Gaze Estimation Using Residual Neural Network

En Teng Wong, Seanglidet Yean, Qingyao Hu, Bu Sung Lee, Jigang Liu, Rajan Deepu
School of Computer Science and Engineering)
Nanyang Technological University
Singapore
{wong0962, seanglid002, qhu003, ebslee, liujg, asdrajan}@e.ntu.edu.sg

***Abstract*—Eye gaze tracking has become an prominent research topic in human-computer interaction and computer vision. It is due to its application in numerous fields, such as the market research, medical, neuroscience and psychology. Eye gaze tracking is implemented by estimating gaze (gaze estimation) for each individual frame in offline or real-time video captured. Therefore, in order to produce the secure the accurate tracking, especially in the emerging use in medical and community, innovation on the gaze estimation posts a challenge in research field. In this paper, we explored the use of the deep learning model, Residual Neural Network (ResNet-18), to predict the eye gaze on mobile device. The model is trained using the large-scale eye tracking public dataset called GazeCapture. We aim to innovate by incorporating methods/techniques of removing the blinking data, applying image histogram normalisation, head pose, and face grid features. As a result, we achieved 3.05cm average error, which is better performance than iTracker (4.11cm average error), the recent gaze tracking deep-learning model using AlexNet architecture. Upon observation, adaptive normalisation of the images was found to produce better results compared to histogram normalisation. Additionally, we found that head pose information was useful contribution to the proposed deep-learning network, while face grid information does not help to reduce test error.**

***Index Terms*—eye track, mobile, ResNet, deep learning**

I. Introduction

Gaze Estimation is the process of determining either the 3D gaze direction or 2D gaze point that a person is looking at while considering the detected eyes in images or videos. In recent years, the gaze estimation is an established and popular research topic in both human-computer interaction and computer vision [1], [2].

The study of gaze estimation continues to expand to various application domains such as psychological research, medical studies, etc [3]. The reason is that gaze greatly links to the cognitive behaviour of an individual [4]–[7]. Moreover, accurate gaze estimation can potentially provide an additional way to predict how humans interact with new technological devices. This can be useful especially to people with certain disabilities.

Gaze estimation research exploits the aspects surrounding the user's eyes. They includes the visual of users (shape of the eye, detected face, head orientation and positioning), surrounded environment (illumination, device's camera orientation), etc. D. W. Hansen and Q. Ji pointed out that the appearance of the eyes changes drastically by the change in yaw angle [8]. Likewise, the perception of eyes varies by adjusting the camera from the top to bottom. Traditionally, the gaze's input is collected using stationary camera such as tobii or webcame [9]. Since hand-held mobile devices is accessible and affordable, gaze estimation research has shifted its attention to the new device. This introduces new variables as well as challenges because the device is no longer still.

Another factor that lead to improvement in gaze estimation accuracy is quantity and quality of data to train the deep-learning model. Rahayfeh et al. analysed the existing techniques of eye tracking and gaze estimation and concluded that the majority of studies did not utilize the variation and scale of the dataset [10]. After that, Krafka et al. have contributed a massive public dataset called GazeCapture [8] addressing the issue. The GazeCapture consists of 1474 unique subjects with more than 2 million samples. It has provided high scalability and high degree of variation since the data were collected via crowdsourcing. The team had also developed and trained a neural network based on the AlexNet architecture [11], called iTracker. The iTracker model was able to achieve a test error (without calibration and fine tuning) of 1.77cm and 1.53cm for mobile phone and tablet device respectively.

This project aims to improve gaze estimation on mobile devices through deep learning, (specifically the Residual Neural Network [12]) using the public dataset GazeCapture. It is to be noted that GazeCapture is a public dataset consisting of facial images and the point of gaze on a variety of Apple devices. It offers scalability by a large margin compared to other existing public dataset on eye-screen information. The contributions in this paper include:

- Removing of incorrect data of blinking frames
- Incorporating normalisation methods for data preprocessing
- Extracting insightful features such as Euler's angles of head pose and face grids.

II. Methodology

A. Data Preprocessing

Fig. 1 illustrates the preprocessing methods that GazeCapture dataset followed before being used as the model's input. These methods aims to clean the dataset (Remove Blinking Frames), extract potential features (Calculate Head Pose), and reduce the complexity space of the image (Normalization).

