A Webcam Artificial Intelligence-Based Gaze-Tracking Algorithm

Saul Figueroa[1][a], Israel Pineda[2][b], Paulina Vizcaíno[3][c], Iván Reyes-Chacón[3][d]
and Manuel Eugenio Morocho-Cayamcela[1,2][e]

[1] *Yachay Tech University, School of Mathematical and Computational Sciences, DeepARC Research Group, Hda. San José s/n y Proyecto Yachay, Urcuquí, 100119, Ecuador*

[2] *Universidad San Francisco de Quito, College of Science and Engineering, Quito, 170901, Ecuador*

[3] *Universidad Internacional del Ecuador, Faculty of Technical Sciences, School of Computer Science, Quito, 170411, Ecuador*

fi

Keywords: Gaze-Tracking, Artificial Intelligence, Human-Computer Interaction, Visual Attention, Gaze Prediction.

Abstract: Nowadays, technological advancements for supporting human-computer interaction have had a big impact. However, most of those technologies are expensive. For that reason, building a webcam gaze-tracking system represents a computationally cost-effective approach. The gaze-tracking technique focuses on tracking the gaze direction and estimating its coordinates over a computer screen to follow user visual attention. This research presents a gaze estimation approach to predict the user's gaze direction using a webcam artificial intelligence-based gaze-tracking algorithm. The purpose of this paper is to train a convolutional neural network model capable of predicting a 3D gaze vector to estimate then the 2D gaze position coordinates over a computer screen. To perform this task, three steps are followed: 1) Pre-processing the input, crop facial, and eye images from the MPIIFaceGaze dataset. 2) Train a customized network based on a ResNet-50 pre-trained on ImageNet for gaze vector predictions. 3) 3D gaze vectors conversion to 2D point of gaze on the screen. The results demonstrate that our model outperforms the state-of-the-art VGG-16 model under the same dataset by up to $\sim 33\%$. Data and source code is available at: https://github.com/SaulFigue/Gaze-tracking-pipeline.git.

1 INTRODUCTION

The human visual attention study is one of the motivations for analyzing eye movements and estimating gaze by computer vision techniques. The gaze-tracking technique aims to track and estimate gaze position from the facial and eye regions of an individual (Liu et al., 2022). Understanding the cognitive process of people's attention and how they react to a specific task helps to comprehend visual attention study. In addition, facial expressions are one dependent variable that improves the analysis of this computer vision task (Dilini et al., 2021).

Although eye-tracking and gaze-tracking techniques can work together to perform multiple applications, we decided to develop a gaze-tracking algorithm. One of the reasons that support our selection is that most eye-tracking systems need specific hardware devices (Holmqvist and Andersson, 2017), which are highly expensive and invasive hardware to acquire (Krafka et al., 2016). On the other hand, there exist gaze-tracking systems that use default laptop webcams (a few of them are described in Section 2), which represent a low-cost implementation for track and estimating gaze direction. Furthermore, gaze-tracking systems involve more features than eye regions when estimating gaze. Incorporating head orientation into the gaze estimation model complements gaze estimation predictions by considering head pose information such as the head declination information(Cazzato et al., 2020).

Currently, new artificial intelligence methods have been introduced into gaze-tracking systems to improve the accuracy of gaze-estimation prediction and its analysis. Deep learning algorithms are usually found in the literature (de Lope and Graña, 2022), and multiple variations of convolutional neural network (CNN) methods are implemented for tracking the gaze (Chen et al., 2020).

[a] https://orcid.org/0000-0001-9395-5903
[b] https://orcid.org/0000-0002-3950-2169
[c] https://orcid.org/0000-0001-9575-3539
[d] https://orcid.org/0009-0002-2731-5531
[e] https://orcid.org/0000-0002-4705-7923

Figueroa, S., Pineda, I., Vizcaíno, P., Reyes-Chacón, I. and Morocho-Cayamcela, M.
A Webcam Artificial Intelligence-Based Gaze-Tracking Algorithm.
DOI: 10.5220/0012759700003753

In *Proceedings of the 19th International Conference on Software Technologies (ICSOFT 2024)*, pages 228-235
ISBN: 978-989-758-706-1; ISSN: 2184-2833

Gaze estimation is a computer vision task that involves converting gaze directions into screen positions. This technique is useful when working with artificial intelligence on a real-time webcam approach by improving human gaze direction prediction with a significant accuracy (Ou et al., 2021).

