Rotation Face Detection Based on Three-Window Convolutional Neural Networks

Mingyan Gao¹, * and Yan Tang¹

¹School of Computer and Information Science, Southwest University, Chongqing; 400715, China

*mycoolbb@gmail.com;

Abstract. In some family photos or special scene pictures, we can find some rotated faces. Most existing methods are based on increasing the features of rotated faces or changing the directions of pictures to augment the training data. However, these methods have their own limitations and can not detect rotated faces accurately. We propose a method based on three-window convolutional neural networks designed by ourselves. We extract the features of faces and change the feature matrices of faces by clockwise rotation and anticlockwise rotation through three-window convolutional layer in order to increase the face features. We retain parameters of the model after training, and convolutional layer replace fully connected layer. According to the heatmap of the sample, our method can predict face region. We carry out the experiments on FDDB and LFW datasets. The experiments on LFW show that our model achieves AUC of 0.9240, and recall of 0.9367 and the experiments on FDDB show that our model achieves recall of 0.9541.

1. Introduction

Increasingly, deep learning is widely used for face detection methods with the development of computer hardware equipment such as GPU and the performance of deep learning face detection methods is sensational. These methods always detect upright faces in the pictures. However, we discover that some rotated faces in the photos such as whole-family photos in our daily life. The focus of this paper is that we propose a method to detect the rotated faces.

In recently, there are three main methods for dealing with the rotated faces, details as follows:

1. New Haar-like features [1] detection belongs to traditional machine learning method. It adds some new detection features on the basis of Haar-like feature, and it offers the feature calculation method and integration method. The features extracted by new Haar-like detection is shown in Figure 1. However, some newly detection features leads to detection error, and we discover that some non-faces have been mistaken for the detection target.

2. Deep Dense Face Detector (DDFD) [2] is a face detector based on the deep convolutional neural networks. It is able to detect a range of orientations faces through a single model which is trained by fine-tuning AlexNet, and yet this method need very large number of training data in order to improve the accuracy of face detection. As a result, it takes a lot of time for data augmentation.

3. Multi-task Cascaded Convolutional Networks (MTCNN) [3] is based on the cascaded convolutional networks. This face detector leverages a cascaded architecture with three stages of designed deep convolutional networks to predict face and landmark location in a coarse-to-fine manner. This framework can adjustment face orientation through facial landmarks localization firstly and get the
bounding box of upright faces, nevertheless, the requirement of face annotations leads to the complexity of the training and testing process and the adjusted pictures can cause severe distortion of the images.

Figure 1. The feature matrix of Haar-like method

Figure 2. Traditional convolutional layer

In this paper, we propose a method called three-window convolutional neural networks based on deep learning. The three-window convolutional layer can rotate facial feature matrices to boost facial feature extraction, as a result, our face detector is able to detect faces from different angles. We test our method on FDDB and LFW datasets, and the results demonstrate that it is an accurate efficient method.

2. Related Work

2.1. Convolutional layer

The convolutional layer [4] is shown in Figure 2. Convolutional layer is able to extract image features, formulated as below:

$$y^{j(r)} = \max(0, b^{j(r)} + \sum_{i} k^{ij(r)} \ast x^{i(r)})$$

(1)

Where $x^i$ is the $i$-th input layer, and $y^j$ is the $j$-th output layer. $k^{ij}$ is the convolutional kernel in the middle of them. $b^j$ is the $j$-th output layer bias. $r$ in Equation 1 means the weights in this region can be shared. $\ast$ denotes convolution as below:

$$(f \ast g) = \sum_{h=0}^{\infty} \sum_{k=0}^{\infty} f(h,k)g(h,k)$$

(2)

It indicates that the corresponding elements in the two matrices are multiplied and accumulated, and we select downsampling operation with max-pooling layer afterwards. Max-pooling operation as follows:

$$y^{j}_{i,k} = \max_{0 \leq m,n \leq s} \{x^{j}_{i-s+m,k-s+n}\} \hspace{1cm} y^{j'}_{i,k} = \max_{0 \leq m,n \leq s} \{x^{j'}_{i-s+m,k-s+n}\}$$

(3)

Where $y^j$ means the $i$-th output map pools over an $s \times s$ non-overlapping local region in the $i$-th input map $x^j$.

Convolutional layer has the advantage of automatically learning facial features, and the deeper the convolutional networks are, the more features it will obtain. We can regard the convolutional operation as feature extraction process.

