Eye Detection with Faster R-CNN

Jiali Cui[1], Fuqiang Chen[1], Duo Shi[1], Liqiang Liu[1]
[1]*North China University of Technology, Beijing, China*

Keywords: Iris recognition, Eye detection, Deep convolutional neural networks, Faster R-CNN.

Abstract: The accuracy of eye detection is crucial to a variety of biometric identification technologies, such as iris recognition. The challenge of eye detection comes from improving accuracy of detection in the case of occlusion or reflection of glasses. In this paper, an eye detection method based on Faster Region-based Convolutional Neural Network (Faster R-CNN) is proposed. The method includes three import parts: convolutional layers, region proposal network (RPN) and detection network. By training monocular and binocular models on the training dataset, the recall of monocular and binocular models on test dataset can reach 96% and 95% respectively, which proves that the proposed method based on Faster R-CNN has high accuracy of detection in the task.

1 INTRODUCTION

Human eyes are important sensory organs and of rich features of faces. With the development of biometric recognition, eye detection is becoming an important technology in computer vision. Eye detection is a technology that locates eyes in images through abstract features, and has been an essential step in some pattern recognition technologies, such as iris recognition and fatigue detection system (Fei Yang, 2013). In iris recognition system (Wildes, 2002), locating the position of eyes is the first step of the whole system, and then pupil and iris can be segmented. Meanwhile, the size of iris can also been obtained by eye detection. In fatigue detection system, it should similarly locate the position of eyes, and then establish eyes model and count the number of blinks. The accuracy of eye detection directly affects the final accuracy of iris recognition and other biometric applications. In face detection systems, eye detection is optional, the accuracy of face detection can be improved while eye detection is adopted.

Due to the applications discussed above, many algorithms of eye detection have been proposed by studying the feature of eyes. The procedures of algorithms contain two steps: model is firstly established by manually selecting the feature of eyes, and then the model is used to classify and locate eyes. classifiers. With the development of neural network, convolutional neural network (CNN) based object detection can automatically design features and In the early 1990s, (Yuille, 1992) proposed a method based on the eyes geometry to detect eyes. In 2005, (Hamouz, 2005) used appearance features to locate eyes. In 2011, (Fei Yang, 2011) proposed a human eye localization algorithm based on texture features. In recent years, some systems locate eyes by facial landmarks, which can also report a good performance. However, algorithms based on geometry, appearance and texture features have terrible performance if the picture is in the case of occlusion or reflection of glasses. When facial landmarks are used to locate eyes, there might be partial face in the task which would fail to detect faces. Moreover, distinguishing left eyes from right eyes was not referred in their work.

In this paper, an eye detection method based on Faster R-CNN is proposed, which locates eyes with high precision, and distinguishes left eyes from right eyes in images which include even part of faces. All the detection results will improve the accuracy and speed of iris segmentation and feature matching.

2 OVERVIEW OF OBJECT DETECTION

Object detection is a major research interest in computer vision. The traditional framework of object detection is manually designing features and classifiers by learning from big data and enable end-to-end train.

Cui, J., Chen, F., Shi, D. and Liu, L.
Eye Detection with Faster R-CNN.
DOI: 10.5220/0008096201110116
In *Proceedings of the International Conference on Advances in Computer Technology, Information Science and Communications (CTISC 2019)*, pages 111-116
ISBN: 978-989-758-357-5

2.1 Traditional methods

In the field of traditional computer vision, object detection algorithms usually contain three parts: choice of detection windows, feature selection and classifier design. Algorithms of choosing detection windows developed form sliding window based on scales to Selective Search or Edge Box which is more efficient to create region proposals because of multi features. As for selecting features, there are many classical methods, such as Local Binary Patterns (LBP), Histogram of Oriented Gradient (HOG) and so on. Support Vector Machine (SVM) or Decision Tree is usually used as classifier in many object detection systems. In general, traditional object detection systems combine the three important parts and perform well in some constrained scene. In this paper, methods based on Faster R-CNN are used as main procedure.

2.2 Evolution of Region-based CNNs

With the development of computing power, deep convolutional neural networks have dominated many tasks of computer vision. Deep learning can extract abstract features by extensive data and repetitively training, which can better express the key information. The region-based convolution neural network has well performances, so it has become the main algorithm in the field of object detection.

