

# Eye Gaze Tracking for Detecting Non-verbal Communication in Meeting Environments

Naina Dhingra<sup>a</sup>, Christian Hirt<sup>b</sup>, Manuel Angst and Andreas Kunz<sup>c</sup>

*Innovation Center Virtual Reality, ETH Zurich, Zurich, Switzerland*

**Keywords:** Eye Gaze, Eye Tracker, OpenFace, Machine Learning, Support Vector Machine, Regression, Data Processing.

**Abstract:** Non-verbal communication in a team meeting is important to understand the essence of the conversation. Among other gestures, eye gaze shows the focus of interest on a common workspace and can also be used for an interpersonal synchronisation. If this non-verbal information is missing and or cannot be perceived by blind and visually impaired people (BVIP), they would lack important information to get fully immersed in the meeting and may feel alienated in the course of the discussion. Thus, this paper proposes an automatic system to track where a sighted person is gazing at. We use the open source software 'OpenFace' and develop it as an eye tracker by using a support vector regressor to make it work similarly to commercially available expensive eye trackers. We calibrate OpenFace using a desktop screen with a  $2 \times 3$  box matrix and conduct a user study with 28 users on a big screen (161.7 cm x 99.8 cm x 11.5 cm) with a  $1 \times 5$  box matrix. In this user study, we compare the results of our developed algorithm for OpenFace to an SMI RED 250 eye tracker. The results showed that our work achieved an overall relative accuracy of 58.54%.

## 1 INTRODUCTION

One important factor of non-verbal communication is that people are often looking at artifacts on the common work space or at the other person when collaborating with each other. Eye gaze provides information on emotional state (Bal et al., 2010), text entry (Majaranta and R  ih  , 2007), or concentration for an object (Symons et al., 2004) given by the user, to infer visualization tasks and a user's cognitive abilities (Steichen et al., 2013), to enhance interaction (Hennessey et al., 2014), to have communication via eye gaze patterns (Qvarfordt and Zhai, 2005), etc. However, such information cannot be accessed by blind and visually impaired people (BVIP) as they cannot see where the other person in the meeting room is looking at (Dhingra and Kunz, 2019; Dhingra et al., 2020). Therefore, it is important to track eye gaze in the meeting environment to provide the relevant information to them.

Eye gaze tracking is locating the position where a person is looking at. This specific spatial position is known as the point of gaze (O'Reilly et al., 2019). It has been employed for research in scan patterns and

attention in human-computer interaction, as well as in psychological analysis. Eye gaze tracking technology is categorized into two categories, i.e., head-mounted systems and remote systems where head-mounted eye trackers are mobile and remote systems are stationary trackers.

Early eye tracking systems were based on metal contact lenses (Agarwal et al., 2019a), while today's eye trackers use an infrared camera and a bright or dark pupil technique (Duchowski, 2007). These techniques locate the pupil's center. The tracker can then locate the target's position on the screen where the person is gazing at using the relative position of the corneal reflection and the pupil center. Other eye trackers which are based on high speed video cameras are more expensive than the infrared based eye trackers, but are also more accurate than webcam based eye trackers (Agarwal et al., 2019a). In such eye trackers, the measurement is done based on deep learning and computer vision applications (Kato et al., 2019), (Yiu et al., 2019).

Based on the possibility to improve the accessibility of non-verbal communication for BVIP, our work will use eye gaze tracking to detect where people are looking at. Based on the availability and known advantages of the systems, the OpenFace and SMI RED 250 were chosen to be included in the analysis. OpenFace is an open-source software for real-time face

<sup>a</sup>  <https://orcid.org/0000-0001-7546-1213>

<sup>b</sup>  <https://orcid.org/0000-0003-4396-1496>

<sup>c</sup>  <https://orcid.org/0000-0002-6495-4327>

embedding visualization and feature extraction that works with webcams. The commercialized SMI RED 250 remote eye tracker comes with the iView software to process the data.

