Free-head appearance-based eye gaze estimation on mobile devices

Liu, Jigang; Lee, Francis Bu Sung; Rajan, Deepu

2019

Liu, J., Lee, F. B. S., & Rajan, D. (2019). Free-head appearance-based eye gaze estimation on mobile devices. 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). doi:10.1109/ICAIIC.2019.8669057

https://hdl.handle.net/10356/83125

https://doi.org/10.1109/ICAIIC.2019.8669057

Free-Head Appearance-Based Eye Gaze Estimation on Mobile Devices

Liu Jigang
School of Computer Science and Engineering
Nanyang Technological University
Singapore
liujg@ntu.edu.sg

Bu Sung Lee, Francis
School of Computer Science and Engineering
Nanyang Technological University
Singapore
EBSLEE@ntu.edu.sg

Deepu Rajan
School of Computer Science and Engineering
Nanyang Technological University
Singapore
ASDRajan@ntu.edu.sg

***Abstract*— Eye gaze tracking plays an important role in human-computer interaction applications. In recent years, many research have been performed to explore gaze estimation methods to handle free-head movement, most of which focused on gaze direction estimation. Gaze point estimation on the screen is another important application. In this paper, we proposed a two-step training network, called GazeEstimator, to improve the estimation accuracy of gaze location on mobile devices. The first step is to train an eye landmarks localization network on 300W-LP dataset [1], and the second step is to train a gaze estimation network on GazeCapture dataset [2]. Some processing operations are performed between the two networks for data cleaning. The first network is able to localize eye precisely on the image, while the gaze estimation network use only eye images and eye grids as inputs, and it is robust to facial expressions and occlusion.**

Compared with state-of-the-art gaze estimation method, iTracker, our proposed deep network achieves higher accuracy and is able to estimate gaze location even in the condition that the full face cannot be detected.

Keywords—Eye gaze estimation, Deep learning, CNN, Eye localization, Gaze location

I. INTRODUCTION

Eye gaze tracking is the process of predicting where a person looks from data captured from various sensors. In recent years, eye gaze tracking has become an important research topic in computer vision. Because eye gaze reflects human's visual attention and therefore can be used to better understand human activities, eye gaze estimation has a wide range of applications in various areas [3], such as computer-human interaction [4][5], psychological research [6][7], medical studies [8][9], gaming industry [10][11], etc. However, existing gaze estimation methods can fail in the conditions of low image quality or poor illumination.

There are two main categories of gaze estimation methods: **model-based** and **appearance-based**. Most early works on gaze estimation are model-based as they predict gaze by using geometric models of the eyes and face [12], which mimics the structure of human head. This kind of eye gaze estimation has been successfully used in commercial gaze tracking systems. However, they usually need complex hardware systems. Appearance-based methods, on the other hand, use the appearances of eyes as input and learn the mapping function from image patches to gaze prediction. Recently, appearance-based methods have become more popular due to the overwhelming development of deep learning methods and huge data collected from mobile devices like mobile phone, pad, etc.

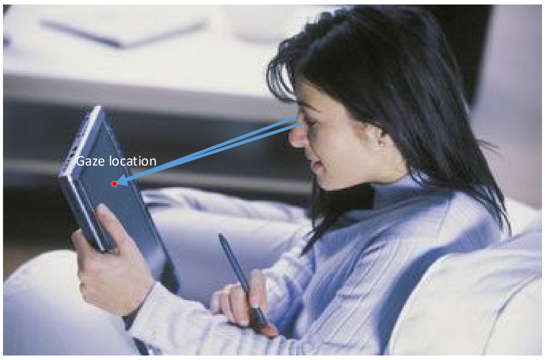


Figure 1. An example of gaze location when a person is looking at a mobile device.

To build a simple, cheap and accurate eye gaze estimation system, we focus on appearance-based gaze estimation with a front-facing camera. Nowadays, mobile devices are widely used, and it is possible to collect huge data. In this work, we proposed a novel architecture， called GazeEstimator composing of two separate convolutional neural networks (CNNs), one for eye localization and another for gaze estimation. Eye localization network is similar to the network architecture of face alignment method [13], which localizes eye landmarks on given images. In this work, we only use eye landmarks as output instead of the whole face. Then eye blink detection and eye region cropping are performed for data cleaning. Finally, both eye image patches and their corresponding grids are input to the gaze estimation CNN.

