Advance Access publication on 12 May 2021 https://doi.org/10.1093/comjnl/bxab043

Gaze Estimation Using Neural Network And Logistic Regression

Yifan Xia[1,2], Baosheng Liang[1,3], Zhaotong Li[1] and Song Gao[1,4,*]

[1]*Institute of Medical Technology, Health Science Center, Peking University, Beijing 100191, P.R. China*
[2]*School of Health Humanities, Peking University, Beijing 100191, P.R. China*
[3]*Department of Biostatistics, School of Public Health, Peking University, Beijing 100191, P.R. China*
[4]*Department of Medical Physics, Health Science Center, Peking University, Beijing 100191, P.R. China*
**Corresponding author: gaoss@bjmu.edu.cn*

Currently, a large number of mature methods are available for gaze estimation. However, most regular gaze estimation approaches require additional hardware or platforms with professional equipment for data collection or computing that typically involve high costs and are relatively tedious. Besides, the implementation is particularly complex. Traditional gaze estimation approaches usually require systematic prior knowledge or expertise for practical operations. Moreover, they are primarily based on the characteristics of pupil and iris, which uses pupil shapes or infrared light and iris glint to estimate gaze, requiring high-quality images shot in special environments and other light source or professional equipment. We herein propose a two-stage gaze estimation method that relies on deep learning methods and logistic regression, which can be applied to various mobile platforms without additional hardware devices or systematic prior knowledge. A set of automatic and fast data collection mechanism is designed for collecting gaze images through a mobile platform camera. Additionally, we propose a new annotation method that improves the prediction accuracy and outperforms the traditional gridding annotation method. Our method achieves good results and can be adapted to different applications.

Keywords: neural network; gaze estimation; logistic regression; machine learning

Received 2 April 2020; Revised 18 February 2021; Editorial Decision 27 March 2021
Handling editor: Fionn Murtagh

1. INTRODUCTION

Gaze estimation is a technique of detecting and obtaining the direction and position of observed gaze through hardware and software algorithm analyses [1–4]. For instance, as illustrated in Fig. 1, a gaze estimation-based application installed in the device would return the predicted gaze location on the screen when the user is looking at the screen. Recently, gaze estimation has been widely used in many scientific research fields, especially in human–computer interaction [5]. With gaze estimation technology, a system was developed that around the needs for region of interest decompression and display in the context of large image interpretation and analysis in science and medicine [6]. Other application fields cover advertising recommender systems [7], assisted driving [8], psychology [9], military [10] and other fields [11, 12]. As the number of scenarios using gaze estimation technology is increasing, more convenient, fast and low-cost methods for gaze estimation are required.

Gaze estimation also has many applications in medical research. An example is the vision screening test, a popular test used in hospital for screening potential vision problems and eye disorders [13]. Typically, during the test, a doctor monitors the eye movement of children when the children are attracted to the test pictures. However, many factors including the non-cooperation of children, limitation of viewing angle and different medical experience of doctors can affect the results of the vision screening test. For the vision screening test, gaze estimation can be utilized to obtain more accurate results in a fast and convenient way. For instance, using the gaze estimation method proposed herein, without human intervention, mobile devices can automatically recognize the gaze direction of children or assess whether the gaze of children

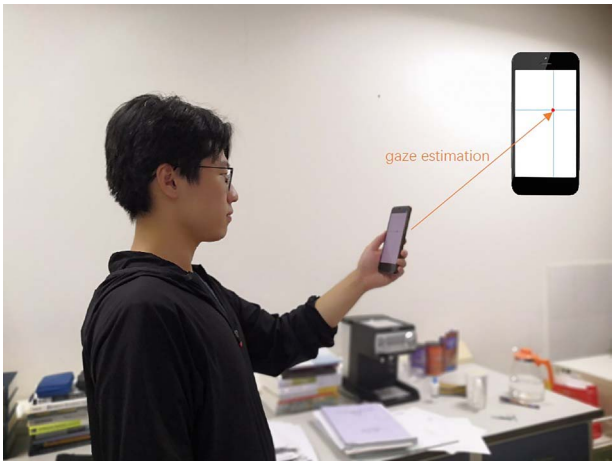

FIGURE 1. An example of gaze estimation application in the daily life, the gaze location is predicted on the screen when a user plays a smart phone.

is located on the testing pictures through the gaze captured by the camera of mobile devices, and can thus aid doctors in obtaining more accurate test results.

In this study, we aim to develop a gaze estimation method specialized for mobile devices such as mobile phones. The main reasons for mobile devices are 3-fold. First, mobile phones are widely used not only in communication but also in daily life, including making online shopping payments, entertainment, telecommuting and identity recognition. Next, mobile phones receive frequent hardware updates, and most smart phones are capable to support computationally intensive algorithms. Moreover, the rapid development of digital cameras in phones has tremendously improved image quality, which enables us to collect reliable gaze images conveniently and economically. Mobile phones supply a convenient platform for gaze estimation and have become popular in recent years. Gaze has shown its advantages in human–computer interaction. When one uses mobile device, gaze serves as a tool for hands-free interaction and has many applications experience as an input modality for tasks including desk control [14], target selection [15] and notification display [16]. Using gaze for interaction control is faster than the mouse for pointing on the screen; the mouse tends to lag behind gaze by more than 100 ms on average, according to the result of experiment about eye and mousecorrelated relationship [17]. The gaze estimation technique can also be used to recommend relevant or similar goods to a potential customer if the gaze of a user is detected to stop on some good. When a user is playing games on a mobile phone, the captured gaze can help the user to control the direction of the target movement. The gaze captured by mobile phone cameras can help users to unlock their phones using password entry [18] and open an application without using hands.

