Adaptive Feature Fusion Network for Gaze Tracking in Mobile Tablets

Yiwei Bao[1] Yihua Cheng[1] Yunfei Liu[1] Feng Lu[1,2,*]
[1]State Key Laboratory of Virtual Reality Technology and Systems, School of CSE, Beihang University, Beijing, China
[2]Peng Cheng Laboratory, Shenzhen, China
{baoyiwei, yihua_c, lyunfei, lufeng}@buaa.edu.cn

***Abstract*—Recently, many multi-stream gaze estimation methods have been proposed. They estimate gaze from eye and face appearances and achieve reasonable accuracy. However, most of the methods simply concatenate the features extracted from eye and face appearance. The feature fusion process has been ignored. In this paper, we propose a novel Adaptive Feature Fusion Network (AFF-Net), which performs gaze tracking task in mobile tablets. We stack two-eye feature maps and utilize Squeeze-and-Excitation layers to adaptively fuse two-eye features according to their similarity on appearance. Meanwhile, we also propose Adaptive Group Normalization to recalibrate eye features with the guidance of facial feature. Extensive experiments on both GazeCapture and MPIIFaceGaze datasets demonstrate consistently superior performance of the proposed method.**

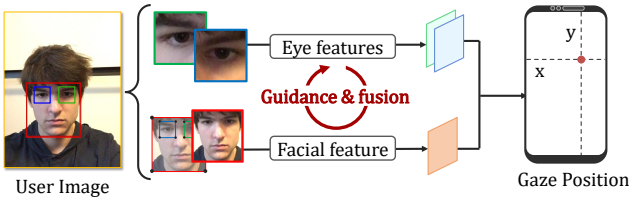

Fig. 1. Illustration of our task and proposed technique. User images are used to estimate gaze positions in mobile tablets. An effective guidance & fusion mechanism is proposed to enhance the feature extraction from both eye regions and facial region.

I. Introduction

As an important indicator of human attention, gaze is found to be useful to diagnose mental condition and predict human intentions. For example, gaze estimation technology is commonly used in human attention diagnosis like fatigue driving [1], [2] and saliency detection [3]–[5]. Gaze has also became a newly-developing human-computer interaction method [6], [7], especially in areas like virtual reality [8], [9].

Up to now, many gaze estimation methods have been proposed. Conventional model-based methods estimate gaze by building 3D eye models. However, they usually require some dedicate devices, like high resolution cameras, RGB-D cameras [10]–[12] and infrared cameras [13]. Recent years, appearance-based methods which directly map gaze from appearance have made great progress. Some appearance-based methods using convolutional neural networks (CNNs) have been proposed and show convincing results. The methods estimating gaze from eye images show reasonable results at the beginning [14], [15]. Later, face images and head pose are found to be helpful for gaze estimation [16], [17]. Meanwhile, several large-scale gaze datasets have also been published for the research of CNN-based gaze estimation [18]–[20].

More recently, CNN-based methods [16], [20], [21] with face and eyes inputs have became popular in gaze estimation since their high accuracy and robustness. Meanwhile, eye structures like iris and pupil are also crucial for human gaze estimation. Most of CNN-based methods always have eye images (or face images which contain eyes) for input. Generally, left and right eyes have identical structure and look at the same targets in most time. The observation of obvious relationships between left and right eyes can help to better use eye images for accurate gaze estimation. Fischer *et al.* [22] and Krafka *et al.* [20] proposed to concatenate feature vectors from both eyes and process them with fully connected layers (FC layers). Cheng *et al.* [23] proposed to utilize two eyes asymmetric by adaptively adjusting evaluation weights of eyes. However, we find 1) it's not sufficient to utilize the relationship between both eyes by concatenating feature vectors directly or adjusting weights of eyes. Especially for gaze estimation in wild settings, which is extremely hard to extract and utilize proper eye features. 2) Although face images are found to be helpful [16] and become a common input which provide critical information like head pose, light condition, individual differences, most methods still treat face and eye images separately. Only few efforts were made to explore face-eye relationship in gaze estimation. Cheng *et al.* [24] proposed to estimate basic gaze direction from face image and refine it with eye images.

To better utilize the similarity of two eye structures and face-eye relationship, we propose Adaptive Feature Fusion Network (AFF-Net). We illustrate the pipeline of AFF-Net in Figure 1. The AFF-Net improves gaze tracking accuracy in two ways. First, the AFF-Net fuses two eye features according to two eye similarity and appearance. Second, the AFF-Net guides eye feature extraction with face appearance characteristics by adaptively recalibrating eye features according to face and eye bounding boxes (we refer to them as Rects) and facial feature. Rigorous experiments show that AFF-Net can produce superior performance against state-of-the-art methods on two

*Corresponding author.

commonly used datasets. In particular, while combining two eye features, we take two eye feature maps from different layers to stack a fused eye feature map and use a combination of convolutional layers and Squeeze-and-Excitation layers (SE layer) to generate final eye feature. Compared with the classic way of concatenating eye feature vectors, stacked eye feature maps reserve more spatial information and better utilize the identical structure of two eyes. SE layers in the fusion process adaptively weight each channel based on cross-channel information, treat two eye features differently according to their appearance. Specifically, 1) to better extract and combine the identical structure of two eyes, we stack a fused eye feature map and use a convolutional layer (conv layer) to generate final eye feature. Then we apply SE layer to adaptively weight each channel based on cross-channel information, treat two eye features differently according to their appearance. 2) we also propose a novel Adaptive Group Normalization (AdaGN) to apply face appearance characteristics guidance for eye feature extraction. In detail, AdaGN takes face, eyes bounding boxes and facial feature as input to scale and shift eye features.