The main contributions of this work are as follows: (1) We developed an overall low-cost eye tracker which uses a webcam and the OpenFace software together with support vector machines to improve accuracy; (2) we designed this eye tracking system to be used in real-time in meeting environments; (3) we performed a user study with 28 users to evaluate the performance of this new eye gaze tracking approach; and (4) we evaluated the effect of users wearing glasses and not wearing glasses on performance.

The motivation of our work is the use of a low cost webcam along with a free open source face feature detection software, since commercial eye tracking systems are usually not available in typical meeting rooms and are very expensive.

This paper is organized as follows: Section 2 describes the state of the art in eye tracking. Section 3 briefly describes the methods and techniques used in our system, while Section 4 gives details about the experimental setup. In section 5, we show an overview of the conducted user study and discusses the achieved results. Finally, Section 6 concludes the paper with future work and improvements of the current system.

## 2 STATE OF THE ART

The advancement of computers and peripheral hardware led to several different applications which are based on gaze interactions. These applications can be divided into various sub-categories: TV panels (Lee et al., 2010), head-mounted displays (Ryan et al., 2008), automotive setups (Ji and Yang, 2002), desktop computers (Dong et al., 2015), (Pi and Shi, 2017), and hand-held devices (Nagamatsu et al., 2010). Numerous researchers have worked on other eye gaze interactions such as public and large displays. (Drewes and Schmidt, 2007) worked on eye movements and gaze gestures for public display application. Another work by (Zhang et al., 2013) built a system for detecting eye gaze gestures to the right and left directions. In such systems, either hardware-based or software-based eye tracking is employed.

### 2.1 Hardware-based Eye Gaze Tracking Systems

Hardware-based eye gaze trackers are commercially available and usually provide high accuracy that

comes with a high cost of such devices. Such eye gaze trackers can further be categorized into two groups, i.e., head-mounted eye trackers and remote eye trackers. Head-mounted devices usually consist of a number of cameras and near-infrared (NIR) light emitting diodes (Eivazi et al., 2018), being integrated in the frame of goggles. Remote eye trackers on the other hand are stationary. We used an SMI RED 250 eye tracker which is one of the remote eye trackers. Using this device, we built an automatic eye gaze tracking system for users sitting in the meeting environment.

### 2.2 Software-based Eye Gaze Tracking Systems

Software-based eye gaze tracking uses features extracted from a regular camera image by computer vision algorithms. In (Zhu and Yang, 2002), the center of the iris is identified using an interpolated Sobel edge detection. Head direction also plays a significant role in the eye gaze tracking, e.g. in (Valenti et al., 2011) where a combination of eye location and head pose is used. In (Torricelli et al., 2008), a general regression neural network (GRNN) is used to map geometric features of the eye position to screen coordinates. The accuracy of GRNN is depending on the input vectors. A low cost eye gaze tracking system is developed in (Ince and Kim, 2011) using the eye pupil center detection and movement. It performed well on a low resolution video but had the drawback of being dependant on head movement and pose. Webcam-based gaze tracking has also been researched in several different works using computer vision techniques (Dostal et al., 2013; Agarwal et al., 2019b). These works implemented the feature detection in a proprietary way, whereas we are implementing our system with an open source software which detects feature with good accuracy and we tuned the mapping from detected features to screen coordinates based on a commercially available eye tracker using support vector machine regression.

## 3 METHODOLOGY

### 3.1 OpenFace

OpenFace (Baltrušaitis et al., 2016) is an open-source software which can be used in real-time for analyzing facial features. The software has various features: facial landmark detection (Amos et al., 2016), facial landmark and head pose tracking, eye gaze tracking, facial action unit detection, behavior analysis (Baltrušaitis et al., 2018), etc. We used OpenFaceOf-

fine out of various applications available in the OpenFace package. OpenFace can analyze videos, images, image sequences, and live webcam videos. Gaze recordings include gaze directions for both eyes separately, an averaged gaze angle, and eye landmarks in two-dimensional image coordinates and in three-dimensional coordinates of the camera's coordinate system. Additionally, the timestamp and the success rate are recorded automatically.