Deep learning is an extremely powerful technique that has developed fast and is widely used in computer vision [19–21], including gaze estimation [22, 23]. iTracker [24] is a convolution neural network (CNN) for gaze estimation, a typical method based on deep learning. Hence, we used the deep learning technique in our proposed gaze estimation method. Furthermore, there have been many open-source datasets for gaze estimation, such as GazeCapture [24], TabletGaze [25], a comprehensive head pose and gaze database [26], ETH-XGaze [27] and RT-GENE [28]. However, these datasets were collected from mixed devices or tablets instead of single phone devices. Also, these datasets lacked the images where the participants' gaze located outside the screen. We wish to develop a gaze estimation method mainly based on mobile phone that could exhibit good precision and is applicable to various applications in daily life. Therefore, we collected our own gaze dataset using mobile phones. The appearance and operating system of device used in this study is similar to the most popular and daily used mobile devices. Besides, the participants were all Chinese. In collecting process, we added temporal information and collected the images when the gaze was located outside the screen, which can be used to improve the model.

Motivated by the existing methods, we proposed a gaze estimation method which combined a neural network and the logistic regression method. Our proposed method relied on a two-stage process. In the first stage, a convolutional neural network with a logistic regression layer processed the input gaze pictures and output estimated probability vectors of bins annotation labels in both horizontal and vertical directions. In the second stage, an additional logistic regression was used for refinement of prediction from the neural network. The proposed two-stage method is different from existing gaze estimation methods, although they share certain similarities. To be specific, for the first stage, a similar method treating the screen horizontally and vertically has been investigated in TabletGaze, where the gaze labels of data also include both horizontal and vertical coordinates on the screen. Different from TabletGaze, we treat gaze direction as horizontal and vertical directions separately and split screen into bins. The labels of horizontal and vertical directions are bins indexes vectors including 1 and 0 instead of one coordinate. The gaze location in the TabletGaze is obtained directly by regression, while the gaze location in the proposed method is obtained though regression using the probability vector output from CNN and setting the threshold. In another paper, Liu [29] studied a logistic regression layer following a CNN for gaze estimation, which also shares certain similarity with our method. In Liu's work, the last classification layer of CNN is replaced with a logistic regression layer using for classification in all raw image pixels. The proposed method by Liu is more similar to the gridding method, which needs $W \times H$ annotation values for each image with screen resolution $W \times H$. Compared with Liu's methods, the proposed method divides screen into bins and makes $(W + H)$/(length of bin) annotation values for each image. Besides, the logistic regres-

FIGURE 2. The scene of gaze data captured from a participant and example collected images from participants with different postures.

sion layer following the CNN in our proposed method is used to produce probability vectors corresponding to the bins instead of directly classification for binary outcome. In the second stage of the proposed method, the probability vectors are then used to fit curves with modified sigmoid function and get target point by setting threshold, which can refine the prediction result.

In this article, we developed a data driven model for gaze estimation based on two-stage process without using hand-engineered features. In specific, we proposed an annotation method that annotates the gaze location by splitting the screen into bins horizontally and vertically instead of pixels. Compared with traditional annotation methods, the proposed annotation method could convert a complex multi-class problem into a binary-class problem in multiple bins, well control the number of labels and improve the accuracy. The first stage of the proposed method is similar to existing works, and the additional logistic regression in the second stage is our major contribution which can further process the output probability vectors for refinement of the gaze prediction.

2. MATERIALS AND METHODS

2.1. Data collection

For gaze estimation, we first designed a set of automatic and fast data collection mechanisms for collecting gaze data that included ordinary images captured by the designed mobile platform camera. In this study, we collected 200 frames for each of the 550 participants. The participants were all Chinese with the age ranging 20–35 years old. It took about 10 minutes for each participant. The data collection was implemented using Samsung Galaxy S8+. The screen resolution was 2220 $\times$ 1080 pixels. The screen size and device type are aligned with the popular devices Chinese used. The light environment of data collection mimics the natural office working scenarios. To collect gaze images, we preset fixed points on the mobile device screen, which enabled us to obtain the ground truth of the gaze points on the screen easily and conveniently. We setup a collecting program, in advance, that provided instructions to the participants. The participants followed the instructions to prepare for the next step or look at the screen. We obtained frames from the camera of the mobile device when the participants looked at the fixed points on the screen. Specifically, once the collecting program started, the fixed points began to appear in order on the screen. There were 25 gaze points in total and each point was presented for 15 seconds on the screen. The camera captured pictures when the participants were instructed to look at the points. To guarantee the quality of the collected images, every step was performed after a voice prompt. Before displaying a new gaze point, there was a break about 5 seconds. To avoid fatigue of participants, the screen automatically turned black so as to give the participants a short break about half minutes when the screen displayed five gaze points. In addition, the participants were encouraged by voice prompt to change their head pose and move their head to a different position relative to the camera. The scenes of participants with different head poses or body postures simulate the scenes of different postures people using phones in daily life. The scene of gaze data captured was displayed in the left panel of Fig. 2. Due to different sitting posture of participants, the distance from participants to screen was not fixed and was ranging from 25 to 60 cm. An illustration is shown in the right panel of Fig. 2. We also captured the images where the gaze was spotted outside the screen for improving the accuracy of gaze estimation. Besides, we captured the data frames with time sequence for adding the time series relation information to our model for future research.