In summary, the contributions of this paper are as follow:

- We propose a novel architecture (*i.e.*, AFF-Net) for gaze tracking. Motivated by the two eye similarity and relationship between eyes and face, AFF-Net equips with better extraction of two eyes' features and face appearance guidance.
- We propose a novel Adaptive Group Normalization layer, which recalibrates eye features according to face appearance characteristics.
- The proposed method outperforms existing state-of-the-arts methods on both GazeCapture and MPIIFaceGaze datasets.

The rest of the paper is organized as following. Section II summarizes the overview of related works. Section III describes the proposed AFF-Net. In Section III, we first introduce the general architecture of AFF-Net. Then, we describe the proposed eye feature fusion scheme and adaptive Group Normalization. Section IV compares the experimental results on two public dataset with several other state-of-the-art methods, conducts ablation studies to evaluate the effectiveness of each component and further analyze the gaze estimation results of proposed AFF-Net.

II. Related Works

As a active research topic, many different approaches have been proposed to address gaze estimation problem. These methods can be categorized into model-based methods and appearance-based methods.

Model-based approaches aim to fit a 3D eye model to the image and calculate gaze via specific geometric constrains [25]. To acquire accurate eye location, distinct eye features like corneal reflection [26], pupil center [27] and iris contour [10], [28] are commonly used. Paper [12] and [11] proposed to estimate gaze using RGB-D cameras. RGB-D cameras are mainly used to obtain the depth of 2D facial landmarks and 3D pupil location in camera coordinate system. Without RGB-D camera, the 3D location of facial landmarks are usually calculated by minimizing projection error between 2D facial landmarks and corresponding points on 3D face model [29]. Wang *et al.* [30] proposed a deformable eye-face model method. The method models a new subject as a linear combination of offline collected eye-face models. This method also avoid head-eye offset vector estimation to improve accuracy and robustness. Wen *et al.* [31] use convergence constraint which allows calibration without knowing exact gaze location and a person independent gaze corrector to reduce system error. Model-based methods are mostly limited by strict user-camera distance or professional equipment like infrared cameras [26] and RGB-D cameras [11], [12]. Consequently, model-based methods are accurate under controlled laboratory environment but less reliable under unconstrained environment.

Appearance-based approaches aim to find the direct mapping function from image appearance to gaze direction or gaze location. Appearance-based methods take gaze estimation as a regression from eye images to gaze direction. Thus, they usually only require single camera to capture user face image. Tan *et al.* [32] proposed to estimate gaze with local linear interpolation. Lu *et al.* [33] proposed an adaptive linear regression, which allows small differences in head rotation, image resolution and blink. Williams *et al.* [34] utilized Gaussian process regression which is able to train on semi-supervised data to estimate gaze direction. Some appearance-based methods directly compute gaze direction based pixel values without training like dimension reduction [35]. Early appearance-based methods usually rely on hand craft features. With dramatic differences of eye appearance under different head orientation, background environment and personal characteristic, it is difficult for hand crafted features to maintain high accuracy in different settings. Thus, the fast developing CNN with strong representational power makes it the most popular method to estimate human gaze.

CNN-based methods attempt to estimate gaze with CNNs, which is trained from several gaze datasets with supervised learning. Sugano *et al.* [36] capture user images with multiple cameras simultaneously and synthetic images with different gaze directions to compose the UT Multiview dataset. Funes *et al.* [37] published the Eyediap dataset. 16 subjects were required to look at both 2D screen targets and 3D floating targets. Zhang *et al.* [19] proposed a CNN with facial weights and published the popular MPIIGaze dataset. Zhu *et al.* [38] process head pose information with gaze transfer layer to increase robustness by eliminate the overfitting of head-gaze correlation that differs for every dataset. Kim *et al.* [39] published near infrared dataset NVGaze, which contains big amount of collected real images and high resolution synthetic images with detailed parameters. Cheng *et al.* [24] proposed a coarse-to-fine way which estimates a basic gaze direction from face images and refine it with more detailed eye images. To address the lack of large scale dataset, Krafka *et al.* [20] collected the GazeCapture dataset with mobile devices and proposed iTracker which takes eyes, face and face grid as input. Based on the GazeCapture dataset, gaze estimators