1) Removal of Blinking Frames: The frames taken for the GazeCapture dataset consists of a 'valid' tag, which indicates whether a face was detected by IOS face detector at the point

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

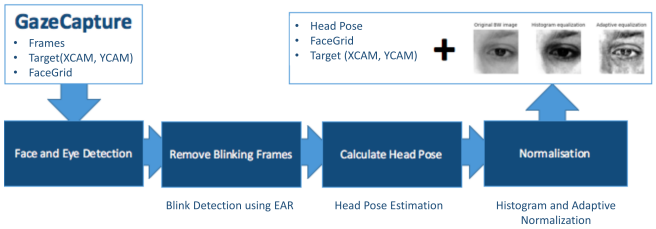

Figure 1: Preprocessing Overview

of photo capturing. However, some frames were taken when the subject was blinking. Thus, eye information in these valid frames were absent. To remove these blinking frames, we calculate the Eye Aspect Ratio (EAR) using the dlib library [13]–[15]. The EAR is a metric that determines a blink using a threshold, and is derived from the width and height of the eye shown in Eq. (1).

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||} \quad (1)$$

p1, p2, .. p6 are eye landmarks detected by dlib. Fig. 2 presents the eye landmarks used for EAR calculation. The EAR threshold used for this project is 0.22, i.e., only valid frames with EAR values above 0.22 are used.

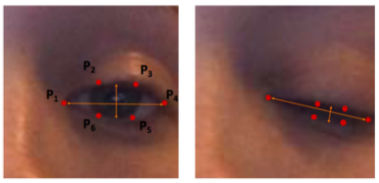

Figure 2: Calculation of EAR using eye landmarks

2) Histogram Normalisation and Adaptive Normalisation: To reduce the complexity, the raw images were normalized to grayscale. The contrast of these grayscale images were subsequently tweaked to improve performances of the neural network. This paper explores two normalization technique: Histogram and Adaptive Equalization. Histogram equalization adjusts the local intensities by flattening the histogram computed from the image and remapping the intensity values. Adaptive normalization is similar to histogram equalization, except multiple histograms were applied to different areas of the image. Fig. 3 shows the processed images after the normalization process.

Figure 3: Processed black and white image using Histogram Normalisation and Adaptive Normalisation

B. Head Pose Estimation

Due to unrestricted user's head movement, head pose could be the additional feature to the training model. Hence, the orientation of head pose was extracted and represented by the Euler's angles: pitch, yaw, and roll (Fig. 4). The landmarks provided by dlib were used to compute the head pose [15].

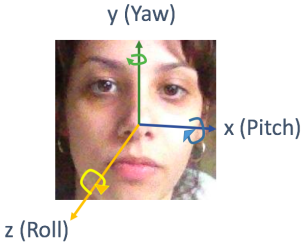

Figure 4: Head Pose using Euler's Angles; Pitch, yaw and roll

C. Residual Neural Network

1) Architecture: Fig. 5 depicts the architecture of the training model using Residual Neural Network (ResNet) [12]. Compared to the AlexNet structure used for iTracker, ResNet uses residual blocks to retain to counter the degradation problem caused by saturated accuracy in deep networks. In this paper, we uses 18 layers of residual blocks for each input. It is then followed by the fully connected layers.

2) Training Details:

- Metrics: To ensure uniform prediction space across the Apple devices with different resolutions and sizes, the distance from the camera to the point on the screen were used instead of the basic xy coordinates of the point on the screen. This metric is provided in the GazeCapture dataset.
- Loss function: Euclidean's distance between predicted $(\bar{x}, \bar{y})$ and target (x, y).
- Validation: 10% data are used as validation dataset.
- Learning rate: Initialised learning rate is 0.001 with decay factor of $sqrt(0.1)$. Minimum learning rate is set to be $0.5e^{-6}$.
- Optimiser: Adams optimiser [16]
- Epochs: 100 epochs.

III. Results and Analysis

A. Effects of removing blinking frames

We compare the test error between before and after removal of blinking frames for both ResNet and re-implemented iTracker (Table I). From Table I , we can see that the test error is reduced for both ResNet and iTracker. Fig. 6 visualize the mapping the predicted point to the target point. This illustration shows that the predicted values are closer to the targets for cleaned data.