This paper proposes a Deep learning-based gaze-tracking algorithm to estimate gaze direction using a pre-trained on ImageNet ResNet-50 model and MPIIFaceGaze public and available gaze-tracking dataset. Furthermore, a comparison between benchmark proposed model for their study of efficiency in real-time webcam gaze tracking system (Gudi et al., 2020) and our proposed model is presented to test our gaze-estimation system effectiveness.

2 RELATED WORKS

2.1 Human Gaze Estimation Approach

Gaze-tracking systems involve many human-face-eyes features. The eye region is one of the main components that help developers to build gaze-tracking systems. There are two types of human gaze estimation approaches.

2.1.1 Model-Based Techniques

Tracking gaze direction through 3D eye models using geometric eye features is one of the objectives of the model-based gaze estimation technique. Gaze points are obtained by calculating the optical and visual axis of eyes (Modi and Singh, 2021). Some researchers obtained desirable results from this model-based technique for gaze-tracking systems. For example, estimating gaze direction using 3D eye models and geometric information to build a model-based gaze estimation system (Kaur et al., 2022).

Facing refraction calculus represents a complex task involving pupils. Therefore, a 3D gaze estimation method based on the iris features is a potential approach (Liu et al., 2020). To estimate the gaze direction, 3D human eye optical axis (OA) reconstruction is performed through the iris and cornea center. Also, a visual axis (VA) is needed to calculate the angle between the OA and VA, known as the kappa angle. This process allows drawing the line of sight using a single light and camera source.

2.1.2 Appearance-Based Techniques

Machine learning and Deep learning are the two most commonly used techniques for building appearance-based gaze estimation systems (Modi and Singh, 2021).

Conventional Appearance-Based Methods. When referring to conventional appearance-based methods, implementations involve a multilevel Histogram of Oriented Gradient (HOG) features extraction approach to infer gaze by processing an eye-appearance features extraction followed by a personal calibration (Martinez et al., 2012). An Adaptive Linear Regression (ALR) method to map high-dimensional eye image features to low-dimensional gaze positions is another approach, where the sub-pixel alignment method helps to solve the face-head motion problem. By this means, the computational cost can be reduced significantly. Additionally, using fixed, smaller training sample sets can help achieve high accuracy, facing some conventional appearance-based challenges (Lu et al., 2014).

Appearance-Based Methods with Deep Learning. The Gaze-Net system, trained on MPIIGaze and re-trained by transfer learning technique on ColumbiaGaze datasets, is a capsule network approach for appearance-based gaze estimation (Mahanama et al., 2020). This network receives eye images as input and returns two results: A reconstructed image and gaze estimation coordinates. The Gaze-net architecture performs a classification training process with 6 classes of gaze direction. The transfer learning process re-trains the gaze estimator with accurate results by considering that one single eye image is enough to estimate gaze reliably. The same conclusion was provided by a study of efficiency in real-time webcam gaze tracking system (Gudi et al., 2020). Their work intended to obtain the best performance in speed and effort terms for estimating gaze. A CNN is used to process face, eyes, and single-eye input images. In addition, the authors find that geometric regression calibration performs better than machine learning and hybrid geometric approaches. A pre-trained on ImageNet VGG-16 neural network model is used to calculate gaze vectors. In addition, the system was tested with MPIIFaceGaze and EYEDIAP datasets. One key finding in Gudi *et al.* work is that using single-eye images (left or right) performs better than using left and right-eye images since it enhances the computational speed and effort for calibration processes.

2.2 Gaze-Tracking Applications

2.2.1 Human-Computer Interaction (HCI)

One of the most attractive applications of gaze-tracking is using gaze and head movement to control computer functions without using hands. A proper implementation based on this purpose involves the face detection algorithm and the iBUG 300-W dataset as the core. Then, developing a simpler mouse assistance system for motor-disable people represents a huge advance in HCI (Anantha Prabha et al., 2022).

One step further is to combine mouse and keyboard functions to develop a real-time HCI gaze tracking system. This approach can be handled by two principal steps: Eye state (opened or closed) recognition and a CNN gaze estimation model to predict the point of gaze (x,y) over the screen (Huang et al., 2021).

