FAU-Gaze: Fast and Accurate User-specific Gaze Estimation Framework

Ye Ding[1], Li Lu[2], Ziyuan Liu[2], Songjie Wu[1], Qing Liao[3,✉]
[1]*School of Cyberspace Security, Dongguan University of Technology, Dongguan, China*
[2]*School of Computer Science and Technology, Dongguan University of Technology, Dongguan, China*
[3]*School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China*
dingye@dgut.edu.cn, 2360890221@qq.com, 516807971@qq.com, 1726580741@qq.com, liaoqing@hit.edu.cn

***Abstract*—Gaze estimation has a wide range of applications such as neuroscience and clinical research. In this paper, we propose and implement a fast and accurate user-specific gaze estimation system, called *FAU-Gaze*. FAU-Gaze supports online real-time training with an inference speed of up to 7-11.5 ms in 100 FPS. Compared with existing models, the kernel model FPGC (Feature-based Personalized Gaze Calibrator) of FAU-Gaze increases the accuracy by 36.4% and 33.7% on MPIIFaceGaze and TabletGaze respectively. By mining each user's potential characteristics, FAU-Gaze can more accurately locate each user's real gaze position. In order to test FAU-Gaze, we also introduce a low-resolution and low-definition laptop gaze estimation dataset TobiiGaze containing 41,000 images. Through our experiments on both TobiiGaze, MPIIFaceGaze, and TabletGaze, the prediction error of FAU-Gaze is reduced to 1.61 cm and the robustness outperforms the state-of-the-art.**

***Index Terms*—deep learning, gaze estimation, eye appearance**

I. Introduction

Gaze reveals human mental state and behavioral activities. Gaze estimation has many applications in neuroscience research [1], human-computer interaction [2], assisted driving [3], market and user research [4], psychology research [5], and online education [6]. With the development of deep learning in computer vision, methods based on Convolutional Neural Network (CNN) [7]–[10] make gaze estimation cheaper and faster. Gaze estimation has two categories: 2D gaze estimation [8], [9], [11], [12] and 3D gaze estimation [13], [14]. In this paper, we focus on 2D gaze estimation: the position (x, y) where the gaze falls on the screen.

A general gaze estimation model cannot explicitly distinguish the gaze stance across different people. Hence, it is challenging to construct a fast and precise gaze estimation framework with personalized calibration. More specifically: **1) Lack of data.** In real-life scenarios, images captured by the front camera of mobile devices may have low resolution and definition, different from open datasets [8], [11], [15] with high resolution and definition. **2) Bad calibration method.** Obtaining additional calibration data requires users to look at a fixed position on the screen. [8] requires 13 fixed calibration points, and [12] requires fewer calibration points ($\leq$5). The above methods make users impatient. **3) Low accuracy.** Personalized calibration is critical to improving accuracy. Support Vector Regression (SVR) [8] and few-shot learning [12] are introduced in recent years but can be further improved.

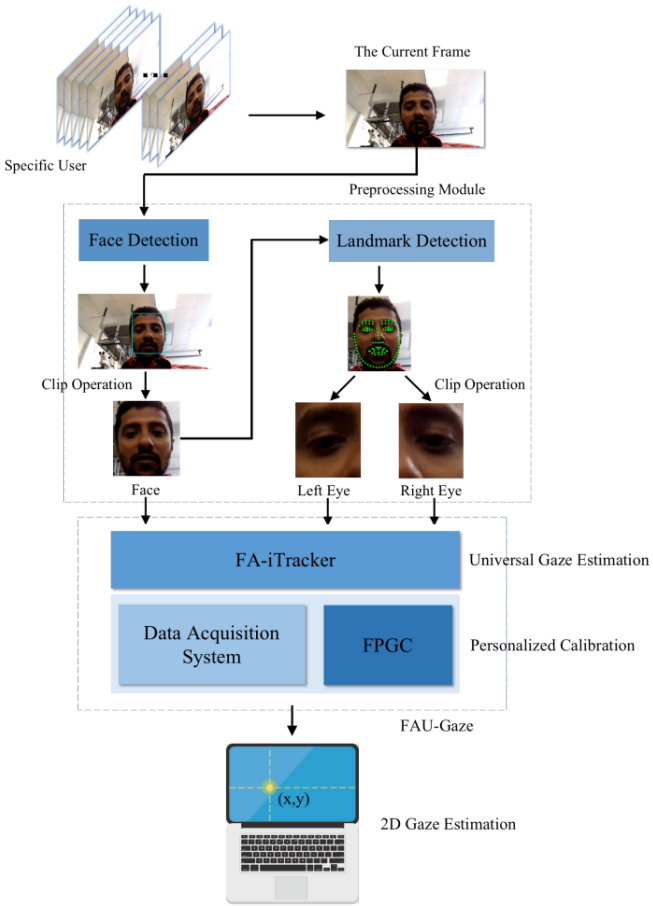


Fig. 1. General working process of FAU-Gaze.

In this paper, we propose a fast and accurate user-specific gaze estimation framework (FAU-Gaze) as shown in Figure 1. The contributions of this paper lie on the following aspects: 1) We use Tobii Pro Fusion to dynamically collect a batch of front camera video data (TobiiGaze) from laptop devices. TobiiGaze contains 20 subjects with 2 minutes of recording time for each subject. Subject is allowed to watch anywhere on the screen, and head posture is not restricted. It is worth mentioning that the resolution and definition of TobiiGaze are more realistic in real-life scenarios. 2) We design a vivid

and flexible method for calibration sample collection. The collected data are cleaned through a multi-rule combination method. 3) We propose an improved universal gaze estimation model FA-iTracker (Fast Accurate iTracker). Compared with previous works, it is more accurate and significantly faster. 4) We propose a personalized gaze calibrator FPGC (Feature-based Personalized Gaze Calibrator), which is independent and supports most general training models with real-time online training. Working with FA-iTracker, FPGC increases the accuracy by 36.4% and 33.7% on MPIIFaceGaze and TabletGaze respectively, and the prediction error on TobiiGaze is only 1.61 cm.

