GEFF: Gaze Estimation with the Fused Features

Haoyu Zhen, Yilin Sun
Shanghai Jiao Tong University, Shanghai, China
{anye_zhen, yilin.sun}@sjtu.edu.cn

Abstract

Gaze articulation plays a vital role in understanding human behaviors. In this project, we transfer the model PIXIE [3] from the human body's reconstruction to gaze estimation. Also, we implement the SimCLR [1, 2] which is a framework for contrastive learning of visual representations. Our code is available at https://github.com/anyeZHY/GEFF.

1. Introduction

The contributions of this project are as follows:

1. We transferred PIXIE[3] model from 3D human body reconstruction to gaze estimation. We modified the network architecture to make it better suited to our task. Now the features of the head fuse with that of the eyes (we call our model GEFF).
2. We implemented SimCLR[1, 2] framework for training deep and complicated network (Currently the SimCLR framework was adapted for GEFF).
3. We applied data augmentation. The methods include but are not limited to flipping the images horizontally, swapping the left eyes and the right eyes, and using masks for model generalization.

2. Fused Feature

Previous work often simply concatenates the features of the eyes and that of the head. Inspired by PIXIE [3], we implement a fusion of them instead. The fusion of features is contingent on the inductive bias that the whole face will imply the information of gaze articulation.

2.1. FE Baseline

FE (Face-Eye) Baseline is our foundation which uses pre-trained ResNet18 and MLPs to build a naive network. In the FE Baseline model we do take both the features from the face and eyes as input and use them to optimize the loss function, yet the connection between face and eyes, or what we define as fused features, is neglected.

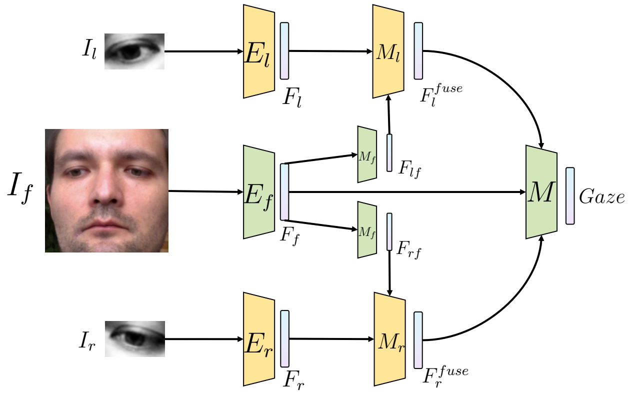

Figure 1. GEFF framework in gaze estimation task

The FE Baseline, despite its relatively poor performance compared with our advanced network architectures, is our benchmark and provides criteria for other networks which apply fused features.

2.2. Vanilla Fusion

The Vanilla Fusion, which applies the core idea of feature fusion, can be viewed as a fundamental version of our upcoming GEFF network. Our idea is that Vanilla Fusion can be viewed as a transition between the baseline and our relatively sophisticated network.

The core idea of fusion is that we generate a fused feature F_{fuse} for each eye by a weighted sum:

$$F_{fuse} = (1 - w)F_{eye} + wF_{face} \tag{1}$$

Here the weight parameter w represents the confidence we assign to face feature F_{face} and eye feature F_{eye} and is fixed equal for both eyes(whereas in GEFF w is learnable and could be different for two eyes).

2.3. GEFF Architecture

The architecture of our GEFF network is inspired by PIXIE[3]. The description of the functions of different components in the network is partly quoted from PIXIE[3] as well. We then built the network by our original work.

GEFF uses the architecture of Figure 1 and is trained end to end. All model components are described below.

Input images: Given an image I with full resolution, we assume a bounding box around the face. (Actually for the MPII dataset the input images are already preprocessed, so here we are describing the general idea of data preprocessing which can be implemented on the ColumbiaGaze dataset) We use this to crop and downsample the image I and get our face I_f, left eye I_l, and right eye I_r.

Feature encoding: We feed $\{I_f, I_l, I_r\}$ to separate expert encoders $\{E_f, E_l, E_r\}$ to extract features $\{F_f, F_l, F_r\}$. We use MLP, ResNet-18, or our pre-trained models from baseline for face and eyes encoders to generate their features respectively.

