IEICE TRANS. FUNDAMENTALS, VOL.E106–A, NO.6 JUNE 2023

PAPER

GazeFollowTR: A Method of Gaze Following with Reborn Mechanism

Jingzhao DAI[†], **Ming LI**[†], **Xuejiao HU**[†], **Yang LI**[†], ***Nonmembers***, ***and*** **Sidan DU**[†a)], ***Member***

SUMMARY Gaze following is the task of estimating where an observer is looking inside a scene. Both the observer and scene information must be learned to determine the gaze directions and gaze points. Recently, many existing works have only focused on scenes or observers. In contrast, revealed frameworks for gaze following are limited. In this paper, a gaze following method using a hybrid transformer is proposed. Based on the conventional method (GazeFollow), we conduct three developments. First, a hybrid transformer is applied for learning head images and gaze positions. Second, the pinball loss function is utilized to control the gaze point error. Finally, a novel ReLU layer with the reborn mechanism (reborn ReLU) is conducted to replace traditional ReLU layers in different network stages. To test the performance of our developments, we train our developed framework with the DL Gaze dataset and evaluate the model on our collected set. Through our experimental results, it can be proven that our framework can achieve outperformance over our referred methods.

key words: *gaze following, transformer encoder, pinball loss function, reborn mechanism, reborn ReLU layer*

1. Introduction

Currently, analyses of the human gaze bring much convenience to our daily lives and have gradually become a hotspot. In the computer vision field, some existing works show that gaze following can be beneficial for inferring human intention, subsequent behaviors [1] and understanding social communication. For instance, in driver security monitoring [2], driver gaze is a crucial element for preventing drivers from car accidents; additionally, gaze analysis has great potential in health-care systems. Concretely, the introduction of gaze estimation approaches efficiently minimizes the cost of medical equipment [3] and helps a lot in monitoring some persons, possibly with autism spectrum disorder (ASD) [4].

To our knowledge, methods for analyzing gaze have evolved over time. In the early time of evolution, gaze directions are predicted mainly based on eyeball movement [5]. Later, the focus of the methods is transferred to estimating gaze targets. Recently, two kinds of samples have been studied: observers-in-image and observers-out-of-image. For samples without observers, existing methods focus on saliency detection [6], [7] or analyses of gaze patterns [8], [9]. For observers-in-image, related works were early defined by Lian et al. [10] as "gaze following". Usually, a more complex framework must be designed for estimating the light of vision from an observer inside an image. Light of vision is a sophisticated gaze cue containing plentiful information. It is closely related to many factors, including human perception [11], social behaviors [1], [12], surrounding persons [13] and the environment [14], [15]. Among these factors, the surrounding scenes are essential for the final performance. In most scenes, the saliency distribution may be rough, which means that different regions in the scene usually obtain different degrees of saliency. This possibly affects the gaze bias of the observer. Moreover, face information from the observer (e.g., face image, head pose, eyeball movement) has a great impact on the field of view (FoV) generation, which can contribute greatly to the explicit location of the gaze target. To our knowledge, most gaze works focus on scene saliency to achieve gaze targets or on observers' information to obtain gaze directions. While gaze-following works are limited, they have great potential. First, approaches for gaze following can decrease the investment in the lab period and equipment. The process of mapping gaze cues directly from images can skip the utilization of eye trackers. Other works have revealed that the carry of eye trackers brings some issues of high-in-cost, heavy-to-participants, hard-in-calibration and so on. Additionally, gaze-following works can handle diverse problems, such as the social communication of multiple observers and gaze behavior forecasting. In this paper, we present a new framework for gaze following using a hybrid transformer encoder (GazeFollowTR). This framework is developed based on GazeFollow [10]. Our developments are inspired by works in [16]–[18]. The developments can be concretely described as follows:

(1) We explore the improvement contributed by the ReLU layers with the reborn mechanism (reborn ReLU). In different modules of our framework, we replace the usual ReLU layers with reborn ReLU layers in the face, heatmap and gaze field modules;
(2) Based on GazeFollow, we develop the module for observers' information. Different from the original module consisting of a ResNet-18 and three fully connected layers (FCs), we apply the hybrid transformer containing a ResNet-18 and a transformer encoder. It is introduced by Cheng et al. [16];
(3) We deem that detecting gaze points is a fine-tuned problem and requires a bound to control error. Concretely, we apply the pinball loss function for the hybrid transformer. In [17], the authors apply the pinball loss in the

Manuscript received June 18, 2022.
Manuscript revised September 30, 2022.
Manuscript publicized November 30, 2022.
[†]The authors are with School of Electronic Science and Engineering, Nanjing University, Nanjing, 210023 China.
a) E-mail: coff128@nju.edu.cn
DOI: 10.1587/transfun.2022EAP1068

Gaze360 model to control the error of estimating gaze directions. To our knowledge, this is the first work that employs the pinball loss function to model gaze points within constrained bounds;

(4) In this paper, we introduce works related to gaze following in Sect. 2. For the proposed methods in our work, we present details in Sect. 3. Then, our experimental results, challenges in the future and conclusion for our work are presented in Sects. 4, 5 and 6, respectively.