The contributions in this work can be summarized as follows, 1) an end-to-end framework for eye gaze estimation is proposed using raw image as input and gaze location as output. 2) Eye localization network can localize both eyes even in the condition of partial face appearing in the image. 3) We challenge the traditional face-eye model for eye gaze estimation task by using only both eyes as input to the gaze estimation network.

II. RELATED WORK

Eye gaze estimation has been studied for decades due to its wide applications. In this section, we provide a brief overview of existing eye gaze estimation methods. Recent

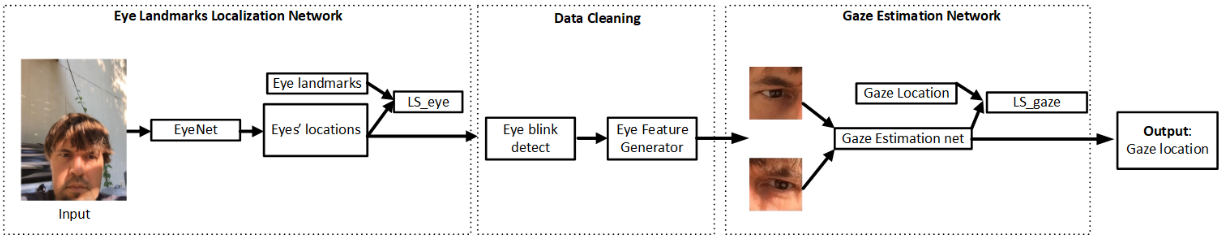


Figure 2. Overview of the proposed gaze estimation method composed of three parts, eye landmarks localization network, eye blink detection and eye region extraction, gaze estimation network.

research on gaze estimation was reviewed by Hansen and Ji [14].

A. Model-based eye gaze estimation

Model-based approaches compute the gaze direction from the eye features based on a geometric model of the eyes [14]. Traditional model-based methods reply on metric information and require camera calibration, geometric model of light sources, camera and eyeball position and orientation [15].

Yang et al. [16] employed one camera and four infrared light sources to capture video, then extracted the corneal glints and the pupil centre according to the grayscale distribution of the video frame. A calibration procedure is added to eliminate the error from the deviation of the optical and visual axes.

A 3D deformable eye-face model was proposed for eye gaze estimation with a single web-camera in [17]. With the deformable eye-face model, eye gaze can be estimated from 2D facial landmarks. A unified calibration algorithm has to reconstruct 3D eye-face model and estimate personal eye parameters for each frame.

Early model-based approaches requires complex equipment and computation, although recent work can works with a single camera, high resolution images are required to predict gaze direction or location.

B. Appearance-based eye gaze estimation

Appearance-based methods directly use image data to estimate gaze information by mapping image data to gaze vector or gaze point. With the development of deep learning methods, promising algorithms for appearance-based gaze estimation using CNNs have been proposed.

Rice TabletGaze dataset [18] is the first dataset collected in unconstrained environment by using mobile devices. The also proposed a regression model for eye gaze estimation. They obtained medium error of 3.17cm.

Krafka and his colleagues [2] introduced an end-to-end eye gaze estimation approach targeting mobile devices. A deep convolutional neural network is trained on a large-scale mobile tracking dataset named GazeCapture. Face, eyes and face grid are prepared beforehand as the inputs to the network. The authors claim that their iTracker model outperforms state-of-the-art methods by a large margin.

An appearance-based gaze estimator is learned from one million synthesised images in [19]. The authors introduced a new way to generate a large number of synthesized images which can be used as training data.

In [20], a spatial weights CNN method is designed for full-face appearance-based gaze estimation. This method is robust to facial appearance variation caused by extreme head pose and gaze directions as well as illumination.

Deng and Zhu [21] used two separate models for head pose and eyeball movement which are connected by a gaze transform layer. This method can overcome head-gaze correlation overfitting.

A simulated and unsupervised learning framework called SimGAN was proposed to refine an eye simulator's output with unlabelled eye image data. The refined eye images are served as training data for eye gaze estimation.

Appearance-based approaches potentially can work on low quality images, and are believed to require large number of labelled training data. In recent years, several eye gaze datasets with a large amount of labelled eye gaze data have been collected in unconstrained environments [2][18][22] as well as synthetic data [19]. Given the success of previous appearance-based eye gaze estimation research and available huge labelled datasets, in this work we also focus on this kind of methods.

