Appearance-Based Gaze Estimation with Deep Neural Networks: From Data Collection to Evaluation

Article · May 2024

DOI: 10.60401/ijabc.9

4 authors, including:

Ankur Bhatt
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
3 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Ko Watanabe
Deutsches Forschungszentrum für Künstliche Intelligenz
37 PUBLICATIONS **136** CITATIONS
SEE PROFILE

Andreas Dengel
Deutsches Forschungszentrum für Künstliche Intelligenz
1,082 PUBLICATIONS **17,534** CITATIONS
SEE PROFILE

Appearance-Based Gaze Estimation with Deep Neural Networks: From Data Collection to Evaluation

Ankur Bhatt* [1], Ko Watanabe [2], Andreas Dengel[3], Shoya Ishimaru [4]
[123]RPTU Kaiserslautern-Landau & DFKI GmbH, Kaiserslautern, Germany
[4] Osaka Metropolitan University, Osaka, Japan

Abstract

Gaze estimation is an important factor in human activity and behavior recognition. The technology is used in numerous applications such as human-computer interaction, driver monitoring systems, and surveillance. Gaze estimation can be achieved using different technologies such as wearable devices or cameras. Estimating gaze using a webcam can indeed be more accessible and convenient compared to methods that rely on specific hardware like infrared cameras. In this paper, we propose a data acquisition approach for modeling appearance-based webcam gaze estimation. We implemented an application to capture gaze points using a common webcam. The application asks to click on the circle displayed on the screen, and whenever the circle is clicked, the face image and the pixel coordinates of the circle are stored. From each of the 17 participants, 50 patterns of face images and pixel coordinate information were collected. The gaze estimation models used were VGG16, ResNet50, EfficientNetB7, and EfficientNetB2. In conclusion, the result of the test set is best for VGG16 (four feature extractors) with an error difference of 2.4 cm. To validate our model, we also applied leave-one-participant-out cross-validation and found that the participant with the smallest error difference is 2.533 cm and the largest error difference is 4.759 cm. The study contributes to proposing the data collection method, the best prediction model, and discovering the difficulty of prediction occurs with human individual differences for webcam-based gaze estimation.

1 Introduction

Gaze estimation plays an essential role in many fields. In previous research, it is used to understand reading behavior [1–3], to develop intelligent textbooks for

[1]ankur.bhatt@dfki.de
[2]ko.watanabe@dfki.de
[3]andreas.dengel@dfki.de
[4]ishimaru@omu.ac.jp

readers [4, 5], and also to analyze confidence or mind wondering while answering questions [6, 7]. The limitation of the above work was that all these studies required specialized hardware to track gaze. In the field of activity behavior computing, replacing the expensive device to estimate the same activity is significant [8]. Gaze estimation using inexpensive devices such as webcams is important.

In gaze estimation, traditional image processing techniques extract features such as pupil position, eye angle, pupil diameter, or gaze direction from eye images. Eye tracking systems, on the other hand, use special cameras or sensors to track eye movements directly. Deep learning has shown great success in various computer vision tasks, including gaze estimation. Gaze estimation has become more convenient with the advent of web cameras compared to the use of skin electrodes in the past [9].

The attached sensor-based method involves sampling the electrical signal from skin electrodes to detect the user's eye movement. The 3D eye model recovery method constructs a geometric model of the eye to determine the direction of gaze. However, it requires the use of special equipment such as infrared cameras.

The 2D feature regression method uses the detected geometric features, such as pupil center and glints, to directly estimate the gaze direction. Like the 3D eye model, the reconstruction method requires using infrared cameras. Funes Mora et al. divides eye images into 15 subregions and computes the sum of pixel intensities in each subregion as features [10]. Appearance-based gaze estimation uses the deep neural network to estimate the gaze point. The main difference between conventional appearance-based methods and deep learning-based methods is that the performance of the conventional appearance-based method drops when it encounters head motion, while the deep learning method can tolerate the head motion. In addition, deep learning methods can extract high-level abstract gaze features from high-dimensional images and learn a highly nonlinear mapping from eye appearance to gaze.