2. Related Work

2.1 Appearance-Based Gaze Estimation

Appearance-based gaze estimation methods can be tracked back to early modeling approaches mapping from eye images to gaze directions. The eye-to-gaze approaches largely depend on high-quality eye images. To ensure data clarity, participants usually needed to wear heavy equipment (eye-tracking glasses [19], [20]) during the data collection. Additionally, data preparation takes a very long period. Annotating on the pupil [19] or eye outline [21] is burdensome. Moreover, wearing glasses usually shelters eyes from view and degrades the performance of pupil detection.

To the best of our knowledge, early eye-based approaches mainly have disadvantages, including information redundancy, independence from samples and high hardware costs. Recently, thanks to the development of deep learning methods, more appearance-based methods have focused on face images with simple annotations. A. Recasens et al. [22] and P. Kellnhofer [23] annotate head pose from observers. In [3], P. A. Dias et al. adopt OpenPose to generate features of facial keypoints, including nose, eyes and ears. In [10] and [24], the authors set a simple head position as input information. In these works, with the supplementary of saliency detection on scenes, proposed face-based approaches can predict not only the gaze direction but also the gaze target.

2.2 Gaze Target Estimation

Recently, estimation of the gaze target has been a hotspot but a challenging issue. From previous works, it can be seen that there are mainly two types of gaze target estimation tasks: gaze point estimation and gaze object estimation. Early works [25], [26], [6] are mainly about generating fixation maps. In each image, no specific observers are contained. The researchers only focus on which part of the image can gain the highest attention from the outside persons. In [10], [13], [24], [14], researchers focus on estimating the specific gaze point. For these works, analyses on both observers and scenes are needed. For gaze object estimation, there is a typical work from Tomas et al. [27]. They propose a new task of gaze object prediction and collect a dataset containing synthetic and real-world images in the retail environment.

2.3 Saliency Detection

Saliency detection and gaze following (gaze estimation for observers inside samples) are different but closely related tasks. Traditional methods [28], [15], [6] for saliency detection focus on generating fixation maps from observers who are out of original images. However, in gaze-following tasks, analyzing the interestingness of the scene [29] is usually applied with a heatmap module [30], [31] to express the saliency region. In the generated heatmap, the gaze points are regarded as the point with the maximum value. It is also worth mentioning that the ground truth heatmaps are yielded as [32] by

$$h(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-g_x)^2+(y-g_y)^2}{2\sigma^2}}, \tag{1}$$

where (g_x, g_y) is the ground truth gaze point, and $h(x,y)$ is the ground truth heatmap generated through a Gaussian filter around (g_x, g_y).

3. Methodology

3.1 Reborn ReLU

Along with the rapid development of deep learning methods, two long-term problems have been raised. First, how can models efficiently make use of feature maps? Moreover, how can redundant information of feature maps be reduced as much as possible? To solve these problems, the rectified linear unit (ReLU) is adopted in many network architectures and reveals its advantages. First, ReLU outputs the maximum value between input and zero, which can be easily computed. Different from tanh and sigmoid activation functions, ReLU can efficiently alleviate problems of gradient vanishing and explosion. Additionally, ReLU can increase the sparsity and linearity of neurons. Within a neural network, ReLU outputs zero when the input value is negative, and the whole feature map becomes sparser after a ReLU layer. However, how ReLU processes negative inputs frequently leads to the dying ReLU problem [33]. During forward propagation, negative neurons will be curtailed, and their derivative will also become zero. Based on our experience, this property usually results in inefficient usage of information. Some existing works proposed the improved ReLU. For instance, leaky ReLU [34] is set with a small positive gradient (0.01) for nonpositive inputs, and in exponential linear units (ELUs) [35], negative inputs are processed thorough the exponential formulation to achieve better convergence. However, these works lose more or fewer advantages of ReLU. In [18], the authors propose a ReLU with the reborn mechanism (Reborn ReLU). Compared to the traditional ReLU layer, it is more effective in information usage, representation ability and channel compensation. Additionally, applying the reborn ReLU can prevent the death of neurons. Equations (2)–(4) show the workflow of one reborn ReLU layer.

$$X_1 = BN(Deconv(-ReLU(-X))), \tag{2}$$

$$X_2 = ReLU(X), \tag{3}$$

$$Y = Compress\,(Contact\,(X_1, X_2)) \tag{4}$$

where "Deconv", "BN", and "ReLU" represent the deconvolution, batch normalization and ReLU layers, respectively. In Eqs. (2)–(4), it can be known that an input map is first fed into two parallel flows to generate X_1 and X_2. Subsequently, we concatenate X_1 and X_2 and then compress them to generate the output map Y.

3.2 Transformer in Gaze Following

In the natural language processing (NLP) field, a transformer consists of an encoder and decoder. For computer vision tasks, the widely applied module is the transformer encoder, which contains multilayer perceptron layers (MLPs), multihead self-attention (MSA) and layer normalization (LN). MSA is the developed self-attention module with multiple subspaces inside it. MLP is a popular module for nonlinear projection. LN can efficiently contribute to fast convergence and training stability.

