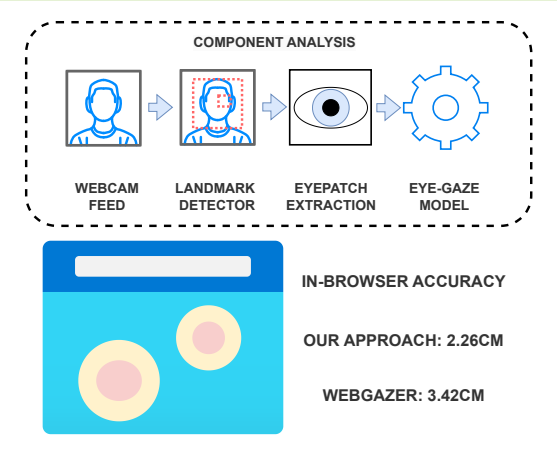
A study on the impact of different components of a traditional webcam-based 2D gaze tracking algorithm

William Blakey, Stamos Katsigiannis, *Member, IEEE*, and Naeem Ramzan, *Senior Member, IEEE*

***Abstract*— Webcam-based 2D gaze tracking algorithms are lightweight and are becoming increasingly used in the fields of medicine, market research and many others. As they become increasingly used, it becomes vital to break down their components to understand their limitations and better explore their practical implications. Key components of the gaze tracking pipeline are the calibration pattern, landmark detector, eye patch generation method, and the final eye-gaze model. Through an experimental framework, this work explores various methods for these components and evaluates the impact of each component on the final performance of an individualised real-time gaze tracking algorithm that is trained and tested on data from single individuals, as opposed to generalised approaches that are trained on data from multiple individuals. Gaze tracking data from users looking at a laptop screen were captured using a webcam and were used for the evaluation of the examined methods. The final proposed pipeline for individualised webcam-based real-time gaze-tracking under "real-world" use cases achieved a 2.26 cm accuracy compared to 3.42 cm for similar approaches. Additional validation on an independent publicly available dataset (EyeDIAP) further supports our findings.**



***Index Terms*— gaze tracking, eye tracking, webcam-based gaze tracking, real-time gaze tracking**

I. Introduction

Gaze tracking technology has gained substantial importance across various industries, spanning from accessibility tools to diagnosing neurological conditions [1], driving user experience decisions for website design [2], medical applications such as eye gaze controlled needle deployment robot and laparoscope [3], to the domain of advertising [4] and learning environments [5]. Initial breakthroughs were powered by the development of highly accurate infrared technology using Pupil Cornea Centre Reflection (PCCR) [6], driving innovation in both hardware and applications. Notably, it led to the creation of tools that empower individuals with impaired speech capabilities, such as those who have lost the use of their mouth and vocal cords, to communicate and drastically improve their quality of life [7], and with more accessible webcam-based approaches these tools may become available for more people at a lower cost.

Webcam-based gaze tracking has become widely used by many researchers by leveraging hardware that is readily available in everyday households. These systems open the door to a wide array of potential applications due to their scalability. Numerous works [8], [9] have conducted rigorous performance evaluation studies, primarily focusing on task-based accuracy, and have consistently found that these low-cost webcam-based trackers are effective in the domains of cognitive and behavioural research, such as studies related to understanding attention in education [10], to Autism Spectrum Disorder (ASD) [11]–[13], or to consumer visual attention to online advertisements [14]. Studies have been conducted to evaluate mobile phone trackers in the context of ASD diagnosis [15]. These studies used these trackers to conduct left and right screen experiments with toddlers [16] for the SenseToKnow app [17]. These contrast other approaches, such as Earlitec, who have published large scale validation studies [18] of infrared approaches with specialised hardware. The presence of camera and infrared approaches for diagnosing ASD should act as an interesting experiment for webcam-based vs infrared-based tracking. Furthermore, webcam-based gaze tracking pipelines, while offering a lower level of precision, excel in their capacity to capture copious amounts of data.

Received X November 2024; Revised X Month Year; Accepted ... Date of current version ...

W. Blakey and N. Ramzan are with the School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley, PA1 2BE, UK (e-mail: b00356762@studentmail.uws.ac.uk, naeem.ramzan@uws.ac.uk).

S. Katsigiannis is with the Department of Computer Science, Durham University, Durham, DH1 3LE, UK (e-mail: stamos.katsigiannis@durham.ac.uk).