This study aims to estimate gaze position from webcam images. To do so, we create our face dataset using our application. By comparing several methods, we can discover the best prediction model. The evaluation of the model is done by leave-one-participant-out cross-validation. Our contributions are as follows:

1. **Gaze data collection application**: We implement an application for webcam gaze data collection. This application can be used on any laptop.
2. **Top-performing gaze estimation model among our range of models**: We compare several deep learning models to identify the best-performing gaze estimation model.
3. **Discovery of the user dependent features**: We discuss characteristics of high and low prediction rate users from the result of the leave-one-participant-out cross-validation.

2 Related Work

Previous appearance-based methods rely on a user-specific mapping function, which requires time-consuming calibration and a fixed environment to collect the user-specific training samples. To minimize the total number of training samples, Williams et al. gave a semi-supervised Gaussian process regression method [11]. However,

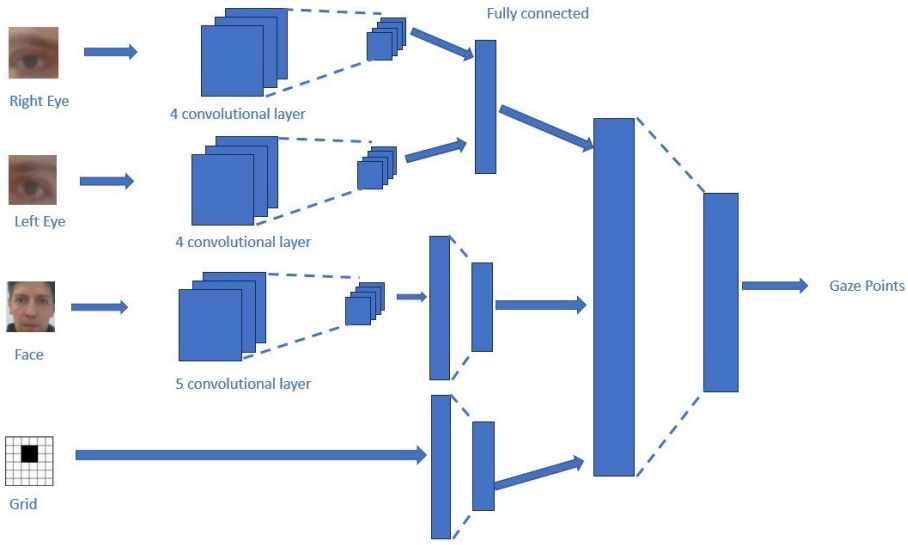


Figure 1: Gaze estimation with face and eye images of a user.

these appearance-based methods show satisfactory performance only under constrained conditions, such as fixed headphones and specific users. Their performance drops when tested in an unconstrained environment.

Deep learning-based methods are used to automatically extract the deep features from the eye images to overcome the disadvantages of conventional appearance-based methods. Zhang et al. proposed the first gaze estimation method to compute the gaze directions using a simple Convolutional Neural Network (CNN), and the performance surpasses most of the conventional appearance-based methods [12]. Inspire by the research, Figure 1 shows an architecture of gaze estimation using face images proposed by Krafka et al. [13]. Another approach is to use video as an input to estimate gaze because it provides more valuable information than images [14]. Gaze estimation from a video (or video frames) involves extracting static features from each frame using a conventional CNN. These static features are then fed into a Recurrent Neural Network (RNN) to capture temporal information.

Convolutional Neural Networks (CNNs) have been extensively applied in various computer vision tasks, including object recognition, image segmentation, and activity recognition, where they exhibit exceptional performance [15–18]. Various CNN methods have been applied to address the gaze estimation task, including supervised, semi-supervised, self-supervised, and unsupervised CNNs. Supervised CNNs are the most common type of gaze regression. In order to supervise the training, the system needs a large dataset, such as MPIIGaze [19], GazeCapture [13], and EyeDiap [10].