2.2.2 Visual Attention

During the last few years, interest in studying the visual attention behavior of people of different ages has increased. Related works investigated the visual intention influence of textual and graphical information images over 30 participants divided into three age groups. Using a real-time low-cost webcam gaze-tracking system helped to determine that younger users tend to give more attention to graphical information images, while older ones lean for textual information (Sabab et al., 2022). On the other hand, OWLET architecture is capable of estimating 127 $\sim$ 7-month-old infants' visual attention by following face, eye, pupil extraction process, ecologically grounded gaze direction, and point of gaze estimation process from the video input of the infant (Werchan et al., 2022).

2.2.3 Cognitive Processes

Multiple researchers have led their studies investigating the cognitive processes of the educational field through eye-gaze-tracking system approaches. For example, using eye-tracking and artificial intelligence to investigate students' motivation while watching Massive Open Online Course (MOOC) lectures (Sharma et al., 2020).

Looking forward, gaze-tracking systems can work as a cognitive analysis tool to examine the relation between students' engagement and facial emotion in combination with eye and head movements information to help teachers make the learning environment effective (Sharma et al., 2023).

3 SYSTEM MODEL AND METHODOLOGY

This section provides detailed implementation properties needed for executing the gaze-tracking system. Additionally, the pipeline of the proposed method is based on the benchmark paper for further comparisons. All the hyper-parameters and architectures used to perform this task are presented.

3.1 Dataset

We use the public MPIIFaceGaze (Zhang et al., 2017) dataset to train the model. It consists of $37,667$ face images of 448×448 pixels from 15 participants. Figure 1 illustrates a sample of facial images of the MPIIFaceGaze dataset.

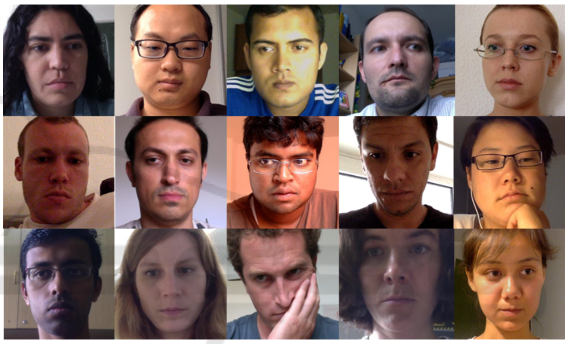


Figure 1: MPIIFaceGaze dataset sample images of the facial region for each participant.

For every participant, 2D and 3D gaze annotations are provided. The relevance of using this dataset is its variety of illumination, skin color, eye accessories, and facial features of participant images. All of these collected images were taken from a laptop webcam source, simulating real-world scenarios. Moreover, the MPIIFaceGaze dataset has an annotation text file for each participant, where gaze location in screen coordinates, facial landmark points, 3D head pose and extrinsic parameters in camera coordinates, gaze origin, and 3D gaze target location are some of the most used information provided by this dataset.

3.2 Proposed Model

The pipeline proposed for the gaze estimation system is presented in Figure 2. Three main building blocks are needed to develop the gaze estimation task based on CNN architectures.

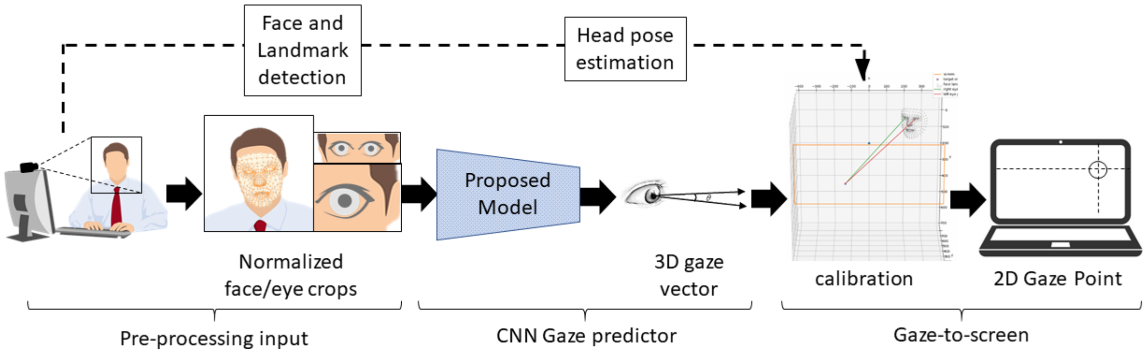


Figure 2: Complete pipeline representation for the artificial intelligence-based gaze-tracking algorithm.