Webcam-based gaze trackers follow two different approaches: (a) *Individualised tracking*, which achieves the best eye tracking results for an individual. Individualised trackers are trained and tested on data from a single individual, thus creating a gaze tracker optimised for the specific individual. (b) *Generalised tracking*, which aims to achieve the best eye tracking for anyone without any calibration for each individual. Generalised trackers are trained on data from multiple individuals and aim to work well for "unseen" individuals.

Individualised approaches such as Webgazer [19] and TurkerGaze [20] have mostly been overlooked in recent gaze tracking research, but not by researchers in other domains who find the disposable models and real-time Javascript implementations very accessible. Generalised approaches such as iTracker [21] have become the de facto state of the art, mainly due to their use of modern neural network architectures. However, there has not been any attempt to perform comparisons between individualised and generalised approaches, thus the gaze tracking community may have moved towards new methods without considering the practical implications and the trade-offs between generalised and individualised approaches. For example, Webgazer's JavaScript implementation has made it convenient and easily usable for both research and "real-world applications". The fact that it can work off training data supplied by the user makes it extremely versatile and it can operate on mobile phones, tablets and desktop/laptop computers with minimal tweaking. Generalised approaches are limited by the constraints of their dataset, whether it be mobile or desktop, and the specifics of the landmark detector used. This can have drawbacks for many researchers looking for gaze trackers to perform a study. The idea of having a calibration-less system is possible with generalised approaches, but in many situations further calibration is required [21], diminishing their benefits over individualised approaches.

This work aims to review the components of individualised pipelines to better understand whether this style of tracker has been overlooked. All gaze tracking pipelines are heavily limited by the components that they use, whether they be individualised or generalised, with both having many similarities. Gaze tracking pipelines typically include a landmark detector, an eye patch extractor and an eye gaze model. Most trackers utilise some form of landmark detection algorithm that provides eye locations for eye patches and face patches to be generated. Even in early iterations of gaze tracking pipelines [20], the impact of jitters from the landmark detector on the overall accuracy of the gaze prediction was evident. Although the large "*Eye tracking for everyone*" [21] dataset has enabled a lot of focus on the gaze model, most works building on it use the eye patches generated within the dataset and ignore the limitations of the components up to the model, such as the landmark detector and the eye patch extractor that have a large impact on the final prediction. Nevertheless, when used in real-time applications, the practical limitations of the specific landmark detector have to be considered. Finding the limitations of the gaze tracking pipeline's components and improving them is vital for the continuous improvement of all gaze tracking algorithms. In this work, we aim to break down many factors of gaze trackers and push for better understanding of the steps leading up to an eye model and the impact those steps can have on the quality of the prediction. This study focuses only on individualised trackers, but as many of the components are shared, the study's conclusions can be interchangeable for generalised trackers.

TABLE I

EXISTING EYE TRACKING APPROACHES AND THEIR ACCURACY

Method	Approach	Accuracy
iTracker [21]	Generalised	1.71cm
Webgazer [19]	Individualised	104 pixels
Turkergaze [20]	Individualised	1.06°
Adler [25]	Generalised	1.3cm
Lemley [26]	Generalised	4.91 °
Akinyelu [27]	Individualised	0.13cm*
TAT [28]	Generalised	1.77cm
iCatcher [29]	Generalised (infants)	99% Discrete (left right)
EFE [30]	Generalised	1.61cm Phone / 2.48cm Tablet
SAGE [31]	Generalised	1.78cm Phone / 2.72cm Tablet
SAGE [31]	Individualised	1.37cm Phone / 2.10cm Tablet
AFF-Net [32]	Generalised	1.62cm Phone / 2.30cm Tablet
Valliappan [33]	Generalised	0.42cm*

*Not separated by training/test, by session, or subject.

To this end, gaze tracking data were collected from 15 individuals using a laptop's web camera in order to simulate a common web browsing use case. The data were processed and analysed using our experimental framework that allowed us to evaluate different combinations of the individualised gaze tracker pipeline with respect to the used landmark detector, eye patch extraction technique and prediction model. Our experimental evaluation included three different landmark detectors (Clmtracker [22], Facemesh [23], Dlib 68 point [24]), three eye patch extraction techniques (basic square, rectangular, angled), and three final regression models (ridge regression, random forest regression, feed-forward neural network). Experimental results showed that a pipeline using a 13-point calibration pattern, the Facemesh landmark detector, the angled eye patches and a feed-forward neural network for regression achieved the best performance, reaching a gaze tracking accuracy of 2.26cm as opposed to the current Webgazer's implementation at 3.42cm with the same data, demonstrating its capability for real-time gaze tracking under the examined use case.