A semi-supervised CNN approach uses labeled and unlabeled images during training [20]. This method incorporates an additional appearance classifier and a head pose classifier to allow feature matching between labeled and unlabeled images. Labeled images from the training set and unlabeled images from the target

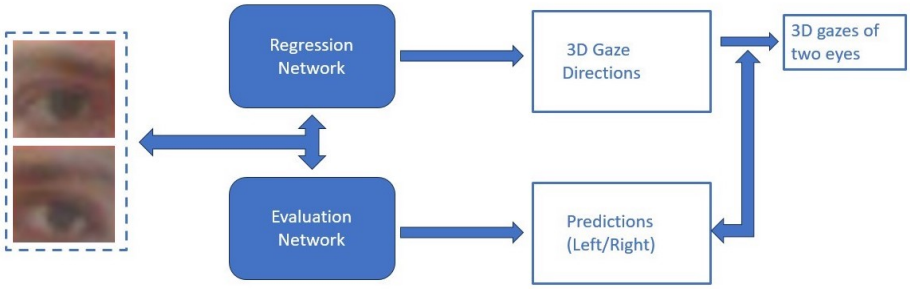


Figure 2: A self-supervised CNN method.

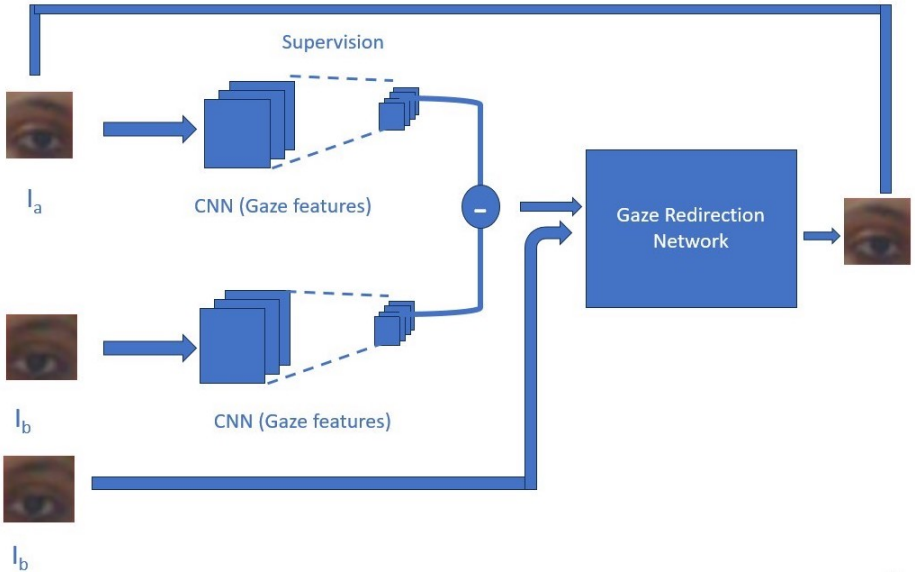


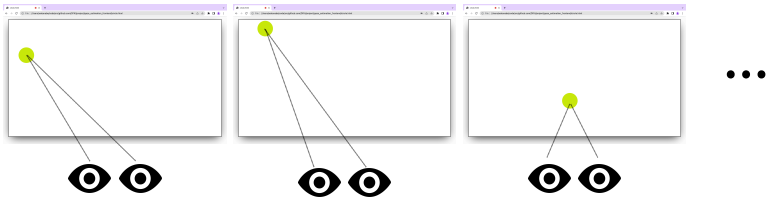
Figure 3: An unsupervised CNN method.

dataset are required to train this model. Unlabeled images are referred to as "targets", while labeled images are labeled as part of the "training set".

A self-supervised CNN method [21] consists of two sub-networks as shown in Figure 2. The regression network estimates gaze using two eye images and generates ground truth for the other network, which enables self-supervision.

An unsupervised CNN method [22] uses a CNN to extract 2D features from eye images as shown in Figure 3. The difference in features between two images, along with one of the eye images, is fed into a pre-trained gaze redirection network. This network generates the other eye image without supervision or labeled data.

((a)) Experiment condition. The participant sits in front of the laptop screen.



((b)) Experiment work-flow. A participant looks at the circle on the screen and clicks with the mouse cursor.

Figure 4: Data collection experimental setting and the workflow.

3 Data Collection

In this Section, we explain the process of collecting the face image and the laptop screen position data. Figure 4 shows an overview of the experimental settings and the data collection workflow. We will explain the background information about the participants in Section 3.1 and the data collection procedure in Section 3.2.