As introduced in [36], [37], different types of features are embedded and passed through layers of the transformer encoder. Concretely, each 2D image is evenly divided into a set of flattened patches. Then, these patches are mapped to image features for the transformer encoder through a linear projection. Subsequently, a learnable embedding, which is similar to BERT's [class] token, is concatenated with the image features. Moreover, another type of feature, the position features for recording the position of each patch, is added into the embedded features. The overall generation of embedded features inputted to a transformer encoder can be shown as (5).

$$x = Concat\left(x_{cls}, x_{img}\right) + x_{pos}, \tag{5}$$

where $x_{cls} \in \mathbb{R}^{1\times D}$, $x_{img} \in \mathbb{R}^{P^2\times D}$, $x_{pos} \in \mathbb{R}^{(1+P^2)\times D}$ and $x \in \mathbb{R}^{(1+P^2)\times D}$ represent the learnable embedding, image features, position features and final embedded features, respectively. P × P is the resolution of each image patch. D is the constant value representing the latent vector size through all layers of the transformer. Concretely, all flattened patches are mapped to D dimensions in Eq. (5). The embedded features are taken as inputs into the transformer encoder. Equations (6)–(7) show how embedded features go through a one-layer transformer encoder, where inputs x and outputs Y have the same dimension.

$$X = MSA(LN(x)) + x, \tag{6}$$

$$Y = MLP(LN(X)) + X. \tag{7}$$

To estimate gaze information, the output of a transformer encoder is inputted into MLPs for gaze regression. Equations (8)–(9) show the gaze regression after the one-layer transformer encoder. In Eq. (9), G(x, y) is the gaze point position in a 2D image. $MLP(Y)$ is the feature matrix. $[0, :]$, where means we choose the first row of $MLP(Y)$ as our estimated gaze position.

$$Y = TransformerEncoder(x), \tag{8}$$

$$G(x, y) = MLP(Y)[0, :]. \tag{9}$$

However, the standard transformer encoder is not efficient enough for gaze estimation. Some patches divided from a face image may include some parts of eyes. This leads to difficulty in gaze prediction. To prevent this, Cheng et al. [16] develop a hybrid transformer, especially for gaze estimation tasks. Different from the vision transformer, each original image is first processed through a convolutional neural network (CNN). Then, the feature is divided into local patches for the transformer encoder.

3.3 Proposed Framework

Figure 1 illustrates the outline of our proposed framework. First, head positions are taken as inputs into a mapping function to obtain the gaze field [10]. Meanwhile, head images go through a hybrid-GazeTR (a ResNet-18 and transformer encoder). Then, outputs from these two flows are concatenated together (shown as "CAT" in Fig. 1) and taken as inputs to the FOV generator. Finally, the generated FOV maps are concatenated with the original image and subsequently input into an FPN to yield a heatmap for finding the gaze point. To concretely describe the process in the top flow for yielding FOV maps, gaze positions and head positions are calculated to obtain normalized gaze directions. Then, these normalized gaze directions are resized and multiplied with obtained the gaze fields. The multiplied maps subsequently go through the "FOV" generator.

It is common that there are three stages in a gaze-following framework: face stage, gaze field stage and heatmap stage. Based on our knowledge, the face stage is usually utilized for processing observers' information and estimating gaze directions. It can significantly affect the subsequent two stages. In the face stage of GazeFollow [10], the authors feed face images and head positions into a module composed of a Resnet-50 and several fully connected layers (FCs). Concretely, the authors take face images as inputs into ResNet-50 and one FC and then concatenate face features with head features encoded by three FCs. From the one-dimensional concatenated features, gaze directions are inferred, and gaze direction fields can be mapped. In contrast, we develop GazeFollow in three aspects. First, we adopt the hybrid transformer (hybrid-GazeTR) for inferring gaze directions. In detail, we input face images and gaze positions into the hybrid-GazeTR. Second, we applied the pinball loss function to efficiently control the error of gaze points. Then, we can infer gaze directions based on the estimated gaze points and given head positions. Finally, we replace ReLU layers with reborn ReLU layers in different stages in our framework.

In addition to the works mentioned above, our framework is also inspired by the multi-learning strategies

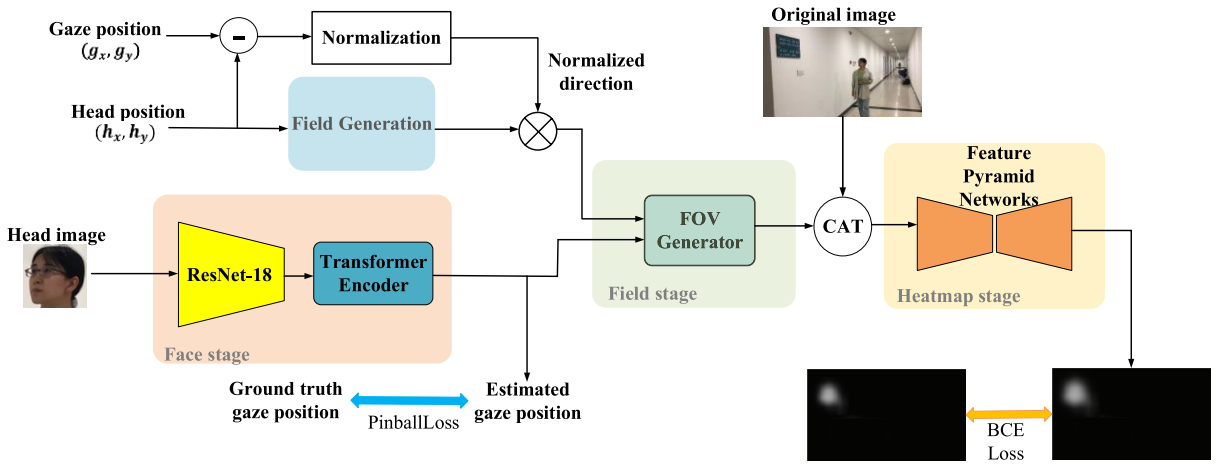

Fig. 1 Our network architecture.