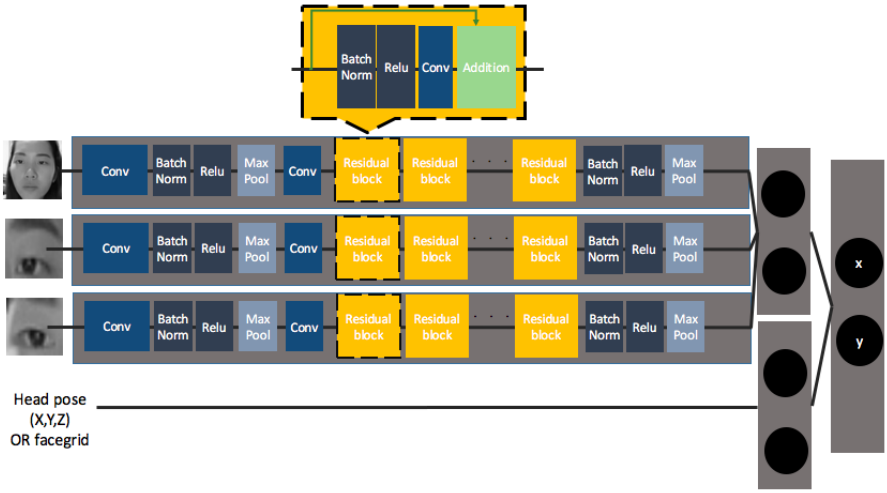

Figure 5: Architecture Diagram

Table I: Training and test error on before and after removal of blinking frames for Resnet and iTracker

	Results			
	ResNet		iTracker	
Frames	150 subjects * 100 frames			
Input	Images of face, left eye and right eye			
Preprocess	None	Cleaned	None	Cleaned
Training Error (cm)	0.84	0.65	2.11	0.13
Test Error (cm)	3.38	3.27	4.11	3.36

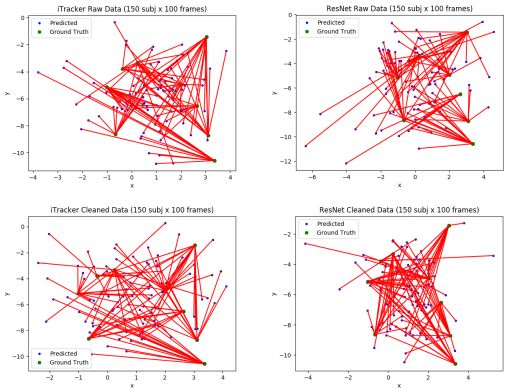

Figure 6: Visualisation of predicted points mapped to target points. Top: Resnet vs iTracker using uncleaned data. Bottom: Resnet vs iTracker using cleaned data

B. *Modes of Normalisation*

We compare the test error between the two types of normalization methods. The results in Table II show that adaptive normalization has helped in reducing the test errors. In contrary, histogram normalization has aggravated the test error. Comparing the images normalized in the histogram and adaptive methods, we believe that the aggravation is due to the decrease in contrast. Histogram normalization has resulted in an low-contrast.

Table II: Training and test error on modes of normalisation

	Results (Resnet)	
Frames	150 subjects * 100 frames	
Input	Images of face, left eye, right eye	
Preprocess	Cleaned + Hist	Cleaned + Adaptive
Training Error (cm)	0.64	0.82
Test Error (cm)	3.77	3.18

C. *Effects of Using Head Pose*

Table III shows the results for including head pose (Euler Angles) as part of training features. It can be seen that inclusion of the Euler angles had reduced the test errors, implying that head pose information was helpful. Figure 7 shows the distribution of head pose. The curve suggests that there were no significant changes in head pose generally. However, the head pose affects the human gaze.

Table III: Training and test error on using head pose with images

	Results (Resnet)
Frames	150 subjects * 100 frames
Input	Images of face, left eye, right eye and head pose
Preprocess	Cleaned + Adaptive
Training Error (cm)	0.1
Test Error (cm)	3.05

D. *Effects of Using Face Grid*

Face grid information was included in the GazeCapture dataset. The face grid encompasses information on location and size of the detected face in the full frame. In contrary to the research on iTracker, introducing face grid information has slightly increased the test error from 3.18 in Table II to 3.22 in Table IV . This could be due to the lower number of nodes in the fully connected layer compared to that in iTracker

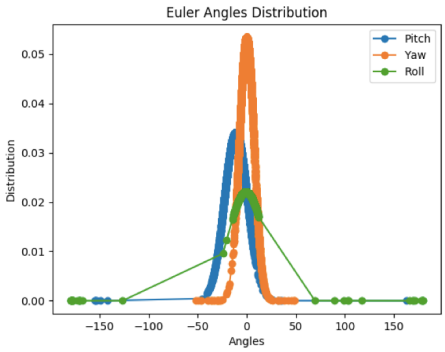

Figure 7: Head Pose using Euler's Angles; Pitch, yaw and roll

that takes in face grid input. Having more nodes in the layer essentially allow the model to learn more features. Thus, by having a lower number of nodes, the proposed architecture was not able to generalise as good as iTracker when the face grid information is included.