II. BACKGROUND

A. 2D gaze trackers

The development of the open source Webgazer [19] has opened the reach and capability of 2D gaze tracking to many. The fact that it is coded in JavaScript enables it to run in the background as people view web pages. The current iteration of Webgazer [34] no longer corresponds to the method described in [19], although the source code suggests that it still follows a similar pipeline design. A high-level description of their proposed gaze tracking pipeline is as follows: (a) detect a face in an image (video frame), (b) apply a landmark detector to that image, (c) find eye patches using the landmarks' coordinates, and (d) apply a ridge regression model to predict the relationship between an eye patch and the corresponding screen location. Although there are many

other aspects, such as gaze filtering, the key parts of [19] are the landmark detector and the eye patch locator and the impact of how different versions of that landmark detector are used in practice. Webgazer has been used extensively for many applications [35], from training to cognitive research.

Although Webgazer certainly popularised this gaze tracking pipeline, it was not the first or only work to utilise such a pipeline. One earlier work [36] used infrared light source and a webcam for a low-cost alternative to infrared gaze tracking and demonstrated the task-based effectiveness compared to infrared. However, one of the first true webcam-based gaze trackers that utilised only the webcam was TurkerGaze [20]. The researchers behind TurkerGaze were the first to introduce the aforementioned pipeline. The authors of TurkerGaze additionally reference jitter in the landmark detector, ClmTracker, and use Kalman filters to compensate.

One of the main foci of many research works is building eye models capable of generalising different participants' eye images. A big step forward in this field is the release of the GazeCapture dataset and their corresponding iTracker model [21]. This model boasts somewhere in the region of 1-2 cm accuracy on the data within the dataset. The dataset was released with coordinates for the eye patches that have been extracted. Lots of research has been built on top of this, some using the provided eye patches, while others using other extraction techniques. However, little has been done to grasp how good these eye extractions have been with respect to the end predictions. Many works use the GazeCapture dataset in combination with many well-established or custom architectures [25], [28], [32], [37]. He et al. [31] used a hybrid of a pre-trained model with a personalised calibration for each participant, therefore fitting into both the generalised and individualised categories, while performing this on their own custom model and on iTracker.

Some works [33] use modern techniques such as fine tuning models trained on the GazeCapture dataset with new data and achieve good results. However, performance is evaluated on the same data used for calibration (training), making comparisons to other works difficult. Another work that achieved extremely high accuracy, [27], demonstrates the need for consistent testing approaches within the eye tracking space. The authors used a 39-point landmark detector as part of their gaze tracking pipeline and a comparable architecture to iTracker, wherein face images are part of the inputs.

There are many ways of separating test and training sets. Some split all the data randomly meaning that a test frame could be sequentially between two training frames, others separate it by participant and others separate by phases. Additionally, the comparison of different approaches should follow a different protocol based on the examined problem. There are two main problems that should be addressed:

- **Individualised tracking:** Achieve the best eye tracking for an individual. Training and testing data for an individual comes from that individual.
- **Generalised tracking:** Achieve the best eye tracking for anyone with no calibration. Training data comes from different people to that which it is tested on.

To this end, methods targeting generalised tracking are not directly comparable to methods targeting individualised tracking, and thus a distinction must be made when evaluating the performance of gaze tracking methods. In the above categories, the approaches explored in the experimentation section of this paper are associated with individualised tracking and not generalised which most of the compared papers are. Another approach is a discrete tracker covered in [29]. This performs tasks such as left and right as opposed to coordinate gaze estimation. These approaches aren't considered in this work.