### 3.2 Eye Tracker SMI RED 250 with iView

The SMI RED 250 comes with the modular design which can be integrated into numerous configurations ranging from a small desktop screen to big television screens or projectors. It utilizes head movement and eye tracking along with the pupil and gaze data to achieve accurate results. SMI claims to have robust results regardless of the age of the user, glasses, lenses, eye color, etc. The system needs to be calibrated which takes a few seconds and maintains its accuracy during the duration of the experiments. It can track eye gaze up to 40 degrees in horizontal direction and 60 degrees in vertical direction. iView software is provided by SMI along with the eye tracker for the data output and processing.

### 3.3 Support Vector Machine Regression

Support vector machines (SVMs) were developed as a binary classification algorithm to increase the gap between different categories or classes from the training set (Suykens and Vandewalle, 1999). SVMs are also used as a regression tool with an intuition to build hyperplanes which are as close as possible to training examples. For more detailed mathematical information on SVMs for regression, refer to (Smola and Schölkopf, 2004). In our system, we will use SVMs to assign measurement values to the predefined classes although such data might not precisely match the training data.

### 3.4 Basic Pipeline

In the experiments, we used the SMI eye tracker with the iView software and compared it to a regular webcam together with the open-source software OpenFace with some adjustments to their output using the aforementioned SVMs. Since the output from OpenFace was given in terms of eye position and eye gaze direction, we used a mathematical and geometrical manipulation to convert these given vectors to screen coordinates.



Figure 1: Setup for the small desktop screen.

For the comparison, two different setups were used: a 22" monitor at 0.6m distance (see Figure 1), and a 65" screen at a distance of 3m to the user (see Figure 4). We also performed a user study with 28 users on this 65" screen. In both cases, the SMI eye tracker and the webcam were placed at a distance of 0.6m to the user. These two different setups were used to prove the robustness of our system as well as to validate our algorithm. It also evaluates that this system can be employed in meeting environments where sighted people might look at two screens placed at different distances having different measurement noise.

### 3.5 Desktop/Small Screen Setup

The setup used for the experiments with the desktop screen is shown in Figure 1. The user had to look at different regions on the desktop screen while his eye gaze was measured by the commercial eye gaze tracker (using iView) as well as by the the webcam (using OpenFace without any correction algorithm). The experiments with this setup showed that the screen coordinates from OpenFace are accumulated at a sub-region of the whole screen as shown in Figure 5.

Because of the accumulation of points, we used an SVM algorithm to convert those coordinates into a similar form as the output coordinates from the iView software. Figure 6 shows the scatterplot for iView and OpenFace coordinates after using the SVM algorithm for correction. We used 70% of the data for training and 30% of the data for testing.

We used a  $2 \times 3$  matrix as shown in Figure 2 to evaluate the manipulated output from OpenFace and SMI RED 250 in terms of 6 classes. The user was told to look at the numbered fields of this matrix and his eye gaze was simultaneously measured using the webcam with OpenFace and using the SMI RED 250 with iView. The measurements stemming from the SMI RED 250 were taken as ground truth. Figure 3

shows the accuracy per box for the  $2 \times 3$  matrix. It is evident that OpenFace performs better for the middle boxes than for the boxes 4 and 6. The results are also shown in Table 1. We achieved an accuracy of **57.69%** on the test data using the SVM algorithm for regressing points from OpenFace to iView coordinates.



Figure 2:  $2 \times 3$  matrix for the comparison of OpenFace and iView after applying the correction with the SVM.

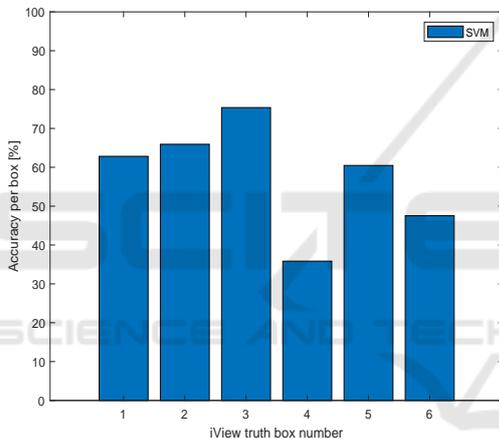


Figure 3: Accuracy per box for OpenFace (with SVM applied) when the iView box number is considered as ground truth.

