HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 1 (2023)

A Method of Disentanglement of Latent Factor Using Geometric Feature for Gaze Estimation Network Training

Seung-woo Ko[1] and Bo Kyoung Park[1#]

[1]Elite Open School LRC Korea
[#]Advisor

ABSTRACT

Since each human eye has different anatomical features, gaze estimation is a very challenging task. Although numerous studies regarding gaze estimation were proposed, there is a need for improving the preciseness in order to facilitate the application of the method to real-world scenarios. To accomplish this goal, I propose a novel training strategy for gaze representation learning. The proposed training method includes two training phases: the autoencoder-based representation learning phase and the gaze estimation network training phase. The proposed training strategy enforces the trained model to disentangle the gaze-related latent code and produce a more accurate gaze estimation. In addition, I also propose and showcase a real-world application that exploits the proposed method in order to prove the practicality of the proposed method. Through the experiment, it is proven that the proposed method shows an outstanding performance compared to other methods on the Gaze360 dataset.

Introduction

Gaze estimation is a technique used for predicting the direction where a person is looking by referring to the person's entire face or eye area. Currently, it is receiving a lot of attention in computer vision fields since it is widely applicable in real-life scenarios such as detecting careless drivers or monitoring online classes. However, since each individual has different anatomical features, it is difficult to produce accurate gaze estimation.

Many studies have applied convolutional neural networks to solve gaze estimation problems. These methods directly predict gaze direction from the features extracted from convolutional layers. However, these methods tend to yield poor results since they engage with entangled features. To solve this problem, representation learning-based methods have widely been studied in order to disentangle the gaze-related latent code.

Cheng et al. proposed PureGaze which purifies the gaze feature for generalizable gaze estimation [1]. This method has shown successful disentanglement of gaze-related features from various characteristics, such as skin color, age, or illumination condition. PureGaze is trained in supervised learning approaches which require gaze annotation for its learning process. However, collecting these gaze annotations is impractical because it is very time-consuming and costly. Recently, Sun et al. proposed CrossEncoder that disentangles the gaze-related code by swapping the latent code [2]. Their method swaps the latent code on eye image pairs (left and right) which are assumed to be consistent such as appearance features. Inspired by the aforementioned CrossEncoder, Gideon et al. proposed a method that extends the latent code swapping mechanism on multi-view video. Their method can significantly leverage the existing dataset sample in disentangling the gaze-related latent code [3].

By taking over the flow of these studies, this paper suggests a novel method that uses representation learning. In this paper, I proposed a gaze estimation that manipulates the latent code in order to disentangle gaze-related features.

The proposed task is divided into two phases: autoencoder-based representation learning and gaze estimation training phase.

In the first phase, the input eye is compressed using an encoder and produces the latent code. The latent code is divided into two subparts: gaze and appearance. After that, a rotation matrix is applied to the gaze-related latent code. The decoder takes the rotated gaze and appearance latent code as input and reconstructs the image that shows the same person keeping their eye in a rotating direction. In the second phase, the trained encoder is used as a feature extractor for the gaze estimation network. The encoder takes the input image and produces the latent code that consists of two subparts: gaze and appearance. The purpose of the second phase is to train the proposed gaze estimation network with features that are disentangled via an encoder. The gaze estimation network takes the gaze part of the latent code and estimates the direction of the gaze represented as yaw and pitch.

Method

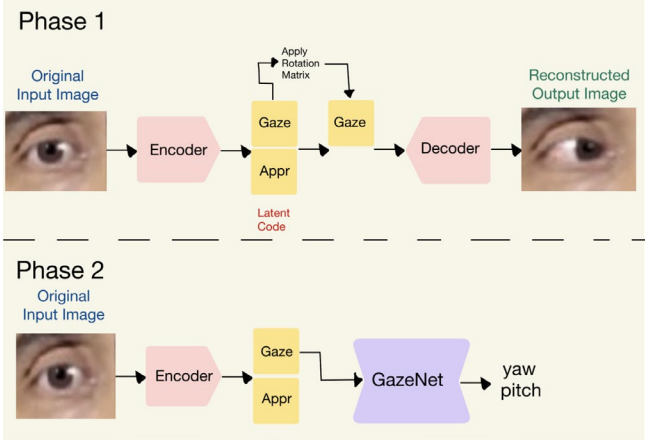

Figure 1. The proposed training pipeline for the Gaze Estimation System

Figure 1 represents the proposed training pipeline for the Gaze Estimation System. There are two different phases: the first phase and the second phase. In the first phase, autoencoder-based representation learning is conducted to disentangle the gaze-related latent code for better representation. The second phase of the proposed training pipeline completes the gaze estimation training by using the features that are disentangled in the first phase and estimates the direction of the gaze direction represented as yaw and pitch. In chapter 2.1, I will explain how I disentangle the gaze-related latent code from the entangled latent code in detail. The second gaze estimation network training phase will be explained in chapter 2.2 and the detailed implementation of the proposed system is explained in chapter 2.3.