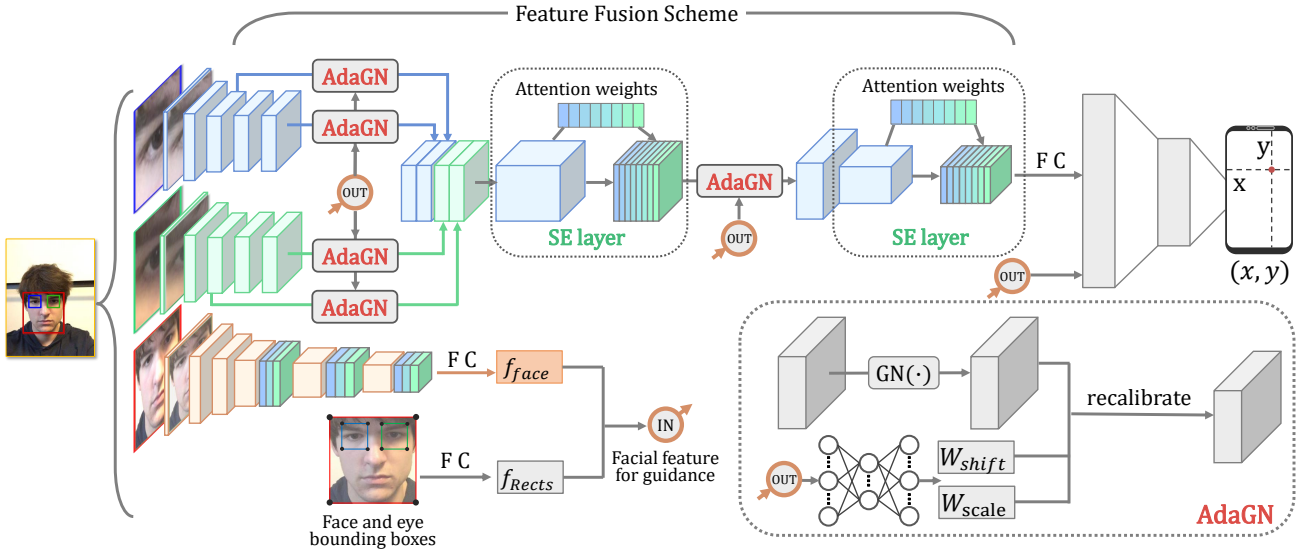

Fig. 2. Structure of proposed AFF-Net. Facial feature is extracted by several conv layers and FC layers. Rects feature is extracted directly by FC layers. The eye stream takes left and flipped right eye images as input and extract fused eye features by proposed stacking architecture and SE layers. AdaGN recalibrates eye features with face appearance characteristics derived from face and Rects features.

under mobile devices settings gains much attention. He *et al.* [40] proposed a lite CNN model without face input and a few shot calibration scheme which further increases accuracy. Guo *et al* [41] proposed a Tolerant and Talented training scheme to get better accuracy and robustness. Thanks to large dataset like GazeCapture and powerful feature extraction ability of CNN, appearance-based methods achieve good accuracy and head-pose independence. The ability to tolerant low resolution images and changing environments also make appearance-based methods more effective in wild settings.

III. Adaptive Feature Fusion Network

A. Overview of the network

The structure of AFF-Net is shown in Figure 2. The AFF-Net has four inputs, which are face image, left and right eye images, top left corner and bottom right corner of face and eye bounding boxes . We propose AFF-Net in order to improve eye tracking accuracy by 1) utilizing the similarity of two eyes appearance to adaptively fuse eye features, 2) guide eye feature extraction with face appearance characteristics. For the first aspect, we stack feature maps from both eyes in channel wise to fuse eye features, which are followed by several convolutional layers. SE layers are used during eye feature extraction to weight features from different eyes according to their appearance. For the second aspect, the pre-processed Rects and facial feature are normalized for guiding eye feature extraction.

B. Squeeze and excitation for eye feature fusion

When fuse eye features, most state-of-the-arts methods concatenate eye feature vectors reshaped from eye feature maps and process them with fully connected layers. However, there are two problems in these methods. First, reshape feature maps to feature vector loses some spacial and cross-channel information, weakening two eyes relationships. Second, fully connected layers are not as powerful as convolutional layers when dealing with images. To handle such problems, we propose an eye feature fusion structure which contains eye feature stacking and SE layer for adaptive eye fusion. As shown in Figure 3, we take eye feature maps from different layers and stack them in channel wise. For the shape and structures of two eyes are identical, it is reasonable to fuse eye

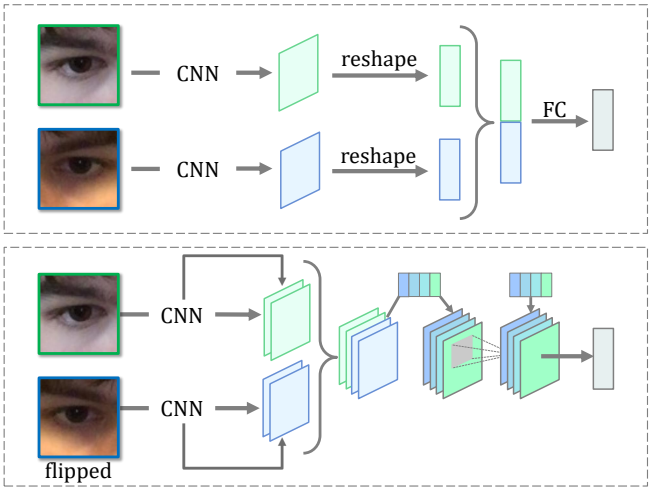

Fig. 3. Classic eye feature fusion structure (top) and proposed eye feature fusion structure (bottom).

features by stacking feature maps together. We take feature maps from lower layer for they reserve more spacial information and feature maps from higher layer for their stronger representational abilities. The fused eye features are followed by convolutional layers to generate final eye feature vector. In this way, two eyes relationships are considers by calculating the final eye feature values according to the corresponding areas in both eyes.

