FastSightNet: Real-time Eye-Tracking-based Human-Computer Interface

1st Bejan Andrei-Paul
Babes-Bolyai University
Cluj-Napoca, Romania
bejan.andrei.official@gmail.com

***Abstract*—Current human-computer interfaces (e.g. mouse, trackpad, or keyboard) have gained in popularity over the years because they offered the simplest solution at the right time while having a reasonable learning curve. The technology has evolved rapidly in the last decades, and there is now the opportunity to move towards new, intelligent, more intuitive, and faster human-computer interfaces. Eye-Tracking represents a promising step in this direction. Recent developments in appearance-based gaze estimation using deep learning have come a long way, and soon they will reach consumer-ready performance. This work introduces FastSightNet, an efficient model based on the MobileNet architecture, evaluated and compared with existing models. Trained on the MPIIFaceGaze dataset, FastSightNet achieves a 5.1° mean angular error with 77 frames per second inference speed. Additionally, GazeTrack is presented, a system that allows real-time evaluation of the trained models on the webcam feed in both 3D world and 2D screen scenarios.**

***Index Terms*—Eye-tracking, Gaze estimation, Deep-learning, MPIIFaceGaze, MobileNet**

I. Introduction

Human-computer interface is a subject that has been studied extensively over the years, but has yet to become exhausted. The current most used human-computer interfaces such as mouse and keyboard in their various forms do a good job enabling people to control their computers but they are not the most intuitive, nor the fastest out there. One action humans perform unconsciously while navigating on their computer is to look directly at the component they want to press or select on the screen. Naturally, this leads to one challenging problem, but very promising idea which is Eye-Tracking.

Eye-Tracking may sound like an impossible problem to solve algorithmically, since it is often unconstrained and it is really hard to create a model that can generalize well across multiple subjects. However, the last decade has shown an accelerated expansion in the machine learning field, culminating with deep architectures that can in fact learn very complex patterns from data and generalize well to unseen data.

The issue being addressed is that current human-computer interfaces can be more intuitive, more efficient, and better suited for humans. The scope of this piece of work is to accelerate the development towards new human-computer interfaces based on Eye-Tracking technology. By providing a solution that works well in real-life scenarios, everyone will benefit, and eventually, a transition from traditional human-computer interfaces that merely accomplish tasks to innovative, more intuitive, and faster ways of interacting with computers can be achieved.

One of the reasons for solving this problem is that current human-computer interfaces have poor accessibility for people with upper-limb disabilities. Besides this, the trackpad usually found on laptops is hard and very slow to use, while a peripheral mouse is another object that people must carry around. Actually, in most real-life scenarios such as web browsing, reading text files, writing emails, and so on, an eye-tracking solution might be a better option, giving up on all the disadvantages of the aforementioned.

This scientific work aims to advance the development of new human-computer interfaces based on Eye-Tracking technology. The key objectives and contributions are:

- Conduct a comprehensive literature review of current Eye-Tracking methods
- Reproduce a state-of-the-art deep learning gaze estimation model
- Propose a new model with improved inference speed
- Compare the proposed model with existing models
- Develop a system to preview the real-time performance of the model in both 3D and 2D scenarios

II. Related Work

The iTracker model, introduced alongside the GazeCapture dataset [11], is a CNN similar to AlexNet [12]. It takes images of the left and right eyes, the face, and a face grid as input and outputs distances from the camera on the X and Y axis. The model achieves an inference time of 0.05 seconds on an iPhone 6s and reports errors of 1.71 cm and 2.53 cm on phones and tablets. By calibrating the model using 13 fixed locations its performance improved to 1.34 cm and 2.12 cm, respectively.

Authors of [17] proposed the MPIIGaze dataset and GazeNet, a multimodal CNN based on the LeNet architecture [13]. It takes an eye image and head angle vector as input and outputs a 2D gaze vector. Trained with L2 loss, the model achieves a 6.3° mean angular error, outperforming other models like Random Forests, kNN, Adaptive Linear Regression, and Support Vector Regression in within-dataset evaluation.

The same authors also proposed a new Spatial Weights CNN model [18], which uses only face images to output 2D gaze on-screen locations and 3D gaze vectors. The model is built on an AlexNet backbone. They normalize the input data [15]

and apply L1 loss for training. The model achieves a 4.2 cm error on 2D gaze estimation and a 4.8° angular error on 3D gaze estimation, exceeding state-of-the-art performance.

The Gaze360 model, introduced in [10], utilizes a ResNet-18 backbone [8] and a bidirectional Long Short-Term Memory (Bi-LSTM) [6] to exploit the temporal continuity of gaze. It processes a sequence of 7 face images to predict a single 3D gaze direction and an error estimation term. The model achieves an 11.1° angular error on the front-facing subset of Gaze360 and 9.9° when trained on Gaze360 and tested on MPIIFaceGaze.