As shown in Table I, there is a variety of metrics and approaches that are used within the field of 2D gaze tracking. This high variety in ways of reviewing the success of a method makes it challenging to evaluate what constitutes state-of-the-art performance and what impedes the evaluated approaches from achieving better performance, as there is no centralised understanding of what the problems are with 2D gaze tracking pipelines. On screen error measured in *cm* seems like the best metric for understanding 2D gaze tracking performance across different hardware because angular error measured in degrees is less understandable in the practical use case of the tracker but more suited to 3D. It is fairly apparent that the same model applied on different devices achieves different results and there seems to be phenomena that the smaller the device's screen the more accurate the results. This is likely an artefact of the training point in relation to the screen size and the way that the models work, amongst other things. Blakey et al. [38] further demonstrated this by evaluating the same self-calibrated tracker on mobiles and desktop computers, noting better results on mobile.

Many review papers consider the many aspects that are reviewed in this work. One thorough review paper looks at different calibration techniques, errors caused by head movement and illumination levels [39]. These were concluded to limit the progression of eye tracking techniques. The inconsistencies within the eye tracking space are reviewed in detail in [40]. Cheng et al. [41] review deep learning approaches in eye tracking and attempt to establish a benchmark by comparing different approaches with the same dataset. The differences established between individualised vs. generalised and mobile vs. desktop suggest that not all approaches will fit into this benchmark. The approaches reviewed in our work are individualised and less comparable. Additionally the same approaches examined on mobile devices and desktops yield different results, as shown in [38].

Ehinger et al. [42] propose a new benchmark for evaluating gaze tracking performance across various settings. While these tests provide a comprehensive and thorough assessment of gaze trackers, they require nearly an hour per participant. This time-intensive process limits the feasibility of benchmarking new approaches, as existing datasets do not include these tests, and collecting new datasets would necessitate acquiring nearly an hour of data per participant.

B. Landmark Detectors

With the rise of facial filters on applications such as Snapchat, Instagram and TikTok, facial landmark detectors have seen a sharp increase in use cases. The landmark detector is one of the key parts of the gaze tracking pipeline

Fig. 1. Examples of different landmark detectors

and as a result, this increase in demand has provided gaze trackers with the choice of many different landmark detection approaches. The release of the ClmTrackr's [22] (Figure 1) JavaScript implementation of a landmark detector based on Saragih et al.'s [43] work on constrained local models fitted by regularised landmark mean-shift was crucial for allowing real-time 2D gaze trackers to be built. The task of generic face fitting has existed for a long time, with approaches for active shape models [44] described in the 90s along with the active appearance model [45], and then shape optimised search [46]. Other approaches are based on regression trees, such as the one underlying the Dlib 68 point landmark detector (Figure 1), which is based on [24]. This approach follows a cascade of regressors that update the facial landmarks' locations on an image, and was trained and evaluated on the HELEN dataset [47], which contains over 2,000 hand-annotated images.

One work that set to address the limitations of previous approaches is Facemesh (Figure 1) [23], a neural network that aims to estimate 468 landmark points in a mesh, as well as in 3D, which is not supported by previous approaches. The means of acquiring the training data for Facemesh involved the creation of synthetic 3D renderings of faces rather than using manual annotation by humans. The depth factor of this model makes it suitable for both AR applications and gaze tracking.

Prior research in facial landmark detection has demonstrated that off-the-shelf approaches achieve sufficiently good performance [23]. The author notes that there seems to be increasing interest in providing depth (a third dimension) in addition to simply tagging the location of landmark points on the screen.

In this work, we compare different landmark detectors with regards to their use case and application within the context of gaze tracking.

C. Eye patch extraction techniques

A large variety of eye patch extraction techniques are used in webcam-based gaze trackers. The basic approach used by Webgazer [19] is to extract a square area around the eye where the height of the eye patch is the same as the width. Turkergaze [20] uses a rectangle where the height of the image is restricted by the location of the upper and lower eye lid, leading to the content of the eye-patch image changing when the user looks down or blinks. iTracker [21] along with other techniques that are built using the "Eye tracking for everyone" dataset utilise a square eye patch with padding around the eye. However, the details of this extraction are not clear. Ansari et al. [37] use a padded oval shape eye patch. This is done by first extracting an eye patch rectangle and then using a mask to remove the oval.

D. Calibration patterns

Different calibration patterns could impact the quality of gaze tracking. Furthermore, it should be noted that there is likely to be an optimal pattern to show people for self-calibrated trackers. Webgazer uses a 9-point calibration pattern along with the corresponding dataset [19], [48]. Each data point needs to be clicked at each location during calibration. TurkerGaze [20] uses a 13-point calibration system. Harezlak et al. [49] conducted a comprehensive study of 29 different calibration points for approaches based on specialised gaze tracking hardware and concluded that a 13 or 11 point calibration pattern is optimal which was used in the experiment in this paper, as shown in Figure 2.