Table 1: The accuracy of OpenFace with SVM using the SMI RED 250 eye tracker as reference. Values are in %.

Box Number	OpenFace with SVM
<b>Overall</b>	<b>57.69</b>
1	62.82
2	65.92
3	75.36
4	35.84
5	60.47
6	47.54

## 4 EXPERIMENTAL SETUP FOR USER STUDY

The setup consists of a demo environment where a sighted person is looking at a screen. We are aiming

to provide the useful information to the BVIP about the location of the sub-region of the screen where a person in the meeting is looking at. We used  $1 \times 5$  boxes in the matrix shown to the user at the time during the user study. In our application, we are concerned about the region of interest of the person gazing at the screen, but not in the particular location. Accordingly, we assume that the screen is divided into 5 sub-parts and aim to provide high accuracy for predicting the sub-part of interest of the user. The experimental setup is shown in 4.

The user was asked to look at a numbered region for few seconds and then the next region number was given. The sequence of region numbers a user had to look at was the same for every user to keep the uniformity. The SMI RED 250 eye-tracker with iView and OpenFace were used to take measurements simultaneously. The data was refined and processed for the sample values which had the same time stamp.



Figure 4: Experimental setup for user study.

## 5 USER STUDY AND RESULTS

We conducted the user study with 28 users which had a mix of people with and without spectacles or lenses. We used the SMART Board® 400 series interactive overlay flat-panel display named as SBID-L465-MP. It has a 65" screen diagonal and its dimensions are: 161.7 cm  $\times$  99.8 cm  $\times$  11.5 cm. The SMI RED 250 eye-tracker with iView and the webcam with OpenFace were used to take measurements simultaneously. We used the same approach as for the small desktop screen. But in this case instead of training the OpenFace points to perform similar to the iView points, we train to compare the performance using the ground truth given by the known positions of the 5 different fields. We used the SVM algorithm to perform a spatial manipulation of the data so that the accumulated

OpenFace data are similar to the iView data as shown in Figures 5 and 6.

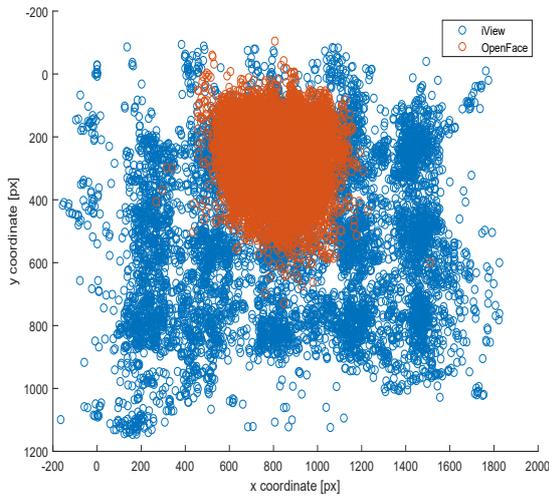


Figure 5: Scatter plots showing the output data from iView and OpenFace. It is shown that the output from the OpenFace is concentrated in the certain area of the the screen.

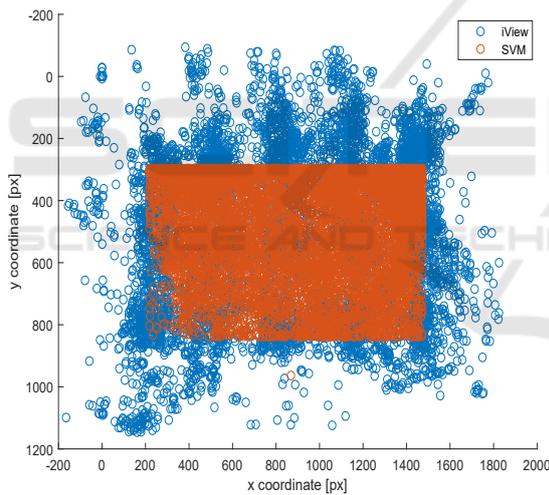


Figure 6: Support vector machines algorithm is used to regress the data in a way so that OpenFace produces results similar to iView.

