FACE DETECTION AND RECOGNITION FOR HOME SERVICE ROBOTS WITH END-TO-END DEEP NEURAL NETWORKS

Wei Jiang, Wei Wang

Robotics Lab, Futurewei Technologies Inc.
Santa Clara, USA

ABSTRACT

This paper proposes an effective end-to-end face detection and recognition framework based on deep convolutional neural networks for home service robots. We combine the state-of-the-art region proposal based deep detection network with the deep face embedding network into an end-to-end system, so that the detection and recognition networks can share the same deep convolutional layers, enabling significant reduction of computation through sharing convolutional features. The detection network is robust to large occlusion, and scale, pose, and lighting variations. The recognition network does not require explicit face alignment, which enables an effective training strategy to generate a unified network. A practical robot system is also developed based on the proposed framework, where the system automatically asks for a minimum level of human supervision when needed, and no complicated region-level face annotation is required. Experiments are conducted over WIDER and LFW benchmarks, as well as a personalized dataset collected from an office setting, which demonstrate state-of-the-art performance of our system.

***Index Terms*—** Face Detection, Face Recognition, Convolutional Neural Network

1. INTRODUCTION

Being able to detect and recognize human faces is essential for home service robots in a variety of applications such as home security and surveillance, as well as personalization and natural user-robot interaction. The recent significant advances in face detection [1, 2, 3] and face recognition [4, 5, 6] by using deep neural networks make it possible to handle several challenging conditions: large pose variations and occlusions, difficult lighting conditions, and poor-quality images with large motion blurs. However, some issues remain unsolved for practical systems to operate in personalized home environments, including limitations in computing powers, and the lack of training data for personalized face identification.

Most current systems decompose the face detection and recognition task into three stages: face detection where faces are localized in an input image, face alignment where the detected face is warped into a 2D or 3D canonical face model, and face recognition where the aligned face is classified into different identities. Each part has been actively studied in the field and near-human performances have been achieved over many benchmark datasets. Also, recent work [4, 5] has shown that deep convolutional neural networks (CNN) can get good recognition performance without explicit face alignment.

In this paper, we propose an end-to-end deep CNN-based face detection and recognition framework for home service robots. We combine the state-of-the-art region proposal network (RPN) based detection [7] and the FaceNet embedding network [4] in an end-to-end framework as described in Figure 1. The main advantages of the proposed system lie in three folds. (1) At test time, the face detection and face recognition modules share computation through sharing the same deep convolutional layers, which significantly reduces the computation memory requirement and computation time. This is essentially important for robotic applications with limited on-robot computing powers. (2) The RPN-based face detection network is powerful in localizing faces with large scale, pose, and lighting variations and with large occlusions. Compared with cascade-CNN [2], the feature map generated by RPN-based detection network has higher capacity for better recognition later on. Compared with the Faceness network designed for face detection [3], the RPN-based face detector shares a common set of convolutional layers with the later recognition module. (3) The FaceNet embedding framework does not require explicit 2D or 3D face alignment, which enables an effective training strategy that trains the detection and embedding networks to generate a unified network with shared convolutional features.

Based on the proposed end-to-end CNN-based face detection and recognition network, we further develop a practical home service robot system with automatic face collection and remote model training and updates, as described in Figure 2. Our system automatically asks for human supervision when needed, *i.e.*, when an unknown identify is introduced to the robot for the first time. Also, only minimum amount of human supervision is required, *i.e.*, providing associated names for the unknown identify to the robot. With automatic data collection using combined face detection and tracking, our system does not require complicated region-level face annotation that is too expensive to acquire in practical applications.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

The final robot system is trained in two stages. Before being deployed into any specific home, the end-to-end network described in Figure 1 is pre-trained on the challenging WIDER face detection benchmark [8] and LFW face recognition benchmark [9]. After deployment, layers of the recognition network are fine-tuned according to newly collected data in the deployed home environment, and successive classifiers are trained to classify different identities in the home. Experiments over FDDB [10] and LFW [9] benchmarks and a self-collected dataset in an office setting demonstrate the effectiveness of our system.