III. GAZEESTIMATOR

With the development of deep convolutional neural network (CNN), appearance-based gaze estimation becomes much easier to implement due to reduced need of calibration and robustness to illumination. This work improves Krafka's approach [2] for eye gaze estimation in unconstrained environments. The overview of the proposed GazeEstimator is shown in Figure 2. There are three main parts in this approach, 1) Eye location network for eye landmark localization 2) Eye feature extraction including eye blink detection and eye region and grid calculation. 3) Eye gaze estimation network. In the proposed method, only eye features are used as an input of the eye gaze estimation network, we believe that the eye features are sufficient to present eye gaze. Due to the variation of facial expressions, face feature may introduce overfitting issue to the network.

A. Eye Localization Network

Krafka [2] used Apple's face detection method to extract face image and eye images. But due to the poor performance of this method, face and eyes were successfully detected only from 1,490,959 frames out of 2,445,504 frames. There are 954,545 frames discarded due to failure in localizing face or eyes on these frames. In this work, we applied Bulat's face localization method [13] as the first module to localize eyes on the original images. This approach allows for stronger relations to be learned without excessively increasing the number of network parameters. This neural network was trained for landmark localization by using 300W-LP dataset [1]. Some eye localization results are shown in Figure 3. Even in the condition of low-light condition or occlusion, both eyes are still detected successfully. The situations in Figure 3 are common when a person uses a mobile device and the camera captures part of his/her face. However, eye gazes in these kind of images cannot be estimated in iTracker [2] due to failure in detecting faces.

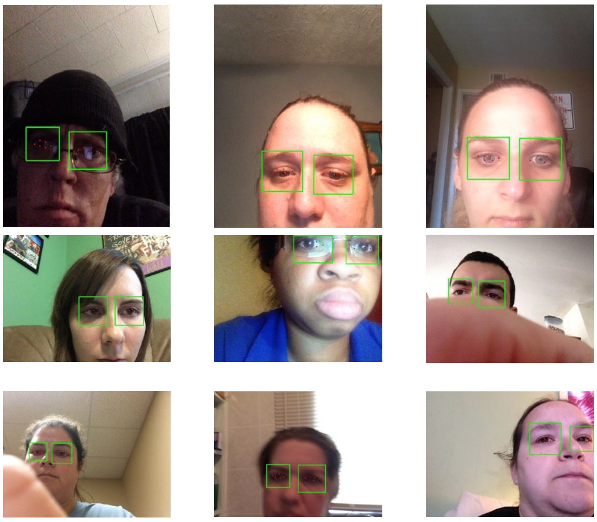
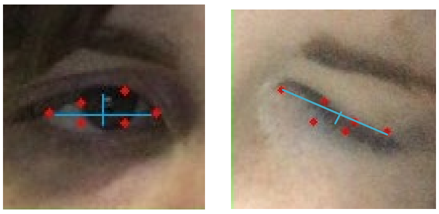


Figure 3. Eye localization results on the frames which are not used in iTrakcer. Even in the condtion of low light (Top left), occlusion (middle right) or part of face is outside the image, eyes still can be detected successfully.

B. Data Cleaning

Each eye is represented by 6 landmarks calculated by eye localization network discussed above. In some frames, the images were captured when the subject was blinking, these images should be removed from the dataset. Examples of open eye and close eye are shown in Figure 4.



(a) (b)

Figure 4. Examples of (a) open eye and (b) closed eye

Based on the work by Soukupová and Čech [23], eye aspect ratio (EAR) is calculated to measure if an eye is open or closed.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \quad (1)$$

where $p_1, \ldots, p_6$ are the coordinates of 2D landmarks shown in Figure 4. In this work, the subject is considered to be blinking in the frame if the EAR value is less than 0.2.

If both eyes are detected and open in the frame, a bounding rectangle of the 6 eye landmarks are calculated for each eye, then a square patch with the bounding rectangle as center is extracted and saved as eye image that serves as one of the inputs of gaze estimation network. Examples of eye square regions are shown as green boxes in Figure 3.

C. Gaze Estimation Network

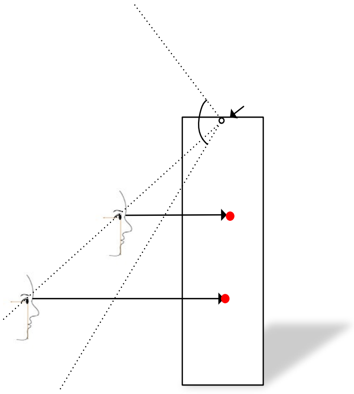


Figure 5. The locations of head in the image captured by the front-facing camera are the same, only the sizes are different. In the condition that the subject keep the same head pose and gaze direction, the gaze points locates in the different positions on the screen of the mobile device.