(MTLs), which have been widely applied for fine-grained classification [38], [39] and food recognition tasks [40]–[43]. The gaze estimation task usually has complex samples. In each sample, observers stand or walk around in diverse scenarios (indoor or outdoor). Without depth given, the detection of gaze points is challenging. Many existing works reveal the efficiency of MTL. Taken together, MTLs for gaze estimation can be in the form of one learning flow and two parallel learning flows. Among existing gaze works, the architectures of [10], [22], [24] are designed as the one learning flow. The authors divide the gaze following task into three subtasks, which are respectively assigned by three modules: the scene module, the head module and the heatmap module. In [44], the authors assign gaze directions and gaze points tasks in two learning flows. In this paper, two subtasks, gaze position estimation and gaze field generation, are learned in two parallel flows. Then in the field stage, these two flows are fused together to further determine the final gaze points and directions in the heatmap stage.

3.4 Pinball Loss Function

The pinball loss function is defined based on its shape, which is similar to the trajectory of a pinball. This function is applied to measure how accurate a quantile forecast is. Different from classical forecasts, which aim to return forecasts based on observed values, the quantile forecast focuses more on the given target. Usually, the lower returned by the pinball loss means that the quantile forecast is more accurate. The pinball loss function L_τ can be formulated as

$$L_\tau(\mathrm{y},\mathrm{z}) = \begin{cases} (y-z)\tau & if\ y \geq z, \\ (z-y)(1-\tau) & if\ z > y, \end{cases} \tag{10}$$

where τ is the given target quantile, z is the quantile forecast and y is the real value. Recently, the pinball loss function has been utilized for pattern classification and optimizing networks. Meanwhile, it has been applied to model the error bounds of gaze directions. Kellnhofer, P. et al. [23] propose the pinball loss function for estimating gaze directions in unconstrained scenes. In our work, we employ the pinball loss function to control the error of gaze points. The process can be formulated as follows.

$$\hat{G}, \Delta\hat{G} = GazeTR\left(i_{face}\right), \tag{11}$$

$$z = \begin{cases} \hat{G} - \Delta\hat{G}, & \tau \leq 0.5 \\ \hat{G} + \Delta\widehat{G}, & \tau > 0.5 \end{cases}, y = G_{gt}, \tag{12}$$

$$L_\tau\left(\hat{G}, \Delta\hat{G}, G_{gt}\right) = \max\left(\tau\left(y-z\right), \left(1-\tau\right)\left(z-y\right)\right), \tag{13}$$

where i_{face} is the face image inputted to the model *GazeTR* (ResNet-18+transformer); $\hat{G}$ and $\Delta\hat{G}$ are the estimated gaze point and the error of gaze points, respectively; and G_{gt} is the ground truth gaze point. In this paper, we compare the performance brought by the pinball loss and L1Loss functions through the evaluation on our collected samples, which can be shown in Table 2.

4. Experiments

In this section, all models are trained with the DL Gaze dataset [10] and then evaluated on our collected set. Due to this difference between these two datasets, the existing methods in [10], [23], [16] achieve degraded performance on our collected set. Compared to the existing methods, our proposed methods can achieve outperformance. In this section, we arrange our experimental results in three aspects. First, we emphasize the efficiency of the transformer and pinball loss function in Table 2. Second, we present the results of different reborn ReLU replacements in three modules of our network architecture. Third, we compare our optimal results with existing works. The evaluation protocols are distance (Dist.) between the ground truth and estimated gaze points, and gaze angle errors (Ang.) between the ground truth and estimated gaze directions. In each table, the optimal results are highlighted in bold and italic.

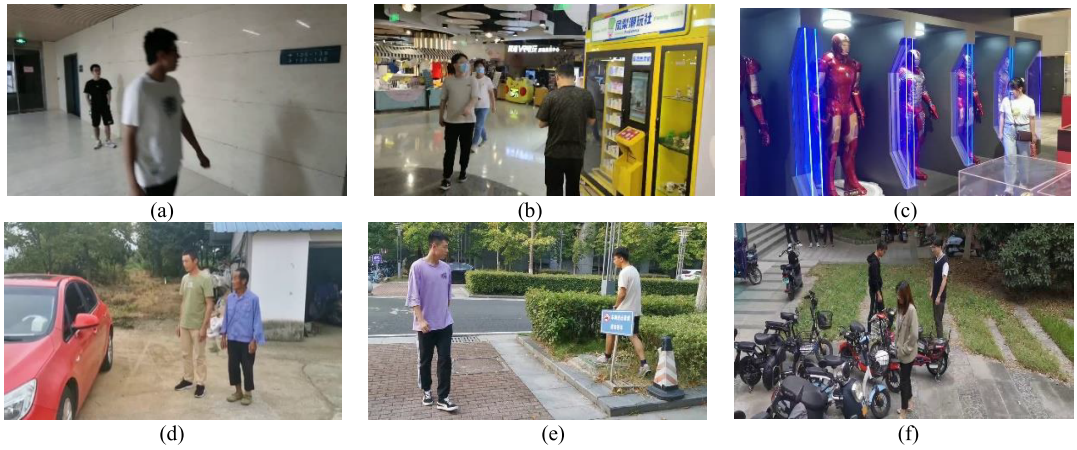

Fig. 2 Our sample image.