2.2. Softmax layer and loss function

According to [2], we design our model to a binary model, at last, we choose softmax layer to compute the probability of face and non-face. Softmax function is formulated as follows:
\[ \sigma_i(z) = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}} \]  
(4)

Where \( i \) is label and \( i \subseteq \{0, 1\} \), and 0, 1 indicate face and non-face respectively. \( j \) also means label and \( j \subseteq \{0, 1\} \). \( z \) derived from the previous layer. \( \sigma(z) \) represents the probability of the sample belonging to class \( i \). We define the loss function as follows:

\[ \hat{I}(y, z) = -\log\left(\frac{e^{z_y}}{\sum_{j=1}^{m} e^{z_j}}\right) = \log\left(\sum_{j=1}^{m} e^{z_j}\right) - z_y \]  
(5)

Where \( y \) denotes the true label of the sample. We minimize the value of loss function on the training data by adjusting parameters.

2.3. Fully convolutional networks

We refer to the paper [5], and we retain the parameters in the model, and we use the convolutional layers to replace the last few full connection layers to construct the fully convolutional networks. The sizes of faces in the images are different, but no matter what sizes of the faces can be received by the fully convolutional networks, and the sliding window in the convolutional layer can detect target on the matrix to achieve pixel-level classification. Above all, it is more efficiency than the traditional detection method.

Feature extraction is carried out by the fully convolutional networks through all convolutional layers as well as the Maxpooling layer. The data size is getting smaller and smaller, and the final result is an \( n \times n \) matrix, and \( n \) is determined by the input image size. This matrix is called the heatmap and we can determine the target region on the basis of the heatmap.

2.4. Non-maximum suppression (NMS) algorithm

In the face detection process, there are many regions where the confidence value exceeds the setting threshold determine the target region. In order to avoid the overlapping of bounding boxes, referring to [6], we use NMS algorithm to screen the candidate boxes. At first, we sort all bounding boxes reaching the setting threshold from high score to low score and we select the box whose score is highest. Secondly, we traverse the remaining bounding boxes, and the box will be deleted if its Intersection-over-Union (IOU) with the highest score bounding box is greater than the threshold value. Otherwise keep it, because it could be another face. At last, we continue to select a highest score in rest of the bounding boxes and deal with it based on the above process.

3. Three-window convolutional neural networks method

Three-window convolutional neural networks improve based on the traditional convolutional neural networks. We add new layer designed by ourselves. Three-window means three kinds of facial features matrices. The first one is extracted through the convolutional layer and keeping its size and direction. The second one is produced by the convolutional layer and clockwise rotation 90\(^\circ\). The third one obtained by convolutional layer and rotate it by 90\(^\circ\) counterclockwise.

As for the tilted faces in the images, it is not difficult to find that no matter what the direction the face in the image is, as long as we rotate the image, we can find the face that is close to the upright face and that is easy to detect. We can see the Figure 3.
Figure 3. Three different directions of pictures. These are the same pictures, but the directions are different. We can find upright face in any of them.

Three-window convolutional networks can increase sample features. The upright face feature matrix, clockwise rotation $90^\circ$ face feature matrix as well as counterclockwise rotation $90^\circ$ face feature matrix are all the facial features, then there is no need to expand data sets or data augmentation, and the trained model will consider that three directions images are faces. In the testing process, the networks load a face image, regardless of its tilted angle, there will always be a matrix among the extracted three matrices close to the upright face.

3.1. Overall Framework

Figure 4. The framework of three-window convolutional networks.

The three-window convolutional networks framework is shown in Figure 4. The networks load data, and the three-window convolutional layer extracts features to obtain three matrices. The Eltwise layer accumulates the elements at the corresponding positions of the three matrices to generate a new matrix. After a series of convolution and Maxpooling down-sampling operations, a new matrix with smaller size is generated.

Two operations are performed on the new matrix. One is to flatten the matrix into a one-dimensional vector and connect it to the fully connected layer. The other is convolution and Maxpooling down-sampling.

The corresponding elements in the result of the two operations are added by Eltwise layer, where the dimensions of the vectors obtained by the two operations must be the same. The combined vector is connected to the fully connected layer. Softmax layer was used to calculate the obtained results and output the result of the networks for sample. $Y_0$ represents the probability value of judging the sample is label 0 (assuming that the label 0 represents the positive sample), $Y_1$ represents the probability value of judging the sample is label 1, where $Y_0 + Y_1 = 1$. Bringing $Y_0$ and $Y_1$ into the loss function mentioned in section 2.