SE layer is a powerful structure for applying attention for different eye features from different channels. SE layer works as Equation 1:

$$\begin{cases} W_{weight} = \sigma(L(GAP(f_{in}))), \\ f_{out} = F_{scale}(W_{weight}, f_{in})), \end{cases} \tag{1}$$

f_{in} stands for the input feature of SE layer. $GAP(\cdot)$ stands for Global Average Pooling layer (GAP layer). Every channel of input feature is compressed to single value by GAP layer. $L(\cdot)$ stands for FC layers and $\sigma\cdot$ stands for Sigmoid function. Then, a weight vector W_{weight} is calculated by FC layers. $F_{scale}(\cdot)$ stands for channel wise multiplication of inputs. W_{weight} is applied to original input to derive final result f_{out}. SE layer allocates attentions to different channels according to cross-channel relationships, which is suitable for two-eye relationship based eye feature fusion.

In the eye feature fusion structure, SE layers are added before and after eye feature stacking. Before eye feature stacking, SE layers are added to dynamically adjust channel-wise features, enhance the representational power of the network [42]. After eye feature stacking, as described above, spacial information and high level features from left and right eyes are stored in different channels of fused feature map. To summary, the SE layers are used for adaptively balancing spatial information and complex features from left and right eye according to two eye appearance (eye feature map values) and relationship (cross-channel information).

Note that right eye images are horizontally flipped in data processing. As we stack two eye features in channel wise, it is vital to ensure the extraction of two eye features are identical. To address such issue, we 1) use shared weights CNN to process eye images, and 2) ensure the appearance of two eyes stays the same by flipping. Through flipping, the position of inner and outer eye corners, the direction of eyebrows are consistent in left and right eye images due to the different eye orientation. This makes it easier for shared weights Eye Net to extract eye features like iris for the different appearance, further improves the final performance.

C. Adaptive Group Normalization

To fuse facial feature for a better gaze estimation, we propose the Adaptive Group Normalization (AdaGN) that adequately utilizes facial feature guidance. Though eye features are vital for gaze estimation, it is also difficult to extract proper eye features for eye appearance changes dramatically due to different factors like head pose, light condition and individual differences. The small region of the eye makes it difficult for the network to recognize all these changing factors. On the contrary, these factors always result in the difference of face location and appearance. Thus, it is necessary to consider face appearance characteristics during eye feature extraction procedure. In CNNs, some normalization layers contains scale and shift operation to enhance the representational power of normalized features. Inspired by them, we guide recalibration in Group Normalization with face appearance characteristics. We propose AdaGN which takes concatenated Rects features and facial features as input to represent face appearance characteristics. Then, AdaGN adaptively adjusts eye feature extraction according to face appearance characteristics by recalibrating eye features. According to different combination of head pose, light condition and other factors reflected at face appearance characteristics, AdaGN calculate scale and shift parameters to adaptively recalbrate eye features.

In detail, we first put Rects through four fully connected layers to generate a 64 dimensional vector which contains information about head translation. Then, the 64 dimensional feature vector is concatenated with facial feature to represent face appearance characteristics. The AdaGN works as follow equations:

$$\begin{cases} [W_{shift}, W_{scale}] = LeakyReLU(L(f_{Rects}, f_{face})), \\ f_{out} = W_{scale} * GN(f_{in}) + W_{shift}, \end{cases} \tag{2}$$

where $L(\cdot)$ stands for fully connected layer, $GN(\cdot)$ stands for normal Group Normalization without scale and shift. f_{in} is the original feature map and f_{out} is the final output of AdaGN. Rects features extracted by FC layers and facial feature extracted by face stream are represented as f_{Rects} and f_{face} in Equation 2. W_{scale} and W_{shift} are scale and shift parameters for each channel of f_{in}. AdaGN in eye net takes the Rects features and facial feature as input and calculate shift and scale by a fully connected layer. We choose Group Normalization rather than Batch Normalization for BN normalizes every channel in batch wise and neglects cross-channel relationship, adding disturbance to the channel weights given by SE layers. In this way, AFF-Net extracts eye features adaptively based on face appearance characteristics, generates more reasonable eye features for accurate gaze tracking.