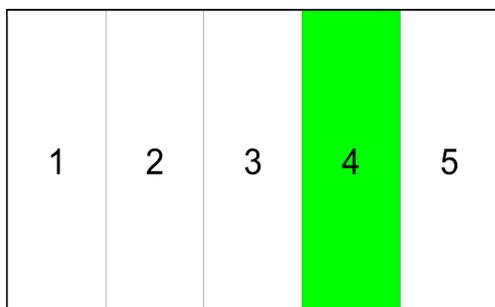


Figure 7: Numbers displayed on the big screen at a distance of 3m from the user.

We asked each user to sit in front of the big screen at a distance of 3 meters. The screen displayed a matrix of  $1 \times 5$  numbers. The users were asked to look at those numbers as shown in Figure 7. Whenever a user looked at some number, that region was highlighted in green. We gave the same sequence of numbers to all 28 users to look at. Then, we saved the raw data from both software, i.e., iView and OpenFace.

Table 2: Accuracy of the SMI RED 250 eye tracker compared to OpenFace with SVM.

Box Number	SMI RED 250	OpenFace with SVM
<b>Overall</b>	<b>81.48</b>	<b>58.54</b>
1	84.03	42.25
2	81.72	62.89
3	78.85	67.46
4	81.04	70.41
5	81.74	49.71

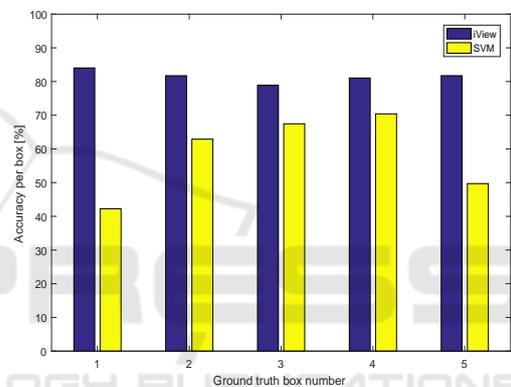


Figure 8: Accuracy per box for all the users.

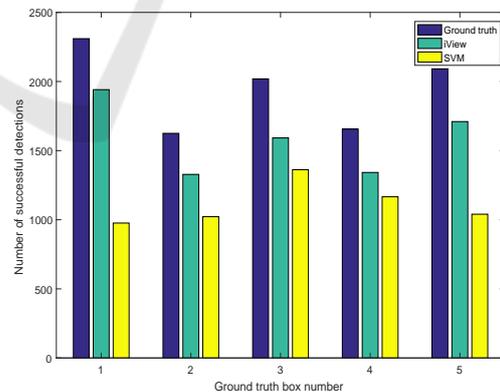


Figure 9: Number of successful detections for the test data in user study.

We analyzed the raw data to know whether the measured position coincides with the original position of the corresponding box number on the screen. We compared the output results from both software at the same time stamps with the ground truth numbers at which the user was asked to look at for a particular time stamp. The average accuracy for iView was