E. Public gaze tracking datasets

There are numerous publicly available gaze tracking datasets: One such dataset is the MPIIGaze dataset [50] (15 users), which is specifically designed for the purpose of "in-the-wild" gaze tracking. This dataset is aimed at predicting eye gaze vectors rather than screen locations meaning the errors estimating the 3D position of the individual will impact the prediction. It practice it cannot be used to thoroughly benchmark gaze predictors because it only presents cropped eye images. Landmark detection and eye extraction is a significant part of gaze tracking, thus removing that part limits any potential study. In addition, models trained on this dataset rely on a specific landmark detector and its constraints. Another is the Columbia dataset [51], which contains data from 56 users in strict lab conditions where head movement is controlled by using a head clamp. The EyeDIAP dataset [32] contains data from 16 users and is also commonly used for 3D gaze tracking. The EOTT dataset [48] focuses on participants in a lab setting. This dataset captures data in a video form and so frames are not strictly tied to ground truth data because only clicks are captured to indicate a person is looking at an item. This could limit how much ground truth data is tied to a frame, as outside of the clicks, the user would have to assume that the participant was looking and there is no guarantee without going into more analysis. The "Eye tracking for everyone" dataset [21] is regularly used for building generalised gaze tracking approaches. This is the largest dataset with data from over 1,500 users. It is specifically tied to mobile phone gaze tracking and most suitable for generalised gaze tracking studies. A different style of gaze tracking dataset is U2Eyes [52], which generates synthetic eye images using a gaming engine.

F. Applications and Use Cases

One important factor when cosidering gaze trackers is ease of use. Webgazer is widely accessible because of its JavaScript implementation and its validation in many applications, such as user research, advertisement and neuroscience. The single

use, disposable nature of the models means that anyone can train and build the gaze tracker during an experiment and get useful results. Models created from the "training" phase data are thrown away at the end of the session. This makes them uniquely suited to maintaining privacy and operating in a variety of conditions, whereas the completeness of the package with a landmark detector, eye patch extractor and gaze model make it easy to work with. Webgazer's original implementation boasted a 4 cm accuracy, although it has since been improved with a new landmark detector, making it to this day the preferred candidate for many applied gaze tracking studies.

One factor that limits many of the generalised gaze trackers from getting widespread usage is the requirement for large scale datasets. Datasets such as MPII Gaze [50] and GazeCapture [21] can limit and restrict the widespread adoption of technologies built on them because they provide the eye patch images. Although GazeCapture also provides the whole images, unlike MPII Gaze, many researchers building gaze tracking technology from this dataset rely on the eye patch images provided. This reliance on a specific landmark detector, which in GazeCapture's case is the internal face detection model on iPhones, means that eye patches extracted through other landmark detectors become less compatible with the models trained. Additionally, as new iterations of the iPhone's face detection model have come out, the models trained may become out of date and not compatible with current technology.

G. Research Obstacles

Some of the main obstacles in gaze tracking research are:

- **Lack of consistent metrics:** Many gaze tracking researchers are limited by a lack consistency across the metrics with which the accuracy is reported. The metrics should be tied to applications such as screen-based or 3D and even within these applications it is sometimes unclear which is best.
- **Dataset homogeneity:** Datasets may heavily tie approaches to hardware. For example, if the dataset uses hardware-based face detection techniques, and other researchers build on this, then face detection techniques are part of the final eye tracking approach, which limits usability.
- **Comparison between approaches:** Approaches such as mobile vs. desktop and generalised vs. individualised are hard to compare against. These comparisons and different protocols can make it challenging to establish which approaches are best for certain situations.
- **Focus on Eye models:** Many approaches do not declare metrics on the whole pipeline of what it takes to get from an image to a prediction. Components such as landmark detectors may not be reported, which makes reproducibility quite challenging.
- **Inconsistent evaluations:** Some evaluations can be done with leave-one-out methods for participants and others use all participants with a test/train split, whilst others use random shuffling. These evaluation strategies are not compatible with each other and leave for ambiguous interpretation as to whether the methods are working as desired.