2.2. Data annotation and pre-processing

Data annotation is necessary for gaze learning tasks. A commonly used method is the gridding annotation method that

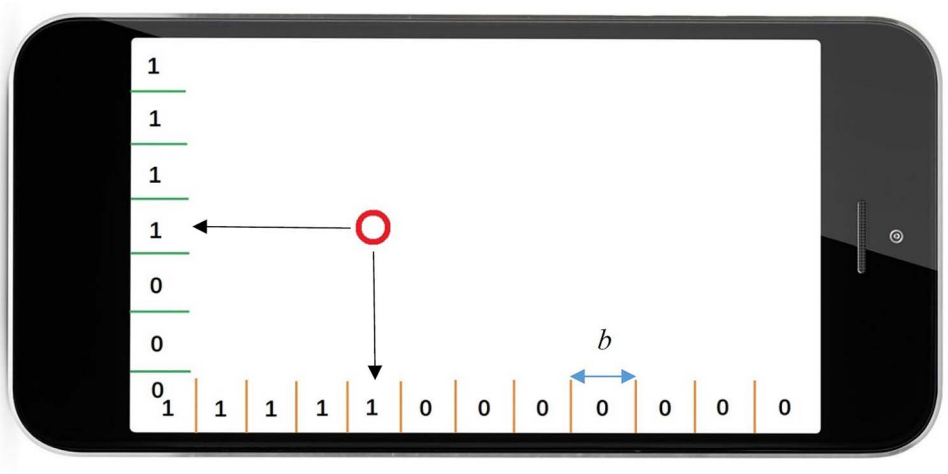

FIGURE 3. In data annotation procedure, the screen was divided into bins in the horizontal direction and the vertical direction.

divides the screen into grids. Denote the width and height of the screen as W and H, respectively, and set the grid width as g. Here, W and H are measured with pixel values. We can select the appropriate g such that W and H can be divided by it. Therefore, we set the grid value of the gaze location as 1 and the others as 0. Finally, this method produces $(W \times H)/g$ grid labels, which may cause a computationally intensive problem in practice or result in a complex model and hence affect the accuracy of the model output.

Considering the drawbacks of regular gridding annotation method, we propose a new annotation method to control the number of labels on the gaze image. We herein propose an annotation method that divides screen into bins instead of grids. Hence, the amount of calculation is reduced significantly, and the predicted gaze location is labeled separately in the horizontal and vertical directions. We divide the screen into horizontal and vertical bins, respectively, and set the bin length equally as b, see Fig. 3. Additionally, we select the appropriate bin length such that W and H can be divided by it. This annotation method produces $(W+H)/b$ bins. Denote their references as $(x_1, x_2, x_3 \ldots x_n)$ and $(y_1, y_2, y_3 \ldots y_m)$, where $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$, $y_j \in \{0, 1\}$, $j = 1, 2, \ldots, m$. In this annotation method, all bin references on the left of gaze location in horizontal direction are labeled as 1. Similarly, all bin references above the gaze location in vertical direction are labeled as 1. The rest bins references are labeled as 0 in horizontal and vertical directions.

Compared with the proposed annotation method, the gridding annotation method yields larger dimension of an annotation vector, producing larger amount of annotation values. The proposed method yields less dimension of an annotation vector but requires more processing steps to obtain the predicted gaze location. With the proposed annotation method, we essentially convert a multi-class problem into a binary-class problem in multiple bins. We herein focus on the new annotation method.

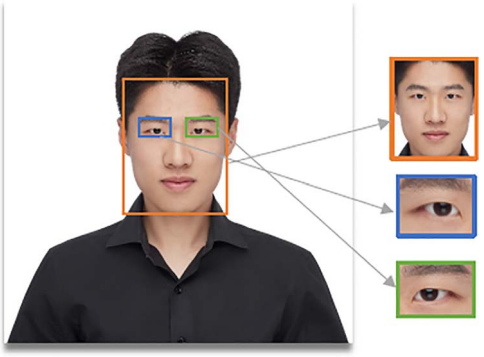

FIGURE 4. The image is preprocessed before training the model, the two eyes and face were detected, and the image was cropped into three parts.

For the training of neural network, we only use the face and eyes patches, so we first pre-process the frames, as illustrated in Fig. 4. For each image, we detected the two eyes and face on the frames employing Haar Cascade Classifiers [30] in Open Source Computer Vision Library (OpenCV) [31] and got the position of top-left corner, width and height of the two eyes boxes and face box. After detecting, we cropped the images of the left eye, right eye and face from the frames according to the box position and width. We resized the box images to the same scale for the convenience of input.

The next two subsections introduce the proposed two-stage procedure of gaze estimation, consisting of using the neural network to output the estimation in two directions and refining the estimation with logistic regression.