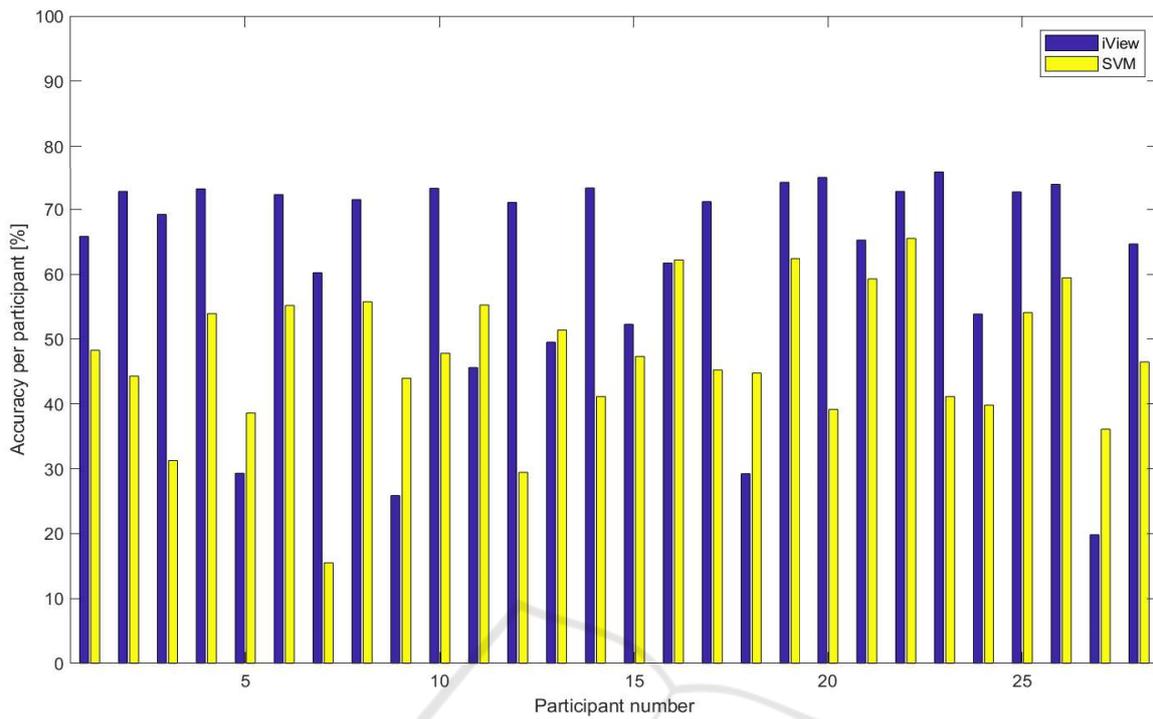


Figure 10: Accuracy per user for the big screen.

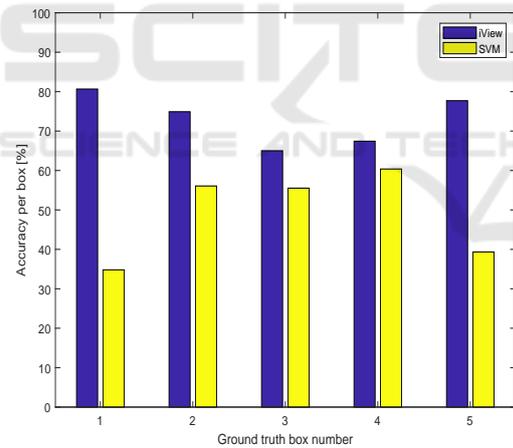


Figure 11: Accuracy per box for users with glasses.

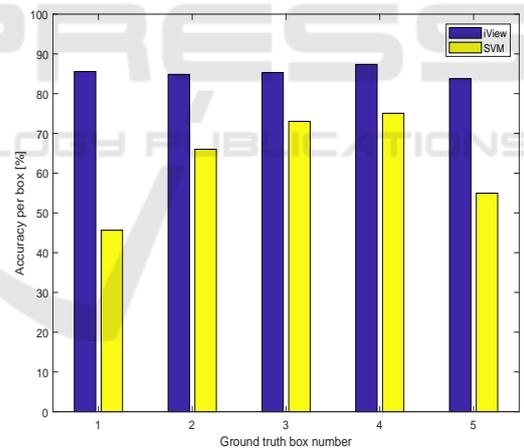


Figure 12: Accuracy per box for users without glasses.

Table 3: The accuracy of the SMI RED 250 eye tracker compared to OpenFace with SVM for users with and without glasses.

	SMI RED 250	OpenFace with SVM
glasses	75.02	50.90
no glasses	85.96	66.79

81.48% and for the OpenFace with SVM was 58.54%.

We further analyzed the accuracy of each box of a matrix as shown in Table 2. We see that the accuracy for the box number 1 and 5 is the lowest for OpenFace, which means that it is unable to recognize

correctly for the eye gaze at the corner boxes, while it performs better for the boxes in the middle. Figure 8 describes the performance comparison of our setup which shows that there is a potential of the open-source software such as OpenFace to work as an eye gaze tracker. Figure 9 shows the number of hits for each out of 5 boxes, which further tells us that the corner boxes, i.e., box number 1 and 5 have lowest number of hits which is in accordance to the accuracy achieved per box as shown in Figure 8.