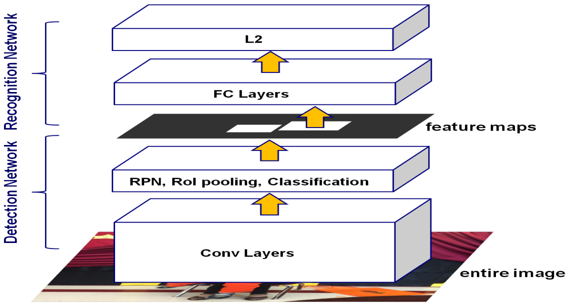


Fig. 1. The unified network structure for end-to-end face detection and recognition.

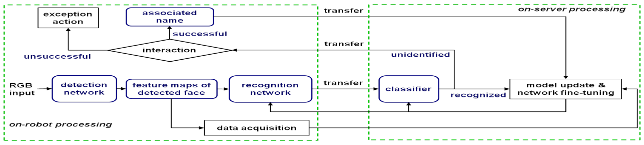


Fig. 2. The practical face detection and recognition framework for home service robots with automatic face collection and remote model training and updates.

2. RELATED WORK

2.1. Face detection

Viola and Jones [11] first introduced fast boosted cascade classifier using Haar-like features. Among the many variants using the cascade structure, state-of-the-art face detection performance was achieved by [12], which combines face detection and alignment in the same cascade framework. By treating a face as a collection of facial parts, deformable part models (DPM) [13, 14] can handle occlusions more robustly than cascade based methods. These two types of approaches dominated the face detection field in the past decade.

Due to the recent advances in object detection and recognition using deep neural networks [7, 15], researchers have revisited the face detection methods using neural networks, which was once actively studied in the field [16]. Recent studies [1] showed that by using the high capacity of deep CNNs, the performance of face detection can be further improved, especially in the challenging conditions with large pose variations and occlusions. Furthermore, the Faceness network was proposed in [3], which was specially designed to discover facial part responses. A cascade CNN framework was developed in [2] for detecting faces at multiple scales. The most recent experiments in [17] show that using the region proposal network [7] as the face detector gives better performance than these specially designed face detection networks.

2.2. Face recognition

Significant advances have been achieved by using deep neural networks for face verification and recognition. Many approaches use a multi-stage framework, where faces are first aligned, either to a canonical frontal view [18] or to a general 3D model [6], and then CNNs are learned in conjunction with a classifier (*e.g.*, SVM) for face classification. Without explicit face alignment, the work of [5] trains a network with combined verification and classification loss to achieve effective recognition. Later on, the FaceNet framework is proposed [4], which learns an optimal feature embedding for face verification, classification, and clustering. It has been shown that the state-of-the-art face recognition performance can be achieved without explicit 2D or 3D face alignment [4].

3. NETWORK STRUCTURE

In this work, we propose an effective face detection and recognition system based on end-to-end deep convolutional neural networks. Figure 1 shows the overall network structure of our system. Specifically, our system is composed of two parts, the detection network and the recognition network. The detection network follows the structure of the faster R-CNN network developed in [7], which is further composed of two modules: the deep fully convolutional network module for region proposal and the fast R-CNN [19] detector that uses the proposed regions. In our network, the last fully connected layer of the fast R-CNN module computes softmax over only 2 categories, face and non-face. All other structures of the fully convolutional and the faster R-CNN network remain the same as in [7].