4.1 Our Dataset

We build a video dataset recorded from six types of real-scenarios as shown in Fig. 2. Concretely, our dataset contains 920 frames clipped from 8 videos (10 fps). In each video, we recorded the activities of 8 volunteers in 8 different scenes. Each volunteer usually looks at a point of the object while walking. We recorded all videos with an iPhone X.

Additionally, we only focus on one volunteer activity in each video and then ask them to annotate true gaze positions. Some sample images are shown in Fig. 2. In Table 1, we show the estimated gaze positions (normalized) achieved from our method and the referred methods. The chosen sample image is image (a) in Fig. 2. Its true gaze position is [0.8726562, 0.25694445].

Table 1 Comparison of gaze positions estimated by our method and the referred methods on our sample image (Fig. 2(a)).

Methods	Estimated gaze positions	Dist.
GazeFollow [10]	[0.75, 0.26]	0.44
Gaze360 [23]	[0.54, 0.50]	0.47
Hybrid-GazeTR (L1Loss) [16]	[0.50, 0.36]	0.33
Ours: Hybrid-GazeTR (pinball loss)	[0.52, 0.43]	0.40
Ours: GazeFollowTR	[0.37, 0.54]	0.47

Table 2 Comparison between LSTM and the transformer and between L1Loss and PinballLoss functions.

Methods	Architecture	Dist.	Ang.
Gaze360 [23]	ResNet-18+LSTM	0.237	74.25
Hybrid-GazeTR (L1Loss) [16]	ResNet-18+Transformer (L1Loss)	0.257	50.08
Hybrid-GazeTR (Pinball Loss)	ResNet-18+Transformer (PinballLoss)	***0.219***	***42.40***

4.2 Transformer vs LSTM

The long short-term memory network (LSTM) and the transformer are all applied in analyzing gaze. LSTM is efficient for capturing and combining temporal and spatial information. Kellnhofer, P. et al. [23] propose a model using bidirectional LSTM for estimating gaze directions in a dynamic environment. For the transformer, Cheng et al. [16] conduct the first work of applying the transformer for predicting gaze directions. They develop a hybrid transformer (hybrid-GazeTR) that is more suitable for gaze samples. Different from the work in [16], we replace the originally employed L1Loss function with the pinball loss function. To emphasize the efficiency of the transformer and pinball loss function, we evaluate Gaze360, Hybrid-GazeTR and our developed GazeTR on our collected samples.

As shown in Table 2, it can be seen that the transformer can contribute more improvement than LSTM in gaze direction estimation. Additionally, employing the pinball loss function can further increase the performance. From the improved performance, it can be first analyzed that the transformer encoder is more efficient than LSTM in gaze following. Moreover, the performance of two different functions may be due to signatures of the pinball loss function, which we mentioned before. The experiments also test our hypothesis that gaze point estimation is a fine-tuned problem that requires a boundary for controlling errors. Therefore, we decide the developed GazeTR (the hybrid GazeTR with a pinball loss function) as a module in the face stage of our framework.

4.3 Reborn ReLU Replacement

Before the experiments, our hypothesis is that replacing ReLU layers with reborn ReLU layers can contribute to the performance. To test our hypothesis, we conduct the reborn ReLU replacements in different modules of our proposed framework. Concretely, the face stage, heatmap stage and field stage are considered for the reborn ReLU replacements. Taken together, eight conditions of reborn ReLU replacements are considered in this paper. The experimental results are shown in Table 3.

Table 3 Evaluation of different Reborn ReLU replacements in our framework.

Reborn ReLU replacement	Dist.	Ang.
Plain Framework	0.203	21.75
Field Stage	0.240	27.42
Heatmap Stage	0.204	22.14
Face Stage	***0.187***	20.80
Field & Heatmap modules	0.297	21.93
Face & Heatmap modules	0.196	20.74
Face & Field modules	0.193	21.25
Face & Heatmap & Field modules	***0.188***	***19.23***

In this paragraph, we describe where all reborn ReLU replacements are conducted in our framework. In the field stage, we replace the ReLU layer just after the generated FoV map. In the heatmap stage, we conduct the replacement in a feature pyramid network (FPN) [45]. Concretely, the last 4 ReLU layers in the top-down pathway of FPN are all replaced as reborn ReLU layers. In the face stage, the replacement is operated in a ResNet-18 module of the hybrid-GazeTR. In ResNet-18, there are four basic blocks before the convolutional layer (the last layer). ReLU layers in each basic block are all replaced as reborn ReLU layers. In our implementation, the parameters can be changed are the number of input channels and output channels. The number of input channels relies on the input maps of a reborn ReLU layer. The number of output channels is always set as 5.