II. Related Work

The 2D gaze estimation function is specifically manifested as using the gaze estimation algorithm to estimate the focus of user's binocular gaze in real-time, that is, the gaze point of the user's eyes on the current two-dimensional plane. This two-dimensional plane can be a mobile phone screen, a pad screen, a laptop screen, and TV screen, etc.

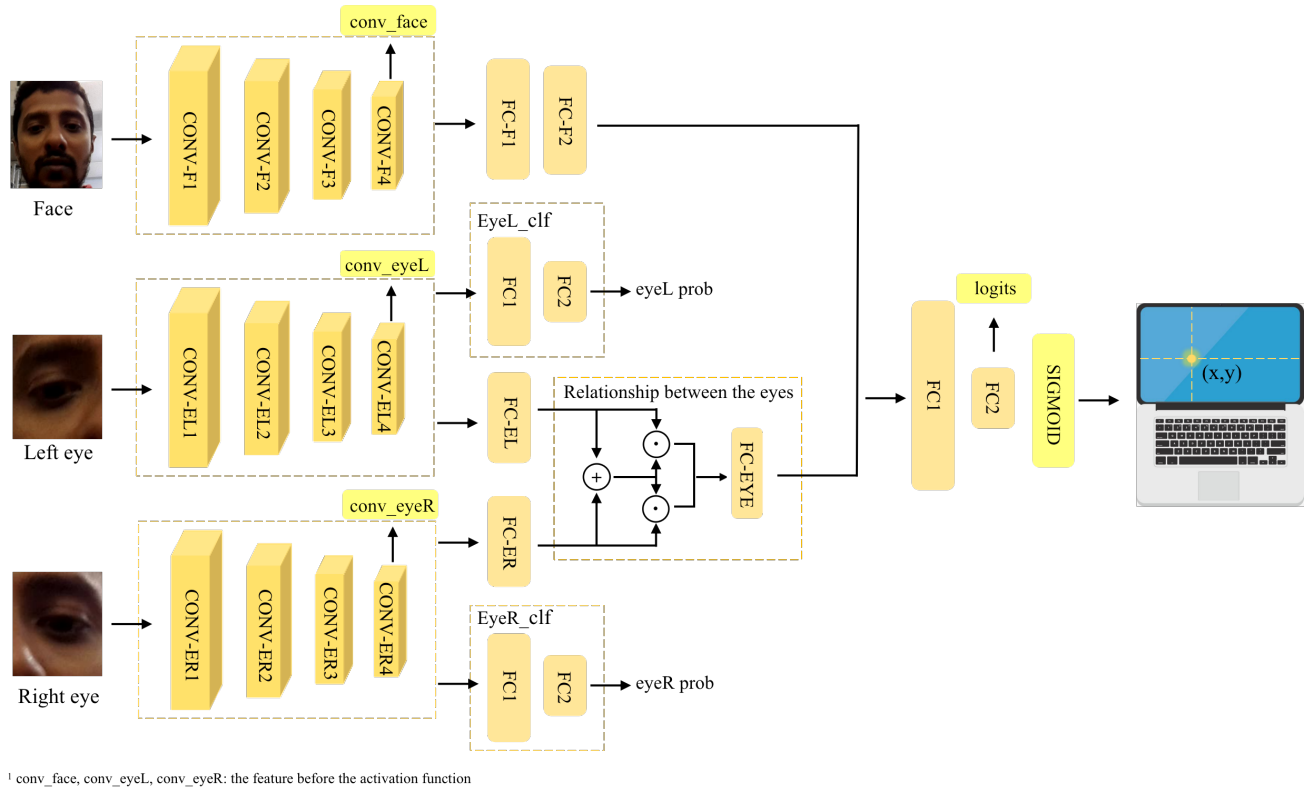
Appearance-based gaze estimation. Gaze estimation methods can be divided into model-based and appearance-based methods [16]. Model-based methods use external light sources to detect the characteristics of the eyes [17], [18], or rely on the establishment of a geometric model of the eye area [19]–[21]. The appearance-based method [8], [9], [11], [12] takes the image captured by the camera directly as input to track the user's gaze. It will not be affected by the image resolution and lighting conditions but needs more specific user training data [7]. Qiong Huang et al. [11] mainly studied gaze estimation on a tablet and proposed a tablet gaze algorithm based on multi-level HoG features and random forest regression. Krafka et al. [8] proposed iTracker. The model has four image inputs: left-eye image, right-eye image, face image, and face grid. In addition, they also collected and released a large public 2D gaze dataset Gaze Capture. Xucong Zhang et al. [9] believe that the other areas of the face except the eyes hide important information that assists the gaze estimation problem. They used the complete face image as the input of the neural network, added the spatial weighting mechanism to the classic CNN network structure, and achieved good results. Junfeng He et al. [12] improved iTracker and proposed a SAGE model structure. The model loses the input of the face grid and only relies on the eye image and eye region landmark, and the left eye image is mirrored and flipped horizontally to make it easier to share weights. Compared with iTracker, the model greatly improves the inference speed.