3.2.1 Pre-Processing Block

In this first building block, facial images are processed to obtain normalized and cropped images before moving to the CNN model gaze predictor. Using the annotation information provided by the dataset, facial features and eye regions can be found easily. Since some of the images inside the dataset present head pose inclination, we perform an alignment process using the 6 landmark points and 3D face model to build rotation and translation matrices. Moreover, before dividing the complete dataset into 3 subsets: Training, testing, and validation, we need to remove those erroneous data from those gaze points not in the screen size. Then, we decide to divide these subsets into 15 participants. 13 are for training, and the missing two are for testing and validation. All of these processes help to get the final normalized face and eye crop images to get into the CNN model.

3.2.2 CNN Gaze Predictor Model

In contrast with the benchmark paper, the pre-trained VGG-16 network architecture is replaced by a ResNet-50 model pre-trained on ImageNet to predict pitch and yaw angles with respect to the camera by using the pre-processed output images. For a further comparison between both models, the same pre-processed MPIIFaceGaze dataset is used for the training process. Figure 3 illustrates the proposed face CNN architecture model. A similar architecture is used for the eyes CNN model.

In addition, the complete architecture is presented in Figure 4, where L-Eye CNN and R-Eye CNN represent the proposed CNN model architecture for eyes images; Eye CNN is the concatenated model for both L and R Eye CNN; FF Eye CNN is the feature fusion process of the Eye CNN block; FC Eye represents the fully connected layer for the FF Eye CNN block; Face CNN is the proposed model architecture for facial images; FC Face represents the fully con-

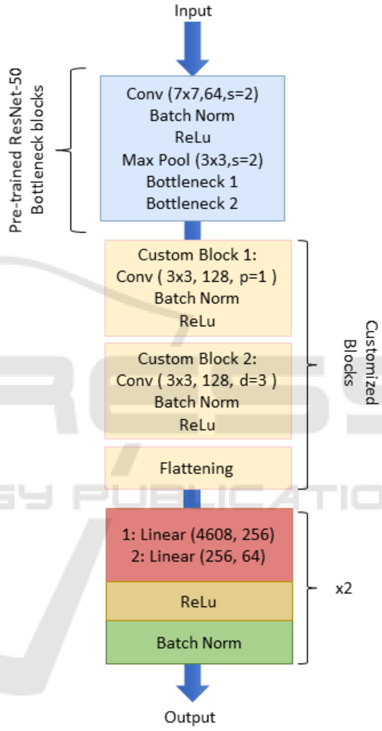


Figure 3: Proposed CNN model for gaze direction prediction: Facial images inputted on a pre-trained ResNet-50 model with the first two Bottleneck stage blocks and two customized convolutional blocks added, plus flatten linear layers.

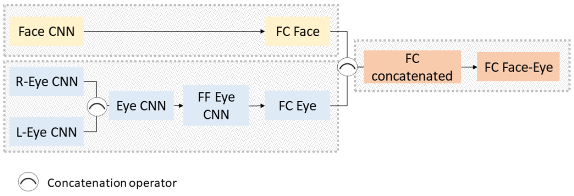


Figure 4: Proposed forward architecture for gaze estimation.

nected layer for Face CNN block; FC concatenated is the combined block after a concatenation process of both face and eye fully connected layers; FC Face-Eye represents the final fully connected layer for all the system model.

For the training process, some parameters are considered: 96×96 pixels pre-processed facial images from the MpiiFaceGaze dataset; 64×96 pixels pre-processed eyes images from the MpiiFaceGaze dataset; 50 epochs; 10^{-5} learning rate value; Adam optimizer.

The ground truth gaze vector is compared with predicted ones to analyze how accurate this model is. Moreover, the mean angular error from the gaze vector of each participant is one of the principal performance metrics to be evaluated. Then, let k be the number of sample images with random gaze target points generated. Random calibration (RC) samples describe the randomly positioned targets over the sample images based on k value. Furthermore, comparing the benchmark and proposed model performances can give us an idea of how accurate our proposed model is compared to previous literature.

3.2.3 Gaze-to-Screen Mapping Method

To translate the coordinates of the gaze vector to the screen, a geometric relation is applied between the camera and the screen coordinates system. For this process, the predicted gaze vector in camera coordinates must be calibrated to gaze points in screen coordinates.