It can be observed from Table 3 that our framework can achieve excellent performance under two conditions, which are shown as the "face stage" and "face & heatmap & field stages" in this table. Concretely, reborn ReLU replacement is operated in ResNet-18 of the "face stage", after upsampling in the feature pyramid network ("heatmap stage"), and in the last layer in the "field module".

4.4 Overall Experimental Comparison

Our method is inspired by GazeFollow [10], Gaze 360 [23] and hybrid-GazeTR [16]. In the beginning, our framework is designed closely to GazeFollow. However, we develop the face module part, which is inspired by hybrid-GazeTR. In addition, thanks to the introduction of the pinball loss function in [23], we replace the L1Loss function of hybrid-GazeTR with the pinball loss function. Moreover, the ReLU layer, which is popular in many networks, is replaced as the novel reborn ReLU layer. After these developments, our method can achieve outperformance over these referred works on our collected dataset, which can be seen in Table 4.

Based on this table, there are several points can be analyzed. For conventional methods, Gaze360 and the hybrid-GazeTR both achieve degraded performance compared to GazeFollow. The factor we analyzed is that the framework of GazeFollow is more comprehensive than the two other methods. More importantly, there is no gaze field stage in Gaze360 and the hybrid-GazeTR. However, we still think these two models can achieve good performance if we apply them as one module in our framework. Therefore, we propose the pinball loss function in Gaze360 and hybrid-GazeTR in our framework to learn face images and gaze positions. As shown in Table 4, after some developments, our GazeFollowTR method can achieve outperformance over our referred methods. Additionally, to keep the evaluation consistency, we follow the same training/testing split with GazeFollow. The results are shown in Table 5.

Table 4 Comparison between our method and the referred methods on our collected dataset.

Methods	Dist.	Ang.
GazeFollow [10]	0.215	26.27
Gaze360 [23]	0.237	74.25
Hybrid-GazeTR (L1Loss) [16]	0.257	50.08
Ours: Hybrid-GazeTR (pinball loss)	0.219	42.40
Ours: GazeFollowTR	***0.188***	***19.23***

Table 5 Comparison between our method and the referred methods on our collected dataset.

Methods	Dist.	Ang.
GazeFollow [10]	0.213	19.94
Ours: GazeFollowTR	***0.174***	***17.40***

5. Challenges in the Future

In this section, we present some challenges in our experiments and our plan in the future. First, self-occlusion and gaze ambiguity [10], [3] are common in monocular images. Concretely, some gaze points are sheltered by other objects that are closer to the camera. Meanwhile, even for participants, data labeling is difficult in clearly deciding where they ever looked. To alleviate this challenge, we learn related literature and find that some existing methods (e.g., the work in [46]) tend to analyze gaze under varied depths and positions. These methods inspire us to think deeply about the solution. In the future, we plan to conduct gaze-following work in immersive videos. In the currently collected videos, it can be observed that issues occurring in monocular images are alleviated.

The second challenge is that datasets for gaze following in immersive videos are not available. To our knowledge, existing datasets for gaze following are predominantly or wholly formed in two dimensions. In addition, immersive datasets related to gaze analyses are almost without observers. Almost all publishers focus on saliency detection. In the future, we plan to build immersive videos including observers and scenes and further develop our framework. We believe that the current challenges will be alleviated in our future work.

6. Conclusion

In this paper, we propose a developed framework called GazeFollowTR. Through the experimental results yielded from GazeFollowTR, we test our three hypotheses. First, the transformer encoder is efficient in the gaze-following task. Second, replacing traditional ReLU layers with a reborn ReLU layer can contribute to performance improvement. Finally, applying the pinball loss function can efficiently control the error of gaze points. Through our experimental results, our hypotheses are tested. We consider some factors leading to the improvement. First, in our framework, the improvement brought by our face stage (hybrid transformer applied with a pinball loss function) can efficiently sharpen the field of view in the next stage, which is also beneficial for the final heatmap stage. Moreover, the reborn ReLU makes full use of the information in a feature map. Compared to the ordinary ReLU layers, we deem the reborn ReLU layers to be more suitable for observers' face images and scene images containing complex information. Finally, to our knowledge, we deem that accurately deciding a gaze point in a saliency region is difficult. Therefore, adopting loss function modeling error bounds is necessary.

Acknowledgments

We acknowledge the computational resources supported by the High-Performance Computing Center of Collaborative Innovation Center of Advanced Microstructures, Nanjing University, and Nanjing Institute of Advanced Artificial Intelligence. In this survey paper, there are no relevant conflicts of interest to declare.

References

[1] P. Wei, Y. Liu, T. Shu, N. Zheng, and S. Zhu, "Where and why are they looking? Jointly inferring human attention and intentions in complex tasks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.6801–6809, June 2018, doi: 10.1109/CVPR.2018.00711.

[2] R.F. Ribeiro and P.D.P. Costa, "Driver gaze zone dataset with depth data," 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), pp.1–5, May 2019, doi: 10.1109/FG.2019.8756592.

[3] P.A. Dias, D. Malafronte, H. Medeiros, and F. Odone, "Gaze estimation for assisted living environments," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.279–288, March 2020, doi: 10.1109/WACV45572.2020.9093439.