Personalization calibration method. Among the appearance-based methods, some studies [8], [12], [22]–[25] have proved that personalized calibration methods can further improve the accuracy of gaze estimation. The personalized calibration methods are suitable for both 2D and 3D gaze evaluation. Krafka et al. [8] used calibration as a post-processing step in iTracker, using a simple SVR model to train the calibration samples of a specific person while keeping the weights of the rest of the network unchanged. Finally, the dimension of the input SVR model is 128. The accuracy of the model is higher when 13 correction points are used, but the performance is poor when only 4 correction points are used. Junfeng He et al. [12] proposed a supervised and personalized method using a small number of labeled calibration points ($\leq$5), and also proposed an unsupervised method based on a heterogeneous teacher-student network with a small number of users unlabelled, and the embedding-based few-sample learning method is trained to improve the accuracy of gaze estimation. And each user only needs 2-5 calibration points. The few-shot method is also used in [26], and additional training samples are generated by synthesizing the eye image of the gaze redirection from the existing reference samples, thereby improving the adaptive ability of gaze estimation. Park et al. [22] used an encoder-decoder structure to learn a latent representation composed of appearance, gaze, and head pose and used a meta-learning algorithm (MAML) [27] to train a gaze estimator for a small number of specific populations. This allows the model to be better generalized to new personnel. And the model needs calibration points ($<$9) to be well adapted. Liu et al. [23] used the bias elimination method to achieve personalization, using a differential neural network to estimate the difference in the gaze direction of the two images. Only in this way can it be guaranteed that the subtraction operation can eliminate the deviation. During the test phase, a small number of calibration points ($\leq$9) are required.

Considering that the gaze deviation is related to people, the Tobii team proposed in their 2019 paper [24] to use the ID information of the sample to learn the deviation in training. The essence of this method is calibration parameters, and the specific implementation is that assign a 6-dimensional parameter vector to each person as the calibration parameters. Yunyang Xiong of the University of Wisconsin-Madison also proposed a similar idea in the CVPR 2019 paper [25]. They decompose the gaze estimation into a fixed component and a random component related to people.

Gaze dataset. We introduce several existing representative 2D gaze estimation datasets: 1) The MPIIFaceGaze dataset [15] collected 15 subjects in the real environment of a laptop, and its features such as illumination and eyes have significant diversity. 2) The TabletGaze dataset [11] is characterized by significant head posture changes. This dataset records the video taken by the front camera when 51 subjects are holding a tablet. Each subject has 4 different postures and provides a total of 35 gaze points. 3) The advantage of GazeCapture dataset [8] over MPIIFaceGaze and TabletGaze is that the number of participants is larger, with a total of 1474 participants. The dataset provides 13 fixed point positions (according to the device direction) and the use of crowd-sourcing methods overcomes the high cost and lack of data changes.

The above datasets all have high image resolution, and they all give subjects a fixed gaze point. However, in actual situations, the user's gaze may be non-fixed, and the front camera of the device may have a shooting effect. It is vague,



[1] conv_face, conv_eyeL, conv_eyeR: the feature before the activation function

Fig. 2. The network structure of FA-iTracker.

and there is no such data as a training set, so we used Tobii Pro Fusion to collect a batch of video data (TobiiGaze) on HP Windows 10. The front camera of the device has a low resolution and low definition. We hope that when it is used for gaze estimation in real situations, it will not be significantly different from the performance on the dataset.

III. FAU-Gaze

The general working process of FAU-Gaze is shown in Fig. 1. FAU-Gaze preprocesses the data before training through face detection and landmark detection. The input of preprocessing is the original image of the current frame, and the output is the face image, the left eye image, and the right eye image, and they are used as the three inputs for the next stage of gaze estimation.

A. FA-iTracker

Fig. 2 illustrates the model structure of FA-iTracker. FA-iTracker has three inputs: face image, left eye image, and right eye image. We use three identical and independent CNNs to extract features from the three inputs respectively. FA-iTracker is an improvement of iTracker [8] with faster speed and higher accuracy. The improvements include: **1) Relationship between eyes.** Compared with iTracker [8] and SAGE [12], we learn the hidden features of the two eyes more deeply, because eye information always estimates the gaze most importantly. We give priority to the independent study of the two eyes and consider the relationship between them, and then learn together. **2) Two categories of eyes.** We also embed an eye classifier layer (EyeL_clf, EyeR_clf) into the eye network, and use the eye class model to determine whether the input eye image (left eye image, right eye image) is positive or negative: positive means that the image is judged to be an eye picture, and the negative category means that the image is judged to be a non-eye picture. The final output probability of two eyes classification (0-1): eyeL prob, eyeR prob. These two probabilities will not have any impact on FA-iTracker but will be used as input for the next stage of personalized calibration. **3) Output normalization.** We pay more attention to the position where the gaze falls on the screen, and we don't even need to know the position beyond the screen size. Therefore, the output of the FC2 layer of the model is normalized to the range of 0-1 using the sigmoid function, and then the real gaze position (x, y) is calculated according to the real screen size.