MCGaze [7] estimates 3D gaze by capturing spatial-temporal interaction among the head, face, and eyes using a ResNet-50-FPN backbone and the transformer architecture.They use different losses for optimizing the clue localization heads and added temporal regularization terms for better temporal modeling. The model achieves a 10.02° angular error on the detectable faces subset of Gaze360 dataset, with an ablation study that confirms the performance boost coming from each component.

The FaZE [14] authors proposed a calibration framework for Few-shot Adaptive GaZE Estimation. They use the MAML meta-learning algorithm and a Disentangling Transforming Encoder-Decoder (DT-ED) architecture to enable person-specific calibration. With arccos loss for training, the model achieves an angular error of 3.18° on GazeCapture and 3.42° on MPIIGaze using as few as three calibration samples, improving state-of-the-art performance.

Authors of [2] proposed L2CS-Net, a CNN-based model with a ResNet-50 backbone and different regression branches for yaw and pitch gaze angles. Combining cross-entropy loss (for classifying the binned prediction) and mean squared error, the model achieves a 3.92° angular error on MPIIFaceGaze and 9.02° on the front-facing subset of Gaze360.

III. Methods

FastSightNet Architecture

The proposed model is called FastSightNet. The motivation behind it is the need for a lightweight model with fast inference time, capable of functioning on low-resource hardware. Training and running other models, such as the L2CS ResNet model [2], is computationally intensive, and it is possible that other more efficient models will work in this scenario as well. The model is based on the MobileNet architecture [9]. The FastSightNet architecture is illustrated in Figure 1. The input of size 3x224x224 is fed through the MobileNetV2 backbone. The output is passed to an adaptive average pooling layer [5] to reduce the feature-map size to Nx1280x1x1, similar to the L2CS-Net model [2]. Finally, the feature-map is linearized and then goes through a fully connected layer and outputs two values (pitch and yaw angles) for each image in the batch.

Dataset

The MPIIGaze dataset [17] was collected by recording participants in daily life scenarios over several months. Participants were located in different locations, at different times, using varying environmental illumination, and having variable head poses. The experiments were recorded using laptops' built-in camera. Every 10 minutes, a software application notified the subject to look at a random sequence of 20 on-screen locations and press enter when they fixated their gaze. A total of 213,659 images were captured from 15 participants. The dataset includes raw images of the subjects as well as normalized eye images.

The MPIIFaceGaze subset provides face images. Labels for eye landmarks, on-screen and 3D gaze targets, estimated 3D head pose, and the position of eyes in the camera coordinate system are provided for each image/subject. The MPIIGaze dataset was chosen because it has the same setup as the intended use-case (e.g. using a laptop built-in camera and predicting 2D screen targets) and offers a good distribution of illumination, head poses, and gaze angles. It is also a medium-sized dataset, allowing for several experiments. Some samples of the MPIIFaceGaze dataset can be observed in Figure 2.

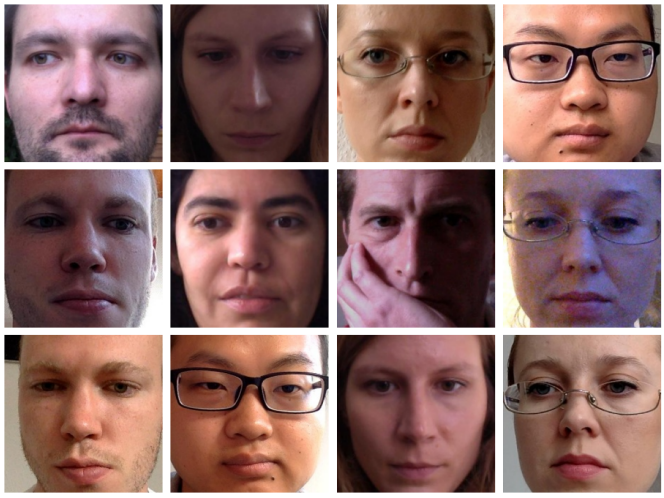


Fig. 2: Random subject face images from the MPIIFaceGaze dataset [18]

Data Pre-processing

The prediction of the 3D gaze direction in spherical coordinates (yaw and pitch) follows the example of [2]. To obtain these coordinates, the pre-processing scripts from [1] are used. Their approach is described in Figure 3. In short, the virtual camera is rotated such that the z-axis points towards the subject, then it is rotated to ensure the person's head is straight, and then the image is scaled to maintain a consistent distance between the subject and the virtual camera.