2.1 Representation Learning

The purpose of the first phase is to disentangle the gaze-related latent code in order to make the trained model more robust against various input gaze samples. First of all, the Encoder takes input image I as input and produces Latent Code C as output. Here, I define the proposed Encoder as Enc: $I \rightarrow C$. Latent Code C is then divided into two subparts: gaze-related code C_{gaze} and appearance code C_{appr}. After that, the rotation matrix is applied to the gaze-related code C_{gaze}, and the Decoder Dec reconstructs the image Irecon using the appearance code C_{appr} and gaze-related code C_{gaze} that contains the application of the rotation matrix. Finally, the reconstructed image Irecon shows the rotated image. For example, in figure 1, the eye in input image I is looking to the left. However, it shows that the same eye in the reconstructed image I_{recon} is looking to the right since the rotation matrix has been applied.

Table 1. Two groups are broken down with age ranges and the difference.

Notation	
Encoder	*Enc*
Decoder	*Dec*
Gaze Estimation Network	*GazeNet*
Latent Code	C
Input Image	I
Reconstructed Image	I_{recon}
Ground Truth Image	I_{gt}
Predicted Gaze	G_{pred}
Ground Truth Gaze	G_{gt}
Gaze-related Code	C_{gaze}
Appearance	C_{appr}

2.2 Representation Learning

The second phase aims to train the proposed gaze estimation network with features that are disentangled via a trained encoder. Same to the first phase's image compression technique, the Encoder *Enc* takes input image I as input and produces latent code C as an output. Since this process is exactly the same as the first phase, the latent code C is divided into two subparts: gaze-related code C_{gaze} and appearance code C_{appr}. Since gaze-related code is the important factor for estimating the direction of the gaze, the Gaze Estimation Network GazeNet takes only the gaze-related code C_{gaze}, flattens it, and estimates the direction of the gaze as either yaw or pitch. This unique representation learning strategy allows the trained encoder to disentangle the gaze-related code thus improving the performance of the overall Gaze Estimation Network.

2.3 Implementation Details

The proposed method uses two types of loss functions to train the system. To train the proposed autoencoder architecture, I use the L_1 loss function which is often used for reconstruction networks [6]. Eq. (1) represents how the L_1 loss function operates with the reconstructed image and its according ground truth.

Equation 1: Equation of L_1 Loss Function operating with the reconstructed image and its according ground truth:

$$L_1 = \left| I - \hat{I} \right|_1$$

where, I indicate the original input image, while the reconstructed image is represented by I.

MSE (Mean Squared Error) is used for training the network in the second phase. Below is the equation for the loss function.

Equation 2: Equation for the loss function:

$$L_{mse} = (G_{gt} - G_{pred})^2$$

here, G_{gt} indicates the ground truth gaze and G_{pred} represents the predicted gaze. Final loss function L is then calculated as Eq. (3).

Equation 3: Calculation of the final loss function:

$$L = L_{mse} + \lambda L_1$$

Through the experiments, the quality of the performance is the highest when λ is 0.9.

The architecture of the proposed autoencoder is heavily based on Resnet [4]. For the Encoder, I exploit vanilla Resnet and for the Decoder, I add a few upsample layers to make the network reconstruct the original image resolution. The proposed Gaze Estimation Network is composed of three-linear layers. Through extensive experiments, I have found that the depth of the Gaze Estimation Network is deep enough to achieve accurate results.

To train the proposed method, I use Adam [5] optimizer with a learning rate of 0.0001. I set the batch size to 128 and train the system for 200 epochs. The initial learning rate is multiplied by 0.1 at 80 and 160 epochs. For data augmentation, a random horizontal flip is used to provide a wide range of dataset distribution to the trained system. Figure 2 shows an example of a random horizontal flip augmentation.