2.3. Stage-I: process gaze image using neural network

In this subsection, the neural network model was used to extract the features of eyes and faces and output coordinate probability

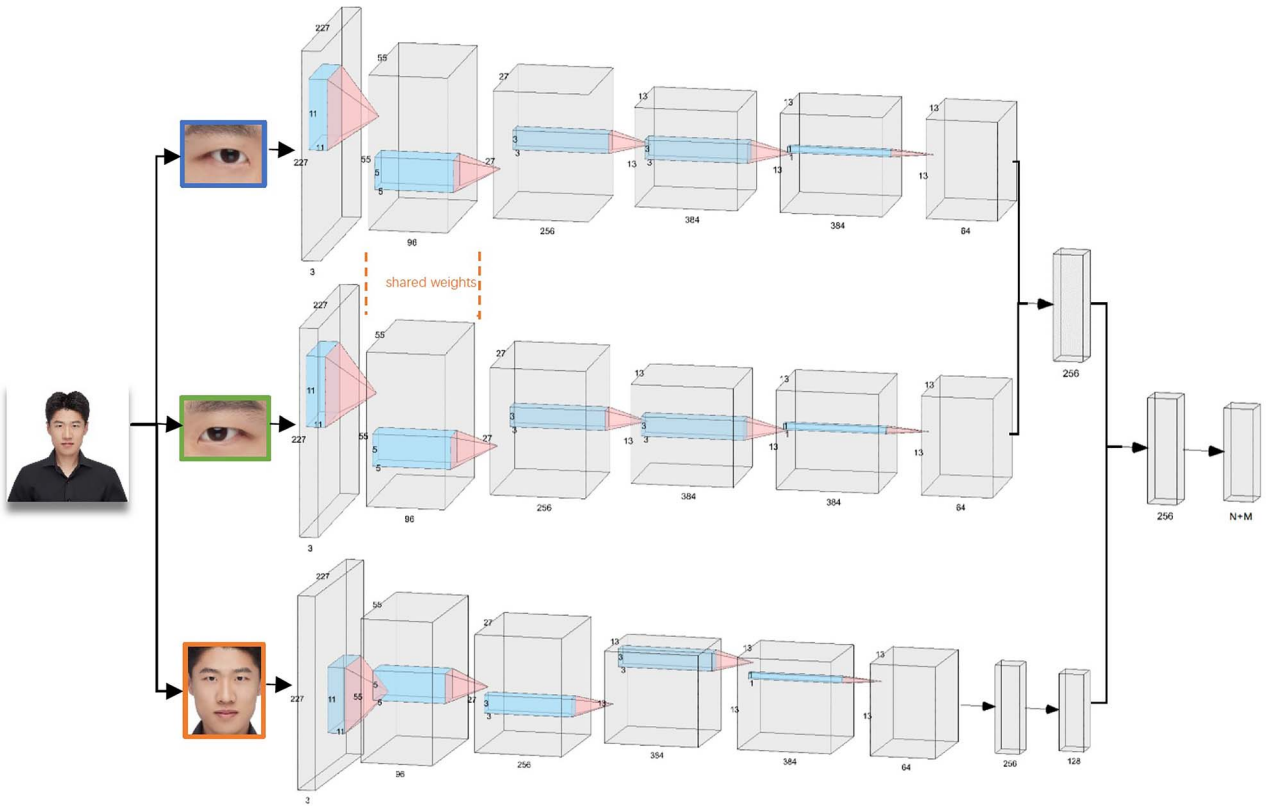

FIGURE 5. The convolutional neural network of the proposed gaze estimation method.

vectors. Frames of the face, left eye and right eye were cropped from the original pictures. Subsequently, these three parts were used as input to the corresponding convolutional layer to extract high-level features. The first two convolutional layers of left eye and right eye shared the same weights. After two convolutional layers and pooling layers, the left eye features and right eye features were independently input into three convolutional layers, respectively. The three convolutional layers were followed by a common fully connected layer. The frame of face was input to five convolutional layers followed by two fully connected layers. Finally, the total features above were joined and then placed into two fully connected layers, as shown in Fig. 5. The inputs contained left eye, right eye and face patches with size 227 $\times$ 227. The size of last two fully connection layers were 128 and $M + N$. For the convolutional layers, each layer contained two hidden layers including batch normalization [32], which have much higher learning rates and are insensitive to initializations in the network and rectified linear units [33], and hence could better learn features for face verification and preserve relative intensities information through multiple layers of feature detectors.

Finally, with a sigmoid function, we obtained vectors output by the neural network. We further divided them into an $M \times 1$ dimensional vector and an $N \times 1$ dimensional vector, respectively, in the horizontal and vertical directions, where M and N are the number of horizontal and vertical bins, respectively. The vector, which is the output from a sigmoid activation function, represents the probability of each bin in which the value is predicted to be 1. Here, we obtain the probability vectors instead of predicted coordinates. It is noteworthy that the traditional neural network outputs a value in a range starting from 0 to the screen width or height if it predicts the coordinates directly, which causes a wide output range compared with that of the proposed model, while our method converts the prediction task into a binary classification problem, hence avoiding such problems as in traditional neural networks.

2.4. Stage-II: fitting network output using modified logistic regression

Based on the neural network output, we used an additional logistic regression for refinement of the estimation results in the second stage. Considering the annotation data of the image is a vector composed of subsequence of 1 and 0 in order, we applied logistic regression with a modified sigmoid function to process the output from stage-I, which corresponds to the activation function of the neural network output layer, i.e. the sigmoid function. The same model was separately applied to the output