D. Point of gaze prediction

In this section, we describe the complete architecture of proposed AFF-Net. For eye streams, feature maps from third and fifth convolutional layers are stacked to forge fused eye feature map. Then, another conv layer produces the final eye feature. AdaGNs are used as normalization layers. For face stream, facial feature is extracted by 6 conv layers. Last three conv layers are followed by SE layers to enhance representational power. For Rects stream, 64 dimensional feature vector generated by four FC layers is used as both the input of AdaGN and final Rects feature. Eye, face and Rects features are concatenated and fed to two FC layers to predict 2D gaze position of screen. We use ReLU as excitation layers for all convolutional layers and LeakyReLU for all FC layers. Similar to [20], we use physical distance related to the camera as label

to achieve device independence. Adam optimizer and Smooth L1 Loss function are used while training.

TABLE I
GAZE ESTIMATION ERROR IN CENTIMETERS COMPARES WITH SOTA METHODS ON THE GAZECAPTURE DATASET.

Method	Phone error (cm)	Tablet error (cm)
iTracker [20]	1.86	2.81
SAGE [40]	1.78	2.72
TAT [41]	1.77	2.66
AFF-Net	**1.62**	**2.30**

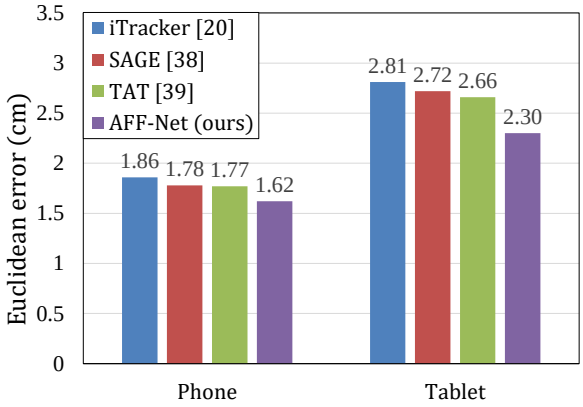

Fig. 4. Gaze estimation error in centimeters compared with SOTA methods on the GazeCapture dataset.

TABLE II
2D GAZE POSITION ESTIMATION ERROR IN CENTIMETERS AND 3D GAZE DIRECTION ESTIMATION ERROR IN DEGREES COMPARED WITH OTHER MAJOR METHODS ON THE MPIIFACEGAZE DATASET.

Method	2D gaze location error (cm)	3D gaze direction error (degree)
iTracker [20]	5.46	6.2
Spatial weights CNN [16]	4.2	4.8
RT-GENE [22]	4.2	4.8
AFF-Net	**3.9**	**4.4**

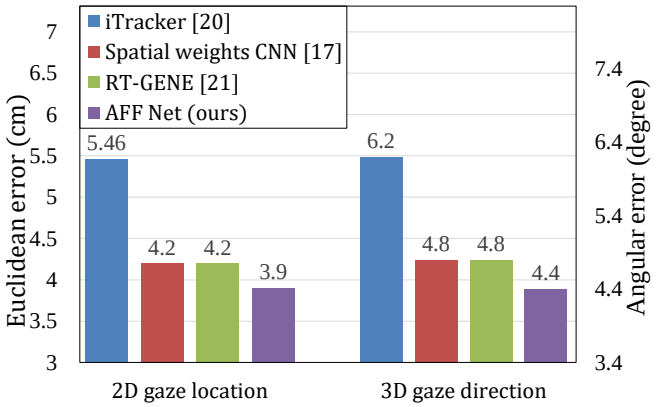

Fig. 5. 2D gaze position estimation error and 3D gaze direction estimation error compared with other major methods on the MPIIFaceGaze dataset.

IV. EXPERIMENTS

A. Setup

1) Dataset: We conduct experiments in two famous dataset with 2D gaze labels: GazeCapture [20] and MPIIFaceGaze [14].

GazeCapture is a large 2D gaze dataset with 2,445,504 images from ∼1,500 subjects. The GazeCapture dataset is collected using crowdsourcing. The dataset is captured by mobile phones or tablets in different orientations. There are 1,490,959 frames have both face and eye detections, which are further divided into 1,251,983 training images, 59,480 validation images and 179,496 test images. GazeCapture provides gaze positions represented as pixel location on screen and physical distance to the camera. We use physical distance to the camera in both training and testing for its device independence.

MPIIFaceGaze contains 213,659 images from 15 subjects. It is a commonly used dataset for gaze estimation problem. MPIIFaceGaze has a larger prediction space than GazeCapture for it is collected by laptops. The dataset provides physical screen sizes and 2D gaze position in pixels. So we convert pixel coordinates of the gaze to physical coordinates relative to the screen.

2) Data processing: For the GazeCapture dataset, we directly crop face and eye images according to face detection results of python face-recognition lib. Face images are resized to 224*224*3, and eye images are resized to 112*112*3. Pixel values are normalized into [0, 1]. For the MPIIFaceGaze dataset, we calculate face and eye bounding boxes from 6 facial landmarks in the dataset label. Specifically, we take 1.7 times eye x coordinates width as eye bounding box size and the average of eye coordinates as eye center. We set the face center as the mid point of the average eye coordinates and the average mouth coordinates. Face bounding box size is $1/0.3$ times eye bounding box size. We use the same image resolution settings as in the GazeCapture dataset. In GazeCapture, we follow the given train and test set. To simulate settings without calibration, we do leave-one-person-out test and average results from all subjects as final performance in MPIIFaceGaze.

