Chapter 4: Face Verification

4.1 Block Diagram

Figure 1: FRdiagram.jpg

General diagram for face verification as shown in figure 1. Input image 1 and image 2 will get from laptop integrated webcam or camera connected via USB port. Image 1 will be the picture of the person who need to be verified, Image 2 will be the picture of the ID card. In pre-processing block, we will using face detection algorithm to get two face from two image. Then align them into the size of input layer of the CNN. Extracting Deep Feature will compute feature of face image and return a feature vector for each image. By calculating norm distance between two vectors, we can evaluate if two face is from the same person.

4.2 Preprocessing

4.2.1 Face detection and Alignment Algorithm

4.2.1.1 Overall framework

The overall pipeline of our approach is shown in figure 2. Given an image, we initially resize it to different scales to build and image pyramid, which is the input of the following three-stage cascaded framework:

**Stage 1:** We exploit a fully convolutional network, called Proposal Network (P-Net), to obtain the candidate facial windows and their bounding box regression vectors. Then candidates are calibrated based on the estimated bounding box regression vectors. After that, we employ non-maximum suppression (NMS) to merge highly overlapped candidates.

**Stage 2:** All candidates are fed to another CNN, called Refine Network (R-Net), which further rejects a large number of false candidates, performs calibration with bounding box regression, and conducts NMS.

**Stage 3:** This stage is similar to the second stage, but in this stage, we aim to identify face regions with more supervision. In particular, the network will output five facial landmarks’ positions.

Figure 2: Overall.jpg

4.2.1.2 CNN architectures.

In [19], multiple CNNs have been designed for face detection. However, we notice its performance might be limited by the following facts:

(1) Some filters in convolution layers lack diversity that may limit their discriminative ability.

(2) Compared to other multi-class objection detection and classification tasks, face detection is a challenging binary classification task, so it may need less numbers of filters per layer.

To this end, we reduce the number of filters and change the 5×5 filter to 3×3 filter to reduce the computing while increase the depth to get better performance. With these improvements, compared to the previous architecture in [19], we can get better performance with less runtime (the results in training phase are shown in Table I. For fair comparison, we use the same training and validation data in each group). Our CNN architectures are shown in Fig. 3. We apply PReLU [30] as nonlinearity activation function after the convolution and fully connection layers (except output layers).

Table 1: Table1FD.jpg

Figure 3: CNNStruct.jpg

4.2.2 ID number detection

In this part, we will consider the given image with ID number on the bottom left of the image (as Fig 4). The following step is showing how to segmenting text from an unstructured scene which will help to improve optical character recognition (OCR).

Fig 4: IDcard.jpg

Step 1: Crop the image keeping the bottom left which contain the ID number.

Fig 5: Cropped Image with ID number on the bottom left quaterID.jpg

Step 2: **Detect candidate text regions using MSER** function to find all the regions within the image and plot these results. Notice that there are many non-text regions detected alongside the text.

Fig 6: Detect candidate text regions. MSERregion.jpg

Step 3: **Remove non-text Regions based on basic geometric properties**

There are several geometric properties that are good for discriminating between text and non-text regions, including:

* Aspect ratio
* Eccentricity
* Euler number
* Extent
* Solidity

Use *regionprops* to measure a few of these properties and then remove regions based on their property values.

Fig 7: After removing non-text Regions rmNontext.jpg

Step 4: **Merge text regions for final detection result**

At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string 'EXIT' vs. the set of individual characters {'X','E','T','I'}, where the meaning of the word is lost without the correct ordering.

One approach for merging individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around these regions. To find neighboring regions, expand the bounding boxes computed earlier with regionprops. This makes the bounding boxes of neighboring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes.

Fig 8: Merge text regions for final detection area expandbox.jpg

Step 5: **Recognize detection text using OCR**

After detecting the text regions, use the ocr function to recognize the text within each bounding box. Note that without first finding the text regions, the output of the ocr function would be considerably more noisy.

Fig 9: Recognition result detectedtext.jpg

4.3 Extracting Deep Feature and Compute distance

Facenet is based on learning a Euclidean embedding per image using a deep convolutional network. The network is trained such that the squared L2 distances in the embedding space directly correspond to face similarity: faces of the same person have small distances and faces of distinct people have large distances.

Once this embedding has been produced, then the aforementioned tasks become straight-forward: face verification simply involves thresholding the distance between the two embeddings; recognition becomes a k-NN classification problem; and clustering can be achieved using off-theshelf techniques such as k-means or agglomerative clustering.

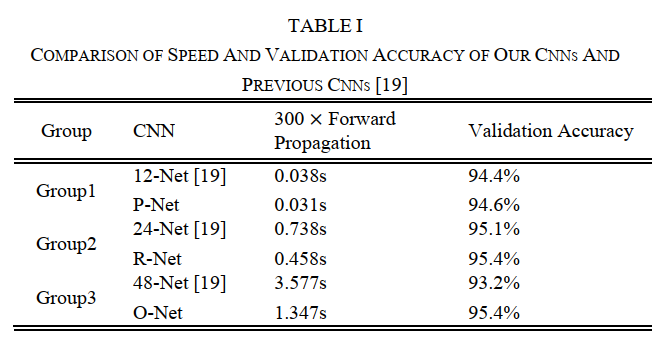
Previous face recognition approaches based on deep networks use a classification layer [15, 17] trained over a set of known face identities and then take an intermediate bottleneck layer as a representation used to generalize recognition beyond the set of identities used in training. The downsides of this approach are its indirectness and its inefficiency: one has to hope that the bottleneck representation generalizes well to new faces; and by using a bottleneck layer the representation size per face is usually very large (1000s of dimensions). Some recent work [15] has reduced this dimensionality using PCA, but this is a linear transformation that can be easily learnt in one layer of the network.

In contrast to these approaches, FaceNet directly trains its output to be a compact 128-D embedding using a tripletbased loss function based on LMNN [19]. Our triplets consist of two matching face thumbnails and a non-matching face thumbnail and the loss aims to separate the positive pair from the negative by a distance margin. The thumbnails are tight crops of the face area, no 2D or 3D alignment, other than scale and translation is performed.

Choosing which triplets to use turns out to be very important for achieving good performance and, inspired by curriculum learning [1], we present a novel online negative exemplar mining strategy which ensures consistently increasing difficulty of triplets as the network trains. To improve clustering accuracy, we also explore hard-positive mining techniques

|  |  |  |
| --- | --- | --- |
| Sample Size | ID detection accuracy | Face Verification accuracy |
| 10 |  |  |
| 20 |  |  |
| 30 |  |  |
| 40 |  |  |
| 50 |  |  |
| 60 |  |  |

which encourage spherical clusters for the embeddings of a single person. As an illustration of the incredible variability that our method can handle see Figure 1. Shown are image pairs from PIE [13] that previously were considered to be very difficult for face verification systems.



[19] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural  
network cascade for face detection,” in IEEE Conference on Computer  
Vision and Pattern Recognition, 2015, pp. 5325-5334

[30] K. He, X. Zhang, S. Ren, J. Sun, “Delving deep into rectifiers: Surpassing  
human-level performance on imagenet classification,” in IEEE International Conference on Computer Vision, 2015, pp. 1026-1034.