After applying the rectification method, the 3D vector to be predicted is defined as having one of its axes perpendicular to the screen plane. Predicting a value for this axis is not valuable, as it often remains constant (given by the normalized distance between the subject and the camera). Therefore, only the other two axes are predicted by transforming the 3D vectors into pitch/yaw rotation movements. As a result, normalized face

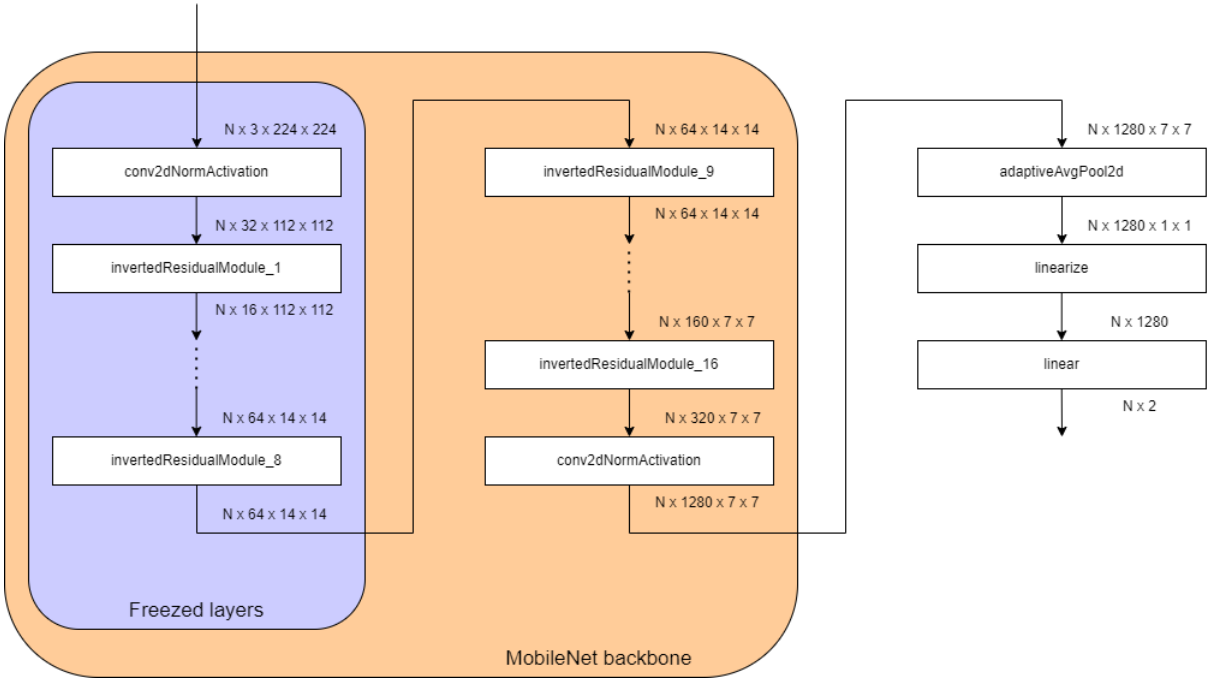


Fig. 1: FastSightNet model architecture

RGB images of size 224x224 and normalized labels for gaze angles (yaw and pitch) are obtained and used for training.

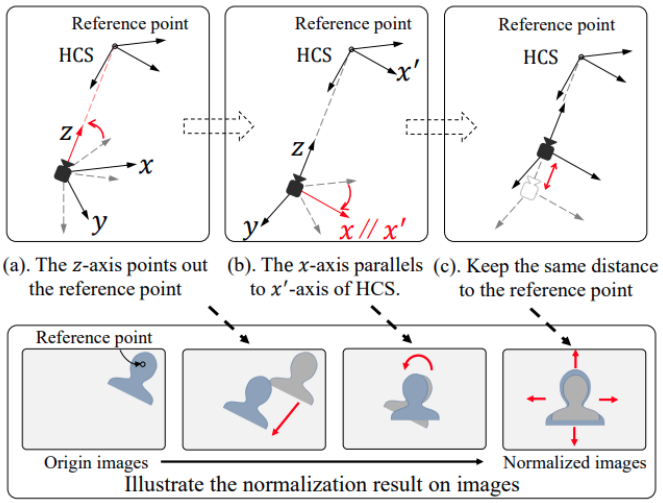


Fig. 3: Data pre-processing for gaze estimation [3]