Screen Calibration. To perform this task, we follow the geometry-based calibration method used in the benchmark paper as a guide. First, individual eye location is estimated in the screen coordinate system (x,y,z) by fixate gaze over a specific (x,y) point on the screen and considering z as the distance between individual and screen. Then, the subject was asked to look at specific points over the screen while the CNN gaze predictor model records eye movements and gaze directions. In addition, we assume that the camera's position is the same as the default laptop's webcam location (top-mid). Once the pitch angle is obtained from the previous process, rotation R and translation T matrices can be calculated by the following equation:

$$\mathrm{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & -\sin(\rho) \\ 0 & \sin(\rho) & \cos(\rho) \end{bmatrix} \tag{1}$$

$$\mathrm{T} = \quad e_{scs} - \mathrm{R} \cdot e_{ccs} \tag{2}$$

Where ρ is the estimated pitch angle, e_{scs} is the eye location in the screen coordinates system, and e_{ccs} is the eye location in the camera coordinates system obtained from eye landmark points. Next, all eye locations on the screen coordinate system can be obtained by using the same rotation and translation matrices. The following equation represents this definition:

$$\hat{e}_{scs} = \mathrm{R} \cdot \hat{e}_{ccs} + \mathrm{T} \tag{3}$$

Where $\hat{e}_{scs}$ represents any eye location in the screen coordinates system and $\hat{e}_{ccs}$ represents any eye location in the camera coordinates system. Finally, from $\hat{e}_{scs}$ we can extract $\hat{x}_{scs}$ and $\hat{y}_{scs}$ mapped values.

4 RESULTS AND DISCUSSIONS

4.1 Mean Angular Error Analysis

Table 1 presents the mean angular error for each one of the participants evaluated on both models.

Table 1: Mean angular error comparison between benchmark network based on VGG-16 architecture and proposed network based on ResNet-50 architecture accuracy with $k = [9, 128]$ random calibration samples.

Participants \Models	Benchmark		Proposed	
	RC k = 9	RC k = 128	RC k = 9	RC k = 128
p00	1.780	1.676	2.102	1.997
p01	1.899	1.777	2.994	2.845
p02	1.910	1.790	1.266	1.193
p03	2.924	2.729	1.879	1.767
p04	2.355	2.239	1.411	1.341
p05	1.836	1.720	1.358	1.284
p06	2.569	2.464	1.769	1.705
p07	3.823	3.599	1.994	1.883
p08	3.778	3.508	2.454	2.304
p09	2.695	2.528	1.313	1.234
p10	3.241	3.126	1.506	1.431
p11	2.668	2.535	1.434	1.365
p12	2.204	1.877	1.391	1.240
p13	2.914	2.753	1.650	1.547
p14	2.161	2.010	1.359	1.280
mean	2.584	2.422	1.725	1.628

Results demonstrate an improvement in terms of mean angular error achieved by the proposed architecture. Moreover, for $k = 9$ RC samples, the benchmark paper results were improved by up to 33.24% comparing 1.725 degrees to 2.584 degrees. In the same way, for $k = 128$ RC samples, the benchmark paper results were improved by up to 32.78% comparing 1.628 degrees to 2.422 degrees.

4.2 Pitch and Yaw Prediction Analysis

A comparison between pitch and yaw predicted values for both models is presented in Figure 5.

Pitch and Yaw prediction values are key outputs for the gaze estimation system. To evaluate those results, scatter relation plots for pitch and yaw prediction of both models are given for one of the participants as a sample test. Figure 5 plots the pitch and yaw representation predicted values over the ground truth for participant 13. One big difference between these two models is that the proposed model based on ResNet-50 architecture performed better than the benchmark model based on VGG-16 architecture. This difference is appreciated from Figure 5b and Figure 5a pitch representations, where the first shows a not desirable relation between these two axes, while the last one keeps a better relation. This analysis is also interpreted for yaw plots in Figure 5d and Figure 5c, where the last one, our proposed model, presents a sharpened relation results in comparison with the benchmark model plot. These results are supported by considering that our customized network uses the first ResNet-50 pre-trained blocks until the second bottleneck, which contains more convolution layers than the benchmark architecture with only four convolution blocks of the VGG-16 pre-trained model. Moreover, the residual information collected by ResNet architecture keeps more feature information useful for learning. This model aims to extract more features from the input data, increasing the information quality and optimizing the gradient backpropagation to avoid losing important information in the process.

4.3 Training Performance Analysis

Figure 6 illustrates the performance during training for the benchmark and proposed model.