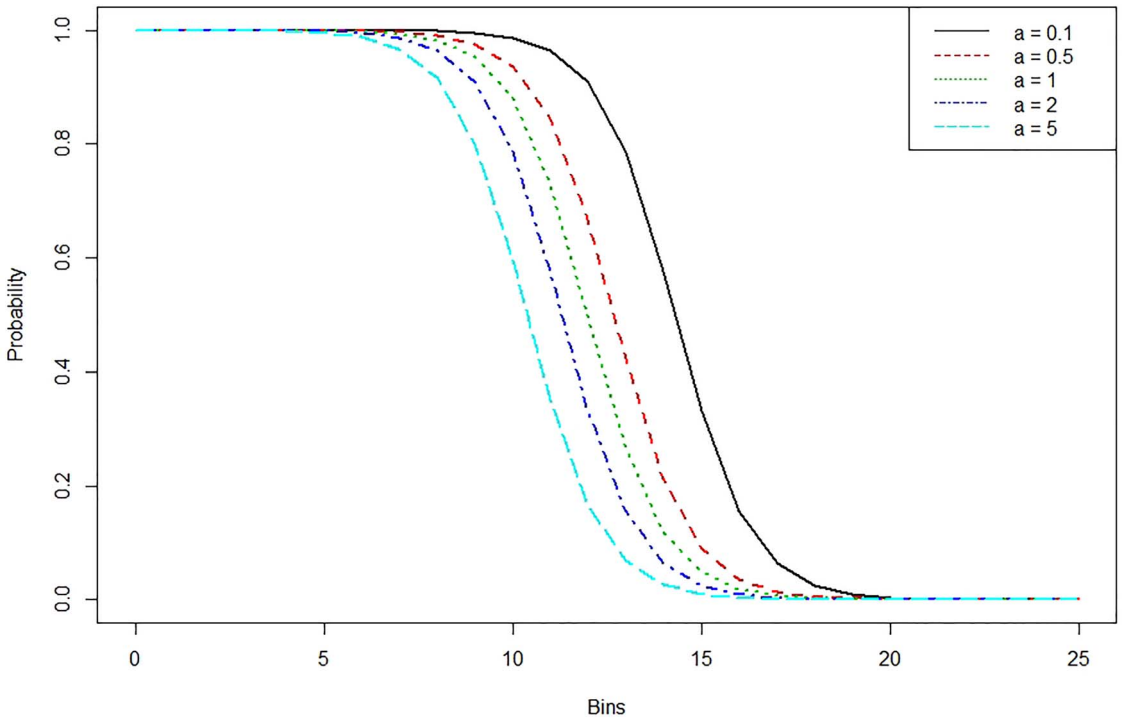

FIGURE 6. The sigmoid function is modified according to the proposed gaze estimation method.

vector in the horizontal and vertical directions. The purpose was to obtain the boundary position of the two subsequences such that the logistic regression model could be used for classification to obtain the target position according to the classified data. We used the vector data of the neural network output from one direction, such as the horizontal direction, as input to fit the sigmoid function, see Fig. 6. The fitting data were a value vector ranking from big to small because the labeled data composed of subsequence of 1 and followed by subsequence of 0. According to the characteristics of fitting data, we modify the sigmoid function as

$$F(x) = \left[1 + a \times \exp\left(x - \bar{x}\right)\right]^{-1}, \tag{1}$$

where $\bar{x} = \sum_{i=1}^{K}\left(x_i p_i\right)$ is the mathematical expectation of the output variable from neural network, p_i represents possibility variable of neural network output and K is the number of variables from one direction. Through $\bar{x}$, we process the variable values by mean normalization. Meanwhile, to match the fitting data with the characteristics of the curve graph, the symbol of variables x in the function was transformed such that the curve flipped left and right. In equation (1), a is a tuning parameter that affects the steepness of the fitting curve but not the output results. Therefore, we can set it freely according to our needs. The method above was applied to the output vector in the horizontal and vertical directions, respectively.

ALGORITHM 1. Find target points in the horizontal direction.

Input: Probability vectors V_k from neural network output.
for $k = 1, 2, \ldots, M$ **do.**
  threshold value: $m \leftarrow (\max\{V_k\} + \min\{V_k\})/2$.
  for $i = 1, 2, \ldots$ **do.**
    if $V_k\ [i] > m$, continue.
    else mutation point: $\mu \leftarrow V_k\ [i]$.
  end
end
output: target point $\leftarrow \mu$.

The process of finding the target points in the horizontal direction is shown in Algorithm 1. We set the median value between the maximum and minimum values of the value range as the threshold value for classification with logistic regression. We iterated through each component of the probability vector and compared it with the threshold value until the point breaking through the threshold value was obtained. We define this point as the mutation point, and target point. The process is similar when finding the target points in the vertical direction.

2.5. Revivification of target point

So far, we have obtained the relative position point using logistic regression. To obtain the actual coordinates of a gaze

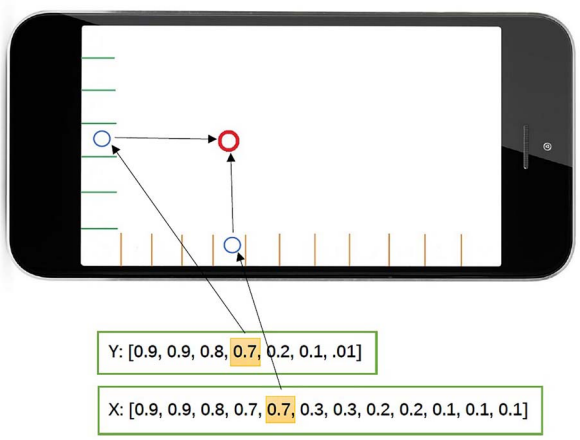

FIGURE 7. The target point is found by revivification procedure.

position on a mobile phone screen, we need to process this relative position point by revivification. Assume that the relative position point indexes i and j of the horizontal and vertical directions are obtained separately. Here, i and j are the references of the horizontal and vertical bins, respectively.

In Fig. 7, the relative points are mapped to the points on the screen. We obtain the actual position coordinates by revivifying the relative position point and calculate the predicted gaze position coordinates using the following equations:

$$T_x = W \times (i - 0.5)\,/M, \tag{2}$$

$$T_y = H \times (j - 0.5)\,/N, \tag{3}$$

where T_x and T_y are the predicted gaze position in horizontal and vertical direction, respectively. Recall that M and N represent the amounts of bins in horizontal and vertical direction, respectively, and W and H represent width and height of the screen.