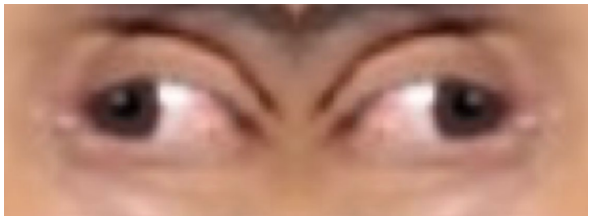

Figure 2. An example of a random horizontal flip

3. Experimental Results

3.1 Dataset

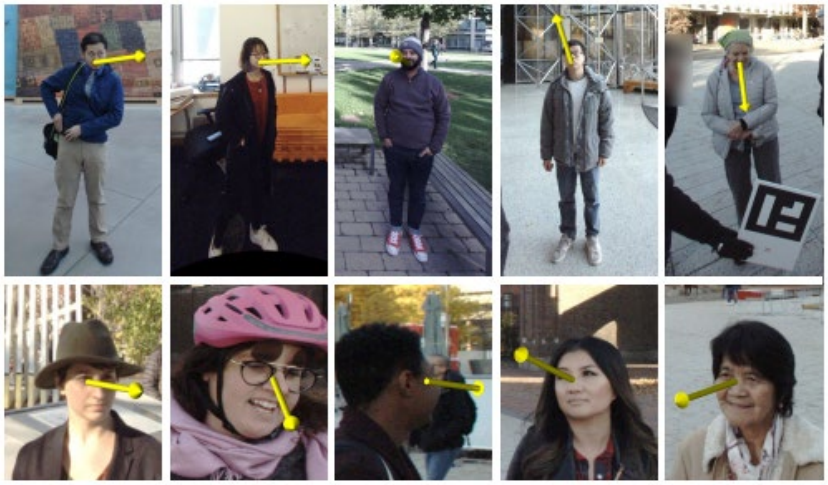

Figure 3. Gaze360 Sample Images.

Figure 3 shows a snippet of the Gaze360 dataset (Petr, et al. 2019) used in this paper. The dataset provides three-dimensional gaze annotation with a 360-degree range, and they were all used for training the proposed model and

testing the quality and efficiency of the model. There are a total of 238 subjects and the samples were collected in an outdoor environment. As shown in Fig. 3, the samples are captured in different types of illumination conditions and contain people of different ages and gender, which makes the gaze estimation problem more challenging. For a fair evaluation comparison, the same test set was used for the proposed method and previous state-of-the-art methods.

3.2 Evaluation Metric

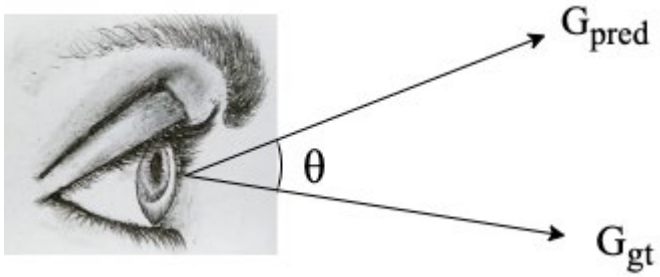

Figure 4. Example of angular error

For the evaluation metric, I used the angular error as shown in Fig. 4. In the figure above, G_{gt} represents the ground truth gaze vector and G_{pred} represents the predicted gaze vector. The angular loss explains how the predicted gaze vector is similar to its according ground truth gaze vector. It is calculated by measuring the value of $\boldsymbol{\theta}$, which is the angular difference between the prediction and the ground truth. The smaller the angular error is, the better performance the proposed model shows. The $\boldsymbol{\theta}$ is calculated as follows.

Equation 4: Angular error equation:

$$\theta = arccos(G_{pred} \cdot G_{gt} / |G_{pred}||G_{gt}|)$$

Where G_{pred} and G_{gt} represent the predicted gaze and ground truth, while indicating dot product. The range of $\boldsymbol{\theta}$ degree would be between 0 and 180, where 0° represents the best case while 180° represents the worst case.

3.3 Comparison with the state-of-the-art methods

Table 2. Angular Error Comparison between state-of-the-art method and the proposed method

Method	Angular Error (degree)
PNP-GA (Liu, et al. 2021)	14.57
PureGaze (Cheng, et al. 2022)	12.89
Ours	10.61

For the comparison methods, I chose two previous state-of-the-art methods; PNP-GA which are proposed by Liu, et al. (Liu, et al. 2021) and Cheng, et al.'s proposed model called PureGaze, (Cheng, et al. 2022) in order to measure the qualitative difference. These methods have comparable results and are proposed relatively recently.