The model parameters of FA-iTracker are much smaller than iTracker [8]. The size of the three inputs of the model is 224×224, and the filter size / number of kernels of the convolutional layers are:

- `CONV-F1`,`CONV-EL1`,`CONV-ER1`: 11×11/32
- `CONV-F2`,`CONV-EL2`,`CONV-ER2`: 5×5/48
- `CONV-F3`,`CONV-EL3`,`CONV-ER3`: 3×3/96
- `CONV-F4`,`CONV-EL4`,`CONV-ER4`: 1×1/16

The sizes of fully-connected layers are: `FC-F1`: 64, `FC-F2`: 32, `FC-EL`: 64, `FC-ER`: 64, `FC-EYE`: 64, `EyeL_clf-FC1`, `EyeR_clf-FC1`: 32, `EyeL_clf-FC2` / `EyeR_clf-FC2`:

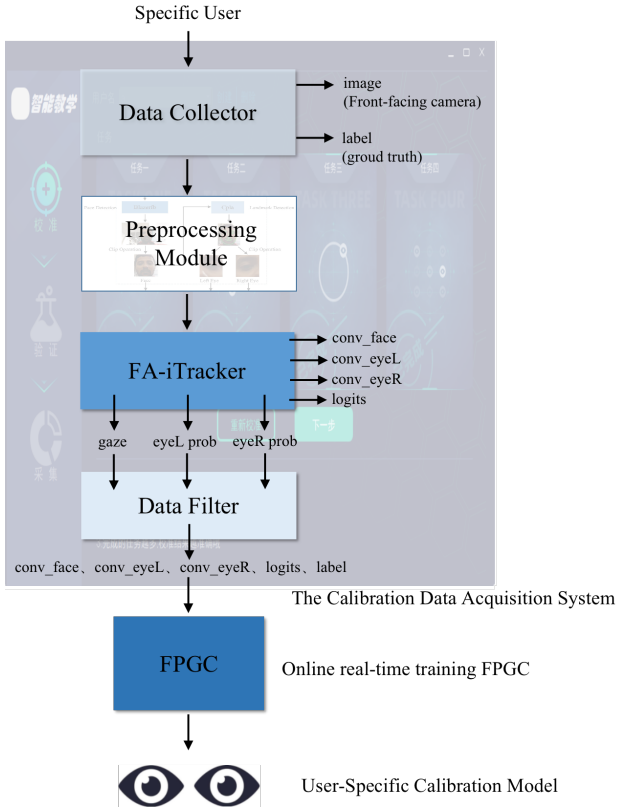


Fig. 3. The personalization calibration of FAU-Gaze.

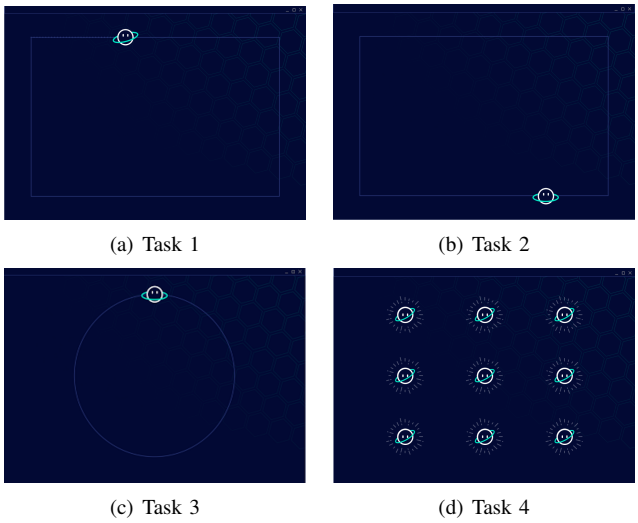
2, `FC1`: 128, `FC2`: 2.

FA-iTracker is an end-to-end CNN-based model. But because of the need to integrate the two-class structure of the eye, during training, we give priority to training the gaze estimator, and specify the node name of the four important features (corresponding to the Figure 2) during the training process conv_face: 7×7×16, conv_eyeL: 7×7×16, conv_eyeR: 7×7×16, logits:2, so that it can be used directly in the subsequent calibration process. Waiting for the training to be completed before proceeding to the eye class training. At this time, only the EyeL_clf and EyeR_clf layers are trained, and the remaining network weights are fixed. In the end, our model can realize gaze estimation and eye estimation classification without any influence on each other.

B. Personalization Calibration

The personalized calibration part of FAU-Gaze includes a calibration sample data acquisition system and a calibrator FPGC. When the user visits for the first time or performs calibration as needed, FAU-Gaze will perform the calibration steps, as shown in Figure 3.

First, we use the calibration sample collector to collect user calibration samples and then use the preprocessing module to process the data and input the results into FA-iTracker to get the sample feature conv_face, conv_eyeL, conv_eyeR, logits, and three output results gaze(x,y), left eye probability(eyeL prob), right eye probability(eyeR prob). At this time, the three output results are input to the calibration sample filter, and after the data is cleaned in a multi-rule combination method, the final effective calibration sample features conv_face, conv_eyeL, conv_eyeR, logits are retained.



(a) Task 1 (b) Task 2

(c) Task 3 (d) Task 4

Fig. 4. Tasks of the calibration sample collector.

After obtaining effective calibration samples, we use the sample features and label (ground truth) to train a user-specific personalized calibrator online, and the final output gaze position (x, y) is close to the user's real gaze position.

1) Acquisition of Sample Data: Below we will separately explain the specific implementation process of the calibration sample collector and the calibration sample filter.

