Neural Network for Eye Gaze Estimation

Mengqi Wang, Guirong Fu, Ming Yi

Abstract

Eye gaze estimation from pictures is an exciting and challenging task. It has wide applications in both commercial and scientific fields. In this project, we take the advantage of deep neural network to deal with this problem. By adapting and combining existing powerful deep neural network model and carefully designed data augmentation, our model improves the validation loss from 6.430 given by the baseline model to 5.017 on the public score board.

Keywords: gaze estimation, deep neural network, computer vision

1 Introduction

Eye gaze estimation plays an important role in human-computer interaction[3], medical diagnose, computer vision and so on. The general goal is to estimate the direction of the eye gaze given a picture including the eye(s). The direction can be expressed by a 2d vector, pitch and yaw or a 3d vector (x, y, z) in the 3d dimension. From this perspective, we can treat this problem as a common prediction problem in the computer vision tasks. Namely, given a picture, we want to predict a 2d or 3d vectors.

Deep neural network has been one of the major methods to deal with picture-based tasks and achieved state-of-the-art performance in many benchmarks[8]. It has shown to be able to learn complex features given sufficient training data. Meanwhile, several typical DNN model architectures have been put forward to and can be easily applied to specific tasks, for example AlexNet, ResNet, VGG-16[7], GoogLeNet, DenseNet and so on. These models not only provide powerful examples on how to build networks but also allow us to use pre-trained parameters to accelerate training on specific tasks.

The backbone of our project is such a classical computer vision model architecture. Namely, we have tried ResNet50 [4], VGG-16[7] and iTracker Model[5]. In addition, we do data augmentation for increasing the generality and robustness of the model. Details will be discussed in the corresponding subsections 1.1, 1.2 and 1.3.

1.1 Model Architecture

Two existing models: VGG and ResNet, which have excellent performance on the classification task on imagenet, are adapted to integrate into our model respectively.

VGG utilizes the depth of neural network well. One of the core idea of VGG is that depths affect the ability of neural networks. It stacks many layers of small 3×3 convolutional kernels in order to extract features. Several small kernels have been proved to outperform a bigger kernel but with fewer parameters[7]. Going back to our project, eye-gaze is determined by certain features reflected by the images, e.g. near infrared corneal reflections, pupil center and iris contours. VGG is a promising candidate that could learn such features.

ResNet, differing from past DNN, includes identity mapping to fix the vanish of gradient with neural network going deeper. Compared to the VGG with similar performance, a ResNet needs much fewer parameters. So we also try out a model based on ResNet.

The backbone takes an image as input and outputs the estimated eye-gaze vector. In addition to the eye image itself, features like head position, face grid could be important features of the real eye-gaze direction. Our complete model architecture1 adapts the one in iTracker[6]. The Basic-CNN module could be VGG or ResNet, working to extract 1d feature vectors given an image input. Feature vectors from different images and "head position" features are then concatenated together and feed to a mutli-fully-connected layer. This multi-FC module is expected to fully merge and utilize information from all the sources and give the final estimation.

1.2 Data Augmentation

Generally, deep learning benefits from the powerful fitting ability due to large amount of parameters. It, on the other side, requires a certain volume of training dataset to learn these parameters. Data augmentation is a common method to increase the amount and variety of data in deep learning fields[9]. Larger training data can increase the generality of models while the increased variety can make the model more robust to noise.

We also give it a shot in our project. When doing a quick scan of the given dataset, we find that certain ratio of images don't have desirable qualities and mainly suffer from the following types of noise:

- motion blur: images are blurred because of motion.
- noisy points: random but noisy points can be seen in the images.
- cutout(block covered): images are partially blocked.
- combined noise: multiple types of noises exits on single image, e.g. images suffer from motion blur and noise meanwhile.

Fig. 2 shows some noise and effects of data augmentation methods.

We apply several data augmentation methods, namely, motion blur, gaussian noise, block covered and combined