3.1 Participants

Our experiment collected data from 17 participants (12 males and 5 females). Along with the gaze points, we recorded background information about them, such as their country of origin and whether they had to wear glasses during the experiment. Out of the 17 participants, nine were from Japan, five were from India, and the rest were from Hungary, Chile, and Morocco. Before the experiment, we obtained participants' general data protection regulation (GDPR) consent. The participants were allowed to opt out of the experiment at any time.

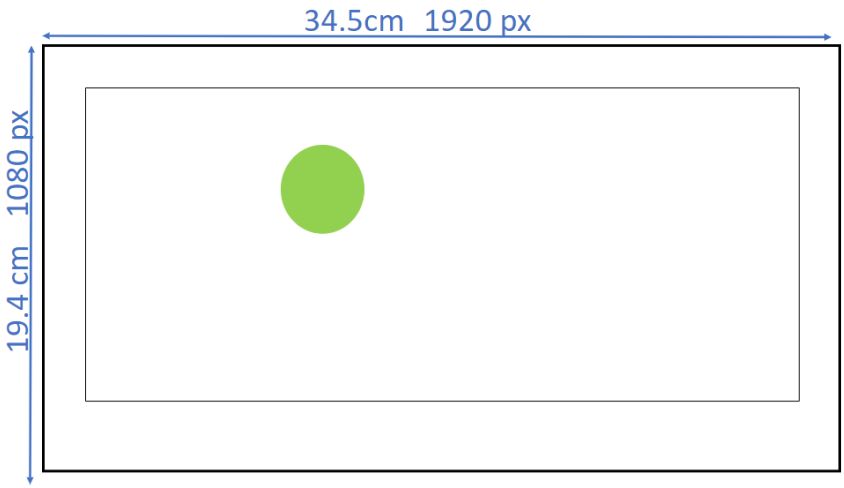


Figure 5: Experiment dimensions.

3.2 Data Collection Procedure

In this study, we conducted an experiment using a single laptop computer. The experiment was conducted in the same room in a controlled manner. Figure 4 shows the data collection experimental setting. The data collection was done in the following procedure.

1. The participants were positioned at an approximate distance of 30 cm from the webcam.
2. The Experiment conductor explains to the participant about the process and the purpose of data collection.
3. Fill out the agreement on the consent form.
4. Sit in front of the laptop and direct their attention towards the circle on the screen.
5. Click on the circle using the mouse cursor.
6. Clicking the mouse triggers the camera to capture an image as well as recorded the pixel coordinates corresponding to the click location.
7. The circle will randomly move to another position on the screen.
8. Repeat Steps 4-6 and when 50 images are saved, the process will end.

The data we collect are the pixel coordinates of the laptop screen and facial images associated with each clicked circle. In total, 50 sets of pixel coordinates and face images are stored for 17 participants, which is 850 sets of pixel coordinates, and face images are collected. Figure 5 shows the dimensions of an experiment laptop. The screen resolution was $1080px \times 1920px$ with a width of $19.4cm \times 34.5cm$.

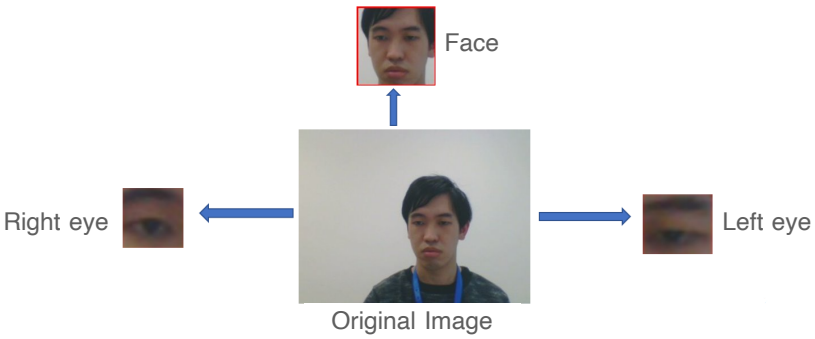


Figure 6: Image extraction of the right eye and left eye and face.