The whole networks keep adjusting parameters to minimize the loss function value, so that the judgment on samples is closer to the true label.
3.2. Three-window convolutional layer

Three-window convolutional layer in networks architecture shown in Figure 5, firstly the input data use convolutional layer for feature extraction, after that then the feature pass to the Convolution-L layer and Convolution-R layer, where Convolution-L is used to rotate the extracted feature matrix $90^\circ$ counterclockwise, whose formula is as follows:

$$y_{h-j,i}^\beta = y_{i,j}^\alpha$$  \hspace{1cm} (6)

Where $y_{(i,j)}^\alpha$ means the pre-rotation matrix, $y_{(h-j,i)}^\beta$ is the post-rotation matrix, $i, j$ are the row number and column number of the matrix, respectively, and $h$ denotes the number of rows of the matrix, while Convolution-R layer is used to rotate matrix $90^\circ$ clockwise, next the generated three matrices of face image features are merged by Eltwise layer, then according to the Figure 4 process forward pass. Such matrix fusion provide three directions of facial features matrices for the entire networks. We output the features extracted from Convolution-L layer and Convolution-R layer, as shown in Figure 6.

![Figure 5. Three-window convolutional layer](image)

3.3. Training and testing

The training dataset we used is AFLW [7]. AFLW dataset is a face dataset containing multiple poses and perspectives, which is very suitable for the research of face detection. This dataset is extremely large, most of which are color pictures and a few are grayscale pictures. The maximum number of training iterations is 60,000, and the learning rate was 0.001.

In accordance with the second section, we retain the trained model parameters and replace the last fully connected layers with convolutional layers. After inputting a face image, the detection result is shown in Figure 7, and we output the heatmap, the heatmap is shown in Figure 7. The region in the picture has higher probability when it is closer to the human face, the color of this region is darker, and the red region is the target with the highest probability. We can also get good detection results when we input face images from different angles, as shown in Figure 8.
We are surprised to find that the detector can also predict landmarks of human face which is reflected by heatmap. It is shown in Figure 9. The contour of eyes and mouth can be seen from the heatmap, which provides a new idea for our subsequent research on the detection of face landmarks.

![Figure 8](image1.png) *Figure 8. Rotate the original image by -90°, 90°. And the detection window can be cropped from the original image.*

![Figure 9](image2.png) *Figure 9. Three-window convolutional networks not only can detect face region, but also can display face landmarks through heatmap.*

4. Experiment

4.1. Datasets

LFW(Labeled Face in the Wild) [8] is a commonly used dataset in the field of face detection, containing 13,233 pictures of 5,749 persons. We rotated half of the faces in the dataset from different angles and randomly cut background images as negative samples.

FDDB(Face Detection Dataset and Benchmark) [9] unconstrained natural scene face detection dataset contains 2,845 images taken from different scenes and different angles, with a total of 5,171 faces.

4.2. Evaluation

The experimental environment is Ubuntu16.04 system, i7-7700 CPU, 4.20GHz, 32G memory, GTX1080 graphics card and 8G graphics memory, and deep learning networks framework is Caffe. We compared Haar-like [1], VJ [10], MTCNN [3], DDFD [2], Cascade-CNN [11] algorithms that have been proposed in recent years.

The evaluations of the algorithm in this paper are AUC and Recall. The calculation method is as follows:

\[ S(d_i, l_j) = \frac{\text{area}(d_i) \cap \text{area}(l_j)}{\text{area}(d_i) \cup \text{area}(l_j)} \]  

(7)

Where \( \text{area}(d_i) \) represents the face region detected by the model, and \( \text{area}(l_j) \) represents the true face region. \( S(d_i, l_j) \geq 0.5 \) means correct detection, and \( S(d_i, l_j) < 0.5 \) means false detection.

With different thresholds, we can obtain different pairs of TPR and FP. ROC curve was drawn according to TPR and FP values, and AUC value means the area under ROC curve. TPR represents the proportion of positive samples detected in all positive samples, FP is the quantities of false positives and Recall represents the recall rate. The calculation formula of Recall is shown as below:

\[ \text{Recall} = \frac{TP}{TP + FN} \]  

(8)

Where TP means the actual positive sample, and the detection result is true; FN represents the actual positive sample, and the detection result is false. FP means that the actual sample is negative and the detection result is true.