In 2014, (Girshick, 2014) first proposed a region-based CNN(R-CNN) for object detection. Compared to tradition algorithms in object detection, it has high accuracy of locating and classifying object. However, it needs many feature extractors and SVM classifiers. The training time is long. To mitigate these problems, two methods, the SPP-Net (Kaiming, 2014) and the Fast R-CNN (Girshick, 2015) have been proposed. Instead of feeding each warped proposal image region to the CNN, the SPP-Net and the Fast R-CNN run through the CNN exactly once for the entire input image. After mapping the proposals to the feature maps of last convolutional layer, each proposal can get scores of classes and coordinates in detection layers. All of R-CNN, SPP-Net and Fast R-CNN rely on the input generic object proposals, which come from selective search (Uijlings, 2013). But it is computationally intensive. To reduce the computational burden of proposal generation, the team of Ren proposed Faster R-CNN (Ren, 2015).

After Faster R-CNN is proposed, it is used in many object detection applications. For example,

(Huaizu Jiang, 2016) proposed face detection method based on Faster R-CNN. They reported state-of-the-art results on two widely used face detection benchmarks, FDDB and the recently released IJB-A.

3 PROPOSED METHOD

Faster R-CNN has become mainstream method in many object detection applications. Faster R-CNN abandons the traditional selective search algorithm to extract proposed regions, and proposes Regional Proposal Network (RPN), which is a fully convolutional network. Faster R-CNN can be deemed as the combination of Fast R-CNN and RPN. Therefore, one network can extract region proposals and locate objects. This network can run faster than R-CNN and Fast R-CNN and can reach 8-9 frames per second in professional GPU, such as NVIDIA TITAN. The structure of Faster R-CNN is shown in Figure 1.

This network contains the following three important parts:

(1) Convolutional layers

At present, there are many CNN models, including ZFNet (Zeiler, 2013), VGGNet (Simonyan, 2014), ResNet (Kaiming, 2016) and so on. In this experiment, VGG16 and ResNet are selected as convolution layers. Finally, comparison in performance of two networks are made. The convolution layer finally outputs feature maps with a size of 1024 *51 * 28. By observing different channels, different features are shown in Figure.2

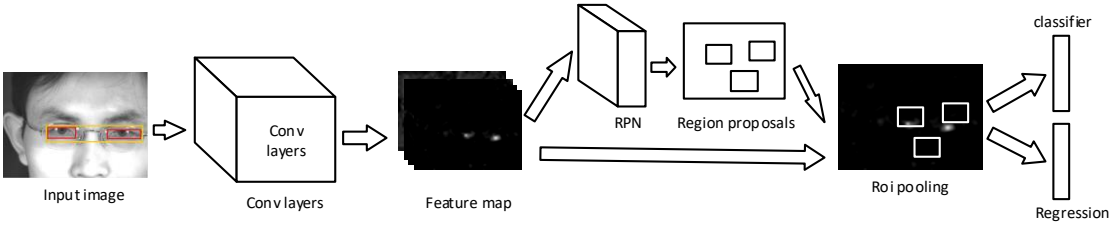


Figure 1: Structure of Faster R-CNN.

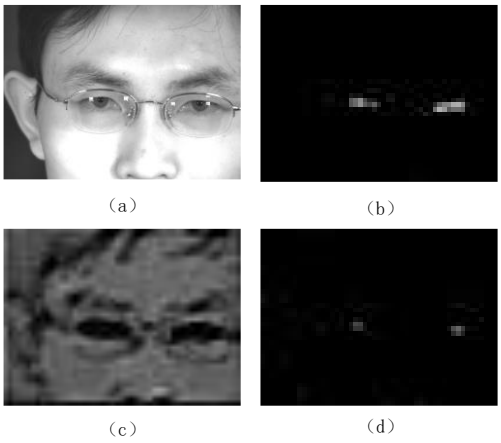


Figure 2: Feature maps of different channels (a) Original (b) Eye area (c) Face area (d) Iris area.