3) Implementation details: The face stream has 6 conv layers. The specific parameters represented as (out channels, kernel size, stride, padding) are [(48, 5, 2, 0), (96, 5, 1, 0), (128, 5, 1, 2), (192, 3, 1, 1), (128, 3, 2, 0), (64, 3, 2, 0)]. 3x3

max pooling layers with stride 2 is used after the second and third conv layers. Two FC layer further compress the facial feature to a 64 dimensional vector. Rects input is processed by four FC layers with output channels as (64, 96, 128, 64).

The eye stream has 5 conv layers with parameters as [(24, 5, 2, 0), (48, 5, 1, 0), (64, 5, 1, 1), (128, 3, 1, 1), (64, 3, 1, 1)]. Max pooling layers in eye stream is the same as in face stream. After feature maps from third and fifth conv layers are fused, a (64, 3, 2, 1) conv extracts the joint two eyes feature map. One FC layer compress the two eyes feature map to a 128 dimensional vector. The final eye, face and Rects feature vectors are concatenated and two FC layers with 128, 2 dimensional outputs derive the gaze position coordinates. For the left and right eye networks share weights, right eye image is flipped to keep the eye orientation consistent. SE layers are added after second, forth, last conv layers and stacking.

The model is trained for 12 epochs on the GazeCapture dataset with Smooth L1 Loss function. The learning rate is 0.001 and reduce to 0.0001 after 8 epochs. Batch size is set to 256. On the MPIIFaceGaze dataset, the model is trained for 17 epochs under same settings. Similar to [20], [40], we add a random shift from 0 to 30 pixels to the face and eye positions while training to improve the robustness of the model. For testing, we report Euclidean distance between prediction and ground truth in centimeters. The whole experiments are implemented using PyTorch. Network parameters are initialized by the default initialization of PyTorch.

B. Performance

The GazeCapture dataset is almost the largest gaze dataset in mobile device. We first conduct performance evaluation of our method on the GazeCapture dataset. We choose three methods for comparison on GazeCapture, which are SAGE [40], TAT [41] and iTracker [20]. To the best of our knowledge, TAT shows the state-of-the-art performance on GazeCapture. We illustrate the result in Figure 4 and list the result in Table I. Our AFF-Net achieves 1.62 cm error on mobile phone captured images and 2.30 cm error on tablet captured images, outperforms state-of-the-art methods. For mobile phone image test, the earliest iTracker has the highest error as 1.86 cm. SAGE and TAT has similar performance around 1.77 cm, improve about 5% from iTracker. Our AFF-Net achieves 1.62 cm error, outperforms other methods significantly. The AFF-Net improves about 8.5% from SAGE and TAT. For the more challenging tablet image test, the error of iTracker is 2.81 cm. Different from mobile phone image test, TAT achieves 2.66 cm error, which is 0.06 cm lower than SAGE. Our AFF-Net also has the lowest error at 2.30 cm, which significantly improves 13.5% from the second best method TAT. As there are only about 15% images are from tablets, the results show that AFF-Net learns proper features faster than other methods. In the mean time, our AFF-Net only has 1.94M parameters. It is 3 times fewer compares to the iTracker which has 6.29M parameters. These experiment results show that AFF-Net has a clear advantage compare with other methods, especially in tablet images which is less in total amount.

TABLE III
EUCLIDEAN ERROR IN CENTIMETERS ON THE GAZECAPTURE DATASET FOR ABLATION STUDY.

Method	GazeCapture	
	Phone (cm)	Tablet (cm)
AFF-Net	**1.62**	**2.30**
without ST	1.67	2.39
without SE	1.68	2.31
without AdaGN	1.69	2.33

To further demonstrate the advantage of our method, We conduct more experiments on the MPIIFaceGaze dataset. We calculate face and eye bounding boxes according to provided facial landmarks and convert the screen pixel coordinates of targets to physical distance. We choose iTracker [20] and Spacial Weights CNN [16] as the compared methods, since they both show outstanding performance in the 2D gaze position estimation task on MPIIFaceGaze. Meanwhile, since MPIIFaceGaze is popular used in 3D gaze direction estimation task, we also select RT-GENE [22] as compared method for providing convinced comparison. The RT-GENE almost shows start-of-the-art performance in 3D gaze direction estimation task on MPIIFaceGaze. In order to provide more comprehensive comparison, we further convert the 2D gaze positions result estimated from our AFF-Net into 3D gaze directions according to provided camera-screen calibration matrix. As can be seen in Figure 5 and Table II, our method achieves the performance of 3.9 cm Euclidean error and 4.4 degree angular error, which significant performs better than other compared methods. Note that, the MPIIFaceGaze dataset is collected from laptop. This result demonstrate that our method also can perform well in the laptop.