The recognition network follows the structure of the deep CNN based FaceNet [4]. Since our goal is to share computation between detection and recognition networks, we assume that both the detection network and the recognition network share a common set of convolutional layers. In our final system, we use the VGG-16 model [20] with 13 sharable convolutional layers. It is worth mentioning that we have tried the Zeiler and Fergus model (ZF) [21], which has 5 sharable convolution layers. However, experiments showed significant performance drop when using ZF network. With VGG-16, through the detection network, a $512 \times 7 \times 7$ feature map is generated after the RoI pooling layer for each face region x, which is treated as inputs to the recognition network. The recognition network is composed of several fully connected layers, followed by an L_2 normalization layer, which computes an embedding $f(x)$, such that the squared distance between all faces of the same identify is small, while the squared distance between a pair of faces from different identifies is large. That is, the embedding feature space R^d is discriminative and a shallow classifier such as SVM trained in this feature space can separate different identities easily.

In our final system, 3 fully connected layers are used in the recognition network, whose input/output parameters are shown in Table 1. The network is learned though minimizing the triplet loss:

$$||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2, \quad (1)$$
$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T}$$

α is a margin that is enforced between positive and negative pairs, x_i^p, x_i^n, and x_i^a represent the positive, negative, and anchor sample, respectively. $\mathcal{T}$ is the set of all possible triplets in the training set.

layer	size-in	size-out	kernel	param
fc1	$512\times7\times7$	$1\times32\times128$	maxout $p=2$	103M
fc2	$1\times32\times128$	$1\times32\times128$	maxout $p=2$	17M
fc3	$1\times32\times128$	$1\times1\times128$		524K
L2	$1\times1\times128$	$1\times1\times128$		0

Table 1. Structure of the recognition network layers.

3.1. Training Strategy

The network described in Figure 1 is first pre-trained before the robot is deployed to any home. After being deployed to a specific home, the recognition network layers are fine-tuned using the newly acquired data from that home environment, while the detection network layers are fixed without change.

The pre-train process has two steps, which pre-train the detection and embedding networks to generate a unified network with shared convolutional features. In the first step, the faster R-CNN network is trained using the recently released WIDER dataset [8], following the approximate joint training process suggested in [7]. This network is initialized with an ImageNet-pre-trained model and fined-tuned over WIDER. The WIDER dataset has high degree of variability in scale, pose and occlusion in the training data, which is close to the actual environment our system operates in. In the second step, a pre-train recognition network is trained using the LFW [9] dataset, whose structure is shown in Figure 3. This pre-train recognition network follows the structure of the FaceNet model, and the last few layers (3 fully connected layers followed by an L2 normalization layer) are the same as the final unified network in Figure 1. In training, the convolutional layers of this network come from the pre-trained detection network model from step 1 and are fixed. The remaining layers are initialized with an ImageNet-pre-trained model and fine-tuned over LFW, using SGD with 0.5 dropout rate. The feature map of this pre-trained network has the dimension $512 \times 7 \times 7$ to match the recognition network layers in the final unified network. After these two steps, the final pre-trained unified detection and recognition network has the detection layers from the pre-trained detection network in step 1, and has the recognition layers from the pre-trained recognition network from step 2.

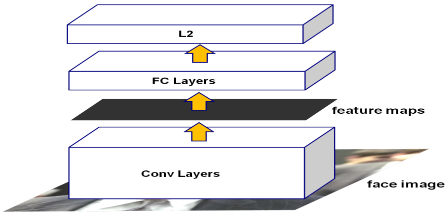


Fig. 3. The pre-train recognition network structure.

4. ROBOT SYSTEM

We develop a practical system for home service robots to automatically operate in deployed homes. As described in Figure 2, the system has two parts, on-robot processing and on-server processing. Our robot is a ClearPath Turtlebot 2 [22] equipped with a Laptop with Intel i7 CPU and GTX 980m GPU, as shown in Figure 4. The color inputs come from an Asus Xtion Pro Live RGBD sensor (VGA res). The face detection and recognition network is on the robot computing platform. In a regular thread, face detection is performed at 1 fps. Once faces are detected, the robot starts to detect faces at 3 fps and track faces at 30 fps, using the kernelized correlation filters (KCF) tracker [23]. The detected faces are fed forward through the recognition network to compute a 128-dim embedding feature for each face, which are transferred to the server for further classification. Compared with directly transferring the RGB sequence to the server and performing all computations on the server, the current system has much less transmission overhead, *i.e.*, 128 float numbers per face versus the original VGA images. Based on our observation, the transmission speed is one main source of response delays and can cause system instability sometimes.