For the appearance-based gaze estimation task by using CNN, the main challenge is to choose the inputs and the architecture of CNN. Eye gaze estimation network in Figure 2 represents our CNN architecture. There are two inputs for each eye, 1) **eye image** and 2) **eye grid**. Eye image is an image patch with eye ball in the centre and resized to 224x224, and eye grid provides the information on eye location and eye size in the frame. One example of left eye grid calculation is illustrated in Figure 6.

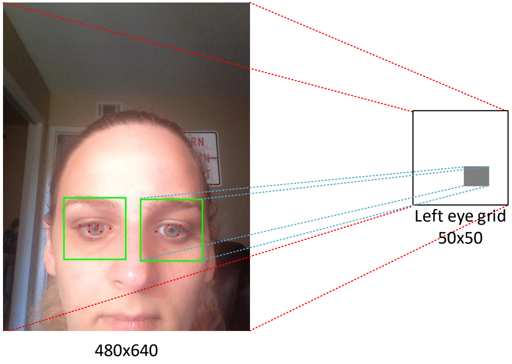


Figure 6. Eye grid calculation. The original frame with size 480x640 is mapped to a image patch with size 50x50. The pixel value of non-left-eye region is set to 255, and that of left eye region is ratio of eye region area to the full frame area in the original image.

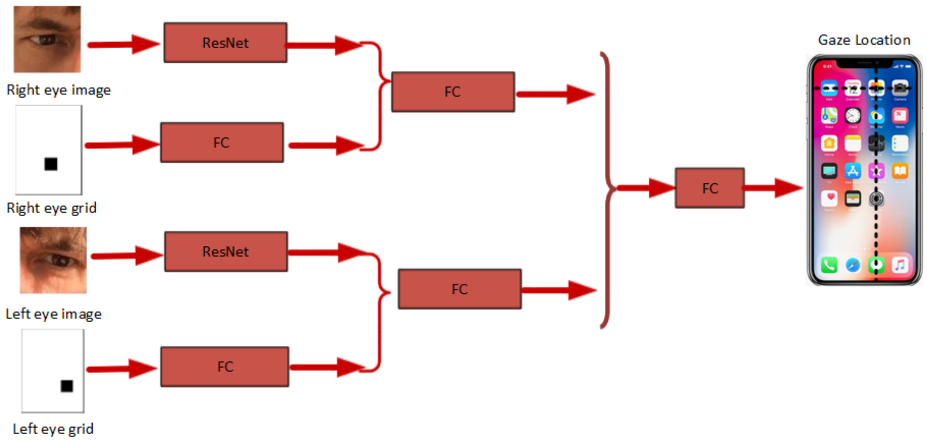
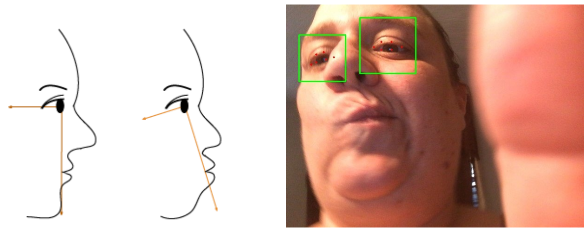


Figure 8. Eye gaze estimation network.

The size of eye grid is crucial for the eye gaze estimation. If it is too small, some intermediate depth information will be lost. One example is illustrated in Figure 5, where the subject keeps the same head pose and gaze direction at hp_1 and hp_2, and the gaze points are located at p_1 and p_2 respectively. If a small size of eye grid is chosen, eye grids are similar for the two cases. In this work, grid size is set to 50x50 and the pixel value of the eye region is set as the ratio of eye area to the full frame area.



(a) (b)

Figure 7. (a) When a person pouts the mouth, the head pose will change [21]. (b) An example of a person pouting the mouth from GazeCapture data.

Features Many appearance-based gaze estimation methods use face image as inputs to CNNs [2][20] because the authors believe that head pose can be calculated from face image. Facial landmark location will change with facial expressions so that it makes head pose unstable [21]. Figure 7 shows that head pose changes when a person pouts the mouth, even when the gaze is kept the same.