(2) RPN

This part is to extract the region proposals. The main process is to map the above 1024-dimensional features to low-dimensional space (512), which is followed by two full connection layers: one is responsible for classification and the other is responsible for regression. In the training stage, anchors are defined as candidate areas that may appear in the original image. In this experiment, the RPN is trained by 256 generated anchors, including 128 positive samples and 128 negative samples. The intersection over unit (IOU) of positive samples is more than 0.7. If the IOU is less than 0.3, those anchors are defined as negative samples.

(3) ROI pooling

This part has two inputs: one is the proposal regions generated by RPN, and the other is the feature maps from convolution layer. The main role is to regress to the coordinate of eye area and class category. ROI Polling is a special version of the space pyramid pool proposed by SPP-Net. It does not use different scales for the feature maps. In this experiment, the convolution features of the ROI region are divided into 7 * 7 meshes, and max pooling in each mesh is made. Finally, a fixed length (49) vector can be got, regardless of the size of the ROI region.

4 EXPERIMENTS AND RESULTS

In this section, the process of experiment based on Faster R-CNN is detailedly introduced. The overall framework of the experiment is shown in Figure.3. In the stage of training, collected infrared pictures are used as training dataset for the purpose of iris recognition. Deep model is trained through the Faster R-CNN. In the stage of test, images from test dataset are sent into trained model, and the outputs are coordinates and score of class.

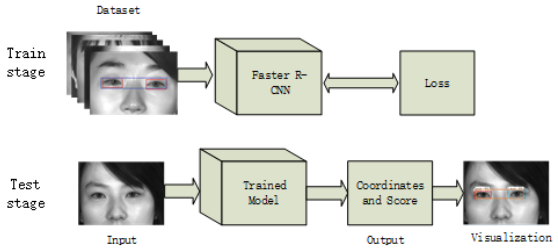


Figure3: System overview.

4.1 Training and Testing

In the experiment, the 1500 infrared images which include varying degrees of occlusion or reflection of glasses are selected in an open dataset (CASIA Iris Image Database Version 4.0) for the purpose of iris recognition. Sample images are shown in the first row of Figure 4. These images are divided into two datasets, namely training and test datasets. 1000 images are labeled as training dataset and the other 500 pictures are used as test dataset. Some original images in training dataset are shown in Figure 4 (a) and the corresponding labels are shown in Figure 4(b) and (c), where black bounding boxes are ground-truth annotations. In this experiment, monocular model and binocular model are studied. Because binocular model has symmetry, it may have more well-marked features in eye detection. However, the premise of detecting binocular model is to detect the monocular model. In order to study the differences of monocular and binocular model in eye detection, monocular model and binocular model are compared in this experiment. There are three labels: ‘left eye’, ‘right eye’ and ‘eyes’.

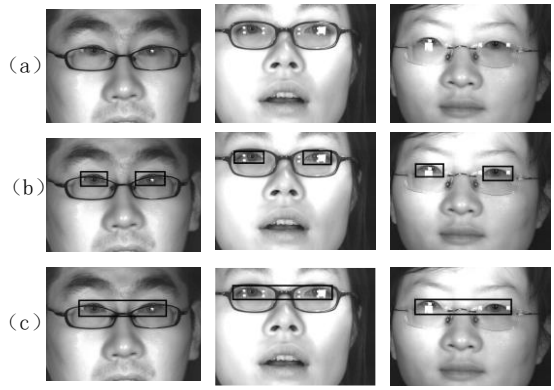


Figure 4: Partial data: (a)original images (b)monocular areas (c) binocular areas.

The size of original images is relatively large (2400 * 1700). In these images, the average size of the eyes area is 1500 * 230. Because the anchor in the network is responsible for adapting to the scale, even if the largest anchor (1024 * 512) can't completely cover the ground truth. To adapt the size of anchor, images in training dataset are normalized: the smallest side of the picture is scaled to 600, and the other side does the same scale, so the anchor can cover all possible cases; in addition, because the number of images in dataset is small, the images are augmented by small angles and translations.

The training process consists of the following two steps:

(1) After training the RPN model based on a pre-trained ImageNet model, detection network is trained by proposed regions generated by RPN.