The experiment was conducted in an approximately 15-minute session in a controlled environment within a closed empty room. This approach ensured consistent lighting conditions and helped minimize any potential background noise so as not to interfere with the gaze data. The laptop was placed in a stable position.

4 Methodoogy

To estimate the gaze points, we used two different methods. The first method uses one feature extractor, and the second uses four different feature extractors.

4.1 Data Preparation

Data preprocessing is a crucial stage. The use of preprocessing techniques improves the quality and suitability of the image data, making it more suitable for subsequent analysis or processing tasks, such as object detection, image classification, or image segmentation. In addition, directly using the raw gaze images for gaze regression increases computational resources and introduces confounding factors such as scene changes. We cropped some essential parts of the images, such as the face and the left and right eyes, from the original images collected during the experiment for each participant, as shown in Figure 6. The images were normalized because normalizing pixel values to a standard range helps to achieve consistency and comparability across different images. In the single feature extractor method, all images (original images taken during the experiment, faces and left and right eyes) are combined into one image. In contrast, the four feature extractors method uses a single image as input.

4.2 Deep Learning Model

Three different backbones were used to compute the gaze points: VGG16, ResNet50, EfficientNetB2, and EfficientNetB7. Different combinations with the backbones were used in terms of image resolution (64×64, 128×128, 256×256), batch size (8, 16, 32), number of trainable layers in the backbone (all, last layer, last two layers, none), and backbones with the same weights as imagenet or without the same weights as imagenet (i.e. training from scratch) We incorporated a learning rate of

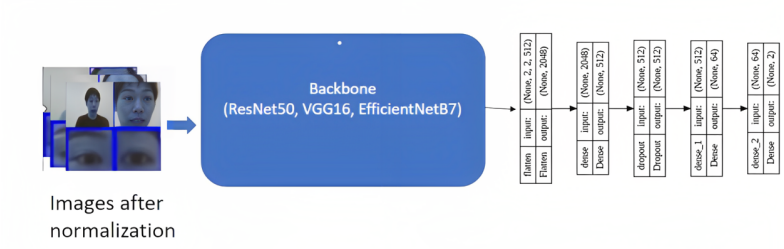


Figure 7: Architecture of the method which uses one feature extractor.

5e-5 alongside the ReduceLROnPlateau callback. This callback plays a vital role by automatically adjusting the learning rate during training and continuously monitoring a specified metric like validation loss. If the monitored metric shows no further improvement, the callback reduces the learning rate accordingly. Additionally, we utilized the "*Adam*" optimizer for our model optimization. The process we undertook can be regarded as an experiment, where we explored various combinations to determine the most effective approach for gaze estimation. Our aim was to thoroughly investigate and compare different combinations of models, considering factors such as top-1 accuracy, top-5 accuracy, the number of parameters, and model depth [23]. By conducting this comprehensive analysis, we sought to identify the combination that would yield the best results for gaze estimation.

The method using a single feature extractor takes a single input constructed by combining the original image, the face, and the left and right eyes. Then this input is fed into a backbone (VGG16, ResNet50, EfficientNetB7). In our selection process, we carefully considered the performance metrics of top-1 accuracy and top-5 accuracy when choosing the VGG16, ResNet50, EfficientNetB7, and EfficientNetB2 models. We aimed to assess whether models with a higher or lower number of parameters yielded better results. Additionally, we took into account the depth of the models to ensure a comprehensive evaluation of their capabilities. Then the feature maps are passed to the fully connected layer, and finally, the network outputs the gaze pixel coordinates as shown in Figure 7.

$$pixel\ coordinates \in \mathbb{R}^2$$

In contrast, the method using the four feature extractors takes four different images as input: the original image, the face, and the left and right eyes. Then, these four images are fed to four different backbones, and the feature maps from each backbone are passed through a concatenation layer, where the feature maps are concatenated. Then the concatenated feature maps are passed to the fully connected layer, and finally, the network outputs the gaze pixel coordinates as shown in Figure 8.

$$pixel\ coordinates \in \mathbb{R}^2$$

4.3 Model Accuracy Comparison

We were interested in determining the individual contribution of each participant to the overall result. To assess the impact of each participant, we used a technique