We assume that one person's two eyes have similar sizes and shapes and the distance L between two eye balls is a constant in 3D space (See Figure 9). Therefore, head orientation can be estimated by the relative positions and sizes of two eyes.

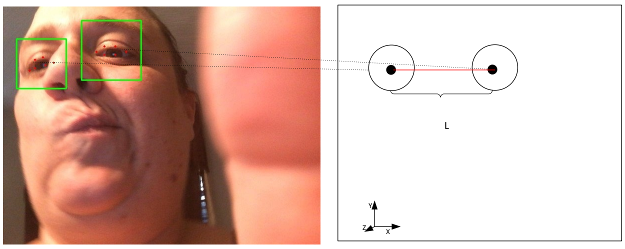


Figure 9. Head pose can be estimated from two eyes.

CNN Architecture In this work, two eye images are fed through ResNet-18 [24] , which can accelerate the speed of training of the deep networks and increase depth of the network resulting in less extra parameters. Two eye grids are fed through fully connected layer without any convolution.

Left eye and left eye grid layers converge upon one fully connected layer after propagating through ResNet and fully connected layer. Similar operation is performed on right eye and right eye grid layers. Finally, all layers are connected by one fully connected layers.

IV. Experiments and Results

A. Data

We use GazeCapture, a large-scale dataset collected by Krafka and his colleagues [2]. They use Amazon Mechanical Turk (AMT) as a platform for recruiting people to use their IOS application which shows dots on a screen at random locations and recording the user's gaze using the front-facing camera.

The dot locations were both random and from 13 fixed locations. The users are forced to change the orientation of their mobile device after every 60 dots. GazeCapture datasets contains a total of 1474 subjects and totally 2,445,504 frames with corresponding dot locations which are gaze points on the screen. 1249 subjects used iPhones while 225 ones used iPads, resulting in a total of 2.1M and 360K frames for each of the devices respectively.

B. Eye Gaze Estimation Network

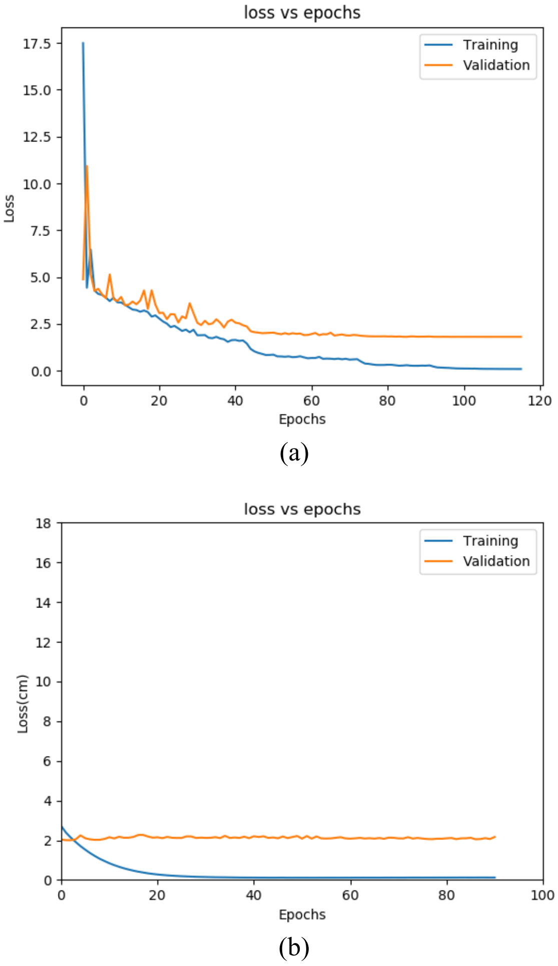
Eye gaze estimation model is implemented using Pytorch [25]. It was trained on GazeCapture dataset for 30 epochs with a batch size of 256. Hyper-parameters we used are listed in Table 1.

Hyper parameter	Value
Learning rate	0.001 (divided by 10 every 5 epochs)
Batch size	256
Momentum	0.9
Optimizer	SGD

Table 1. Hyperparameters

iTracker vs GazeEstimator

To evaluate the performance of GazeEstimator, we use the same data as iTracker, totally 1,490,959 frames to train and validate the network. Figure 10 compared the training loss and validation loss between iTracker and our GazeEstimator. The loss error is defined as the distance between predicted gaze position and the ground truth. The validation error for iTracker is around 2.49cm after 45 epochs, while our method's validation error is converged to 2.1cm during the first epoch. Because ResNet is employed, the proposed network converged must faster than iTracker and achieved smaller prediction error, that is, the predicted eye gaze location is closer to the ground truth.