Table IV: Training and test error on using face grid with images

	Results (Resnet)
Frames	150 subjects * 100 frames
Input	Images of face, left eye, right eye and head grid
Preprocess	Cleaned + Adaptive
Training Error (cm)	0.38
Test Error (cm)	3.22

Discussion

In this research, we have proven the viability of using ResNet as deep learning approach to gaze estimation in handheld devices. Though preprocessing has helped in lowering test errors for gaze estimation, these preprocesses take a significant large amount of time and are not yet feasible to be implemented in handheld devices. We believe that finding the optimal contrast can lower the test errors.

Furthermore, to be able to train the full dataset can be beneficial to generalization performances of the network as well.

Lastly, we believe the EAR threshold configured can be improved by computing an optimal EAR threshold for each subject. This is because each subject have different eye shapes and those with narrower eyes tend to have smaller EAR, resulting in many frames from the individual to be rejected.

Conclusion

In this paper, we proposed the use of Residual Neural Network (ResNet) to predict gaze point in handheld device using the massive public dataset, GazeCapture. From the result, the proposed Resnet framework achieved 3.05cm average error comparing with 4.11cm average error from the recent gaze tracking model AlexNet architecture. In addition to the deep learning architecture, the performance improvement was the result of exploring the preprocessing techniques such as removing blinking frame, applying normalization, and inputting features such as head pose and face grid. The effect of removing blinking frames were proven useful as it provide the bettter convergence between the predicted points and the target points. Secondly, the adaptive normalization is beneficial in generalization performance of the network. Studying the input features, head pose showed improvements to generalization performance in ResNet while face grid information did not help to reduce test error.

Acknowledgment

We wish to acknowledge the resources from Nanyang Technological University as well as the funding supports from the Undergraduate Research Experience on CAmpus (URECA) programme and AcRF MoE Tier 1 grant entitled, Eyegaze estimation using deep appearance in natural environment.

References

[1] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction," in *Advances in physiological computing*. Springer, 2014, pp. 39–65.

[2] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[3] P. A. Punde, M. E. Jadhav, and R. R. Manza, "A study of eye tracking technology and its applications," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*. IEEE, 2017, pp. 86–90.

[4] Ö. Battal, T. Balcıoğlu, and A. D. Duru, "Analysis of gaze characteristics with eye tracking system during repeated breath holding exercises in underwater hockey elite athletes," in *Biomedical Engineering Meeting (BIYOMUT), 2016 20th National*. IEEE, 2016, pp. 1–4.

[5] R. Y. Cristanti, R. Sigit, T. Harsono, D. C. Adelina, A. Nabilah, and N. P. Anggraeni, "Eye gaze tracking to operate android-based communication helper application," in *Knowledge Creation and Intelligent Computing (IES-KCIC), 2017 International Electronics Symposium on*. IEEE, 2017, pp. 89–94.

[6] T. Falck-Ytter, S. Bölte, and G. Gredebäck, "Eye tracking in early autism research," *Journal of neurodevelopmental disorders*, vol. 5, no. 1, p. 28, 2013.

[7] S. Goyal, K. P. Miyapuram, and U. Lahiri, "Predicting consumer," in *2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*. IEEE, 2015, pp. 126–129.

[8] 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*, 2016, pp. 2176–2184.

[9] Y. Liu, B.-S. Lee, and M. McKeown, "A new reconstruction method in gaze estimation with natural head movement," in *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*. IEEE, 2017, pp. 219–222.

[10] A. Al-Rahayfeh and M. Faezipour, "Eye tracking and head movement detection: A state-of-art survey," *IEEE journal of translational engineering in health and medicine*, vol. 1, 2013.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] 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*, 2016, pp. 770–778.

[13] R. G. J. Janssen, L. Rothkrantz, I. P. Wiggers, and I. H. Geers, "Real time eye blink detection using a configurable processor," Ph.D. dissertation, Doctoral dissertation, TU Delft, Delft University of Technology, 2010.

[14] Opencv library. https://opencv.org/.

[15] Dlib.net. dlib c++ library. http://dlib.net/.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.