On the server, each face is classified into various identities using its embedding feature. For consistently unrecognized face, an exception thread is initiated, where the robot asks the unknown person for verification. For family members or friends who know how to verify, a name will be acquired from the person to be associated with the face data. Then a new classifier is trained for the new person. The training data come from the detection thread at 3 fps, which continues until the face tracker is lost. If the verification failed, the unknown face is treated as potential stranger and alerts will be sent to the owner. For recognized faces, the new face data will be used as additional training data to update classifiers or fine-tune the recognition network layers on the server. In practice, the fine-tuning only happens when many training data (*e.g.*, 3000) have been newly acquired for many identities (*e.g.*, 8).

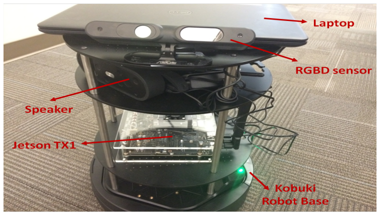


Fig. 4. The robot platform. The Jetson TX1 performs navigation tasks. The laptop performs detection/recognition tasks.

5. EXPERIMENTS

We first evaluate the pre-trained end-to-end detection and recognition network in Figure 1 before the system is deployed to any specific homes, and then evaluate the robot system in Figure 2 after deployment.

5.1. Evaluating the pre-trained network

The face detection network is pre-trained over the challenging WIDER dataset [8], which contains 159,424 training faces. The large variations in pose, scale, facial expression, occlusion, and background clutters are close to our practical problems. The trained model is then evaluated over the FDDB benchmark [10]. Extensive experiments have been conducted by Jiang and Learned-Miller in [17] on using the faster R-CNN structure as the face detector, which showed the comparable or even better performance compared with other state-of-the-art deep face detectors: Faceness [3] and cascade CNN [2]. Our experiments show similar findings, and due to space limits, we will not list the results here. One of our observations is that when the number of proposals drops from 500 to 100, the detection shows stable performance. Using as few as 100 proposals, 85.7% true positive rate (with 50 false positive in test set) can still be achieved. Therefore, we use 100 proposals in the final system as a tradeoff between accuracy and speed. Our un-optimized code runs roughly at 5 fps.

We evaluate the pre-train face recognition network described in Figure 3 over the LFW dataset [9]. Similar to experiments in [4], we use nine training splits to select an L_2-distance threshold, and perform a simple classification using the threshold on the tenth test split, *i.e.*, same or different identify when L_2 distance is smaller or larger than the threshold. Our experiments give a classification accuracy of $97.61\% \pm 0.23$ when using the fixed center crop.

5.2. Evaluating the deployed robot system

The robot system described in Figure 2 was deployed into an office setting for 1 week (5 days), where the network and classifiers were learned to identify 8 people in the office. When people were introduced to the robot for the first time, we asked that only one people showed up in front of the robot at a time. Each day, people who are in the office were asked to go to the robot for about 2-3 minutes. A total of 9,820 faces were collected in 5 days for training. For fine-tuning the recognition network, we augmented the original face images by applying random flip, random rotation and adding random noise. For evaluation, five 5-minute videos was recorded at different locations in the office where 10 identities in total were exposed to the robot system, including the 8 people in the training stage, 1 new person and 1 new face from a picture frame. We manually annotated the detected faces in the test video and computed the recognition mAP. Figure 5 shows the mAPs with different numbers of training data collected each day. Figure 6 gives some examples of the recognition result. As we can see, images in our application often have motion blur, with large variance in pose, scale, lighting, facial expression, and occlusion. Besides, due to its height, the robot tends to look up at people, which adds difficulty to recognition. In general, the proposed system can perform reasonably well despite such challenges. The system improved its performance each day and the final mAP reached 87.67% over the tested 10 identities after the 1 week deployment.