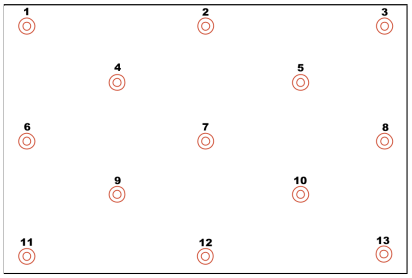


Fig. 2. Calibration pattern. Numbers were not shown during the experiment.

III. Methodology

To evaluate the factors that affect the performance of real-time webcam-based gaze trackers, we implemented an experimental framework that allowed us to evaluate different combinations of the individualised gaze tracker pipeline with respect to the used landmark detector, eye patch extraction technique, and prediction model. Various combinations of the examined techniques were evaluated on real data captured from 15 individuals in order to propose an individualised gaze tracker that outperforms the state of the art.

A. Data collection

In this work, we opted to capture new data for our study for two main reasons: (i) Collecting data that would allow for eye movement, such as saccades and fixations for other experiments. (ii) Controlling some of the "in the wild" variations, such as lighting and hardware, so that the results can be controlled. A limitation of many publicly available datasets is that the data was not captured to have a 1:1 frame to gaze point. A video would typically have a single reference point and it is difficult to assert that the user was looking at the target the entire time. In our approach, the face image and eye image were captured in a strict loop, thus each image has a direct link to the ground truth target location. Furthermore, in order to enable an exploration of various gaze tracking approaches, data needs to be collected in a way that allows variations in the gaze tracking pipeline to be tested in a consistent manner. To this end, we used the same device to collect data from users watching stimuli in a similar lighting environment. Collecting data in this fashion allows modules within the gaze tracking pipeline to easily be switched out with alternative modules as shown in Figure 3, such as changing the landmark detector, or the methodology for extracting the eye patches, or the final regression model.

The data were therefore collected from different individuals in a similar setting, with participants being presented with stimuli that allowed for ground truth data to be collected. To this end, 15 participants (8 male, 7 female, between the age of 20 and 60, all professionals that utilise technology in some capacity every day) with normal or corrected to normal vision were asked to participate. This number of participants

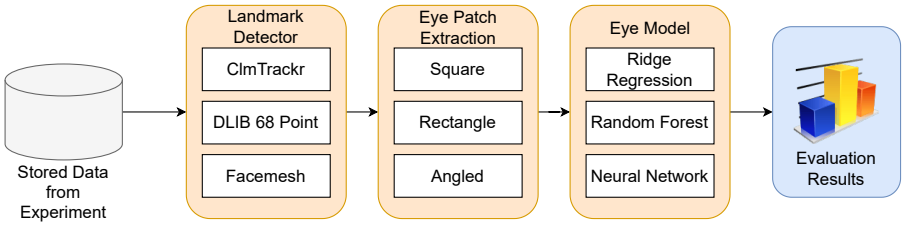


Fig. 3. Gaze tracking pipeline evaluation framework.

is consistent with widely used gaze tracking datasets, such as MPII Gaze [50] (15 participants) and EyeDIAP [32] (16 participants). All participants were asked to sit approximately 50 cm in front of a 15" 2012 MacBook Pro laptop with a built-in webcam, 16GB of RAM and an i7 processor. The built-in camera was recording at 30 frames per second for the whole duration of each experiment and each video frame was stored as an individual image in lossless PNG format to avoid video compression and ensure that syncing issues did not occur between ground truth target locations and eye images. The MacBook Pro (15 inch, 2012) model was selected for our experiment as it is a good example of typical consumer device in terms of screen size and web camera quality. Furthermore, the number of participants being 15 was in line with other research in the field, such as in the gaze tracking studies of [32], [53], [54].

In order to keep the conditions constant, the room was set up with identical lighting. The time of the day was kept the same and the laptop was always on the same position on the same table. The individual was facing the window to ensure the best lighting conditions. This setup remained constant for all individuals. The participants were also instructed to remain a normal user distance from the laptop and although variations would be present in distance to the camera and the screen, these were kept to a minimum through these instructions.

Data acquisition was conducted in two phases, leading to capturing approximately 1000 frames per participant. In Phase 1, participants were asked to look at the laptop screen as a calibration pattern was shown. More specifically, the participants were instructed to follow a target around the screen as it stopped in the locations shown in Figure 2. The target moved at a constant velocity to each calibration point and waited for 28 frames at each location. During each time step in the recording (30 fps), a frame was captured and was strictly tied to the location of the target on the screen. No additional processing or data cleaning was applied to the captured frames.