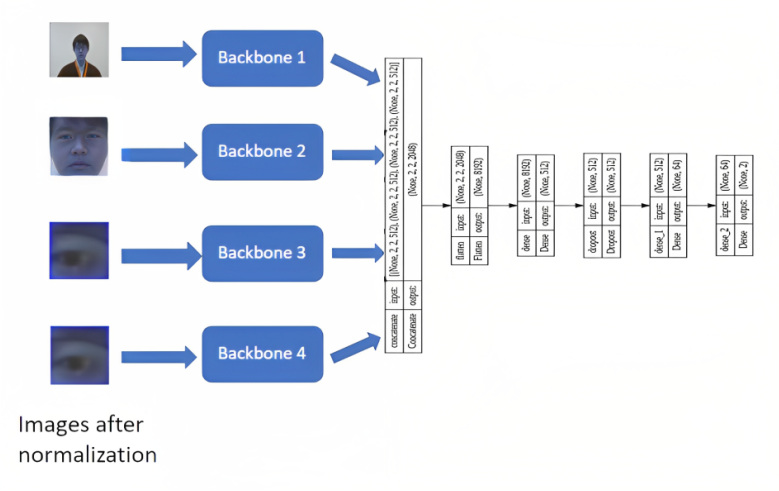


Figure 8: Architecture of the method which uses four feature extractors.

known as leave-one-participant-out cross-validation. In this method, one participant is excluded from the training set and used as the test set, while the remaining participants are included in the training set.

We used the root mean square error matrix to evaluate the error difference between the ground truth gaze points and the predicted gaze points. Note that we used pixel coordinates and computed the error difference in centimeters. In order to convert the pixel error difference to the centimeter error difference, we made some simple calculations. The following calculations refer to PPC: Pixels Per Centimeter, SWR: Screen Width Resolution, SWL: Screen Width Length, and RMSE: Root Mean Square Deviation, respectively. SWR and SWL are our experimental screen-dependent variables, as explained previously in Figure 5.

$$PPC = SWR/SWL = 1920(px) \div 34.5(cm) = 55.65 \approx 56(px/cm)$$

$$RMSE(cm) = RMSE(px) \div PPC = RMSE(px) \div 56(px/cm)$$

5 Result

In this Section, we explain the result of comparing each deep learning model and leave-one-participant-out cross-validation.

Table 1 shows the result using one feature extractor from the different combinations of settings, and it presents the best two results from each of the backbone or feature extractors. We found that VGG16 with image resolution 64×64, batch size 16, and all trainable layers with the same weight as imagenet produces the best result with an error difference of 2.489 cm.

Table 2 shows the result for each backbone with the best setting using the four-feature extractors method. It can be seen that the VGG16 with the image resolution 64×64, batch size 32, and only the last two trainable slices with the same weight

Table 1: The best results using one feature extractor method.

Model	Image Resolution	Batch Size	Trainable layer of Backbone	RMSE (px)	RMSE (cm)
EfficientNetB7	64×64	32	All	271.490	4.848
EfficientNetB7	64×64	32	Last	262.567	4.688
ResNet50	64×64	8	All	152.236	2.718
VGG16	64×64	32	All	147.168	2.628
ResNet50	64×64	16	All	146.577	2.617
VGG16	64×64	16	All	139.425	**2.489**

Table 2: Comparison of best results using four feature extraction method.

Model	Resolution	Batch Size	Trainable layer of Backbone	RMSE (px)	RMSE (cm)
EfficientNetB2	64×64	32	All	213.906	3.819
ResNet50	64×64	32	All	141.319	2.523
VGG16	64×64	32	Last two	134.419	**2.400**

as the imagenet setting again outperformed the others and gave the best result with an error difference of 2.400 cm.

Table 3 summarizes the result of the cross-validation where one participant is evaluated based on the remaining participants. In this method, we have used the same setting as the best method with four feature extractors, i.e. VGG16 backbone with image resolution 64×64, batch size 32, and only the last two trainable layers with the same weight as the imagenet. The lowest error difference is 2.533 cm and the highest error difference is 4.759 cm among the participants.