PNP-GA achieves an angular error of 14.57° while the proposed method achieves an angular error of 10.61°. This shows that the proposed method outperforms the method of Liu, et al. by achieving a 3.96° lower angular error. The proposed method is also superior to PureGaze which achieves an angular error of 12.89°. The proposed method achieves 2.28° lower. Overall, the proposed method is more accurate than the previous state-of-the-art methods.

I attribute the superiority of the proposed model to the autoencoder-based representation learning strategy which is conducted to disentangle the gaze-related latent code for better representation. By applying the rotation matrix to the latent code, the proposed training strategy allows the trained model to successfully disentangle the gaze-related latent code which contains important information about the direction of the input eye.

3.4 Ablation Study

To prove the effectiveness of each component of the proposed method, I conducted the ablation study by measuring the difference in the degree of error between the full model and the ablation models. This comparison allows estimating the amount of contribution that the proposed autoencoder and rotation matrix make.

Table 3. Angular error comparison between full model and ablation models

Method	Angular Error (degree)
baseline (first ablation model)	14.20
baseline + autoencoder (second ablation model)	13.76
baseline + autoencoder + rotation matrix (full model)	10.61

Table 3 shows the difference in angular error between the full model and ablation models. The first ablation model is trained on regular CNN (Convolutional Neural Networks) without an autoencoder which directly predicts the yaw and pitch. I refer to this as a baseline. For the second ablation model, I keep the baseline and remove the rotation matrix technique from the proposed method. Finally, the full model is trained as explained in chapter 2.1. and 2.2.

The first ablation model achieves 14.20% of angular error, while the full model achieves 10.61% of angular error. Compared to the first ablation model, the full model achieves 3.59% fewer errors. By compressing the input image using the encoder, and reconstructing it using the decoder, an autoencoder forces an encoder to extract the important features. Also, by applying the rotation matrix to the latent code, the model becomes able to disentangle the gaze-related feature which contains important information about the direction of the input eye.

Also, the second ablation model achieves an 13.76% of angular error. Compared to the full model, the second ablation model yields 3.15% more errors. Although the ablation model has an autoencoder, it achieves more errors since its latent code is still entangled compared to the latent code extracted using the full model. By applying the rotation matrix to the latent code, the model can learn to disentangle the gaze-related feature which enables it to accurately estimate the direction of the input eye by providing important information. This successful disentanglement process provides rich features and makes the latter gaze estimation task more accurate.

Conclusion

In this study, I proposed the gaze estimation system which disentangles the gaze-related feature from the input eye for accurate gaze direction. The proposed method was trained in two phases. In the first phase, an encoder extracts the latent code of an input eye into two parts: gaze and appearance. Then, the rotation matrix is applied to the gaze-related latent code. In the second phase, the gaze estimation network takes the gaze part of the latent code to estimate the direction of the gaze as yaw and pitch. This unique training strategy enforces the encoder to successfully disentangle the gaze-related latent code. For the evaluation metric, I used angular error which measures the angle difference between the predicted gaze vector and the ground truth gaze vector. Through the comparison with the state-of-the-art methods, it is shown that the proposed method outperformed the comparison methods by achieving the angular error of 10.61 which is less than the results of the comparison methods. To further prove the accuracy of the proposed method, an ablation study was conducted. As a result, it is proven that each proposed idea helped with increasing the performance of the gaze estimation system. Furthermore, I plan to develop a more accurate system that minimizes angular error in order to easily expand the proposed systems to the real-world scenario and provide better feasibility.

References

[1] Cheng, Y., Bao, Y., & Lu, F. (2022, June). Puregaze: Purifying gaze feature for generalizable gaze estimation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 1, pp. 436-443).

[2] Sun, Y., Zeng, J., Shan, S., & Chen, X. (2021). Cross-encoder for unsupervised gaze representation learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3702-3711).

[3] Gideon, J., Su, S., & Stent, S. (2022). Unsupervised Multi-View Gaze Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5001-5009).

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[5] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[6] Park, S. C., Park, M. K., & Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. IEEE signal processing magazine, 20(3), 21-36.

[7] Kellnhofer, P., Recasens, A., Stent, S., Matusik, W., & Torralba, A. (2019). Gaze360: Physically unconstrained gaze estimation in the wild. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6912-6921).

[8] Liu, Y., Liu, R., Wang, H., & Lu, F. (2021). Generalizing gaze estimation with outlier-guided collaborative adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3835-3844).