C. Ablation Study

In this section, we conduct ablation experiments to prove the advantage of each proposed method. We focus on three component described above: stacking (*ST*), SE layers for eye feature fusion (*SE*) and adaptive Group Normalization (*AdaGN*). Specifically, to ablate SE layers, we simply remove all SE layers in the AFF-Net and keep other components the same. This architecture is denoted as *No SE* in III. To ablate AdaGN, the normal GN without extra input is used instead of AdaGN with Rects input. When ablates stacking, we carefully ensure the number of conv layers keeps the same for fair comparison. We remove the stacking architecture and directly move the conv layer after stacking to the end of eye stream. Then, the feature maps from left and right eyes are reshaped to feature vectors and processed by FC layers like classic methods.

The results of ablation study is shown in Table III. Complete AFF-Net with all three components achieves the highest accuracy. Removing of stacking architecture increases error on both phone images and tablet images by 0.05 cm and 0.09 cm respectively. Ablation of AdaGN causes a 0.07 cm error

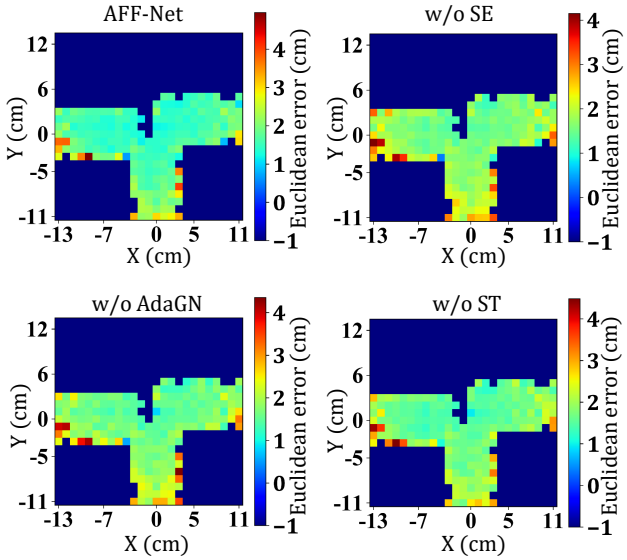

Fig. 6. Spacial heat maps of Euclidean error in centimeters on the GazeCapture dataset for different models.

increase on phone images. Model without SE layers also has a 0.06 cm error increase on phone images, while the accuracy on tablet images remains nearly the same. The results show that removing any one of three components will increase the Euclidean error by about 4%, proves the effect of proposed stacking architecture, SE layers and AdaGN.

D. Result Analysis

In this section, we further analyze the result of different models presented in ablation study on phone images of the GazeCapture dataset. Figure 6 shows heat maps of gaze estimation error in centimeters for different models. The camera is at the (0, 0) in every heat map. Gaze locations result in only three branches for GazeCapture does not contain phone test images with upside down orientation. As can be seen for all four heat maps, gaze estimation error increases as gaze location moves away from camera. The error of AFF-Net clearly increases slower than other three ablated networks, proving stronger robustness for different gaze location.

Figure 7 shows the curve of gaze estimation error relative to reciprocal of face width. Specifically, we set X axis as the reciprocal of face width relative to the shorter axis of original image. We choose it as X axis for it generally reflects the user-camera distance as face appears smaller when user moves away from camera, although it is disturbed by camera length and individual differences. As shown in Figure 7, the AFF-Net evidently outperforms other models, especially in images with very small face size. This results show the better robustness of AFF-Net against extreme cases.

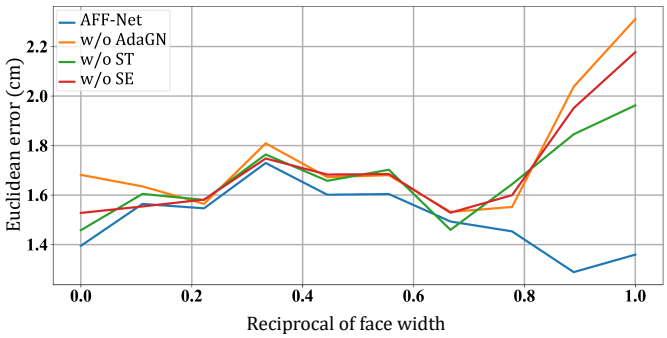

Fig. 7. Curve of Euclidean error in centimeters on GazeCapture dataset relative to the reciprocal of face width.

V. Conclusion

In this paper, we propose an accurate appearance-based gaze estimation method named AFF-Net. The proposed AFF-Net improves gaze tracking accuracy by adaptively fusing two eye features and face appearance characteristics guided eye feature extraction. In particular, we propose a stacking architecture with SE layers to fuse two eye features. Inspired by the identical structure of two eyes, we stack feature maps from different eyes and derive final feature map by convolutional layers. SE layers are added in the fusion process to adaptively weight two eye features according to their appearance. In addition, we also propose to calculate shift and scale parameters in Group Normalization based on face appearance characteristics to realize eye feature recalibration. The AFF-Net which combines above methods achieves state-of-the-art performance on both GazeCapture and MPIIFaceGaze dataset, proves the effectiveness of proposed network.