Figure 10 shows the accuracy of each of the 28 users in our user study. This shows that for certain

users our system outperformed the iView SMI RED 250 eye tracker.

We also evaluated the results for the users wearing glasses and without glasses as shown in Figure 11 and Figure 12. It is evident that both systems work better with the user without glasses compared to users with glasses. Table 3 details the accuracy for the compared systems for users with and without glasses.

## 6 CONCLUSION

We worked on the eye gaze location detection on the screen for team meetings to help BVIP to get immersed in the conversation. We built a prototype of an automatic eye gaze tracking system which can be available at low cost using an open source software 'OpenFace'. We geometrically converted the eye gaze vectors and eye position coordinates to screen coordinates and manipulated those coordinates using an SVM regression algorithm to work in a similar manner to the commercially available SMI 250 RED eye tracker. We used a small desktop screen with  $2 \times 3$  box matrix to calibrate our proposed system for eye gaze tracking.

In our user study, we evaluated our automatic system with 28 users. We found out that our system works quite comparable to the SMI RED 250 eye tracker for the numbers which are inside the box matrix on the screen but inversely for the corner boxes on the screen which led to the accuracy difference between the proposed system and the SMI RED 250. We compared the performance of users with spectacles and without spectacles which showed that the users with spectacles had less accuracy than without spectacles which might be due to the extra reflection due to presence glasses.

In future work, we will convert the output in such a way that it can be made accessible to BVIP by audio or haptic feedback. We will also work on improving the accuracy of our system using neural networks which have proven to perform better than the classical computer vision techniques in other problems.

## ACKNOWLEDGEMENTS

This work has been supported by the Swiss National Science Foundation (SNF) under the grant no. 200021E 177542 / 1. It is part of a joint project between TU Darmstadt, ETH Zurich, and JKU Linz with the respective funding organizations DFG (German Research Foundation), SNF and FWF (Austrian Science Fund). We also thank Dr. Quentin Lohmeyer

and Product Development Group Zurich for lending us the SMI RED 250 eye tracker.

## REFERENCES

- Agarwal, A., JeevithaShree, D., Saluja, K. S., Sahay, A., Mounika, P., Sahu, A., Bhaumik, R., Rajendran, V. K., and Biswas, P. (2019a). Comparing two webcam-based eye gaze trackers for users with severe speech and motor impairment. In *Research into Design for a Connected World*, pages 641–652. Springer.
- Agarwal, A., JeevithaShree, D., Saluja, K. S., Sahay, A., Mounika, P., Sahu, A., Bhaumik, R., Rajendran, V. K., and Biswas, P. (2019b). Comparing two webcam-based eye gaze trackers for users with severe speech and motor impairment. In Chakrabarti, A., editor, *Research into Design for a Connected World*, pages 641–652, Singapore. Springer Singapore.
- Amos, B., Ludwiczuk, B., Satyanarayanan, M., et al. (2016). Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6.
- Bal, E., Harden, E., Lamb, D., Van Hecke, A. V., Denver, J. W., and Porges, S. W. (2010). Emotion recognition in children with autism spectrum disorders: Relations to eye gaze and autonomic state. *Journal of autism and developmental disorders*, 40(3):358–370.
- Baltrušaitis, T., Robinson, P., and Morency, L.-P. (2016). Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE.
- Baltrušaitis, T., Zadeh, A., Lim, Y. C., and Morency, L.-P. (2018). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE.
- Dhingra, N. and Kunz, A. (2019). Res3atn-deep 3D residual attention network for hand gesture recognition in videos. In *2019 International Conference on 3D Vision (3DV)*, pages 491–501. IEEE.
- Dhingra, N., Valli, E., and Kunz, A. (2020). Recognition and localisation of pointing gestures using a RGB-D camera. *arXiv e-prints*, page arXiv:2001.03687.
- Dong, X., Wang, H., Chen, Z., and Shi, B. E. (2015). Hybrid brain computer interface via bayesian integration of eeg and eye gaze. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 150–153. IEEE.
- Dostal, J., Kristensson, P. O., and Quigley, A. (2013). Subtle gaze-dependent techniques for visualising display changes in multi-display environments. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 137–148. ACM.
- Drewes, H. and Schmidt, A. (2007). Interacting with the computer using gaze gestures. In *IFIP Conference on Human-Computer Interaction*, pages 475–488. Springer.
- Duchowski, A. T. (2007). Eye tracking methodology. *Theory and practice*, 328(614):2–3.
- Eivazi, S., Kübler, T. C., Santini, T., and Kasneci, E. (2018). An inconspicuous and modular head-mounted