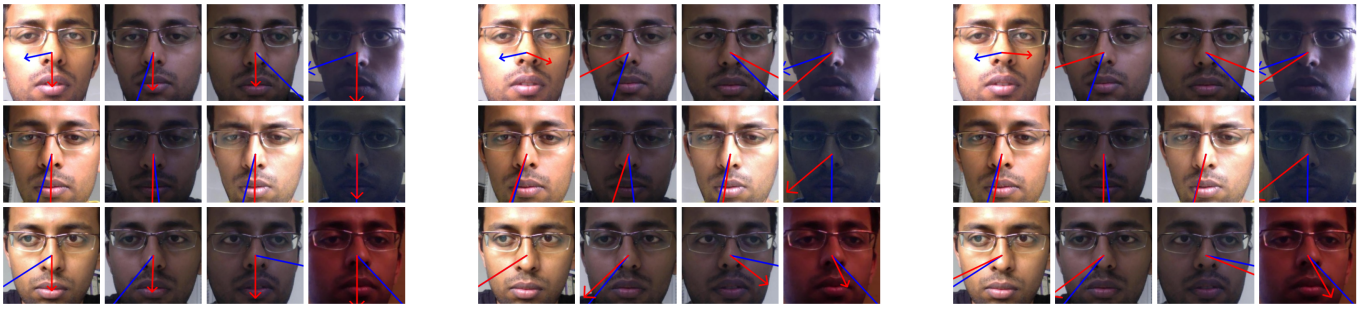
Training and Tuning

The first 9 layers of the backbone were frozen since they were trained on the ImageNet dataset and had already learned most of the small patterns. However, the rest of the backbone and the additional layers at the end were trained on the MPIIFaceGaze dataset. For training, the mean square error defined in Equation 1 was used as the loss function, and the mean angular error in Equation 2 was used as the evaluation metric. The leave-one-subject-out evaluation was performed on subject 10. The training set contains 42,000 samples (3,000 samples per subject for 14 subjects), while the test set contains 3,000 samples (1 subject).

$$\text{MSE}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

$$L_{\text{angular}} = \arccos\left(\frac{\mathbf{g}\cdot\hat{\mathbf{g}}}{|\mathbf{g}||\hat{\mathbf{g}}|}\right) \tag{2}$$

After conducting manual hyperparameter tuning on the learning rate, batch size, number of epochs, random seed, and comparing training runs on different folds, it was concluded that the best performing stable model was trained using a learning rate of 0.00001, batch size of 32, 10 epochs, and the Adam optimizer. The model achieved 4.6° mean angular error on the test set. The loss trend during training can be observed in Figure 5, and the angular error trend can be seen in Figure 6. The model's performance was also evaluated after each epoch of training by inferencing on the same 12 selected images from the evaluation set. The improvements over time can be seen in Figure 4.



(a) After 1 epoch (b) After 5 epochs (c) After 10 epochs

Fig. 4: Labeled random samples from the evaluation set. Blue arrow is the groudtruth, red arrow is the prediction. Visual evaluation on the test set during the training process of the FastSightNet model. MobileNetV2 backbone. Trained on the MPIIFaceGaze dataset with 42000 training samples and 3000 testing samples. Evaluation on subject 10. Batch size 32. Learning rate 0.00001. 10 epochs. Adam optimizer. Trained on an NVIDIA GeForce GTX 1060.

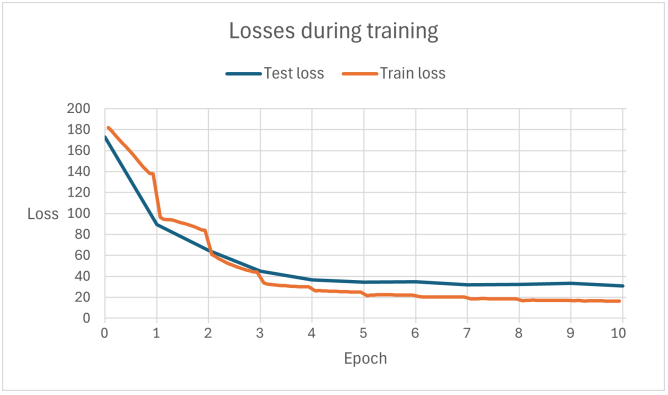


Fig. 5: Loss trend during training of the FastSightNet model. MobileNetV2 backbone. Trained on the MPIIFaceGaze dataset with 42000 training samples and 3000 testing samples. Evaluation on subject 10. Batch size 32. Learning rate 0.00001. 10 epochs. Adam optimizer. Trained on an NVIDIA GeForce GTX 1060.

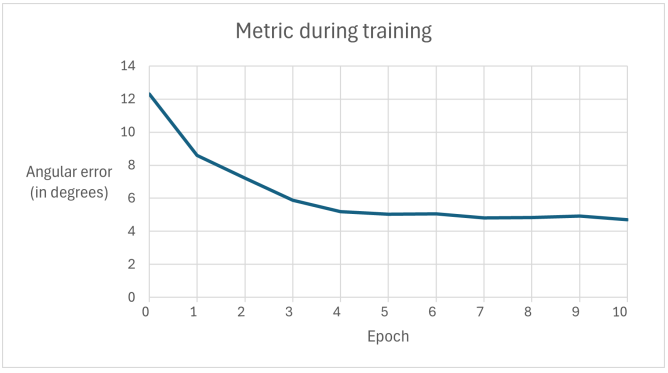


Fig. 6: Metric trend during training of the FastSightNet model. MobileNetV2 backbone. Trained on the MPIIFaceGaze dataset with 42000 training samples and 3000 testing samples. Evaluation on subject 10. Batch size 32. Learning rate 0.00001. 10 epochs. Adam optimizer. Trained on an NVIDIA GeForce GTX 1060.