In Phase 2, to allow for the evaluation of gaze tracking techniques on a simulated "real-world" gaze tracking scenario, we repeated the aforementioned data acquisition process and collected gaze tracking data after the initial calibration (Phase 1). Gaze tracking models should not be tested in a typical random train/test split of the dataset. The test set must be taken from some time after the training and validation data were acquired in order to simulate a test of a real world tracker, whereas if a random train/test split is used, the models will be trained on video frames between frames used for calibration which will artificially boost the trackers performance. By capturing the test data at a later time from the initial calibration, we simulate what happens when the tracker is used in practice, when the head has moved slightly and the lighting is somewhat different.

The evaluation took place on what many researchers consider a "fixation test". Participants were not instructed to view webpages but rather a set of targets, i.e. the calibration points in this study. This is so that a ground truth location can be compared against the gaze point of the user. Fixation tests are common in research for assessing the success of a gaze tracker [42], [48]. It must also be noted that due to the participants being instructed to look at the gaze location for each calibration point, all the ground truth data is assumed to not include saccades.

This study was conducted according to the guidelines of the University of the West of Scotland University Ethics Committee. All participants were informed about the scope of the study and about how the captured data would be used, as well as regarding their option to withdraw from the study at any time and request deletion of their data. All participants provided consent for participating in the study and using the data for research purposes. Data are stored and handled according to the University's policy.

B. Experimental Framework & Evaluation

The collected data was then used to evaluate the performance of an individualised real-time gaze tracker. As shown in Figure 3, our framework allowed us to evaluate the performance of different landmark detectors and eye patch extraction methods. Each step in the pipeline was evaluated independently by keeping the rest of the steps the same as for the original Webgazer [19] pipeline. Consequently, at each experiment, gaze tracking performance was only affected by the different method used for the examined step. Finally, the combination of the best performing methods for each step in the pipeline were also evaluated using random forest regression and neural network-based regression in place of the final ridge regression step of the original pipeline. Once the best components were established through the experimental framework, the final best performing configuration was evaluated against state-of-the-art gaze trackers. It must be noted that our study focuses on individualised real-time webcam-based gaze tracking, thus all methods examined in this work were able to be trained and used in real-time, and computationally heavy offline methods were not considered.

The framework's individual components were assessed by the gaze tracker's accuracy. As the problem space is binocular, the assessment is based on a model using both eyes and a single convergence point on the screen. Based on an in-depth review of eye/gaze tracking accuracy metrics [38], we opted to assess gaze tracking performance in terms of screen-based accuracy, measured in centimetres (cm). The per eye accuracy is defined as $GazeAcc = \sqrt{E_x^2 + E_y^2}$, where lower $GazeAcc$ values indicate better gaze tracking performance and $E_x = |x - \hat{x}|$, $E_y = |y - \hat{y}|$, (x, y) are the actual coordinates of the convergence point on the screen, and $(\hat{x}, \hat{y})$ are the predicted coordinates of the convergence point. This error is initially measured in pixels. However, as pixels vary by screen, cm should be used in order for the performance results to be compared with other trackers. This can be achieved by simply multiplying the error value by the ratio of pixels on screen to the width of the screen in cm. In order to compute the binocular on-screen accuracy, the average gaze for both eyes was calculated and the difference to the ground truth was calculated as the accuracy. Lower screen error and therefore lower GazeAcc will result in a better gaze tracker performance unlike most accuracy measures in other fields.

During the main study, monocular accuracy is considered for evaluating the individual components of the gaze tracking pipeline, whereas binocular accuracy was considered for the comparison of the best performing pipeline against the state of the art. This is done because it is important for the accuracy per eye to be optimised before the eyes together are considered in more practical use cases such as comparing the performance of the overall pipeline against other methods. Gaze trackers will generally perform better when averaging the error for both eyes meaning that the binocular error is likely better. To this end, individual components of the gaze tracking pipeline, such as the landmark detector and eye patch extraction, need to be optimised for an individual eye rather than allowing the binocular error to inflate their scores.