Since gaze direction precision involves angle metrics, and to better compare benchmark gaze vector results, we decided to evaluate the angular error by plotting it over each iteration in the training process to evaluate the proposed model performance. Additionally, a train-loss plot is generated to check if the loss error over the training process keeps decreasing.

Both results are visualized in Figure 6. Figure 6a shows that between $[0-20]$K training iterations, the angular error keeps over 3 for both models, representing bad accuracy results. However, after more iterations, this angular error decreased. The sharpened light-purple curve (bottom one) better illustrates this result and decreases faster than the benchmark training process (light-blue). In the same way, this light-purple curve in Figure 6b follows a desired decreasing tendency result for the loss values after each iteration in the training process compared with benchmark one.

5 CONCLUSIONS

In conclusion, this research proposes an artificial intelligence-based gaze-tracking algorithm that uses a built-in laptop webcam to predict the user's point of gaze on a computer screen. The proposed model involves a customized ResNet-50 network pre-trained on ImageNet, enhancing the performance and accuracy compared to traditional models such as the one used in benchmark one. This architecture, trained on the MPIIFaceGaze dataset, allows us to build a 3D gaze vector predictor that is translated to 2D screen coordinates with the help of a calibration process.

The results indicate that the proposed system can accurately predict the user's gaze direction. Moreover, the proposed model outperforms the benchmark by up to $\sim 33\%$ in terms of mean angular error, increasing our gaze-tracking system's reliability and significantly improving the human-computer interaction experience.

Undoubtedly, the potential application of gaze-tracking systems bridges the gap in how we interact with computers in a hands-free system.