Regarding to these results, we found that VGG16 performs well for appearance-based gaze estimation. Using VGG16 to perform leave-one-participant-out cross-validation, we got a mean error of 3.375 ± 0.891 cm.

6 Discussion and Future Work

In this Section, we will discuss the results shown in Section 5. Regarding the experiment, the performance of multi-feature extractors was observed to surpass that of methods employing single-feature extractors. Each feature extractor is designed to capture specific information from the input data. We can leverage the diverse information they offer by combining multiple feature extractors. Each extractor can focus on distinct aspects or patterns in the data, resulting in a more comprehensive representation. By extracting features from multiple extractors and merging the outputs using fusion techniques such as averaging, concatenation, or

Table 3: The result of Leave-One-Participant-Out cross-validation.

User ID	Gender	Glasses	RMSE (px)	RMSE (cm)
P1	Male	Yes	150.066	2.679
P2	Male	No	150.339	2.684
P3	Male	No	149.632	2.672
P4	Male	No	148.674	2.654
P5	Male	No	148.029	2.643
P6	Female	No	145.878	2.604
P7	Female	No	149.390	2.667
P8	Female	Yes	141.857	**2.533**
P9	Male	No	266.543	**4.759**
P10	Male	No	259.303	4.630
P11	Male	Yes	257.719	4.602
P12	Male	No	251.260	4.486
P13	Female	No	249.370	4.453
P14	Male	Yes	230.259	4.111
P15	Male	No	190.853	3.408
P16	Female	Yes	165.613	2.957
P17	Male	No	159.160	2.842
Mean			189.056	3.375
Standard Deviation			49.911	0.891

advanced methods like attention mechanisms, the overall performance can be enhanced, leading to a more robust representation. Contrary to the initial assumption, the results indicated no significant difference in error rates between individuals who wore glasses and those who did not. This finding is intriguing as it demonstrates the broad applicability of our model across various users, including individuals who wear glasses. Individual differences among participants, such as eye shape, size, and movement patterns, can affect the accuracy of gaze estimation. Additionally, factors like fatigue, or blinking frequency can introduce variability in the estimation results. By delving into these aspects in future research, we can gain a deeper understanding of the factors influencing error differences in participants' gaze estimations using web cameras. This knowledge can contribute to the development of more accurate and robust gaze estimation methods in various ap-

plications.

For future work, there are several aspects we aim to address. Firstly, we intend to enhance the model's robustness by collecting additional data from diverse backgrounds, including variations in room lighting and zoomed-in and zoomed-out images. Secondly, evaluating the VGG16 model on publicly available datasets. In order to achieve a robust model for appearance-based eye tracking, it is significant for the next task. Thirdly, expanding the participant pool and gathering data from more individuals is another task we plan to undertake. Fourthly, instead of random appearances of circles on the screen, we can modify the experiment by controlling the number of circles in each quadrant to ensure data balance and mitigate potential issues. Lastly, an important aspect of our future endeavors involves designing publicly available gaze prediction software.

Additionally, in the future, there is potential to explore enhancements in the computational and memory costs associated with the method that employs four feature extractors. Furthermore, the estimation of gaze can be done using transformers in the future. By leveraging transformers-based models, we can potentially improve the accuracy and performance of gaze estimation in our system.

7 Conclusion

This research presents an approach to collect data for modeling webcam gaze estimation based on appearance. A total of 17 participants were involved, and we collected 50 patterns of face images along with corresponding pixel coordinate information. The findings reveal that utilizing a VGG16 backbone with four-feature extractors yields the most accurate results for gaze estimation. Through leave-one-participant-out cross-validation analysis, we observed that the participants' root mean square deviation ranged from 2.533 cm to 4.759 cm, with an average error value of $3.375 \pm 0.891 cm$. Our future work will expand our data collection efforts to encompass diverse datasets. This expansion aims to enhance the robustness of our model for gaze estimation.

Acknowledgment

This work was supported by DFG International Call on Artificial Intelligence "Learning Cyclotron" (Project Number: 442581111).

References

[1] Shoya Ishimaru, Tilman Dingler, Kai Kunze, Koichi Kise, and Andreas Dengel. Reading interventions: Tracking reading state and designing interventions. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, page 1759–1764, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450344623. doi: 10.1145/2968219.2968271. URL https://doi.org/10.1145/2968219.2968271.