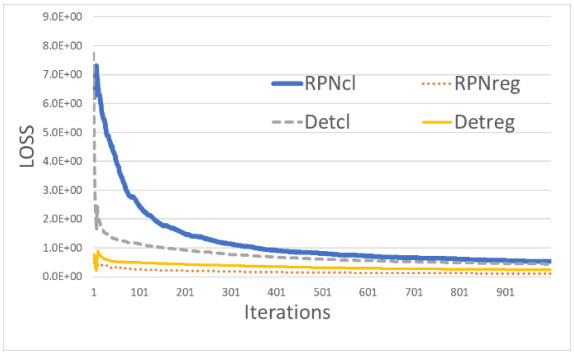
(2) The convolution layer is initialized with the first stage trained weights instead of pre-trained ImageNet model, and then the RPN is trained again. Next, the proposed regions generated by RPN are used to train the entire detection network. The process of convolution layer achieves the weight of sharing.

The training code is run on a personal computer with an Intel CPU I5-4590 of 3.30GHz and an NVIDIA GTX 650 Ti GPU with 2GB memory. During training stage, loss values of RPN and detection net are recorded. The detailed work is shown in chapter of 4.2.

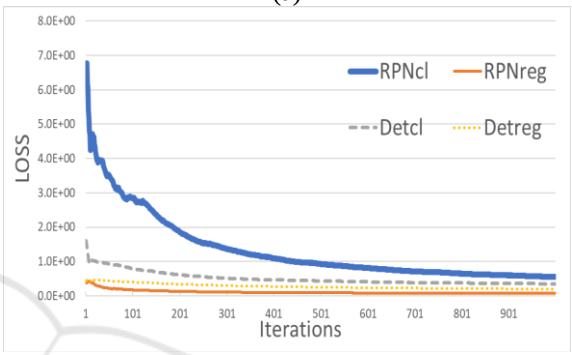
After training, the trained model is used to detect eyes on the test dataset. During test stage, the accuracy and speed of different models are recorded.

4.2 Comparison and Analysis

In this experiment, three networks based on CNN, which include VGG-based Faster R-CNN, ResNet-based Faster R-CNN and You Only Look Once (YOLO) (Redmon, 2015), are trained on the training dataset. Firstly, the convergence speed and test speed based on VGG16 and ResNet are compared, while the accuracy of monocular model and binocular model are also compared. In the training stage, the loss function convergence of ResNet-based Faster R-CNN is shown in Figure 5(a), while the loss function convergence of VGG-based Faster R-CNN is shown in Figure 5(b).



(a)



(b)

Figure 5: Downward trend of loss function (a)ResNet (b)VGG16.

Figure 5 contain four kinds of loss function of the decline: 'RPNcl' denotes the loss function of the RPN network classification; 'RPNreg' denotes the RPN network regression loss function; 'Detcl' denotes the classification loss function of the whole detection network; 'Detreg' denotes the whole network regression loss function.

In order to compare equally the convergence rate, the total convergence speed of the three networks are compared in the 1000 iterations. The result is shown in Figure 6.

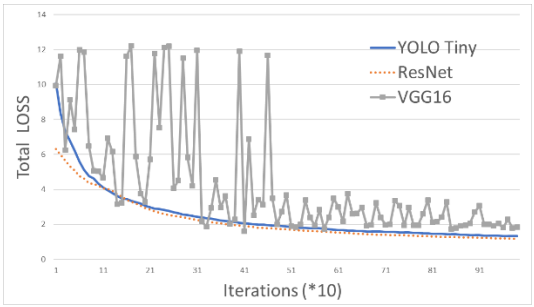


Figure 6: The convergence rate of loss function.

From Figure.5 and Figure.6, we can see that in the same iterations, the convergence speed of ResNet is

faster than others. Finally, ResNet-based Faster R-CNN and VGG16-based Faster R-CNN are trained 10k iterations and YOLO is trained 50k iterations. The loss of final results is that ResNet-based Faster R-CNN get a smallest loss. After training, the performance of those networks is compared on the test dataset, which is evaluated by the term of recall and speed. The recall is defined as in (1).

$$Recall = TP/(TP+FN) \quad (1)$$

TP represents the number of correct results, and FN represents the number of instances that should be correct but not be predicted. The speed is defined as the average speed of processing each picture. The comparison of these performance is shown in Table 1

Table 1: Comparison of test time and recall.