3. EXPERIMENT

In this section, we introduced the datasets on which our method performed and detailed the settings in our experiment. Then, we showed our results using two annotation methods. Besides, we evaluated the performance of the proposed method on our own dataset and the GazeCapture dataset. We also analyze the prediction errors of our method.

3.1. Setup

For the training data of the gaze model, we divided the gaze dataset into training sets and test sets. The participants were divided into 11 groups: six groups for the training set, two groups for validation and three groups for the test set, as listed in Table 1. The participants were enrolled into groups in sequence so as to balance the gender proportion, wearing glasses proportion of participants for each group. Finally, we obtained 50 participants in each group.

TABLE 1. Dataset divided for gaze estimation neural network train.

Data	Train	Validation	Test
Frames	60 000	20 000	30 000
Participant groups	6	2	3

Also, for comparison, we performed our method on GazeCapture dataset. This dataset was captured with iphone and ipad, which contains 1 490 959 frames from 1471 subjects. The dataset was divided into train, validation and test set. The train, validation and test set consisted of 1271, 50 and 150 subjects, respectively.

The network input contained three parts, including the left eye frames, right eyes frames and face frames. The input frames size was 227 × 227. With our gaze dataset, we set the bin number of horizontal and vertical direction as 111 and 54, respectively. With GazeCapture dataset, we set bin number as 100 both in the horizontal and vertical directions. The model was implemented using Python, Pytorch programming framework and compute unified device architecture (CUDA). In the training procedure, the initial learning rate was 0.001, and stochastic gradient decent optimizer with a momentum of 0.9 and a weight decay of 0.0001 was used. The model training ran on Ubuntu 18.04 with 12 3.2GHz i7–8700 CPUs, 32GB memory, additionally with two GPUs including NVIDIA GeForce RTX 2070 and TESLA K40.

3.2. Results and comparison

We evaluate the accuracy of the proposed method via the error from the ground truth location to the gaze predicted location on the screen. We pre-set the fixed points on the mobile device screen and regard the points location as the ground truth when participants watching the points. We compute the error using the following formula:

$$\text{Error} = \sqrt{(T_x - x_0)^2 + (T_y - y_0)^2}, \tag{4}$$

where (x_0, y_0) represents the ground truth location coordinates, and (T_x, T_y) is the predicted gaze location. To assess the proposed annotation method, we compared its performance with the gridding annotation method. For the gridding annotation method, the logistic regression in stage-II is not required for processing after the neural network, so we can obtain the target point through a neural network directly. We evaluate the maximum, minimum and mean errors in the models using two annotation methods. The results were evaluated in centimeters.

TABLE 2. Prediction error comparison based on two annotation methods using two datasets. (unit: cm.)

Dataset	Max. error		Min. error		Mean error		
	M1	M2	M1	M2	M1	M2	M3
GazeMP	2.23	2.20	1.78	1.69	2.04	1.96	
GazeCapture	2.87	2.56	2.12	1.98	2.42	2.23	2.05
GazeCapture(iphone)	2.52	2.38	1.83	1.74	2.20	2.09	1.86
GazeCapture(ipad)	3.44	3.28	2.71	2.54	3.11	2.96	2.81

Note: GazeMP represents our collected dataset. M1 and M2 represent models with the gridding annotation method and the proposed annotation method, respectively. The max. error and min. error were evaluated by groups. The results of M3 are reported in the GazeCapture paper [24].

Specifically, we denoted the model with the gridding and the proposed annotation methods as M1 and M2, respectively. In Table 2, the first row is the results by M1 and M2 performed on our collected dataset. We evaluated the maximum error and minimum error by participant group. The maximum error and minimum error showed the largest and smallest participant group mean error in all groups. The mean error of M2 was smaller than that of M1. M2 also yields better prediction results in terms of the maximum error and minimum error. The neural network in M2 has fewer output quantities than M1; it reduces the quantity of predicted targets. In addition, M2 used the sigmoid function where a high relevance exists between the variables, while M1 did not. Hence, M2 will undoubtedly improve the prediction accuracy. We checked the participant groups with big error in our dataset. We found that there were more participants wearing glasses in the group with big prediction error. It may be necessary to balance the proportion of participants wearing glasses both in training dataset and test dataset and in all participant group.

We also performed the two annotation methods on GazeCapture dataset [34] for comparison. The results are shown in the second row of Table 2. The mean error of M1 was 2.42 cm. However, the mean error of M2 was 2.23 cm, which was better than M1. Also, maximum error and minimum error of M2 were better than that of M1. Neither M1 or M2 has ideal performance on GazeCapture dataset, compared with the results reported in the GazeCapture paper [24].

For GazeCapture data, the results using proposed annotation method also outperform that by the gridding method on both ipad platform and iphone platform. Specifically, the gaze prediction mean error by M1 and M2 on iphone data was 2.20 and 2.09 cm, respectively, much better than that performed on ipad data. One possible explanation is that the gaze location in our proposed method is predicted to be located in a square area with width equal to bin length, and hence, the error is related to the bin length. Since our predicted gaze location is the center point of the square area and the bins, number we set was equal on the ipad and iphone, so the bin width of ipad is larger than that of iphone, which may cause the larger error on the ipad than that on the iphone.