[4] Y. Fang, H. Duan, F. Shi, X. Min, and G. Zhai, "Identifying children with autism spectrum disorder based on gaze-following," 2020 IEEE International Conference on Image Processing (ICIP), pp.423–427, Oct. 2020, doi: 10.1109/ICIP40778.2020.9190831.

[5] K. Kikuchi, H. Takahira, R. Ishikawa, E. Wakamatsu, T. Shinkawa, and M. Yamada, "Development of a device to measure movement of gaze and hand," IEICE Trans. Fundamentals, vol.E97-A, no.2, pp.534–537, Feb. 2014, doi: 10.1587/transfun.E97.A.534.

[6] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," IEEE Trans. Image Process., vol.27, no.10, pp.5142–5154, 2018, doi: 10.1109/TIP.2018.2851672.

[7] A. Volokitin, M. Gygli, and X. Boix, "Predicting when saliency maps are accurate and eye fixations consistent," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[8] H. Takahira, R. Ishikawa, K. Kikuchi, T. Shinkawa, and M. Yamada, "Analysis of gaze movement while reading E-books," IEICE Trans Fundamentals, vol.E97-A, no.2, pp.530–533, Feb. 2014, doi: 10.1587/transfun.E97.A.530.

[9] S. Yeamkuan and K. Chamnongthai, "Fixational feature-based gaze pattern recognition using long short-term memory," 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp.1–4, Dec. 2020.

[10] D. Lian, Z. Yu, and S. Gao, "Believe it or not, we know what you are looking at!," Computer Vision – ACCV 2018, pp.35–50, Springer International Publishing, Cham, 2019.

[11] B. Liu and K. Arakawa, "A method for generating color palettes with deep neural networks considering human perception," IEICE Trans. Fundamentals, vol.E105-A, no.4, pp.639–646, April 2022, doi: 10.1587/transfun.2021SMP0011.

[12] L. Fan, W. Wang, S.-C. Zhu, X. Tang, and S. Huang, "Understanding human gaze communication by spatio-temporal graph reasoning," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[13] N. Zhuang, B. Ni, Y. Xu, X. Yang, W. Zhang, Z. Li, and W. Gao, "MUGGLE: MUlti-stream group gaze learning and estimation," IEEE Trans. Circuits Syst. Video Technol., vol.30, no.10, pp.3637–3650, 2020, doi: 10.1109/TCSVT.2019.2940479.

[14] Y. Fang, J. Tang, W. Shen, W. Shen, X. Gu, L. Song, and G. Zhai, "Dual attention guided gaze target detection in the wild," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.11385–11394, June 2021, doi: 10.1109/CVPR46437.2021.01123.

[15] E. Chong, N. Ruiz, Y. Wang, Y. Zhang, A. Rozga, and J.M. Rehg, "Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency," Computer Vision – ECCV 2018, pp.397–412, Springer International Publishing, Cham, 2018.

[16] Y. Cheng and F. Lu, "Gaze estimation using Transformer," 2022 26th International Conference on Pattern Recognition (ICPR), pp.3341–3347, Aug. 2022.

[17] R. Koenker, Quantile Regression (Econometric Society Monographs), Cambridge University Press, Cambridge, 2005.

[18] Z. Cai, K. Huang, and C. Peng, "Reborn mechanism: Rethinking the negative phase information flow in convolutional neural network," arXiv preprint arXiv:2106.07026v2, 2021.

[19] M. Tonsen, X. Zhang, Y. Sugano, and A. Bulling, "Labeled pupils in the wild: A dataset for studying pupil detection in unconstrained environments," ACM, pp.139–142, 2016.

[20] T. Fischer, H.J. Chang, and Y. Demiris, "RT-GENE: Real-time eye gaze estimation in natural environments," European Conference on Computer Vision, pp.339–357, 2018.

[21] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky, "DeepWarp: Photorealistic image resynthesis for gaze manipulation," ECCV, pp.311–326, 2016.

[22] A. Recasens Continente, A. Khosla, C. Vondrick, and A. Torralba, Where are They Looking?, Advances in Neural Information Processing Systems (NIPS), 2015.

[23] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp.6911–6920, Oct.–Nov. 2019, doi: 10.1109/ICCV.2019.00701.

[24] E. Chong, Y. Wang, N. Ruiz, and J.M. Rehg, "Detecting attended visual targets in video," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.5395–5405, June 2020, doi: 10.1109/CVPR42600.2020.00544.