(a)

(b)

Figure 10. Training loss and validation loss. (a) iTraker (b) GazeEstimator.

Eye Grid

Eye grid represents the eye location and size in the original image. Small size of eye grid leads to certain depth information lost. We have created eye grid with size 25x25 and 50x50 separately, and feed the eye grid with eye image into the proposed eye gaze estimation network. Figure 11 shows the results on training loss and validation loss. The top figure is the results by using eye grid with size 25x25, the validation error is 2.33cm, while the validation error is 2.17cm by using eye grid with size 50x50. Because the eye grid is fed through fully connected layers directly, if its size is too big, it will need more parameters and bring computational burden. In this work, we choose the eye grid with size 50x50.

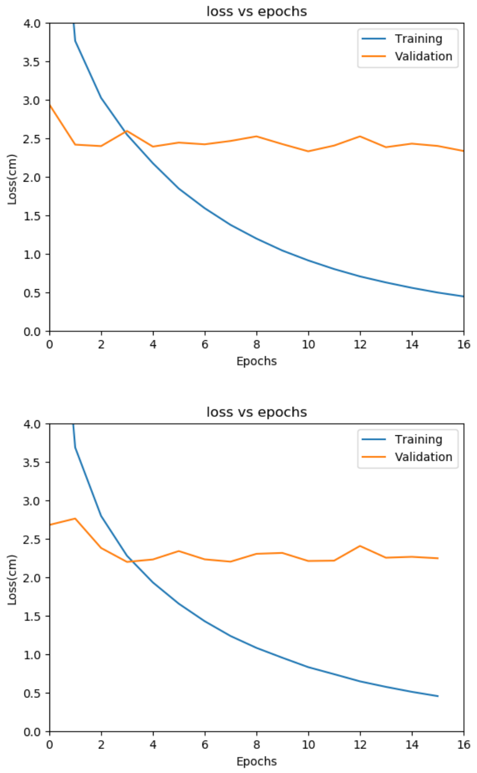


Figure 11. Training and validation loss. (a) Grid size 25x25 (b) Grid size 50x50

The Whole Framework

The whole framework of GazeEstimator is also evaluated on the GazeCapture data. After eye localization and data cleaning, 2,221,723 frames are selected. Then, we divide the data into training set, validation set and test set with 1,777,378 frames, 222,172 frames, and 222,173 frames respectively.

Figure 12 shows the training error and validation error of GazeEstimator. The validation error is as low as 1.25cm, much smaller than that of iTracker (2.44cm).

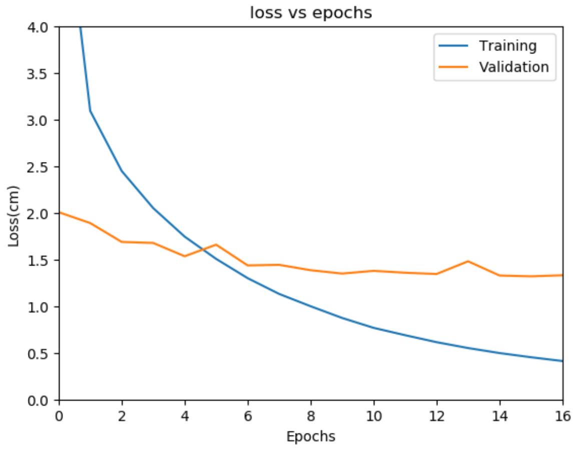


Figure 12. Training loss and validation loss of the proposed eye gaze estimation method.

Model	Overall ave. error	Mobile phone ave. error	Tablet ave. error
iTracker	2.44cm	2.04cm	3.32cm
GazeEstimator	1.25cm	1.14cm	1.92cm

Table 2. Comparison of eye gaze estimation results between GazeEstimator and iTracker.

The trained model is also evaluated on the mobile phone data, tablet data and mixed data separately. Table 2 compares the estimation results of iTracker and GazeEstimator. Compared to iTracker, GazeEstimator reduces error by 1.2cm for overall data, 0.9cm for mobile data and 1.4cm for tablet data.