Data Post-processing

The main use-case of the gaze estimation model is to use it as a human-computer interface. For instance, the goal might be to move the cursor on the screen as indicated by the gaze estimation model. To perform this action, it is necessary to translate from 3D output vectors to 2D points of gaze on the screen. Inspired by the work in [3], their methods are reproduced here. Firstly, the gaze vectors are intersected with the screen plane. It is assumed that the person's face is centered on the camera at a distance of 50 cm and perpendicular to the screen. After performing the intersection, a point location in the camera coordinate system is obtained. Next, the gaze point relative to the top-left corner of the screen is determined. The hard-coded position of the camera relative to the top-left corner of the screen is used, and a rotation is performed to express the gaze point in the screen coordinate system. Millimeters are used as units, and the conversion into pixels is not a difficult problem. A diagram of this method can be observed in Figure 7.

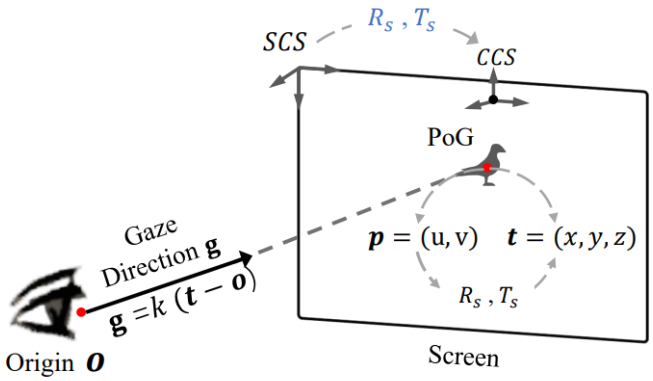


Fig. 7: 3D to 2D conversion [3]

IV. Discussion

To compare the results with other methods, the L2CS-Net model [2] was reproduced. The L2CS-Net is based on a ResNet-50 architecture [8]. A ResNet-18 version of the L2CS-Net was also trained and tuned because of its faster inference time.

The performance of all three models was evaluated using leave-one-subject-out cross-validation. Results can be seen in Figure 8. It can be observed that FastSightNet achieved the best performance on subject 0 with a 3.14° angular error. The performance trend is comparable with the L2CS-Net model. FastSightNet performed better than L2CS-Net on subject 11. Overall, FastSightNet achieved a mean angular error of 5.1° compared to L2CS-Net's error of 3.9°, resulting in a decrease in performance of 1.2°.

The advantages of the FastSightNet model are its inference time, training time, and model size, as shown in Table I. FastSightNet shows a 70% decrease in inference time, a 95% decrease in training time, and a 90% decrease in the number of parameters compared to the reproduced L2CS-Net model. FastSightNet exceeds the goal of 60 frames per second, making it usable in day-to-day workflows from the inference time perspective. On the other hand, the reproduced L2CS-Net is not ideal for real-time use cases since it only runs at 24 frames per second. As one can see there is a trade-off between the accuracy of the model and the inference time. For instance, FastSightNet shows a decrease in performance of 30%, but on the other hand, the inference time improved by over 70%.

TABLE I: Model performance comparison

Model	Inference time	Training time	Parameters
L2CS-Net (reproduced)	0.041s (24 fps)	10h10min	23,628,932
L2CS-Net (tuned ResNet-18)	0.017s (59 fps)	3h45min	11,206,788
FastSightNet (own)	**0.013s (77 fps)**	**30min**	**2,226,434**

V. GazeTrack system

A system was devised to preview the performance of gaze estimation models. The system operates in real-time on the webcam feed, utilizing RetinaFace [4] to detect the face of the subject in the frame. There are two types of previews: real-time webcam feed enhanced with 3D gaze vectors (see Figure 9a) and a tracing task where the user is requested to track a blue dot across the screen, with the gaze estimation shown in red (see Figure 9b).