Network	Speed	Recall	
		Monocular	Binocular
ResNet	0.61s	96.3%	95.3%
VGG16	0.94s	94.8%	92.4%
YOLO(Tiny)	0.12s	47.4%	63.5%

From the comparison, the ResNet has better performance on extracting features than VGG16. ResNet-based Faster R-CNN has a faster convergence rate and has higher accuracy. YOLO has good performance on ImageNet, but it can't work so well in detecting some small object. In eye detection, eye may be too small to make YOLO perform well in monocular model. In addition, monocular model has a higher accuracy than binocular model in region-based CNNs. That proves that binocular model has symmetry, but monocular model is easier to detect in the Faster R-CNN model.

Finally, monocular model of ResNet-based Faster R-CNN is selected. The model is used on test dataset. Some results of visualization are shown in Figure 7.

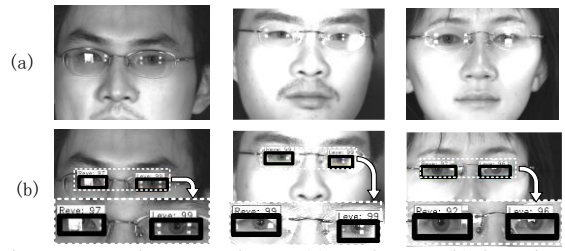


Figure 7: Results on test dataset (a) test images (b) detection results.

5 CONCLUSIONS

Inspired by improving the convenience of iris recognition, an eye detection method based on Faster R-CNN is proposed, which is robust to the case of occlusion or reflection of glasses. Recall of the monocular model can reach 96%. The experiments show that the Faster R-CNN can also be well applied in eye detection.

However, the method still has two main drawbacks: generalization ability and processing speed. Most of eyes in training and test dataset are on clear front face, and the accuracy of detection would be relatively low on dataset with blurred images. In the future, some improvements will be made to solve the drawbacks, such as enlarging training dataset and changing the structure of network.

ACKNOWLEDGEMENTS

This work is supported by the National Key R&D Program of China under Grant 2017YFB0802300 and Beijing Education Commission (KM201510009005).

REFERENCES

Fei Yang et al., 2013, Robust eyelid tracking for fatigue detection. IEEE International Conference on Image Processing, pp.1829-1832.

Wildes R P , 2002, Iris recognition: an emerging biometric technology[J]. Proceedings of the IEEE, 85(9):1348-1363.

Alan L. Yuille et al., 1992, Feature extraction from faces using deformable templates. International Journal of Computer Vision, 8(2), pp.99–111.

Hamouz M. et al., 2005, Feature-based affine invariant localization of faces. IEEE Trans. Pattern Analysis and Machine Intelligence, 27(9), pp.1490–1495.

Fei Yang et al., 2011, Eye localization through multiscale sparse dictionaries. Conference on Automatic Face and Gesture Recognition, pp.514–518.

Girshick et al., 2014, Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR, pp.580–587.

Girshick R., 215, Fast R-CNN. ICCV, pp.1440–1448.

Kaiming.He et al., 2014, Spatial pyramid pooling in deep convolutional networks for visual recognition. ECCV, pp.346–361.

Uijlings J.R et al., 2013, Selective search for object recognition. International Journal of Computer Vision, 104(2), pp.154–171.

S. Ren et al., 2015, Faster R-CNN: towards real-time object detection with region proposal networks. NIPS, pp.91–99.
Huaizu Jiang et al., 2016, Face Detection with the Faster R-CNN[J], pp.650-657.
Zeiler M D et al., 2013, Visualizing and Understanding Convolutional Networks[J], pp.818-833.
K. Simonyan and A. Zisserman., 2014, Very deep convolutional networks for large-scale image recognition. CORR, pp.1409.
Kaiming He et al., 2016, Deep Residual Learning for Image Recognition. Computer Vision and Pattern Recognition. IEEE, pp.770-778.
CASIA Iris Image Database Version, more information is seen http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp
Redmon J, et al., 2015, You Only Look Once: Unified, Real-Time Object Detection, pp.779-788.