Calibration sample collector: We designed four different calibration tasks, including static and dynamic. The static method is similar to [8], [12], while the dynamic method [28] refers to moving the calibration point on the screen, and at the same time, dynamically collecting the user's current gaze state when the user's gaze is required to follow the target movement track. This method is more vivid and interesting, and can also capture the user's dynamic information. As is shown in Figure 4, The following four tasks are introduced separately: **1) Rectangular task 1.** The movement direction of the small planet is clockwise, starting from the upper left corner, moving around the rectangle at a uniform speed. The user only needs to follow the moving planet closely. **2) Rectangular task 2.** Same as task 1, but the starting point of the movement of the small planet is the lower right corner. The user only needs to follow the moving planet closely. This is because the data collected in task 1 may be invalidated by the data filtering rules at some edges, and taking another corner point as the starting point of the movement can ensure that relatively complete edge point data can be collected. **3) Circular task.** The movement direction of the small planet is clockwise, and it moves in a circle at a uniform speed. The user only needs to follow the moving planet closely. This task can collect some non-edge data. **4) Timed task.** Small planets will appear at 9 designated points. When one point is lit, no other points will appear. At

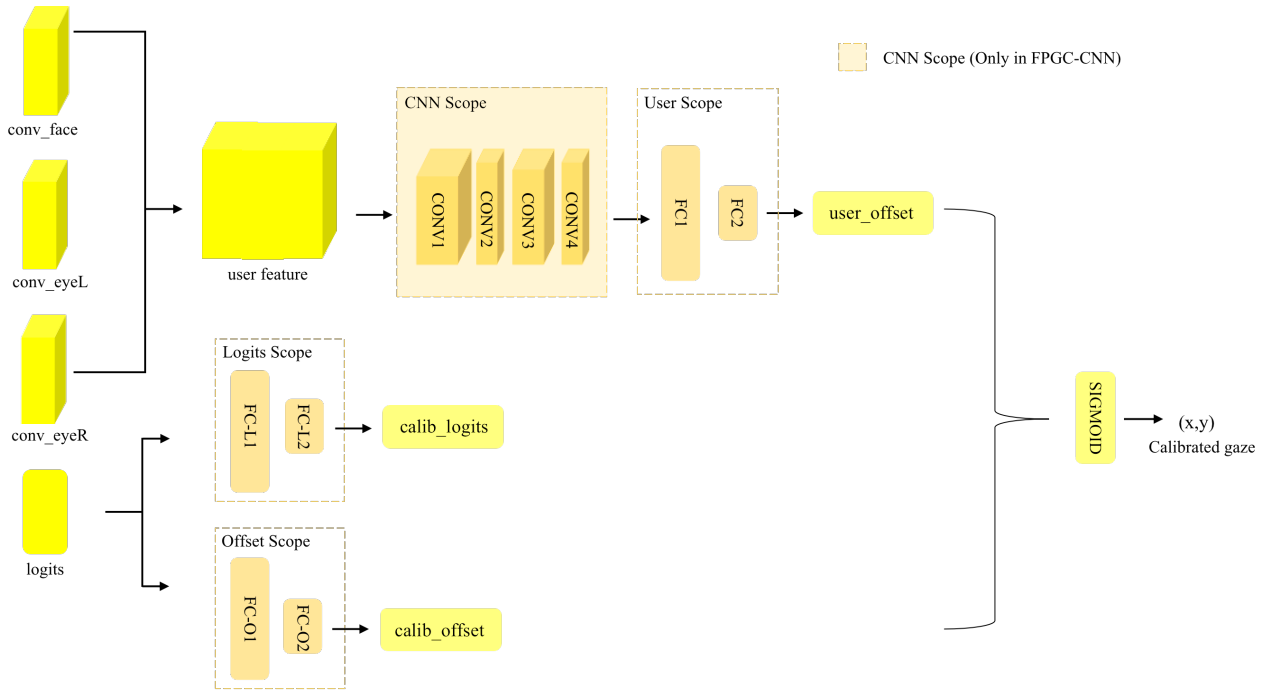


Fig. 5. The FPGC network architecture.

each location, the small planet will light up for 3-5 seconds. This is to collect some corner points.

Calibration sample filter: The key to filtering is to determine whether the user is paying attention to the moving target on the screen and to filter out valid data for the next stage of correction training. We will adopt a multi-rule joint data cleaning method. The four rules are introduced below: 1) Face detection + facial landmark detection + FA-iTracker: judge whether a pair of valid eyes can be detected in the current frame, the threshold of eye classification probability is 0.35, and discard samples with a threshold value of less than 0.35. 2) Refer to [28] to calculate the correlation coefficient between the result sequence of FA-iTracker inference and the target point sequence. We use 1-2 seconds as the time sliding window length. If the calculated correlation coefficient is greater than 0.3, the collected sample sequence is considered to be a valid sequence. 3) Specify a distance threshold of 8 cm, if the euclidean distance between FA-iTracker's inference result and the real target is greater than 8 cm, the frame sample is considered invalid. 4) Discard fuzzy samples. For fixed-point tasks, we discard samples that are about 0.5-1 seconds in the head movement transition phase to avoid introducing misjudgments. Because it takes some time for people to react when the highlights are switched. For other tasks, refer to [28], discard the samples of the first 0.8 seconds and the last 0.2 seconds of the task.

2) FPGC Model Architecture: The calibration process in [8] only adjusts the weight of the last layer of the general model, which will lose the previous user characteristic information. We propose a feature-based personalized calibrator FPGC. The features here include face features (conv_face), left eye features (conv_eyeL), and right eye features (conv_eyeR). They can not only identify each user but also have uniqueness. Adding these features to the calibration process can make better use of specific information about the user's face and eyes and provide more help for gaze correction.

Our calibration model is separated from the general model, which makes the calibration process more room to play. It is based on CNN. It is simple, fast, and supports end-to-end training. In the process of implementing FPGC, we conducted many experiments. First, we proposed the V1 version named FPGC-FC, which used some fully connected layers and achieved good calibration results. Then considering that the user's feature information is presented in the form of images, and CNN is more conducive to extracting user features, so we then propose the V2 version. Compared with the V1 version, it only adds a few layers of CNN, and we named it FPGC-CNN, which further improves the accuracy of the personalized calibration. The two versions are described in detail below.