Feature fusion (moderator): A moderator is implemented as a MLP which gets various features $\{F_f, F_l, F_r\}$ extracted by our encoders as input. We train two types of moderators, $\mathcal{M}_f$ and $\mathcal{M}_l(\mathcal{M}_r)$. We first feed F_f to moderator $\mathcal{M}_f$ and split the output as new features F_{lf} and F_{rf}. We then feed $F_p, F_{pf}(p \in \{l, r\})$ to $\mathcal{M}_p$ and fuse them with a weighted sum:

$$F_p^{fuse} = (1 - w_p)F_p + w_p F_{pf} \tag{2}$$

$$w_p = \frac{1}{1 + \exp(-t \cdot \mathcal{M}_p(F_p, F_{pf}))} \tag{3}$$

where w_p represents the encoder's confidence and t is a hyper-parameter.

Gaze decoding(moderator): Having generated all fused features $\{F_l^{fuse}, F_r^{fuse}\}$, we would use another moderator $\mathcal{M}$ to combine them with face feature F_f to decode our $Gaze$ for further regression.

2.4. Training Losses

Simply, we use L-1 loss to train our model in gaze estimation task.

$$\mathcal{L}_1 = \|y_{\text{pred}} - y\|_1 \tag{4}$$

3. SimCLR Framework

With the advent of the deep and complicated eyes' encoder, the performace of our models are saturated severely. To ameliorate this effect, we use SimCLR [1, 2] framework to pretrain our model.

The SimCLR framework is shown in Figure2. The components are described below.

Data Augmentation: The data augmentation process takes face, left eye and right eye $\{f, l, r\}$ as input and provide two group of similar images $\{\tilde{f}_i, \tilde{l}_i, \tilde{r}_i\}$ and $\{\tilde{f}_j, \tilde{l}_j, \tilde{r}_j\}$ for our further implementation.

Feature Encoding: We feed the augmented data $\{\tilde{f}_i, \tilde{l}_i, \tilde{r}_i\}$ and $\{\tilde{f}_j, \tilde{l}_j, \tilde{r}_j\}$ to two identical encoders to extract features $\{F_r^{(i)}, F_l^{(i)}, F_f^{(i)}\}$ and $\{F_r^{(i)}, F_l^{(i)}, F_f^{(i)}\}$ respectively. We reserve larger face feature for the sake of feature projection.

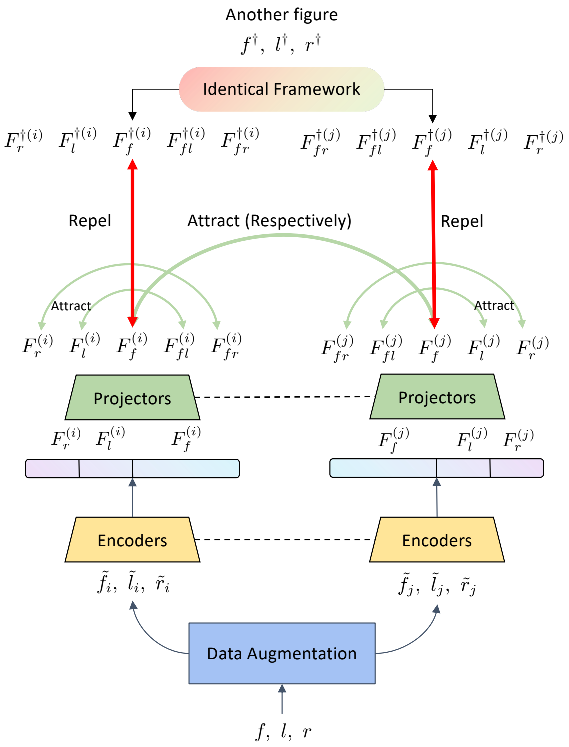
Another figure
$f^\dagger,\ l^\dagger,\ r^\dagger$
Identical Framework
$F_r^{\dagger(i)}$ $F_l^{\dagger(i)}$ $F_f^{\dagger(i)}$ $F_{fl}^{\dagger(i)}$ $F_{fr}^{\dagger(i)}$
$F_{fr}^{\dagger(j)}$ $F_{fl}^{\dagger(j)}$ $F_f^{\dagger(j)}$ $F_l^{\dagger(j)}$ $F_r^{\dagger(j)}$
Repel
Attract (Respectively)
Repel
Attract
Attract
$F_r^{(i)}$ $F_l^{(i)}$ $F_f^{(i)}$ $F_{fl}^{(i)}$ $F_{fr}^{(i)}$
$F_{fr}^{(j)}$ $F_{fl}^{(j)}$ $F_f^{(j)}$ $F_l^{(j)}$ $F_r^{(j)}$
Projectors
Projectors
$F_r^{(i)}$ $F_l^{(i)}$ $F_f^{(i)}$
$F_f^{(j)}$ $F_l^{(j)}$ $F_r^{(j)}$
Encoders
Encoders
$\tilde{f}_i,\ \tilde{l}_i,\ \tilde{r}_i$
$\tilde{f}_j,\ \tilde{l}_j,\ \tilde{r}_j$
Data Augmentation
$f,\ l,\ r$