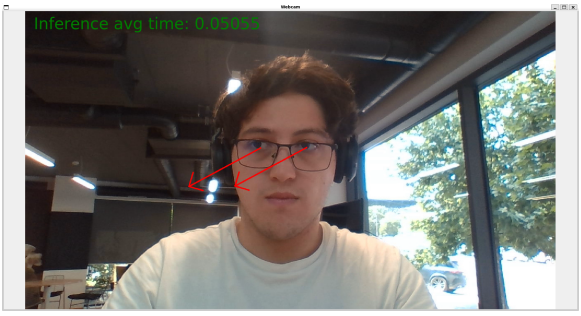
Model performance across diverse conditions

The FastSightNet model performance was explored across multiple subjects, different lighting conditions, and other common scenarios in day-to-day life. Several human subjects were asked to test the system. From an ethical perspective, approval to use their faces was collected through a Google form. Results are shown in Figure 10. The model appears to generalize well across different subjects.

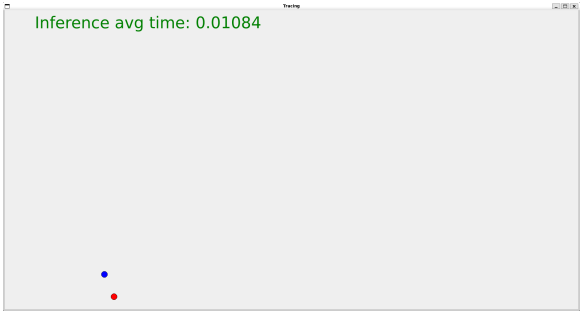
The system's performance in various lighting conditions was also evaluated. Figure 11 shows inference results for subjects in medium/low luminosity environments and with light sources placed at different angles relative to the subject. The model performs well under these conditions, possibly due to the wide range of luminosity conditions in the training set. Additionally, reflections from eye-glasses do not significantly affect performance.

Lastly, the model's performance was evaluated for subjects not centered in the frame, frames containing multiple subjects, blurry images, and different head angles. For reference see Figure 12. When multiple faces are detected in the frame, gaze prediction is performed only on the first identified subject. An improvement would be to perform gaze estimation on the subject with the largest face area. The 3D gaze estimation works well even when the camera is not centered or the head is tilted, although the intersection with the 2D screen is affected because of the assumptions about the subject's position relative to the camera.

A visual estimation of the prediction error in the 2D tracking demo indicates that predictions are usually 1-7 cm off, but can be up to 10 cm off when looking at the bottom center of the screen. The average prediction error is estimated to be around 5 cm. An improvement would be to display the exact distance between the target and the prediction on screen. For the system to work efficiently as a human-computer interface (e.g. to control the cursor with the gaze), the prediction error should be further reduced.



(a) Enhanced real-time webcam. Gaze prediction is shown as 3D gaze vectors (red arrows). In the top left corner the average inference time is shown.



(b) Tracing demo. A target (blue dot) is shown on top of the white canvas. The gaze prediction (red dot) and the average inference time are also shown.

Fig. 9: Different types of GazeTrack views

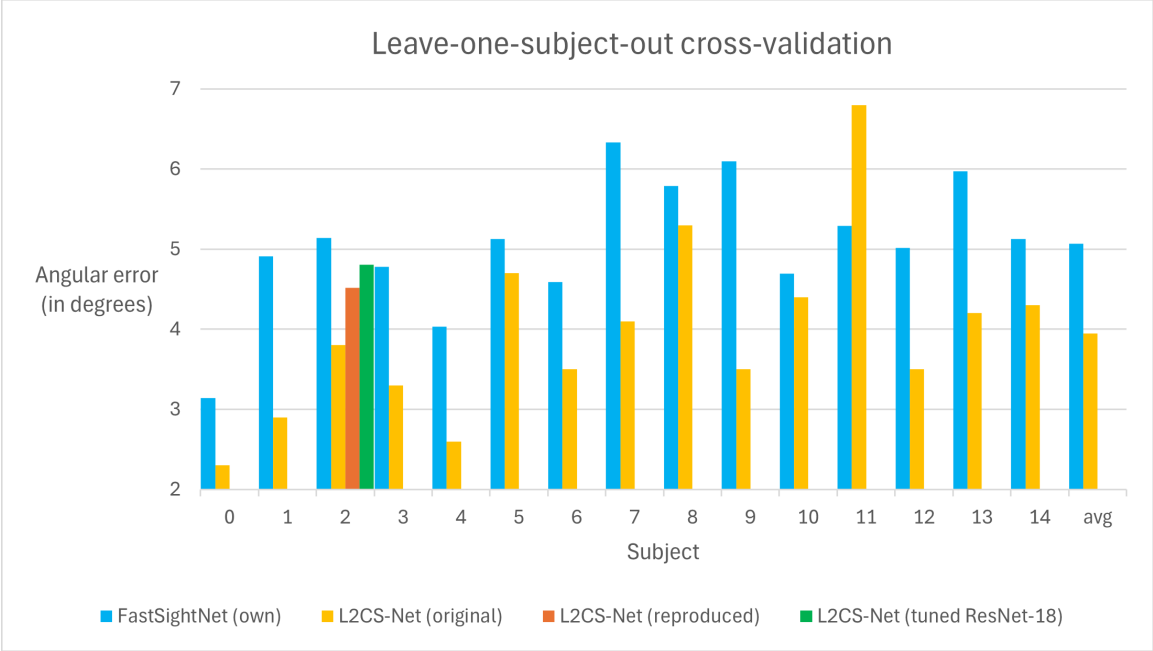
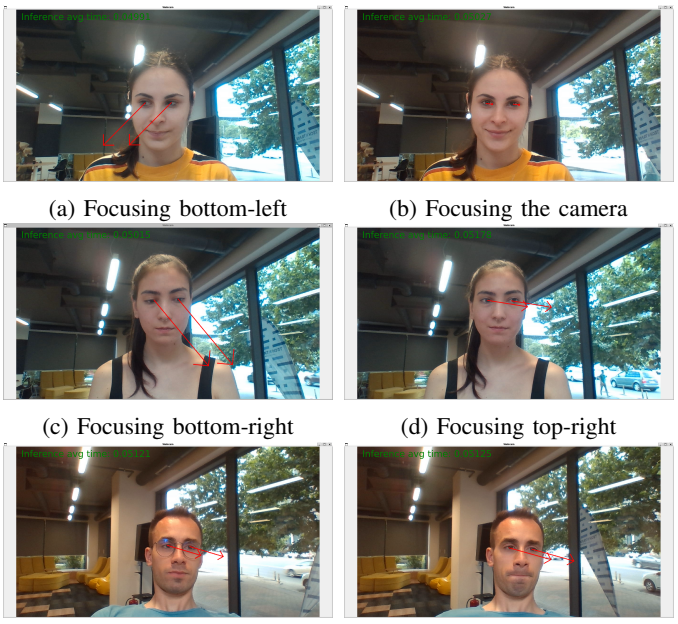