4.3. Results

Firstly, our experiment is carried out on LFW dataset. We rotated half of the images in the dataset. Figure 10 shows the comparison between our detection method and other detection methods. It can be
clearly seen that the area under the curve of the three-window convolutional detection model is larger than that of other detection methods, which proves that our method is superior to other detectors.

![Figure 10. ROC curves of face detectors on LFW dataset.](image1)

![Figure 11. ROC curves of face detectors on FDDB dataset.](image2)

Table 1 shows the specific experimental results. The AUC of our detection method reaches 0.9240, higher than other detector values, and the recall value reaches 0.9367, which is the highest. Table 1 shows the specific experimental results. The AUC of our detection method reaches 0.9240, higher than other detector values, and the recall value reaches 0.9367, which is the highest.

Table 1 shows the experimental results of our detector compared with other detectors on the FDDB dataset. We do not rotate the images in FDDB dataset, and the result shows that the area under the curve of our detector was second only to that of the MTCNN detector, and the detection effect is better than most detectors, slightly lower than that of the MTCNN detector.

We calculate the Recall value of each model, and the experimental result is shown in Table 2. Our detector Recall value reaches 0.9541, which is the highest among all detectors. The highest result of Recall shows that our detector can accurately detect faces in images regardless of the angle.

| Method       | AUC  | Recall |
|--------------|------|--------|
| Our method   | 0.9240 | 0.9367 |
| MTCNN        | 0.8744 | 0.9178 |
| DDFD         | 0.8765 | 0.9098 |
| Haar-like    | 0.8228 | 0.8970 |
| CascadeCNN   | 0.8507 | 0.9001 |
| VJ           | 0.7161 | 0.8123 |

| Method       | Recall |
|--------------|--------|
| Our method   | 0.9541 |
| MTCNN        | 0.9237 |
| DDFD         | 0.9179 |
| Haar-like    | 0.7460 |
| CascadeCNN   | 0.9480 |
| VJ           | 0.8410 |

5. Conclusion
We propose a detection method for rotated faces based on three-window convolutional neural networks, which can accurately detect rotated faces in images. This method adds a three-window convolutional layer to the traditional convolutional neural networks, rotates the face feature matrix clockwise and counterclockwise, and combines the face feature matrix of the upright face for model training. In this paper, we compare the three-window convolutional networks model with MTCNN, Haar-like, VJ, DDFD and CascadeCNN detection models on two data sets. The experimental result shows that the three-window convolutional networks model is superior to other models in the detection of rotated faces. Moreover, this model is a bi-classification model with simple training and there is no need to introduce complex regression calculation and other annotation.

In addition, the efficiency of face detection in images still needs to be improved. The existing methods, whether fully convolutional networks or sliding window, have their limitations. How to use
the accurate classification model and detect faces in the image more efficiently is the research focus of the next work.

Acknowledgements
Thanks to Yan Tang for her guidance in the experiment process, and thanks to the reviewers for their help and discussion on this work.

Reference
[1] Xiaoke, L. , Xiehua, S. , & Yongxia, Z. . (2009). Multi-angle face detection based on new haar-like feature. Computer Engineering, 35(19), 195-197.
[2] Farfade, S. S. , Saberian, M. , & Li, L. J. . (2015). Multi-view face detection using deep convolutional neural networks.
[3] Zhang, K. , Zhang, Z. , Li, Z. , & Qiao, Y. . (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503.
[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
[5] Long, J. , Shelhamer, E. , & Darrell, T. . (2014). Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(4), 640-651.
[6] Neubeck, A., & Gool, L. V. (2006). Efficient Non-Maximum Suppression. International Conference on Pattern Recognition.
[7] Martin Köstinger, Wohlhart, P. , Roth, P. M. , & Bischof, H. . (2011). Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization. IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011. IEEE.
[8] Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008, October). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In Workshop on faces in Real-Life Images: detection, alignment, and recognition.
[9] Jain, V., & Learned-Miller, E. (2010). Fddb: A benchmark for face detection in unconstrained settings (Vol. 88). Technical Report UM-CS-2010-009, University of Massachusetts, Amherst.
[10] Viola, P. , & Jones, M. J. . (2004). Robust real-time face detection. International Journal of Computer Vision, 57(2), 137-154.
[11] Li, H. , Lin, Z. , Shen, X. , Brandt, J. , & Hua, G. . (2015). A convolutional neural network cascade for face detection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society.