Figure 2. SimCLR framework in gaze estimation task

Feature Projection: We use projectors to generate more features from the face and eyes. Specifically, the features will pass fully connected layers and a activation function. For the face feature F_f, the projector will generate $\{F_f, F_{fl}, f_{fr}\}$, combined with the features from eyes $\{F_l, F_r\}$, we finally collected two larger groups of features $\{F_l^{(i)}, F_r^{(i)}, F_f^{(i)}, F_{fl}^{(i)}, f_{fr}^{(i)}\}$, $\{F_l^{(j)}, F_r^{(j)}, F_f^{(j)}, F_{fl}^{(j)}, f_{fr}^{(j)}\}$.

Feature Attraction and Repulsion: To illustrate how the features within and between groups attract and repel each other, first we introduce a similarity function between vectors and a contrastive loss function.

The similarity of vector $\boldsymbol{u}$ and $\boldsymbol{v}$ read

$$\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^T \boldsymbol{v}}{\|\boldsymbol{u}\| \|\boldsymbol{v}\|} \tag{5}$$

Then we introduce Contrastive Loss Function

$$l_{\boldsymbol{z}}(i, j) = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{k \neq i} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)} \tag{6}$$

where $\mathbb{1}$ is a indicator. Then the contrastive loss is shown as

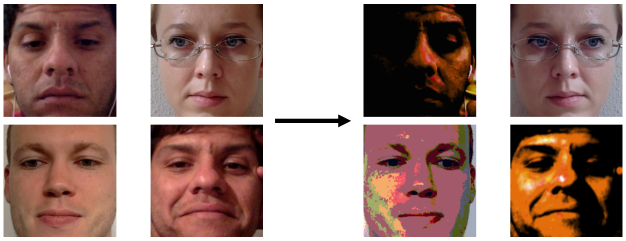

Figure 3. Left: original images. Right: transformed images.

follows.

$$\mathcal{L}(\boldsymbol{z}) = \frac{1}{2N}\sum_{k=1}^{N}\Big[l_{\boldsymbol{z}}(2k-1,2k) + l_{\boldsymbol{z}}(2k,2k-1)\Big] \quad (7)$$

The function $\mathcal{L}$ is defined on a single pair of features between groups, for example $F_f^{(i)}$ and $F_f^{(j)}$. Our goal is to let features between groups i and j attract each other respectively, and let features for the same eye within a group, F_l, F_{fl} and F_r, F_{fr} attract other as well. We use an identical framework and get another figure as input, the features from different figures should repel each other. The feature attraction and repulsion are equivalent to optimizing over the contrastive loss function $\mathcal{L}$:

$$\begin{aligned}\mathcal{L} = \quad & \mathcal{L}\Big([\boldsymbol{F}_f^{(i)};\boldsymbol{F}_f^{(j)}]\Big)+ \\ & \lambda_1\mathcal{L}\Big([\boldsymbol{F}_l^{(i)};\boldsymbol{F}_l^{(j)}]\Big) + \lambda_1\mathcal{L}\Big([\boldsymbol{F}_r^{(i)};\boldsymbol{F}_r^{(j)}]\Big)+ \\ & \lambda_2\mathcal{L}\Big([\boldsymbol{F}_l^{(i)};\boldsymbol{F}_{fl}^{(i)}]\Big) + \lambda_2\mathcal{L}\Big([\boldsymbol{F}_r^{(i)};\boldsymbol{F}_{fr}^{(i)}]\Big)+ \\ & \lambda_2\mathcal{L}\Big([\boldsymbol{F}_l^{(j)};\boldsymbol{F}_{fl}^{(j)}]\Big) + \lambda_2\mathcal{L}\Big([\boldsymbol{F}_r^{(j)};\boldsymbol{F}_{fr}^{(j)}]\Big)\end{aligned} \quad (8)$$

where $[a;b]$ means the vertical concatenation of a and b. (E.g., $a \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{k\times n} \implies [a;b] \in \mathbb{R}^{(m+k)\times n}$)

4. Experiments

We train our models on 2 datasets MPIIGaze [5] and ColumbiaGaze [4].

4.1. Implementation Details

4.1.1 Data Augmentation

In training part, we implement 2 traditional data augmentation methods: random color jitter (w.r.t. the probability $p = 0.6$) and random grayscale (w.r.t. $p = 0.2$). Additionally, we design some special ways for gaze estimation tasks to augment data:

Flipping the images horizontally. As shown in the upper right corner of the figure 3, we flipped the face and eyes horizontally. This approach is naturally concomitant with exchanging left and right eye images.

Using masks on eyes. To generalize our model, we implement the 'Mask' trick on the images of eyes. For any given $\mathcal{I}_{\text{eye}}$ which belongs to $\mathcal{I}_{\text{face}}$, we first set a random number $p \sim \text{Uniform}(0,1)$ and transform $\mathcal{I}_{\text{eye}}$ by following rules:

$$\text{Mask}(\mathcal{I}) = \begin{cases} \text{Gaussian Noise} & \text{if} \quad p \in (0, 0.05] \\ \text{Crop from } \mathcal{I}_{\text{face}} & \text{if} \quad p \in (0.05, 0.2] \\ \mathcal{I}_{\text{eye}} & \text{if} \quad p \in (0.2, 0.6] \\ T(\mathcal{I}_{\text{eye}}) & \text{otherwise} \end{cases} \quad (9)$$

where T denotes the traditional method for data augmentation.

4.1.2 Warming Up and Fine-tuning

To reach the best model, we train GEFF by fine-tuning the baseline. The backbone parameters will not be updated until the epoch is greater than 30.

4.2. Results

The criterion of gaze estimation task is angular loss which is defined as:

$$\mathcal{L}_{\text{ang}}(\boldsymbol{y}, \boldsymbol{y}_{\text{pred}}) = \arccos(\text{sim}(\boldsymbol{y}, \boldsymbol{y}_{\text{pred}})) \quad (10)$$

which will be used for model selection.

4.2.1 Gaze Estimation

MPIIGaze. Figure 4 and table 3 shows the experiment results on MPIIGaze dataset. The interpretation of the name are as follows:

- Base: ResNet without eye encoders
- GEFF-MLP: GEFF model whose eye encoder is MLP
- GEFF-RF: eye encoder is ResNet.

Models	Angular Loss (Average)	
	Cross Validation	Test on Folder 10-14
GEFF (**our full**)	**3.982**	**4.960**
Vanilla Fuse	4.220	5.170
Naive ResNet	4.104	7.213

Table 1. Results on MPIIGaze dataset

We find a bad performance when testing folder 7 containing black people. To ameliorate this, we use the SimCLR framework and fine-tune the downstream network. The training results are shown as figure 6. It is obvious that GEFF+SimCLR performs best, which means our model is less sensitive to the color of skins.