References

[1] H. S. Yoon, N. R. Baek, N. Q. Truong, and K. R. Park, "Driver gaze detection based on deep residual networks using the combined single image of dual near-infrared cameras," *IEEE Access*, vol. 7, pp. 93 448–93 461, 2019.

[2] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-time imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[3] D.-P. Fan, M.-M. Cheng, J.-J. Liu, S.-H. Gao, Q. Hou, and A. Borji, "Salient objects in clutter: Bringing salient object detection to the foreground," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 186–202.

[4] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 1, pp. 20–33, 2017.

[5] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang, "Inferring salient objects from human fixations," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[6] X. Zhang, Y. Sugano, and A. Bulling, "Everyday eye contact detection using unsupervised gaze target discovery," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 193–203.

[7] T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst, "Exploring natural eye-gaze-based interaction for immersive virtual reality," in *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2017, pp. 36–39.

[8] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke, "Perceptually-based foveated virtual reality," in *ACM SIGGRAPH 2016 Emerging Technologies*, 2016, pp. 1–2.

[9] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze prediction in dynamic 360 immersive videos," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5333–5342.

[10] K. Alberto Funes Mora and J.-M. Odobez, "Geometric generative gaze estimation (g3e) for remote rgb-d cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1773–1780.

[11] X. Xiong, Z. Liu, Q. Cai, and Z. Zhang, "Eye gaze tracking using an rgbd camera: a comparison with a rgb solution," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 1113–1121.

[12] L. Sun, Z. Liu, and M.-T. Sun, "Real time gaze estimation with a consumer depth camera," *Information Sciences*, vol. 320, pp. 346–360, 2015.

[13] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *IEEE TRANSACTIONS on biomedical engineering*, vol. 54, no. 12, pp. 2246–2260, 2007.

[14] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4511–4520.

[15] R. Ranjan, S. De Mello, and J. Kautz, "Light-weight head pose invariant gaze tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2156–2164.

[16] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 51–60.

[17] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Gaze estimation from eye appearance: A head pose-free method via eye image synthesis," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3680–3693, Nov 2015.

[18] 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, no. 5-6, pp. 445–461, 2017.

[19] 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.

[20] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2176–2184.

[21] Z. Chen and B. E. Shi, "Appearance-based gaze estimation using dilated-convolutions," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 309–324.

[22] T. Fischer, H. Jin Chang, and Y. Demiris, "Rt-gene: Real-time eye gaze estimation in natural environments," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 334–352.

[23] Y. Cheng, X. Zhang, F. Lu, and Y. Sato, "Gaze estimation by exploring two-eye asymmetry," *IEEE Transactions on Image Processing*, vol. 29, pp. 5259–5272, 2020.

[24] Y. Cheng, S. Huang, F. Wang, C. Qian, and F. Lu, "A coarse-to-fine adaptive network for appearance-based gaze estimation." in *AAAI*, 2020, pp. 10 623–10 630.

[25] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Transactions on biomedical engineering*, vol. 53, no. 6, pp. 1124–1133, 2006.

[26] A. Nakazawa and C. Nitschke, "Point of gaze estimation through corneal surface reflection in an active illumination environment," in *European Conference on Computer Vision*. Springer, 2012, pp. 159–172.

[27] 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.

[28] F. Lu, Y. Gao, and X. Chen, "Estimating 3d gaze directions using unlabeled eye images via synthetic iris appearance fitting," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1772–1782, 2016.

[29] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International journal of computer vision*, vol. 15, no. 1-2, pp. 123–141, 1995.

[30] K. Wang and Q. Ji, "Real time eye gaze tracking with 3d deformable eye-face model," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1003–1011.

[31] Q. Wen, D. Bradley, T. Beeler, S. Park, O. Hilliges, J.-H. Yong, and F. Xu, "Accurate real-time 3d gaze tracking using a lightweight eyeball calibration," *Computer Graphics Forum (proceedings of EUROGRAPHICS)*, 2020.

[32] K.-H. Tan, D. J. Kriegman, and N. Ahuja, "Appearance-based eye gaze estimation," in *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.* IEEE, 2002, pp. 191–195.

[33] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 10, pp. 2033–2046, 2014.

[34] O. Williams, A. Blake, and R. Cipolla, "Sparse and semi-supervised visual mapping with the sˆ 3gp," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 230–237.

[35] F. Lu, X. Chen, and Y. Sato, "Appearance-based gaze estimation via uncalibrated gaze pattern recovery," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1543–1553, 2017.

[36] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3d gaze estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1821–1828.

[37] K. A. Funes Mora, F. Monay, and J.-M. Odobez, "Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 255–258.

[38] W. Zhu and H. Deng, "Monocular free-head 3d gaze tracking with deep learning and geometry constraints," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3143–3152.

[39] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[40] 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 *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[41] T. Guo, Y. Liu, H. Zhang, X. Liu, Y. Kwak, B. In Yoo, J.-J. Han, and C. Choi, "A generalized and robust method towards practical gaze estimation on smart phone," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[42] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.