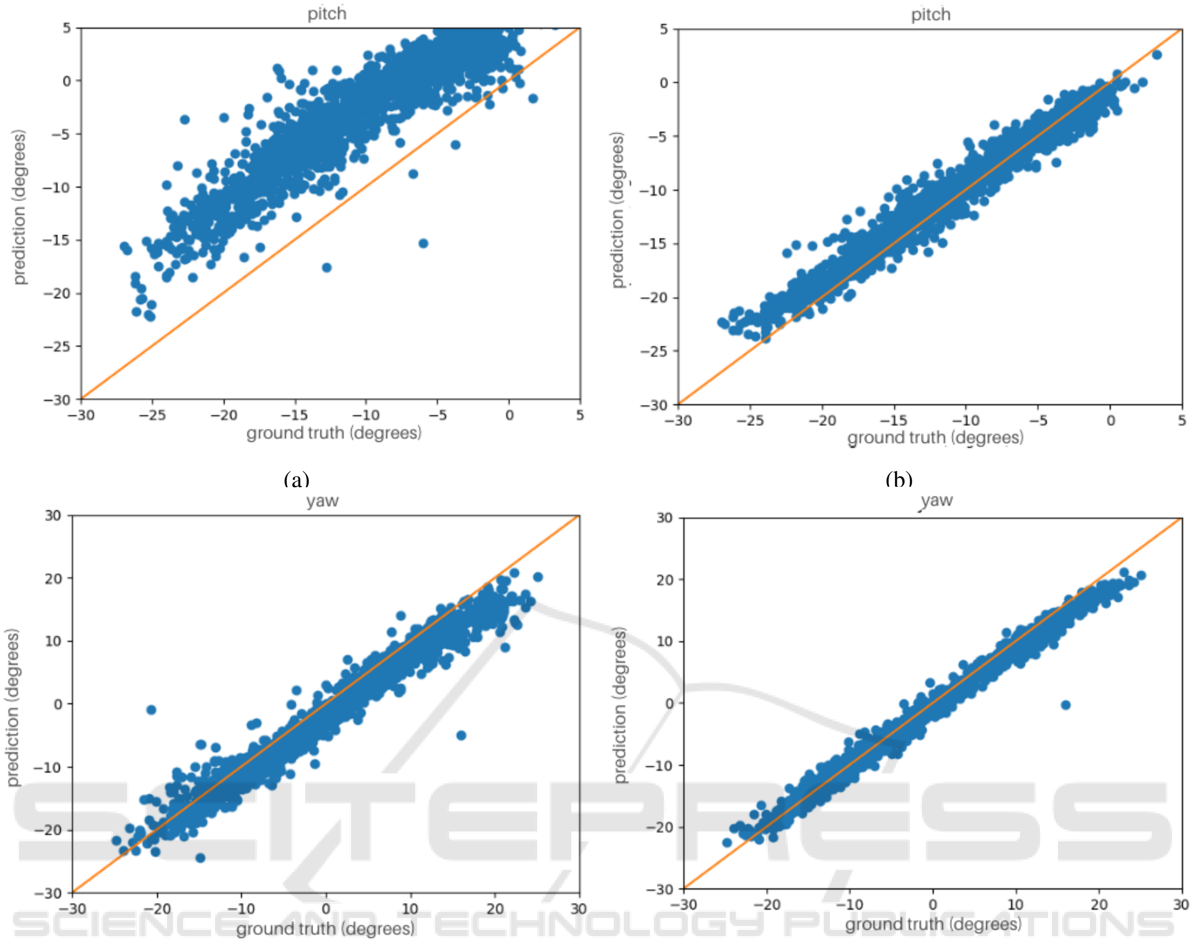
6 LIMITATIONS FOUND

In fact, each built-in laptop webcam has its camera parameters: intrinsic and extrinsic. This leads to the calibration process being unique and dependent on the webcam used in the implementation. Then, the rotation matrix R and translation vector T depend on the camera's features.

During the point of gaze estimation test, the distance between the laptop's camera and the user was fixed at $50-60cm$. However, the accuracy of the gaze estimation predictions for different distance ranges was not covered by the proposed system in this work.

7 FUTURE WORKS

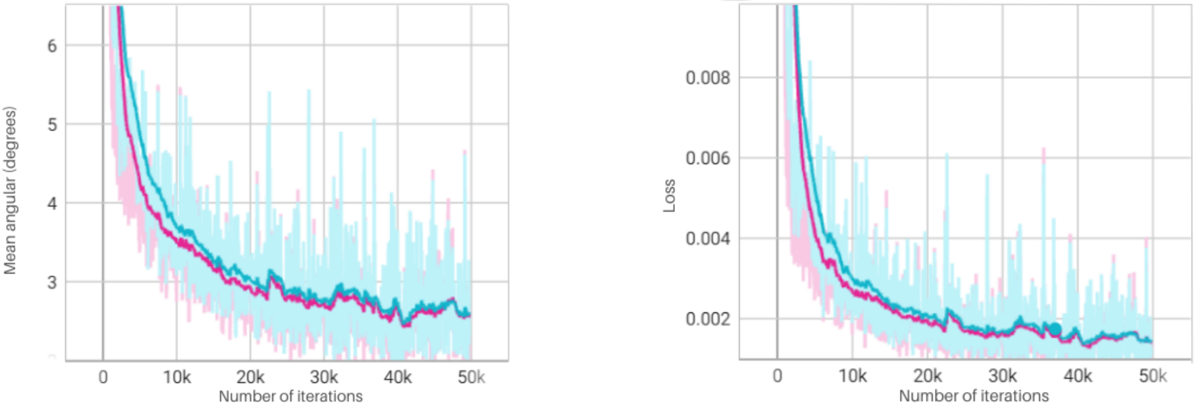
We encourage readers and all practitioners in this research area to address the limitations identified by our



(a) (b)

(c) (d)

Figure 5: Pitch and Yaw scatter plot relation between predicted output and ground truth values for participant 13. (a) and (c) represent the accuracy between the predicted gaze vector and the ground truth vector in degrees by using the benchmark customized VGG-16 trained model; (b) and (d) represent the accuracy between the predict gaze vector and the ground truth vector in degrees by using the proposed trained model.



(a) (b)

Figure 6: Graph representation of the training process using both models, the light-blue curve (upper one) refers to the benchmark model, and the light-purple curve (bottom one) refers to the proposed model. (a) Train/angular-error plot between 50K iterations and angular error values for $y-$axis; (b) Train/loss plot for 50K iterations.

system. They can replicate this proposed implementation using their own laptops' webcams and evaluate the accuracy of their systems. Additionally, varying the distance between the camera and the users' eyes is an important factor to be tested during experiments. Furthermore, implementing a new customized architecture based on the methodology provided in this paper could be a proper approach in this field. An initial starting point to improve the results presented in this work will be to modify some hyper-parameters to conduct more in-depth tests on the proposed architecture.

Finally, we find gaze-tracking algorithm applications useful, and we certainly will continue developing some of those described in Section 2.2 as part of our future work and invite readers to explore a wide range of applications, such as a shooter game aim assistant, mouse assistant controller for disabled people to use in driving a vehicle, or even neuromarketing studies based on visual attention analysis as a solution for products or ad placement in websites like e-commerce or social media pages.

