp. 38–44.
[26] C. Garcia and M. Delakis, "A neural architecture for fast and robust face detection," in *Proc. 16th Int. Conf. Pattern Recognit.*, vol. 2, 2002, pp. 44–47.
[27] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block LBP representation," in *Proc. Int. Conf. Biometrics*. Berlin, Germany: Springer, 2007, pp. 11–18.
[28] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Sep. 2014, pp. 1–8.
[29] C. Zhang, J. C. Platt, and P. A. Viola, "Multiple instance boosting for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1417–1424.
[30] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 671–686, Apr. 2007.
[31] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 236–243.
[32] R. Xiao, L. Zhu, and H.-J. Zhang, "Boosting chain learning for object detection," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 709–715.
[33] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2879–2886.
[34] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 720–735.
[35] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2497–2504.
[36] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[37] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May/Jun. 2017, pp. 650–657.
[38] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3676–3684.
[39] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, and X. Hu, "Scale-aware face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 3, Jul. 2017, pp. 6186–6195.
[40] P. Hu and D. Ramanan, "Finding tiny faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1522–1530.
[41] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 617–624.
[42] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Gool, "Random forests for real time 3D face analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 437–458, 2013.
[43] C. BenAbdelkader, "Robust head pose estimation using supervised manifold learning," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 518–531.
[44] J. Lu and Y. P. Tan, "Ordinary preserving manifold analysis for human age and head pose estimation," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 2, pp. 249–258, Mar. 2013.
[45] B. Ahn, J. Park, and I. S. Kweon, "Real-time head orientation from a monocular camera using deep neural network," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 82–96.
[46] R. S. Ghiass, O. Arandjelović, and D. Laurendeau, "Highly accurate and fully automatic head pose estimation from a low quality consumer-level RGB-D sensor," in *Proc. ACM 2nd Workshop Comput. Models Social Interact., Hum.-Comput.-Media Commun.*, 2015, pp. 25–34.
[47] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks)," in *Proc. Int. Conf. Comput. Vis.*, vol. 1, no. 6, 2017, pp. 1021–1030.
[48] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3D solution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 146–155.
[49] A. Kumar, A. Alavi, and R. Chellappa, "KEPLER: Keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May/Jun. 2017, pp. 258–265.
[50] B. Shi, X. Bai, W. Liu, and J. Wang, "Face alignment with deep regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 183–194, Jan. 2018.
[51] R. Caruana, "Multitask learning," in *Machine Learning*. Springer, 1998, pp. 95–133.
[52] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 109–122.
[53] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 94–108.
[54] G. Levi and T. Hassncer, "Age and gender classification using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 34–42.
[55] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 447–456.
[56] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
[57] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3626–3633.
[58] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1215–1223.
[59] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.
[60] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland:Springer, 2014, pp. 391–405.
[61] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5525–5533.

[62] M. Köstinger, 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 *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 2144–2151.
[63] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. UM-CS-2010-009, 2010, p. 8, vol. 2, no. 7.
[64] X. Zhu and D. Ramanan, "FaceDPL: Detection, pose estimation, and landmark localization in the wild," vol. 1, no. 2, p. 6, 2015.

HAO WU received the M.S. and Ph.D. degrees in control theory and control engineering from Shandong University in 1997 and 2011, respectively. She is currently an Associate Professor at Shandong University. She has authored nearly 20 journal and conference papers. Her research mainly focuses on navigation of service robot and human behavior understanding.

KE ZHANG received the B.Sc. degree in automation from Henan University, Kaifeng, China, in 2016. He is currently pursuing the master's degree with the School of Control Science and Engineering, Shandong University, Jinan, Shandong, China. His research interests include face detection and facial attribute analysis.

GUOHUI TIAN received the M.S. degree in industry automation from Shandong University, Jinan, China, in 1993, and the Ph.D. degree in automatic control theory and application from Northeastern University, Shenyang, China, in 1997. He was a Post-Doctoral Researcher with the Engineering Department, Tokyo University, from 2003 to 2005. He is currently a Professor with the School of Control Science and Engineering, Shandong University. His research mainly focuses on service robots and smart space.