Fig. 8: Cross-validation comparison between FastSightNet and L2CS-Net models. FastSightNet is based on the MobileNetV2 backbone. It was trained for 10 epochs, batch size 32, learning rate of 0.00001. The original L2CS-Net [2] as well as the reproduced version are based on a ResNet-50 backbone. They were trained for 50 epochs, batch size 16, and a learning rate of 0.00001. The optimized version of L2CS-Net is based on a ResNet-18 backbone and it was trained for 50 epochs, with a batch size of 64, and a learning rate of 0.0001.

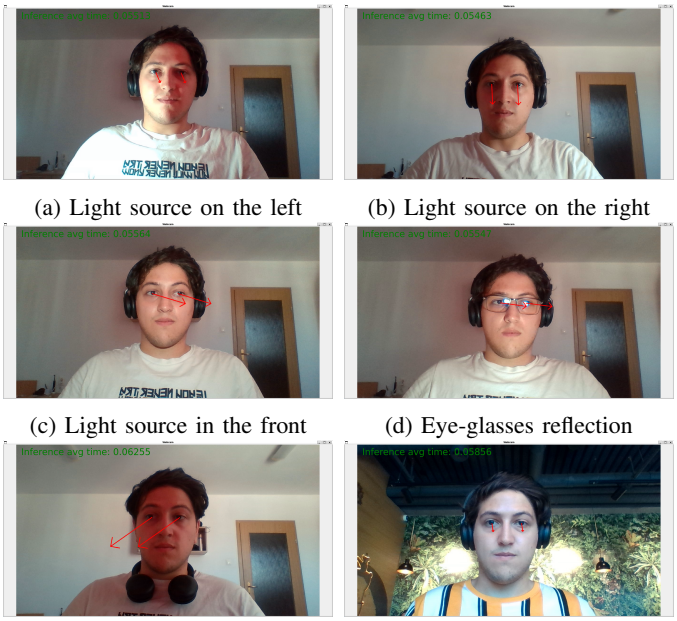


(a) Focusing bottom-left (b) Focusing the camera

(c) Focusing bottom-right (d) Focusing top-right

(e) Focusing top-right (with eye-glasses) (f) Focusing top-right (without eye-glasses)