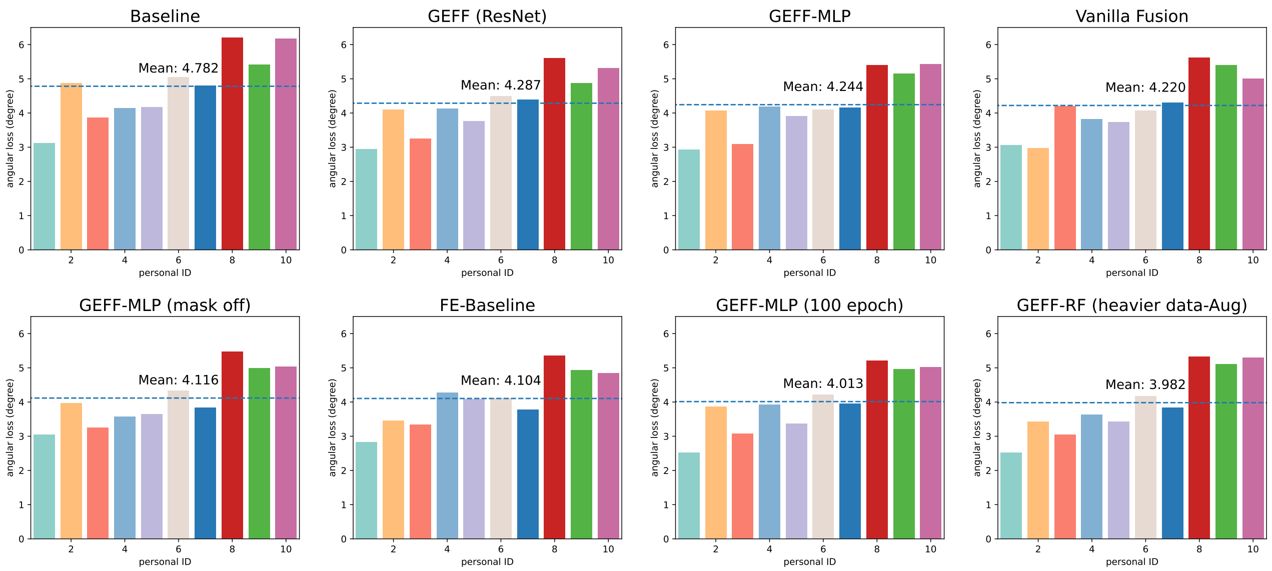

Figure 4. Cross validation on MPIIGaze dataset.

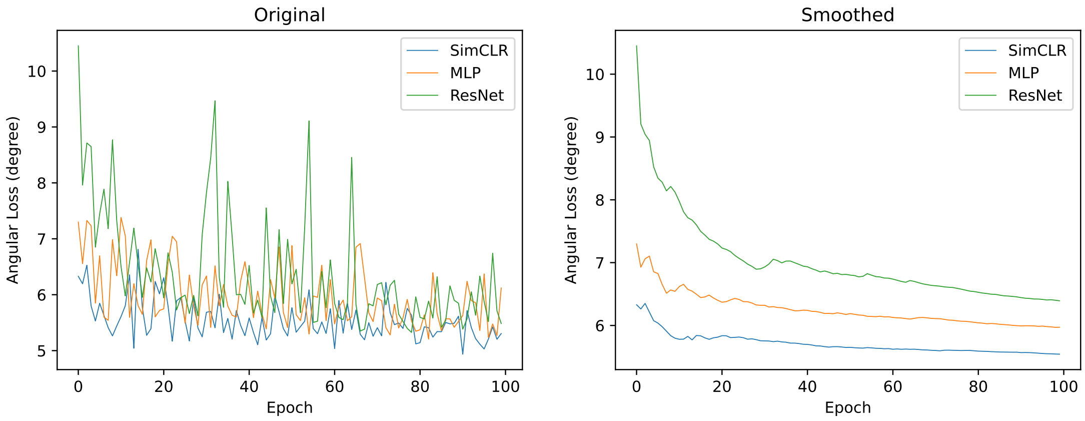

Figure 5. Test results on folder-7, MPIIGaze dataset (GEFF).