- eye tracker. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, page 106. ACM.
- Hennessey, C. A., Fiset, J., and Simon, S.-H. (2014). System and method for using eye gaze information to enhance interactions. US Patent App. 14/200,791.
- Ince, I. F. and Kim, J. W. (2011). A 2D eye gaze estimation system with low-resolution webcam images. *EURASIP Journal on Advances in Signal Processing*, 2011(1):40.
- Ji, Q. and Yang, X. (2002). Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-time imaging*, 8(5):357–377.
- Kato, T., Jo, K., Shibasato, K., and Hakata, T. (2019). Gaze region estimation algorithm without calibration using convolutional neural network. In *Proceedings of the 7th ACIS International Conference on Applied Computing and Information Technology*, page 12. ACM.
- Lee, H. C., Luong, D. T., Cho, C. W., Lee, E. C., and Park, K. R. (2010). Gaze tracking system at a distance for controlling iptv. *IEEE Transactions on Consumer Electronics*, 56(4):2577–2583.
- Majaranta, P. and Riih , K.-J. (2007). Text entry by gaze: Utilizing eye-tracking. *Text entry systems: Mobility, accessibility, universality*, pages 175–187.
- Nagamatsu, T., Yamamoto, M., and Sato, H. (2010). Mobigaze: Development of a gaze interface for handheld mobile devices. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3349–3354. ACM.
- O'Reilly, J., Khan, A. S., Li, Z., Cai, J., Hu, X., Chen, M., and Tong, Y. (2019). A novel remote eye gaze tracking system using line illumination sources. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 449–454. IEEE.
- Pi, J. and Shi, B. E. (2017). Probabilistic adjustment of dwell time for eye typing. In *2017 10th International Conference on Human System Interactions (HSI)*, pages 251–257. IEEE.
- Qvarfordt, P. and Zhai, S. (2005). Conversing with the user based on eye-gaze patterns. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–230. ACM.
- Ryan, W. J., Duchowski, A. T., and Birchfield, S. T. (2008). Limbus/pupil switching for wearable eye tracking under variable lighting conditions. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 61–64. ACM.
- Smola, A. J. and Sch lkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.
- Steichen, B., Carenini, G., and Conati, C. (2013). User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 317–328. ACM.
- Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.
- Symons, L. A., Lee, K., Cedrone, C. C., and Nishimura, M. (2004). What are you looking at? acuity for triadic eye gaze. *The Journal of general psychology*, 131(4):451.
- Torricelli, D., Conforto, S., Schmid, M., and D'Alessio, T. (2008). A neural-based remote eye gaze tracker under natural head motion. *Computer methods and programs in biomedicine*, 92(1):66–78.
- Valenti, R., Sebe, N., and Gevers, T. (2011). Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 21(2):802–815.
- Yiu, Y.-H., Aboulatta, M., Raiser, T., Ophey, L., Flanagan, V. L., zu Eulenburg, P., and Ahmadi, S.-A. (2019). Deepvov: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of neuroscience methods*.
- Zhang, Y., Bulling, A., and Gellersen, H. (2013). Sideways: a gaze interface for spontaneous interaction with situated displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 851–860. ACM.
- Zhu, J. and Yang, J. (2002). Subpixel eye gaze tracking. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 131–136. IEEE.