Fig. 10: FastSightNet model performance on different subjects

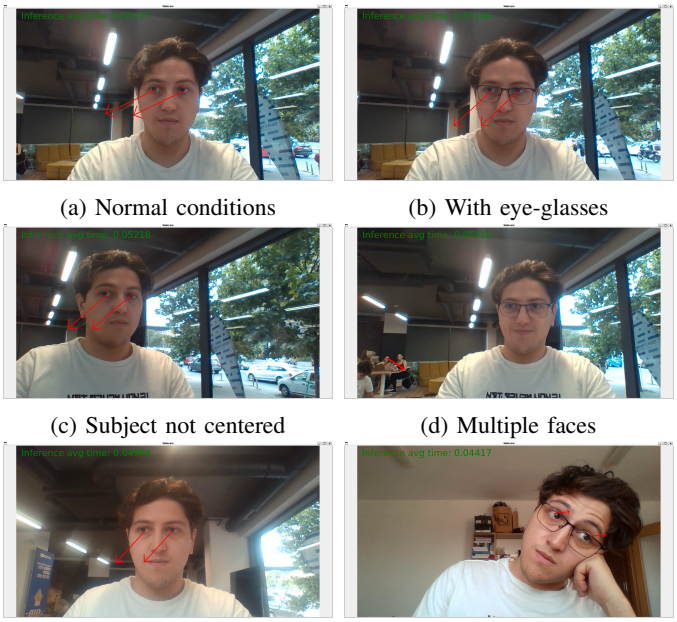


(a) Light source on the left (b) Light source on the right

(c) Light source in the front (d) Eye-glasses reflection

(e) Low luminosity (f) Medium luminosity

Fig. 11: FastSightNet model performance in different lighting conditions



(a) Normal conditions (b) With eye-glasses

(c) Subject not centered (d) Multiple faces

(e) Blurry image (f) Tilted head pose

Fig. 12: FastSightNet model performance in complex environment and subject conditions

VI. Conclusion

In this work, the 3D appearance-based gaze estimation problem was approached by first reproducing, training, tuning, and evaluating existing state-of-the-art models such as the L2CS-Net [2]. Next, a new model called FastSightNet, based on the more efficient MobileNet architecture [9], was proposed. Hyperparameter tuning was performed, and the best model was compared to the original L2CS-Net model. The FastSightNet model achieved a 5.1° mean angular error however, it is 90% smaller in size and can be trained 95% faster, while the inference speed for FastSightNet is 77 frames per second compared to the reproduced L2CS-Net speed of 24 frames per second. The proposed model might be better suited for larger systems where the trade-off between inference speed and accuracy is crucial.

The GazeTrack system was proposed, which enables users to evaluate different models in both 3D and 2D use-cases. During the evaluation phase it was concluded that the proposed solution has good performance but is not yet suitable for real-world scenarios due to the high prediction error of more than 5 cm in 2D screen coordinates. However, the idea that a MobileNet architecture is able to learn the complex patterns of the gaze estimation task was validated.

Future Work

The limitations of the MPIIFaceGaze dataset in terms of gaze angle distribution (30° horizontally and 20° vertically) were identified, motivating a move towards other datasets such as the ETH-XGaze [16] dataset, which has extreme gaze angles and might offer better results in unconstrained real-life scenarios. Validating the model on a larger dataset is also an important step.

For the system to be practically viable in real-life scenarios (e.g. to control the cursor with the gaze), solutions with better performance while maintaining the same inference speed should be researched. A step forward would be to experiment with different layer configurations for the FastSightNet model or freeze fewer backbone layers which can improve the model's performance on the new task by enabling the model to also adjust these initial weights.

References

[1] Gazehub. https://phi-ai.buaa.edu.cn/Gazehub/. Accessed: 2024-04-23.

[2] Ahmed A. Abdelrahman, Thorsten Hempel, Aly Khalifa, and Ayoub Al-Hamadi. L2cs-net: Fine-grained gaze estimation in unconstrained environments. *CoRR*, abs/2203.03339, 2022.

[3] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *CoRR*, abs/2104.12668, 2021.

[4] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *CoRR*, abs/1905.00641, 2019.

[5] Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review. *CoRR*, abs/2009.07485, 2020.

[6] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional LSTM networks for improved phoneme classification and recognition. In Wlodzislaw Duch, Janusz Kacprzyk, Erkki Oja, and Slawomir Zadrozny, editors, *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, 15th International Conference, Warsaw, Poland, September 11-15, 2005, Proceedings, Part II*, volume 3697 of *Lecture Notes in Computer Science*, pages 799–804. Springer, 2005.

[7] Yiran Guan, Zhuoguang Chen, Wenzheng Zeng, Zhiguo Cao, and Yang Xiao. End-to-end video gaze estimation via capturing head-face-eye spatial-temporal interaction context. *IEEE Signal Process. Lett.*, 30:1687–1691, 2023.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[10] Petr Kellnhofer, Adrià Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6911–6920. IEEE, 2019.

[11] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra M. Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2176–2184. IEEE Computer Society, 2016.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[14] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9367–9376. IEEE, 2019.

[15] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1821–1828. IEEE Computer Society, 2014.

[16] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 365–381. Springer, 2020.

[17] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4511–4520. IEEE Computer Society, 2015.

[18] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2299–2308. IEEE Computer Society, 2017.