ColumbiaGaze. Figure 6 shows the results on the Columbia dataset. In this dataset, only 105 images are provided for each folder and there are 56 folders in total, so we do not use cross-validation. We use the same notations about models on the MPIIGaze dataset, and the validation loss is stunningly insightful.

As Figure 6 shows, the loss over FE-Baseline with the most simple structure would descend quickly. But the Vanilla Fusion model would eventually outperform it and wins in the long run. Finally, our GEFF-MFP model will show its dominance as it can both descend quickly and reach the lowest loss.

4.2.2 Domain Adaptation (Trial)

We use our best model for domain adaptation. The results can be found in Table 2

Methods	Angular Loss (Average)
Train on MPII, test on Columbia	13.166
Train on Columbia, test on MPII	14.859
Using SimCLR, test on Columbia	13.00+

Table 2. Results of Domain Adaptation

We then apply the SimCLR method and test on the Columbia dataset, as it turns out, the improvement is not that significant.

This result gives us possible research ideas. Although the SimCLR method seems to not prove an improvement good enough, should image on the Columbia dataset be allowed to us, we would be able to train a better model with SimCLR. We would only need a small number of images from Columbia, say about 100 images. Then by using MPI-

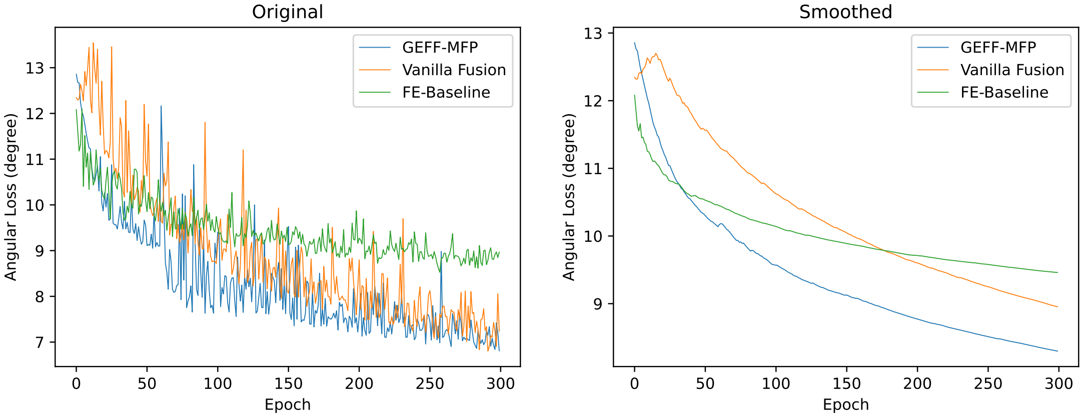

Figure 6. Caption

IGaze and Columbia together we would finally be able to do a better domain adaptation.

5. Conclusion

By solid research and numerous experiments, we finally draw the following conclusions:

1. Our original GEFF network with full technique outperforms all the other models on the gaze estimation task. To be specific, GEFF reaches an average angular loss of 3.982 with cross-validation and 4.960 on the test folders 10-14.
2. The SimCLR framework we adapted for training our sophisticated GEFF network is effective for finetuning the downstream network. To be specific, SimCLR can boost the performance of our models which apply fused features. Moreover, SimCLR can help to ameliorate the bias caused by the different colors of skins.
3. Data augmentation techniques represented by random color jitter, random grayscale, image flipping, and masks over eyes are helpful for model training. Inventing novel methods of data augmentation which are suitable for certain tasks is crucial for finding the best model.
4. Domain adaptation remains a difficult task, especially when adapting between two datasets with different image sizes and features. The potential of SimCLR on the domain adaptation task is promising,

Acknowledgement

We drew our inspirations from the papers [3, 1, 2] and the course CS231n, Stanford University.