V. CONCLUSION

In this paper, we proposed a full framework for eye gaze estimation, called GazeEstimator. Eye landmarks are first estimated on the input image by hierarchical binary CNNs, followed by data cleaning which removes frames with closed eyes or no eye detected. Finally, eye images and eye grids are fed through eye gaze estimation network. Compared with the state-of-the-arts methods, the proposed approach achieved higher accuracy for eye gaze point estimation.

VI. ACKNOWLEDGMENT

We wish to acknowledge the funding support for this project from AcRF Tier 1 grant entitled, Eyegaze estimation using deep appearance in natural environment.

REFERENCES

[1] X. Zhu, xiaoming Liu, Z. Lei, and S. Z. Li, “Face Alignment In Full Pose Range: A 3D Total Solution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] K. Krafka, A. Khosla, P. Kellnhofer, and H. Kannan, “Eye Tracking for Everyone,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[3] A. T. Duchowski, “A breadth-first survey of eye-tracking applications,” *Behav. Res. Methods, Instruments, Comput.*, 2002.

[4] A. Bulling and H. Gellersen, “Toward mobile eye-based human-computer interaction,” *IEEE Pervasive Comput.*, 2010.

[5] H. (2010) Drewes, “Eye Gaze Tracking for Human Computer Interaction,” *Thesis*, 2010.

[6] M. L. Mele and S. Federici, “Gaze and eye-tracking solutions for psychological research,” *Cogn. Process.*, 2012.

[7] K. K. W. Kampe, C. D. Frith, R. J. Dolan, and U. Frith, “Psychology: Reward value of attractiveness and gaze,” *Nature*, 2001.

[8] D. Novak and R. Riener, “Enhancing patient freedom in rehabilitation robotics using gaze-based intention detection,” in *IEEE International Conference on Rehabilitation Robotics*, 2013.

[9] M. S. Atkins, G. Tien, R. S. A. Khan, A. Meneghetti, and B. Zheng, “What do surgeons see: Capturing and synchronizing eye gaze for surgery applications,” *Surg. Innov.*, 2013.

[10] J. S. Agustin, J. C. Mateo, J. P. Hansen, and A. Villanueva, “Evaluation of the potential of gaze input for game interaction,” *PsychNology J.*, 2009.

[11] H. Heo, E. C. Lee, K. R. Park, C. J. Kim, and M. Whang, “A realistic game system using multi-modal user interfaces,” *IEEE Trans. Consum. Electron.*, 2010.

[12] R. Valenti, N. Sebe, and T. Gevers, “Combining head pose and eye location information for gaze estimation,” *IEEE Trans. Image Process.*, 2012.

[13] A. Bulat and Y. Tzimiropoulos, “Hierarchical binary CNNs for landmark localization with limited resources,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[14] D. W. Hansen and Q. Ji, “In the Eye of the Beholder: A Survey of Models for Eyes and Gaze,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.

[15] H. R. Chennamma and X. H. Yuan, “A survey on eye-gaze tracking techniques,” *Indian J. Comput. Sci. Eng.*, vol. 4, no. 5, pp. 388–393, 2013.

[16] X. H. Yang, J. D. Sun, J. Liu, X. C. Li, C. X. Yang, and W. Liu, “A remote gaze tracking system using gray-distribution-based video processing,” *Biomed. Eng. Appl. Basis Commun.*, vol. 24, no. 3, pp. 217–227, 2012.

[17] K. Wang and Q. Ji, “Real Time Eye Gaze Tracking with 3D Deformable Eye-Face Model,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[18] Q. Huang, A. Veeraraghavan, and A. Sabharwal, “TabletGaze: unconstrained appearance-based gaze estimation in mobile tablets,” *arXiv Prepr. arXiv1508.01244*, 2015.

[19] 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*, 2016.

[20] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, “It’s Written All over Your Face: Full-Face Appearance-Based Gaze Estimation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[21] H. Deng and W. Zhu, “Monocular Free-Head 3D Gaze Tracking with Deep Learning and Geometry Constraints,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[22] Q. Huang, A. Veeraraghavan, and A. Sabharwal, “TabletGaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets,” *Mach. Vis. Appl.*, 2017.

[23] J. Cech, “Real-Time Eye Blink Detection using Facial Landmarks,” *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague*, 2016.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] N. Ketkar, “Introduction to PyTorch,” in *Deep Learning with Python*, 2017.