[2] Shoya Ishimaru, Kai Kunze, Koichi Kise, and Andreas Dengel. The wordometer 2.0: Estimating the number of words you read in real life using commercial eog glasses. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, page 293–296, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450344623. doi: 10.1145/2968219.2971398. URL https://doi.org/10.1145/2968219.2971398.

[3] Shoya Ishimaru and Andreas Dengel. Arfled: Ability recognition framework for learning and education. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, UbiComp '17, page 339–343, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450351904. doi: 10.1145/3123024.3123200. URL https://doi.org/10.1145/3123024.3123200.

[4] Shoya Ishimaru, Syed Saqib Bukhari, Carina Heisel, Jochen Kuhn, and Andreas Dengel. Towards an intelligent textbook: Eye gaze based attention extraction on materials for learning and instruction in physics. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, page 1041–1045, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450344623. doi: 10.1145/2968219.2968566. URL https://doi.org/10.1145/2968219.2968566.

[5] Shoya Ishimaru, Nicolas Großmann, Andreas Dengel, Ko Watanabe, Yutaka Arakawa, Carina Heisel, Pascal Klein, and Jochen Kuhn. Hypermind builder: Pervasive user interface to create intelligent interactive documents. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, page 357–360, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359665. doi: 10.1145/3267305.3267667. URL https://doi.org/10.1145/3267305.3267667.

[6] Shoya Ishimaru, Takanori Maruichi, Koichi Kise, and Andreas Dengel. Gaze-based self-confidence estimation on multiple-choice questions and its feedback. In *Proceedings of the 2020 Symposium on Emerging Research from Asia and on Asian Contexts and Cultures*, pages 8–8, 2020.

[7] Iuliia Brishtel, Anam Ahmad Khan, Thomas Schmidt, Tilman Dingler, Shoya Ishimaru, and Andreas Dengel. Mind wandering in a multimodal reading

setting: Behavior analysis & automatic detection using eye-tracking and an eda sensor. *Sensors*, 20(9):2546, 2020.

[8] Kohei Adachi, Paula Lago, Tsuyoshi Okita, and Sozo Inoue. Improvement of human action recognition using 3d pose estimation. *Activity and Behavior Computing*, pages 21–37, 2021.

[9] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark, 2021.

[10] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc 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*, ETRA '14, page 255–258, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327510. doi: 10.1145/2578153.2578190. URL https://doi.org/10.1145/2578153.2578190.

[11] Oliver Williams, Andrew Blake, and Roberto 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)*, volume 1, pages 230–237. IEEE, 2006.

[12] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, 2015. doi: 10.1109/CVPR.2015.7299081.

[13] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[14] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6912–6921, 2019.

[15] Chenhao Chen, Yutaka Arakawa, Ko Watanabe, and Shoya Ishimaru. Quantitative evaluation system for online meetings based on multimodal microbehavior analysis. *Sensors and Materials*, 34(8):3017–3027, 2022. doi: 10.18494/SAM3959.

[16] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3367–3375, 2015.

[17] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th international conference on mobile computing, applications and services*, pages 197–205. IEEE, 2014.

[18] Ko Watanabe, Tanuja Sathyanarayana, Andreas Dengel, and Shoya Ishimaru. Engauge: Engagement gauge of meeting participants estimated by facial expression and deep neural network. *IEEE Access*, pages 1–1, 2023. doi: 10.1109/ACCESS.2023.3279428.

[19] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):162–175, 2019. doi: 10.1109/TPAMI.2017.2778103.

[20] Kang Wang, Rui Zhao, Hui Su, and Qiang Ji. Generalizing eye tracking with bayesian adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11907–11916, 2019.

[21] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 100–115, 2018.

[22] Yann-Aël Le Borgne and Gianluca Bontempi. Unsupervised and Supervised Compression with Principal Component Analysis in Wireless Sensor Networks. *Accepted at the Workshop on Knowledge Discovery, colocated with the 13th International Conference on Knowledge Discovery and Data Mining*, 2007.

[23] François Chollet et al. Keras. `https://keras.io`, 2015.