3.3. Prediction error analysis

The results indicate the gaze prediction errors at different locations, see Fig. 8. According to the error distribution, we observed that a larger error primarily occurred when the points were close to the margin of the screen. The error increased with the distance to the screen center. The result of the center point was not the best of all prediction results. Good predicted results were shown on points close to the center point. Observing the distribution of errors, we discovered a poor gaze prediction on points close to the margin of the screen, which were related to our collected dataset. Because we only collected a small number of frames of people looking at points around the margin of the screen, which might adversely affect the gaze prediction on the point around the margin.

Our gaze estimation method did not perform perfectly in prediction error and the possible explanations are as following. Our collected dataset was collected on the mobile phones with our designed data collecting mechanisms, while the GazeCapture dataset was captured on mixed platforms including iphone and ipad with different screen size. Besides, we have collected high-quality images in our dataset and design the proper annotation method according to our data. When annotating the data, we adjusted the bin width according to the different size of screen on our collected dataset instead of setting the fixed bins numbers on GazeCapture dataset, which may affect the prediction results. In addition, the predicted gaze point is located at the center point in a square area with width equal to the bin length, which makes our method more robust.

In the future, we will include information of facial coordinate position [35], head pose [36, 37] and time sequence information, such as eye optical flow, as the inputs of a neural network to further reduce the prediction error and hence improve the prediction accuracy of the proposed model.

4. CONCLUSIONS

In this article, we herein proposed a two-stage gaze estimation method based on a neural network and logistic regression,

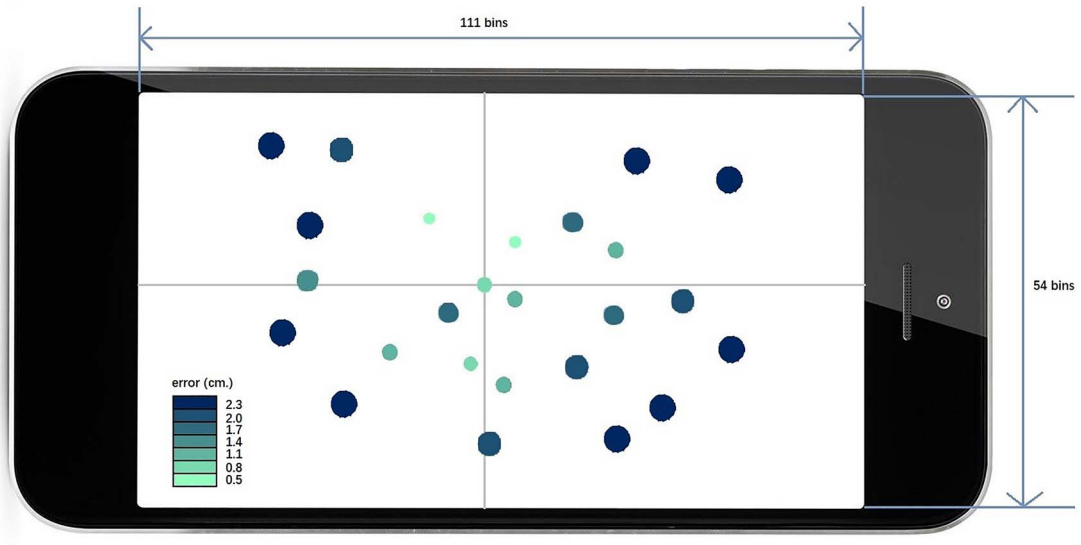

FIGURE 8. The plot of gaze prediction errors at different locations. Bin number of horizontal and vertical direction is set as 111 and 54. Set the center of screen as the coordinate origin. The size of the circle point represents the error at a distance. The bigger the size, the larger is the error.

where the neural network was used to process the input pictures and output predicted probability vectors of gaze labels and the logistic regression was used for refinement of prediction from the neural network. We also designed a dataset collecting mechanisms and built our own dataset. The proposed method could be widely used in various mobile devices without additional hardware or systematic prior knowledge. We demonstrated that the two-stage gaze estimation method combined with a new annotation approach significantly improved the gaze estimation accuracy. Furthermore, by changing the annotation bins, we could adjust the bin length to different accuracy needs for applications.

DATA AVAILABILITY

Currently, the data underlying of this article cannot be shared publicly due to privacy concerns and ongoing study. Once the project is ended, the data will be publicly available, with the ethical authorization, in the near future. The original data should only be used for scientific research purpose. The implementation of the proposed method will also be publicly available on *GitHub*.

FUNDING

National Natural Science Foundation of China (11901013, 12075011); Beijing Natural Science Foundation (1204031, 7202093); Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Hansen, D.W. and Ji, Q. (2010) In the eye of the beholder: a survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**, 478–500.
[2] Sugano, Y., Matsushita, Y. and Sato, Y. (2013) Appearance-based gaze estimation using visual saliency. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**, 329–431.
[3] Sugano, Y., Matsushita, Y. and Sato, Y. (2014) Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1821–1828.
[4] Zhang, X., Sugano, Y., Fritz, M. and Bulling, A. (2015) Appearance-Based Gaze Estimation in the Wild. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4511–4520.
[5] Jacob, R.J.K. and Karn, K.S. (2003) Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. In Hyona, Radach and Deubel (eds.) *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, pp. 573–605, Elsevier Science, Oxford, England.
[6] Farid, M., Murtagh, F. and Starck, J.L. (2002) Computer display control and interaction using eye-gaze. *J. Soc. Inf. Disp.*, **10**, 289–293.
[7] Safaa, A., Sophie, L.B. and Olivier, D. (2018) Face presence and gaze direction in print advertisements. *J. Advert. Res.*, **58**, 443–455.
[8] Ishikawa, T., Baker, S., Matthews, I. and Kanade, T. (2004) Passive Driver Gaze Tracking with Active Appearance Models. In *11th Word Congress on ITS in Nagoya*, pp. 100–109.
[9] Rayner, K. (1998) Eye movements in reading and information processing: 20 years of research. *Psychol. Bull.*, **124**, 372–422.