[25] H.R. Tavakoli, F. Ahmed, A. Borji, and J. Laaksonen, "Saliency revisited: Analysis of mouse movements versus fixations," 2017 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), pp.6354–6362, July 2017, doi: 10.1109/CVPR.2017.673.
[26] W. Jian and Z. Xinbo, "Analysis of eye gaze points based on visual search," 2014 International Conference on Orange Technologies, pp.13–16, Sept. 2014, doi: 10.1109/ICOT.2014.6954665.
[27] H. Tomas, M. Reyes, R. Dionido, M. Ty, J. Mirando, J. Casimiro, R. Atienza, and R. Guinto, "GOO: A dataset for gaze object prediction in retail environments," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp.3119–3127, June 2021, doi: 10.1109/CVPRW53098.2021.00349.
[28] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand, "Where should saliency models look next?," Computer Vision – ECCV 2016, pp.809–824, Springer International Publishing, Cham, 2016.
[29] A. Borji and L. Itti, "CAT2000: A large scale fixation dataset for boosting saliency research," ArXiv, vol.abs/1505.03581, 2015.
[30] M. Kümmerer, L. Theis, and M. Bethge, "Deep gaze I: Boosting saliency prediction with feature maps trained on ImageNet," Computer Science, 2014.
[31] J. Pan, C. Canton, K. Mcguinness, N.E. O'Connor, and X. Giro-I-Nieto, "SalGAN: Visual saliency prediction with generative adversarial networks," arXiv preprint arXiv:1701.01081v3, 2017.
[32] T. Pfister, J. Charles, and A. Zisserman, "Flowing ConvNets for human pose estimation in videos," 2015 IEEE International Conference on Computer Vision (ICCV), pp.1913–1921, Dec. 2015, doi: 10.1109/ICCV.2015.222.
[33] Z. Hu, Y. Li, and Z. Yang, "Improving convolutional neural network using pseudo derivative ReLU," 2018 5th International Conference on Systems and Informatics (ICSAI), pp.283–287, Nov. 2018, doi: 10.1109/ICSAI.2018.8599372.
[34] A.L. Maas, "Rectifier nonlinearities improve neural network acoustic models," International Conference on Machine Learning (ICML), vol.30, 2013.
[35] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," Computer Science, 2015.
[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929v2, 2020.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," arXiv:1706.03762, 2017.
[38] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[39] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4476–4484, July 2017, doi: 10.1109/CVPR.2017.476.
[40] S. Jiang, W. Min, L. Liu, and Z. Luo, "Multi-scale multi-view deep feature aggregation for food recognition," IEEE Trans. Image Process., vol.29, pp.265–276, 2020, doi: 10.1109/TIP.2019.2929447.
[41] J. He, Z. Shao, J. Wright, D. Kerr, C. Boushey, and F. Zhu, "Multi-task image-based dietary assessment for food recognition and portion size estimation," 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp.49–54, Aug. 2020, doi: 10.1109/MIPR49039.2020.00018.
[42] S. Jiang, W. Min, Y. Lyu, and L. Liu, "Few-shot food recognition via multi-view representation learning," ACM Trans. Multimedia Comput. Commun. Appl., vol.16, no.3, pp.1–20, 2020, doi: 10.1145/3391624.
[43] H. Liang, G. Wen, Y. Hu, M. Luo, P. Yang, and Y. Xu, "MVANet: Multi-tasks guided multi-view attention network for Chinese food recognition," IEEE Trans. Multimedia, vol.23, pp.3551–3561, 2021, doi: 10.1109/TMM.2020.3028478.
[44] D. Lian, L. Hu, W. Luo, Y. Xu, L. Duan, J. Yu, and S. Gao, "Multiview multitask gaze estimation with deep convolutional neural networks," IEEE Trans. Neural Netw. Learn. Syst., vol.30, no.10, pp.3010–3023, Oct. 2019, doi: 10.1109/TNNLS.2018.2865525.
[45] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," CoRR, vol.abs/1612.03144, 2016, doi: 10.48550/arXiv.1612.03144.
[46] S. Mochiduki, R. Watanabe, H. Takahira, and M. Yamada, "Analysis of head movement during gaze movement with varied viewing distances and positions," IEICE Trans. Fundamentals, vol.E101-A, no.6, pp.892–899, June 2018, doi: 10.1587/transfun.E101.A.892.

Jingzhao Dai received the B.S. degree in Electronics and Communication Engineering from Zijin college, Nanjing University of Science and Technology, Nanjing, China, in 2017, and the M.S. degree from Electrical and Computer Engineering, Purdue University Northwest, Hammond, US, in 2019. She is currently pursuing the Ph.D. degree in degree in electronic science and technology withhe Nanjing University, Nanjing, China. Her research interests include imaging processing, computer vision, and machine learning.

Ming Li received the B.S. degree in School of electronic science and engineering from Nanjing University, Nanjing, China, in 2017. He is currently working toward Ph.D. degree in electronic science and technology with the Nanjing University. His research interests include image processing, computer vision and artificial intelligence.

Xuejiao Hu received the B.S. degree in Computer Science and Technology from Binjiang college, Nanjing University of information science & Technology, Nanjing, China, in 2017, and the M.S. degree from School of Information Sciences and Technology, Nanjing Agriculture University, Nanjing, China, in 2019. She is currently pursuing the Ph.D. degree in electronic science and technology with the Nanjing University, Nanjing, China. Her research interests include computer vision, machine learning, and artificial intelligence.

Yang Li received the B.S. and M.S. degrees in mechanical engineering from Southeast University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electronics engineering from Nanjing University, Nanjing, China, in 2006. He is an Associated Professor with School of Electronic Science and Engineering, Nanjing University, China, since 2009. His research interests include digital signal processing and computer vision.

Sidan Du received the B.S. and M.S. degrees in electronic engineering from Xidian University, Xian, China in 1984 and 1987, respectively, and the Ph.D. degree in physics from Nanjing University, Nanjing, China, in 1997. She is currently a Professor in the school of Electronic Science and Engineering, Nanjing University. Her research interests include computer vision and signal processing.