FPGC-FC: The model structure is shown in Figure 5. First, directly merge the user feature information (face feature, left eye feature, right eye feature), and then use the full connection (User Scope) to predict the user's specific offset (user_offset). Of course then logits features are input into two different fully connected layers (Logits Scope, Offset Scope), where Logits Scope predicts the correction result of the gaze (calib_logits), which corrects the general model's gaze output logits, and Offset Scope predicts the correction of the gaze offset (calib_off), it predicts an error offset. Using two correctors has a more powerful correction ability. Parameter settings: `FC1`: 1024, `FC2`: 2, `FC-L1`: 2048, `FC-L2`: 2, `FC-O1`: 2048, `FC-O2`: 2.

FPGC-CNN: After merging the user feature information,

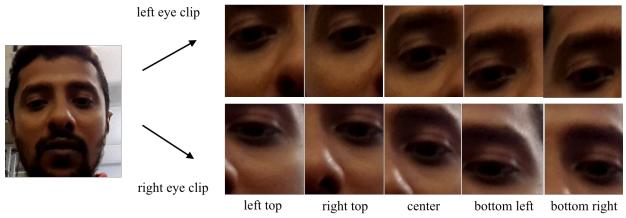


Fig. 6. Eye image five-fold data enhancement.

first use CNN to extract deeper features, and then use the same structure as FPGC-FC, as shown in Figure 5 after adding CNN Scope. Parameter settings: `CONV1`: 5×5/256, `CONV2`: 3×3/16, `CONV3`: 3×3/128, `CONV4`: 3×3/16.

IV. Evaluation

A. Evaluation Setup

1) Data Preparation: We use three datasets in this paper: MPIIFaceGaze, TabletGaze, and TobiiGaze. MPIIFaceGaze has a total of 15 subjects, containing about 36,000 images, and uses laptops of various screen sizes. Among the 51 subjects of TabletGaze, there are video recordings of 40 available subjects. We parsed the videos according to 35 gaze points. The screen size of the tablet used is (22.62 cm, 14.14 cm). TobiiGaze included 20 subjects, and each subject recorded a video for 2 minutes. A total of 41,000 images were produced after processing. The screen size of the HP Windows 10 device used was (29.4 cm, 16.5 cm). In order to make full use of the data set, we separated the left eye from the right eye in the face pictures, to enhance the data we make five-fold eye images respectively, as shown in Figure 6.

2) Evaluation Metric: In order to evaluate our model accurately and fairly, we use the euclidean distance evaluation index to measure the error between the ground-truth and the estimated position of gaze. Taking into account the differences in screen size and using the distance between phones, tablets, and laptops, we have shown the evaluation results of two different devices, tablets, and laptops, including the evaluation results of the uncalibrated model and the final evaluation results after calibration.

3) Implementation Details: We use TensorFlow to implement all CNN models. When training FA-iTracker, the batch size is 144, the Adam optimizer is used for 100,000 iterations, the initial learning rate is 0.001, and after every 20K iterations, the learning rate decay strategy is adopted, and the decay rate is 0.1. When training FPGC online, we use the min batch training method with a batch size of 64. The initial learning rate is 0.0001, and after about 300-500 iterations, the training is completed.

B. Evaluation Result

1) MPIIFaceGaze: First, we randomly divided 15 subjects into 14 for training and 1 for testing. It can be seen from Table I that FA-iTracker reduces the average error from 4.57 cm (iTracker) and 4.20 cm (Full-face) to 4.02 cm. After FPGC-FC calibration (9 calibration points), it reduces to 2.51 cm. After

TABLE I
Evaluation Results on the MPIIFaceGaze Dataset

Model		# of pts	Laptop (ME in cm)
iTracker [8]		0	4.57
Full-Face [9]		0	4.20
FAU-Gaze	FA-iTracker	0	**4.02**
	FA-iTracker+ FPGC-FC	3	3.58
		5	2.92
		9	**2.51**
	FA-iTracker+ FPGC-CNN	3	3.17
		5	2.57
		9	**2.22 (-36.4%)**

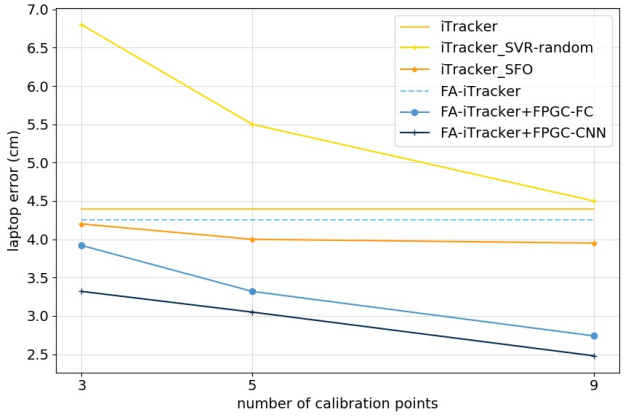


Fig. 7. Comparison of FA-iTracker and iTracker.

FPGC-CNN calibration (9 calibration points), it is greatly reduced to 2.22 cm. And we give the accuracy change process under a different number of calibration points: FPGC-FC drops from 3.58 m (3 calibration points) to 2.92 cm (9 calibration points) and then down to 2.51 cm (13 calibration points), FPGC-CNN dropped from 3.17m (3 calibration points) to 2.57 cm (9 calibration points) and then down to 2.22 cm (13 calibration points).

Then, we divided 15 subjects into 10 for training and 5 for testing. Figure 7 shows the experimental results. It can be seen intuitively that the average error of FA-iTracker is slightly lower than that of iTracker, while the error after calibration by FPGC-CNN significantly lower than iTracker-SVR-random [8], iTracker-SFO [12], finally reduced to 2.48 cm (9 calibration points).