The computations in this paper were run on the cluster supported by the high-performance computing platform of SJTU Student Innovation Center.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 5

[2] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 1, 2, 5

[3] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Collaborative regression of expressive bodies using moderation. In *International Conference on 3D Vision (3DV)*, 2021. 1, 5

[4] B.A. Smith, Q. Yin, S.K. Feiner, and S.K. Nayar. Gaze Locking: Passive Eye Contact Detection for Human?Object Interaction. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 271–280, Oct 2013. 3

[5] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015. 3

Appendix

Here we list the result of various models on the MPIIGaze dataset. Table 3 shows the performance of different models with various parameters and methods. Note that here we do not apply cross-validation, instead, we randomly select images from each folder to make up a validation set, and randomly shuffle all the other images not selected to generate the training set.

Although we did not apply cross-validation, the ablation study still can cast some light on the effects of different methods and help us find the model which can outperform all the other models.

Models	Configuration							Valid		
	Name	Eyes' Encoder	Data Aug	LR Scheduler	Weight / t	Flip	Pretrain	Min	40-100	80-100
GEFF		MLP	✗	None	-1.0	0.0	✗	**1.826**	2.3086	2.1290
		MLP	✓	None	-1.0	0.0	✗	2.243	2.8895	2.6782
		MLP	✓	(20, 0.5)	-2.0	0.0	✗	1.970	2.4032	2.2772
		MLP	✓	(20, 0.5)	1.0	0.0	✗	2.008	**2.1511**	**2.1154**
	M-Full	MLP	✓	(30, 0.5)	1.0	0.5	✗	2.059	2.3472	2.2857
		ResNet	✗	None	-1.0	0.0	✗	2.094	2.5950	2.4061
		ResNet	✓	None	-1.0	0.0	✗	2.330	2.7046	2.6154
	R-Full	ResNet	✓	(30,0.5)	1.0	0.5	✗	2.227	2.2816	2.2555
		MLP	✓	(20, 0.5)	1.0	0.0	✓	1.965	2.1579	2.0956
	M-F-P	MLP	✓	(20, 0.5)	1.0	0.5	✓	**1.351**	**1.3714**	**1.3689**
		ResNet	✓	(20,0.5)	2.0	0.5	✓	1.597	1.7863	1.7930
	R-F-P	ResNet	✓	(20,0.5)	1.0	0.5	✓	1.449	1.4757	1.4735
Vanilla Fusion		MLP	✗	None	0.2	0.0	✗	1.863	2.1570	2.0912
		MLP	✓	None	0.2	0.0	✗	2.091	2.6733	2.3215
		MLP	✓	(20, 0.1)	0.2	0.0	✗	2.004	2.1591	2.1663
		MLP	✓	(20, 0.1)	0.1	0.0	✗	2.247	2.3386	2.3326
	M-Full	MLP	✓	(30, 0.5)	0.2	0.5	✗	**1.740**	**2.1073**	**1.9345**
		Resnet	✓	None	0.2	0.0	✗	2.388	2.9681	2.6670
		Resnet	✓	(20, 0.1)	0.2	0.0	✗	2.236	2.3018	2.3046
	R-Full	Resnet	✓	(30, 0.5)	0.2	0.5	✗	1.977	2.2647	2.1073
		MLP	✓	(20, 0.5)	0.1	0.5	✓	**1.522**	**1.5874**	**1.5824**
	M-F-P	MLP	✓	(20, 0.5)	0.2	0.5	✓	1.553	1.6238	1.6305
		MLP	✓	(20, 0.5)	0.2	0.0	✓	1.677	1.8721	1.7674
	R-F-P	Resnet	✓	(30, 0.5)	0.2	0.5	✓	1.559	1.8476	1.7797
Baseline	Vac	None	✗	None	None	0.0	✗	1.762	1.9956	1.8916
	Aug	None	✓	None	None	0.0	✗	1.824	2.1104	2.0905
	Lr	None	✗	(20, 0.2)	None	0.0	✗	1.697	**1.7640**	**1.7464**
	AugLr	None	✓	(20, 0.7)	None	0.0	✗	**1.692**	1.8538	1.7802

Table 3. Ablation studys on MPII dataset.