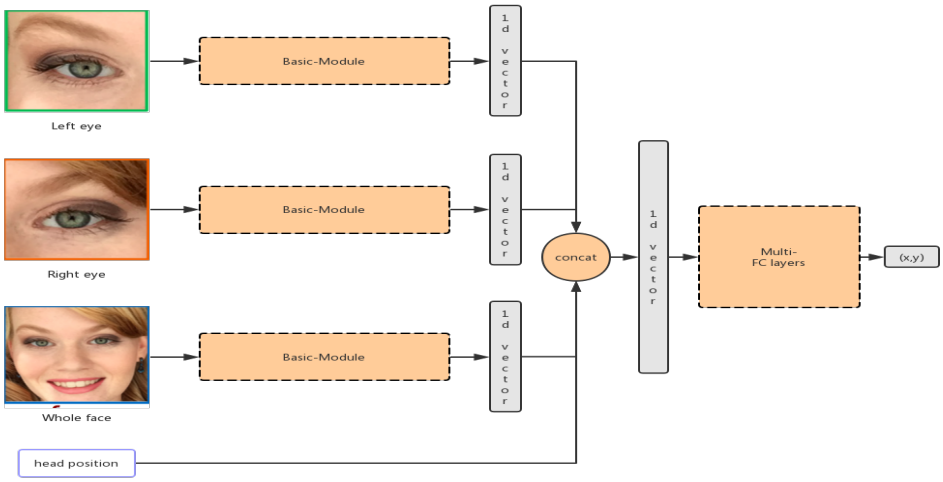

Figure 1. General Model Architecture

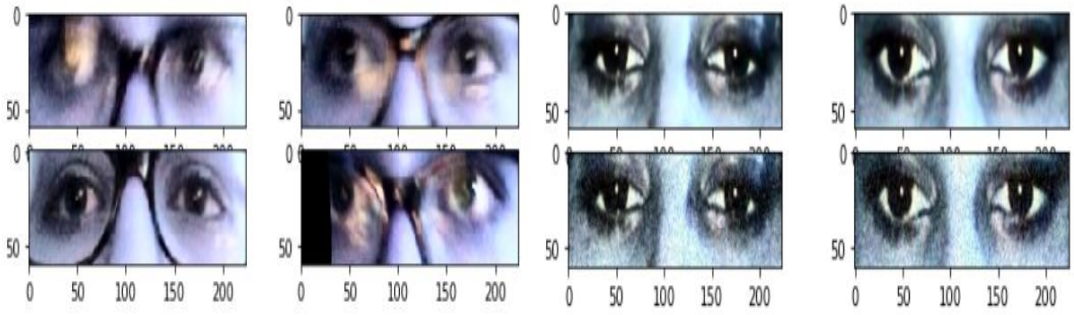

Figure 2. Left two columns: Images suffer from several kinds of noise. The two images in the first row shows motion blur effect. The image at the left-side of the second row shows noise while the one at the right-side shows block covered and motion blur. Right two columns: Original images are in the first row while the ones with Gaussian noise augmented are in the second row.

augmentation according to the significant noises when reading the data. Besides, we use gray images instead of 3-channel RGB images. Colors don't benefit much to the eye-gaze estimation but could bring unexpected noise in our case. In general, the model performance isn't improved much. It could be the reason that the training data is noisy enough and the augmentation doesn't increase the variety obviously.

1.3 Loss Function

As discussed in the paper[1], "At any moment, we cannot expect the same accuracy for two eyes, and either eye has a chance to be more accurate". When feeding two eyes' images into the model and getting two estimations, the ideal action is to take the more accurate one as the final result. We also implement a AR-E network structure following this paper. However, we haven't seen a satisfying improvement. This remains to be a promising direction for future work.

2 Experiments

2.1 Datasets

The neural network is trained on GazeCapture dataset and tested on MPIIFaceGaze dataset. Both datasets are collected in real-world setting with challenging conditions including low illumination, motion blur and occlusion due to head pose or eyeglasses.

One set of image per person includes one whole-face image with the size of 224×224, one eye-region image as the part of the whole-face image with the size of 224×60, one

left-eye and one right-eye image split from the eye-region image with the size of 90×60. Also, for each set of image, we also have features of face grid and head direction. Totally, there are 500 images each from 200 people in GazeCapture for training. For the validation dataset, we have 100 images each from 50 people in GazeCapture and 500 images each from the 15 people in MPIIGaze.

2.2 Model Settings

The overall model architecture is illustrated in Fig.1. The detailed settings of each module are listed below:

- Basic-Module: we use the module ResNet50 and VGG16 in keras.application.
- 1D-vectors: the feature vector of face flattened from the ResNet/VGG output goes through 3 dense layers (4096; 4096; 1000); The feature vector of left-eye /right-eye vector flattened from the ResNet/VGG output goes through 2 dense layers(1024; 512). Both are combined with relu activation.
- Combined Multi-FC layers: Processed feature vectors and headpose vector are concatenated. Then, it goes through 4 dense layers(1024; 1024; 512; 128), with dropout rate 0.5. The output is a 2-length vector as prediction.

We initialize the weight parameters with the weights of the pre-trained model on ImageNet[2] from Keras API.

The settings of hyper-parameters are:

- Initial learning-rate = 1e-4
- batch-size = 16
- epochs = 25
- optimizer: ADAM

Learning rate is multiplied by 0.1 after 60000 iterations (batches), which results in a twice manual decay on the learning rate in the whole training process. The residual error being optimized is angular error which gives a better performance over the RMSE.

2.3 Error Analysis

With respect VGG16, the training error starts from around 10 even without training on one batch. After thousands of batches, the validation error goes under 5 and it stays around 4.5 when the training process finished, while the training error stays around 0.5. However for ResNet50, the training error starts at a large value. The reason can be that the ResNet50 have much less fully connected layers compared to VGG16. The output from the convolution block is already quite selective for a classification task. As we starts employ ResNet50 when it is closed to the deadline, unfortunately, we don't have enough time and GPU resources to explore more on this module.

3 Discussion

3.1 Drawback

The difference between testing loss (4.651) and validation loss (5.017) may be a sign of overfitting. One possible reason is the low stride step which is 1 pixel by default in Keras API. Potential solutions are larger stride step, regularization and early stopping.

3.2 Future Work

We have several ideas yet to be implemented. Firstly, The proposed model can be adapted to integrate many existing models , e.g. *InceptionResNetV2, ResNet152V2* which has been proved to have good performance on the classification task on imagenet. Then the data augmentation can be more carefully designed to reflect the real world scenario, like the illumination, camera angle. Furthermore, An asymmetric error framework, which takes a better feature vector from the two eyes as the input for the fully connected layers, could also be beneficial.

Acknowledgments

To ETH AIT lab for the skeleton code and guidance of the project.

References

[1] Y. Cheng, F. Lu, and X. Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression: 15th european conference, munich, germany, september 8–14, 2018, proceedings, part xiv. 2018.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[3] A. T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag, Berlin, Heidelberg, 2007.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[6] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. M. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. *CoRR*, abs/1606.05814, 2016.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] R. K. Sinha, R. Pandey, and R. Pattnaik. Deep learning for computer vision tasks: A review. *CoRR*, abs/1804.03928, 2018.

[9] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research Applications*, ETRA '16, page 131–138, New York, NY, USA, 2016. Association for Computing Machinery.