2) TabletGaze: Table II shows the experimental results of dividing 40 subjects into 32 for training and 8 for testing on TabletGaze. It can be seen that for the uncalibrated model, FA-iTracker is the best performer. It changes the average error from 3.63 cm (MPIIGaze), 3.17 cm (TabletGaze), and 3.09 (iTracker) are reduced to 2.91 cm. For the calibrated model, 13 calibration points are also used. FPGC-FC and FPGC-CNN reduce the average error from 2.58 cm (iTracker-SVR) to 2.01 cm and 1.71 cm. And the error performance of FPGC-FC and

TABLE II
EVALUATION RESULTS ON THE TABLETGAZE DATASET

Model		#of pts	TabletGaze (ME in cm)
MPIIGaze [7]		0	3.63
TabletGaze [11]		0	3.17
iTracker [8]		0	3.09
iTracker-SVR [8]		13	2.58
FAU-Gaze	FA-iTracker	0	**2.91**
	FA-iTracker+ FPGC-FC	3	2.87
		5	2.46
		9	2.26
		13	**2.01**
	FA-iTracker+ FPGC-CNN	3	2.77
		5	2.02
		9	1.85
		13	**1.71 (-33.7%)**

TABLE III
EVALUATION RESULTS ON THE TOBIIGAZE DATASET

Model	TobiiGaze (ME in cm)
Full-Face [9]	4.93
TabletGaze [11]	4.18
MPIIGaze [7]	4.02
SAGE [12]	4.36
iTracker [8]	3.94
FA-iTracker	**3.83**

TABLE IV
COMPARISON OF FPGC AND CALIBRATION MODELS ON TOBIIGAZE

Model		#of pts	TobiiGaze (ME in cm)
iTracker [8]	iTracker-SVR	9	2.81
		13	2.71
	iTracker-SFO	9	2.77
		13	2.46
	iTracker+ FPGC-FC	9	**2.04**
		13	**1.79**
	iTracker+ FPGC-CNN	9	**1.73**
		13	**1.58**
SAGE [12]	SAGE-SVR	9	3.13
		13	3.02
	SAGE-SFO	9	3.02
		13	2.55
	SAGE+ FPGC-FC	9	**2.04**
		13	**1.88**
	SAGE+ FPGC-CNN	9	**2.01**
		13	**1.80**
FAU-Gaze	FA-iTracker+ SVR	9	2.98
		13	2.96
	FA-iTracker+ SFO	9	2.67
		13	2.39
	FA-iTracker+ FPGC-FC	9	**2.09**
		13	**1.76**
	FA-iTracker+ FPGC-CNN	9	**1.76**
		13	**1.61**

FPGC-CNN when the number of calibration points is 3, 5, 9, and 13 are respectively given in the table.

3) TobiiGaze: We divided its 20 subjects into 18 for training and 2 for testing. These 18 will be jointly trained with MPIIFaceGaze and TabletGaze, and will be tested on the remaining two. We respectively give the comparison results of the uncalibrated model and the calibrated model. Table III shows the evaluation results of FA-iTracker and other uncalibrated models on the TobiiGaze dataset. It can be seen that FA-iTracker performs best, with an average error of 3.83 cm. However, these uncalibrated models cannot meet the actual needs of TobiiGaze, which explains the impact of errors caused by low resolution and low definition in the actual situation, and once again proves the importance of personalized calibration. Table IV shows the evaluation results of the FPGC and other calibration models SVR [8], SFO [12] on the TobiiGaze data set at 9 and 13 calibration points respectively.

We performed calibration experiments on three uncalibrated models of iTracker, SAGE, and FA-iTracker. The experimental results show that SVR has a limited ability to improve accuracy, and its performance is unstable. The error of the FA-iTracker model after SVR calibration can only be reduced to 2.98 (9 calibration points) and 2.96 (13 calibration points). SFO's performance is slightly better than SVR, but when the accuracy of the uncalibrated model is low, it can't play a big role. In the SAGE model, SFO only reduces the error to 3.02 (9 calibration points), and 2.55 (13 calibration points).

FPGC can provide higher performance for any uncalibrated model, even in the case of low accuracy of the uncalibrated model, it can also reduce the error to less than 2 cm. In the iTracker model, FPGC-FC reduces the average error to 2.04 cm (9 calibration points), 1.79 cm (13 calibration points), and FPGC-CNN further reduces the average error to 1.73 ms (9 calibration points), 1.58 cm (13 calibration points). In the SAGE model, FPGC-FC reduces the average error to 2.04 cm (9 calibration points), 1.88 cm (13 calibration points), FPGC-CNN reduces the average error to 2.01 cm (9 calibration points), 1.80 cm (13 calibration points). In the FA-iTracker model, the error is reduced to 2.09 cm (9 calibration points) and 2.00 cm (13 calibration points) after correction by FPGC-FC, after correction by FPGC-CNN, better results have been achieved, and the error was reduced to 1.76 cm (9 calibration points), 1.61 cm (13 calibration points). Experimental results prove that FPGC is independent of the uncalibrated model, and it can provide higher performance for any calibrated model.

Under normal circumstances, the effect of 13 calibration points is the best, which shows that the personalized calibration process based on fine-tuning is still data-driven, its quantity and quality are very important, so it requires a more cumbersome data collection process. We compared the effects

TABLE V
INFERENCE TIME OF FAU-GAZE ON CPU

FAU-Gaze		Inference time on CPU (ms)
FA-iTracker		7-8
FPGC	FPGC-FC	<1
	FPGC-CNN	3-3.5
Total		7-11.5

of different calibration tasks on the results, using 9 and 13 calibration points as the standard: (1) Among the four tasks, any execution of one of them can achieve the accuracy of 9 calibration points. (2) If any two tasks are performed, the accuracy of 13 static calibration points can be achieved, and almost has reached the upper limit of calibration. Therefore, our data collection system solves this problem to a large extent. While facilitating the collection of more data, it also brings a better experience for users.