REFERENCES

Anantha Prabha, P., Srinivash, K., Vigneshwar, S., and Viswa, E. (2022). Mouse assistance for motor-disabled people using computer vision. In *Proceedings of International Conference on Recent Trends in Computing*, pages 403–413. Springer.

Cazzato, D., Leo, M., Distante, C., and Voos, H. (2020). When I look into your eyes: A survey on computer vision contributions for human gaze estimation and tracking. *Sensors*, 20(13):3739.

Chen, H.-H., Hwang, B.-J., Wu, J.-S., and Liu, P.-T. (2020). The effect of different deep network architectures upon cnn-based gaze tracking. *Algorithms*, 13(5):127.

de Lope, J. and Graña, M. (2022). Deep transfer learning-based gaze tracking for behavioral activity recognition. *Neurocomputing*, 500:518–527.

Dilini, N., Senaratne, A., Yasarathna, T., Warnajith, N., and Seneviratne, L. (2021). Cheating detection in browser-based online exams through eye gaze tracking. In *2021 6th International Conference on Information Technology Research (ICITR)*, pages 1–8. IEEE.

Gudi, A., Li, X., and van Gemert, J. (2020). Efficiency in real-time webcam gaze tracking. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 529–543. Springer.

Holmqvist, K. and Andersson, R. (2017). Eye tracking: A comprehensive guide to methods. *paradigms and measures*.

Huang, J., Zhang, Z., Xie, G., and He, H. (2021). Real-time precise human-computer interaction system based on gaze estimation and tracking. *Wireless Communications and Mobile Computing*, 2021.

Kaur, H., Jindal, S., and Manduchi, R. (2022). Rethinking model-based gaze estimation. *Proceedings of the ACM on computer graphics and interactive techniques*, 5(2).

Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., and Torralba, A. (2016). Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184.

Liu, J., Chi, J., Hu, W., and Wang, Z. (2020). 3d model-based gaze tracking via iris features with a single camera and a single light source. *IEEE Transactions on Human-Machine Systems*, 51(2):75–86.

Liu, J., Chi, J., Yang, H., and Yin, X. (2022). In the eye of the beholder: A survey of gaze tracking techniques. *Pattern Recognition*, page 108944.

Lu, F., Sugano, Y., Okabe, T., and Sato, Y. (2014). Adaptive linear regression for appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):2033–2046.

Mahanama, B., Jayawardana, Y., and Jayarathna, S. (2020). Gaze-net: Appearance-based gaze estimation using capsule networks. In *Proceedings of the 11th augmented human international conference*, pages 1–4.

Martinez, F., Carbone, A., and Pissaloux, E. (2012). Gaze estimation using local features and non-linear regression. In *2012 19th IEEE International Conference on Image Processing*, pages 1961–1964. IEEE.

Modi, N. and Singh, J. (2021). A review of various state of art eye gaze estimation techniques. *Advances in Computational Intelligence and Communication Technology: Proceedings of CICT 2019*, pages 501–510.

Ou, W.-L., Kuo, T.-L., Chang, C.-C., and Fan, C.-P. (2021). Deep-learning-based pupil center detection and tracking technology for visible-light wearable gaze tracking devices. *Applied Sciences*, 11(2):851.

Sabab, S. A., Kabir, M. R., Hussain, S. R., Mahmud, H., Hasan, M., Rubaiyeat, H. A., et al. (2022). Vis-itrack: Visual intention through gaze tracking using low-cost webcam. *arXiv preprint arXiv:2202.02587*.

Sharma, K., Giannakos, M., and Dillenbourg, P. (2020). Eye-tracking and artificial intelligence to enhance motivation and learning. *Smart Learning Environments*, 7(1):1–19.

Sharma, P., Joshi, S., Gautam, S., Maharjan, S., Khanal, S. R., Reis, M. C., Barroso, J., and de Jesus Filipe, V. M. (2023). Student engagement detection using emotion analysis, eye tracking and head movement with machine learning. In *Technology and Innovation in Learning, Teaching and Education: Third International Conference, TECH-EDU 2022, Lisbon, Portugal, August 31–September 2, 2022, Revised Selected Papers*, pages 52–68. Springer.

Werchan, D. M., Thomason, M. E., and Brito, N. H. (2022). Owlet: An automated, open-source method for infant gaze tracking using smartphone and webcam recordings. *Behavior Research Methods*, pages 1–15.

Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. (2017). 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*, pages 51–60.