[10] Jeevitha, D.V., Murthy, L.R.D., Saluja, K.S. and Biswas, P. (2018) Operating different displays in military fast jets using eye gaze tracker. *J. Avia. Tech. Eng.*, **8**, 31–50.
[11] Majaranta, P. and Bulling, A. (2014) Eye Tracking and Eye-Based Human Computer Interaction. In *Advances in Physiological Computing*, pp. 39–65. Springer, London.
[12] Morimoto, C.H. and Mimica, M.R.M. (2005) Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.*, **98**, 4–24.
[13] Chou, R., Dana, T. and Bougatsos, C. (2011) Screening for visual impairment in children ages 1-5 years: update for the USPSTF. *Pediatrics*, **127**, e442–e479.
[14] Jacob, R.J.K. (1990) *What You Look at Is What You Get: Eye Movement-Based Interaction Techniques*. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pp. 11–18.
[15] Stellmach, S. and Dachselt, R. (2012) *Look & Touch: Gaze-Supported Target Acquisition*. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pp. 2981–2990.
[16] Garrido, J.E., Penichet, V.M.R., Lozano, M.D., Quigley, A.J. and Kristensson, P.O. (2014) *AwToolkit: Attention-Aware User Interface Widgets*. In *Proc. of the 2014 International Working Conf. on Advanced Visual Interfaces*.
[17] Xu, P., Sugano, Y. and Bulling, A. (2016) Spatio-Temporal Modeling and Prediction of Visual Attention in Graphical User Interfaces. In *Proc. of the 2016 CHI Conf. Human Factors in Computing Systems*, pp. 3299–3310.
[18] Bulling, A., Alt, F. and Schmidt, A. (2012) *Increasing the Security of Gaze-Based Cued-Recall Graphical Passwords using Saliency Masks*. In Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, pp. 3011–3020.
[19] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) Imagenet classification with deep convolutional neural networks. *Commun. ACM*, **25**, 1097–1105.
[20] Jana, R. and Basu, A. (2017) Automatic Age Estimation from Face Image. In *International Conf. on Innovative Mechanisms for Industry Applications*, pp. 87–90.
[21] Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E. (2018) Deep learning for computer vision: a brief review. *Comput. Intell. Neurosci.*, **2018**, 7068349.
[22] Sewell, W. and Komogortsev, O. (2010) Real-Time Eye Gaze Tracking with an Unmodified Commodity Webcam Employing a Neural Network. In *International Conf. on Human Factors in Computing Systems*, pp. 3739–3744.
[23] Baluja, S. and Pomerleau, D. (1993) Non-Intrusive Gaze Tracking using Artificial Neural Networks. In Cowan J.D., Tesauro, G. and Alspector, J. (eds.) *Advances in Neural Information Processing Systems*, (NIPS) 6. 1994. pp. 753–760. Morgan Kaufmann Publishers, San Francisco, CA.
[24] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W. and Torralba, A. (2016) Eye Tracking for Everyone. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2176–2184.
[25] Huang, Q., Veeraraghavan, A. and Sabharwal, A. (2017) TabletGaze: a dataset and baseline algorithms for unconstrained appearance-based gaze estimation in mobile tablets. *Mach. Vis. Appl.*, **28**, 445–461.
[26] Weidenbacher, U., Layher, G., Strauss, P.M. and Neumann, H. (2007) A Comprehensive Head Pose and Gaze Database. In *IET International Conf. on Intelligent Environments*, pp. 455–458.
[27] Zhang, X., Park, S., Beeler, T., Bradley, D., Tang, S. and Hilliges, O. (2020) ETH-XGaze: A Large Scale Dataset for Gaze Estimation under Extreme Head Pose and Gaze Variation. In *European Conf. on Computer Vision*.
[28] Fischer, T., Chang, H.J. and Demiris, Y. (2018) RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In *The European Conf. Computer Vision (ECCV)*, pp. 334–352.
[29] Liu, N., Han, J., Zhang, D., Wen, S. and Liu, T. (2015) Predicting Eye Fixations using Convolutional Neural Networks. In *2015 IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 362–370.
[30] Viola, P. and Jones, M. (2001) Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 511–518.
[31] Open source computer vision library, http://sourceforge.net/projects/opencvlibrary/.
[32] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the 32nd International Conf. on International Conference on Machine Learning*, pp. 448–456.
[33] Nair, V. and Hinton, G.E. (2010) Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conf. on Machine Learning*, pp. 807–814.
[34] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W. and Torralba, A. (2016) *GazeCapture.* http://gazecapture.csail.mit.edu.
[35] Zhu, Z. and Ji, Q. (2005) Eye Gaze Tracking under Natural Head Movements. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 918–923.
[36] Lu, F., Okabe, T., Sugano, Y. and Sato, Y. (2014) Learning gaze biases with head motion for head pose-free gaze estimation. *Image Vis. Comput.*, **32**, 169–179.
[37] Valenti, R., Sebe, N. and Gevers, T. (2011) Combining head pose and eye location information for gaze estimation. *IEEE Trans. Image Process.*, **21**, 802–815.