Figure 6 also shows some cases where the system missed or misclassified some faces in tough situations, e.g., with difficult or extreme pose, lighting, face size. Performance needs to be further improved for a practical system to operate well all the time. Certain engineering work can be done to improve the performance, such as by using better detection and tracking frameworks, or by combining the ego-motion information with detection and recognition to choose better views. We will explore such possibilities in our future work.

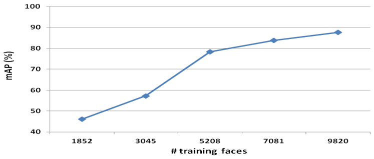
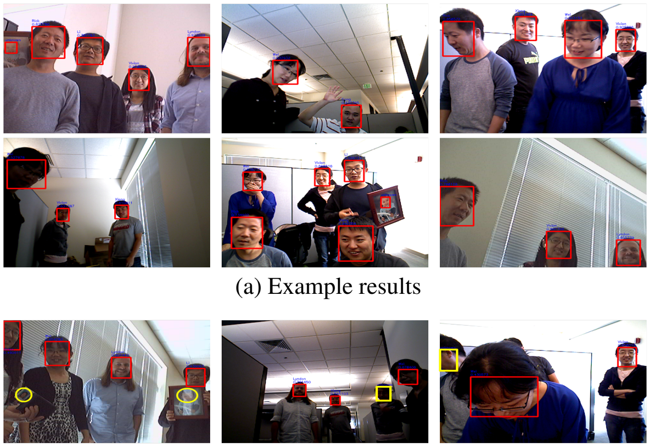


Fig. 5. mAP with different numbers of data obtained after each day.



(a) Example results

(b) Challenging cases

Fig. 6. Example results. Red boxes are correctly detected and labeled faces. Yellow circles are missed faces. Yellow boxes are misclassified faces.

6. CONCLUSION

We propose an end-to-end face detection and recognition framework based on deep CNN, by combining the RPN based detection and the deep embedding networks. Through sharing computation using the same convolutional layers between detection and recognition, the required computation is largely reduced at test time. An effective training strategy is used to generate a unified network. We also develop a practical robot system, which automatically asks for a minimum level of human supervision when needed, without requiring region-level face annotation. Experiments over challenging benchmarks as well as a personalized dataset demonstrate the effectiveness of our system.

7. REFERENCES

[1] C. Farabet, M. Saberian, and L. Li, "Multi-view face detection using deep convolutional neural networks," in *ICMR*, 2015, pp. 643–650.

[2] H. Li, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," *CVPR*, 2015.

[3] S. Yang, P. Luo, C. Loy, C, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *ICCV*, 2015.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CVPR*, 2015.

[5] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representation are sparse, selective, and robust," *CoRR*, 2014.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," *CVPR*, 2014.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *NIPS*, 2015.

[8] S. Yang, P. Luo, C.C. Loy, and X. Tang, "Wider face: A face detection benchmark," *CVPR*, 2015.

[9] E. Learned-Miller, G.B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," *Advances in Face Detection and Facial Image Analysis*, pp. 189–248, 2016.

[10] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[11] P. Viola and M. Jones, "Robust real-time face detection," *IJCV*, 2004.

[12] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," *ECCV*, 2014.

[13] M. Mathias, M. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," *ECCV*, 2014.

[14] X. Zhu and P. Dollar, "Face detection, pose estimation, and landmark localization in the wild," *CVPR*, 2012.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[16] M. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," in *TPAMI*, 2002.

[17] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," *Arxiv: https://arxiv.org/pdf/1606.03473*, 2016.

[18] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Recover canonical-view faces in the wild with deep neural networks," *CoRR*, 2014.

[19] R. Girshick, "Fast R-CNN," *ICCV*, 2015.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[21] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," *ECCV*, 2014.

[22] Turtlebot2, "Open source personal research robot," https://www.clearpathrobotics.com/turtlebot-2-open-source-robot/.

[23] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *CVPR*, 2014.