C. Run-time Performance

FAU-Gaze framework satisfies real-time requirements on CPU devices, Table V shows the inference speed of each part of the FAU-Gaze frameworks on CPU. It can be seen that it is a fast framework. The online training times of FPGC-FC are 5 s and the inference speed is less than 1 ms, while the online training times of FPGC-CNN are 30-40 s and the inference speed is 3-3.5 ms.

V. CONCLUSION

In this paper, we propose a lightweight and robust gaze estimation framework FAU-Gaze. In particular, we focus more on the personalization problem and propose a CNN-based end-to-end personalized calibration model FPGC, which effectively eliminates the user's gaze deviation to a large extent. The average error on the MPIIFaceGaze and TabletGaze datasets is reduced to 2.22 cm and 1.71 cm respectively, and the real-time inference time is 7 ms-11.5 ms. In addition, the TobiiGaze collected by us solves the problem of low-resolution and low-definition of the front camera in actual situations. In the end, FAU-Gaze's prediction error on TobiiGaze is reduced to 1.61 cm. We believe that FAU-Gaze makes it possible to popularize gaze estimation.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China under grant no. U19A2067 and 61976051.

REFERENCES

[1] J. R. Bardeen and T. A. Daniel, "An eye-tracking examination of emotion regulation, attentional bias, and pupillary response to threat stimuli," *Cognitive Therapy and Research*, vol. 41, pp. 853–866, 2017.

[2] M. Zhao, H. Gao, W. Wang, and J. Qu, "Research on human-computer interaction intention recognition based on eeg and eye movement," *IEEE Access*, vol. 8, pp. 145 824–145 832, 2020.

[3] S. Alletto, A. Palazzi, F. Solera, S. Calderara, and R. Cucchiara, "Dr (eye) ve: a dataset for attention-based tasks with applications to autonomous and assisted driving," in *CVPR Workshops*, 2016, pp. 54–60.

[4] J. Mou and D. Shin, "Effects of social popularity and time scarcity on online consumer behaviour regarding smart healthcare products: An eye-tracking approach," *Computers in Human Behavior*, vol. 78, pp. 74–89, 2018.

[5] Z. Ma, S. Vickers, H. Istance, S. Ackland, X. Zhao, and W. Wang, "What were we all looking at? identifying objects of collective visual attention," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 3, pp. 547–560, 2016.

[6] O. Navarro, A. I. Molina, M. Lacruz, and M. Ortega, "Evaluation of multimedia educational materials using eye tracking," *Procedia-Social and Behavioral Sciences*, vol. 197, pp. 2236–2243, 2015.

[7] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *CVPR*, 2015, pp. 4511–4520.

[8] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *CVPR*, 2016, pp. 2176–2184.

[9] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *CVPR Workshops*, 2017, pp. 51–60.

[10] X. Zhang, M. X. Huang, Y. Sugano, and A. Bulling, "Training person-specific gaze estimators from user interactions with multiple devices," in *UIST*, 2018, pp. 1–12.

[11] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets," *Machine Vision and Applications*, vol. 28, pp. 445–461, 2017.

[12] J. He, K. Pham, N. Valliappan, P. Xu, C. Roberts, D. Lagun, and V. Navalpakkam, "On-device few-shot personalization for real-time gaze estimation," in *ICCV Workshops*, 2019, pp. 0–0.

[13] C. Palmero Cantarino, O. V. Komogortsev, and S. S. Talathi, "Benefits of temporal information for appearance-based gaze estimation," in *ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1–5.

[14] S. Nonaka, S. Nobuhara, and K. Nishino, "Dynamic 3d gaze from afar: Deep gaze estimation from temporal eye-head-body coordination," in *CVPR*, 2022, pp. 2192–2201.

[15] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 162–175, 2017.

[16] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478–500, 2009.

[17] J. Liu, J. Chi, W. Hu, and Z. Wang, "3d model-based gaze tracking via iris features with a single camera and a single light source," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 2, pp. 75–86, 2020.

[18] Z. Zhu, Q. Ji, and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *ICPR*, 2006, pp. 1132–1135.

[19] J. Chen and Q. Ji, "3d gaze estimation with a single camera without ir illumination," in *ICPR*, 2008, pp. 1–4.

[20] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, 2011.

[21] D. W. Hansen and A. E. Pece, "Eye tracking in the wild," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[22] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, "Few-shot adaptive gaze estimation," in *ICCV*, 2019, pp. 9368–9377.

[23] G. Liu, Y. Yu, K. A. F. Mora, and J.-M. Odobez, "A differential approach for gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 1092–1099, 2019.

[24] E. Lindén, J. Sjostrand, and A. Proutiere, "Learning to personalize in appearance-based gaze tracking," in *ICCV Workshops*, 2019, pp. 0–0.

[25] Y. Xiong, H. J. Kim, and V. Singh, "Mixed effects neural networks (menets) with applications to gaze estimation," in *CVPR*, 2019, pp. 7743–7752.

[26] Y. Yu, G. Liu, and J.-M. Odobez, "Improving few-shot user-specific gaze adaptation via gaze redirection synthesis," in *CVPR*, 2019, pp. 11 937–11 946.

[27] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[28] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Pursuit calibration: Making gaze calibration less tedious and more flexible," in *UIST*, 2013, pp. 261–270.