C. Landmark detection

Three landmark detection methods were evaluated using our experimental framework: (a) Clmtracker [22], which has been used by both the original Webgazer [19] and Turkergaze [20]. (b) Facemesh [23], a neural network-based landmark detector that is used in the most recent version of Webgazer. (c) the Dlib 68 point landmark detector, which is based on [24]. The Facemesh landmark detector was used as the baseline for our pipeline. Blinks were not removed as part of the process as some of the landmark detectors do not have built-in blink detectors meaning that an additional step would be needed to the pipeline.

D. Eye patch extraction

The following three eye patch extraction techniques were evaluated using our experimental framework. It must be noted that eye patches were computed assuming the eye image coordinates shown in Figure 4. Additionally the different eye patch shapes can be seen in Figure 5.

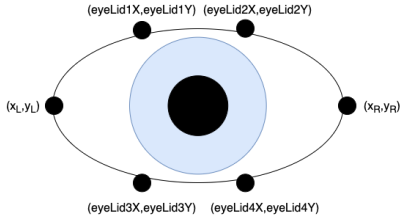


Fig. 4. Eye coordinates used for eye patch extraction.

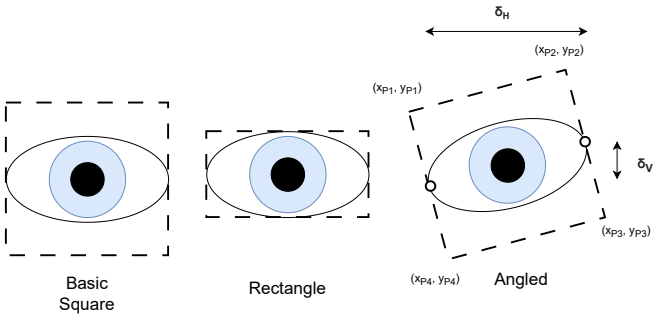


Fig. 5. Eye patch shapes.

1) Basic square: A square eye patch is the simplest form for an eye patch and is used in works such as Webgazer [19]. It is extracted as follows:

$$width = |x_L - x_R| \tag{1}$$

$$height = width \tag{2}$$

$$x_{bb} = x_L \tag{3}$$

$$y_{bb} = |y_L + (0.5 \cdot width)| \tag{4}$$

x_L, x_R, y_L, and y_R denote the x and y coordinates of the top (L)eft and (R)ight corner of the eye, whereas the coordinates (x_{bb}, y_{bb}) define the bounding box's top left corner. The square eye patch was used as the baseline for our pipeline.

2) Rectangular: This approach aims to remove as much noise as possible and was used by TurkerGaze [20]. It is extracted as follows:

$$width = |x_L - x_R| \tag{5}$$

$$height = |max(eyeLid1X, eyeLid2X) - min(eyeLid3X, eyeLid4X)| \tag{6}$$

$$x_{bb} = |x_L + (0.5 \cdot width)| \tag{7}$$

$$y_{bb} = max(eyeLid1X, eyeLid2X) \tag{8}$$

(x_{bb}, y_{bb}) denotes the bounding box's top left corner.

3) Angled: We propose a new approach that keeps the angle of the head consistent so that when the head moves it is possible to keep the same consistent eye patches. The eye patch has a square shape angled by the line connecting one eye corner to the other eye corner. In order to extract this angled square, the four corner points of the square eye patch are computed as follows:

$$\delta_H = x_L - x_R \tag{9}$$

$$\delta_V = y_L - y_R \tag{10}$$

$$(x_{P1}, y_{P1}) = (x_L + \frac{1}{2}\delta_V, y_L - \frac{1}{2}\delta_H) \tag{11}$$

$$(x_{P2}, y_{P2}) = (x_L - \frac{1}{2}\delta_V, y_L + \frac{1}{2}\delta_H) \tag{12}$$

$$(x_{P3}, y_{P3}) = (x_R - \frac{1}{2}\delta_V, y_R + \frac{1}{2}\delta_H) \tag{13}$$

$$(x_{P4}, y_{P4}) = (x_R + \frac{1}{2}\delta_V, y_R - \frac{1}{2}\delta_H) \tag{14}$$

where the coordinates (x_{P1}, y_{P1}), (x_{P2}, y_{P2}), (x_{P3}, y_{P3}), and (x_{P4}, y_{P4}) denote the bounding box's top left, top right, bottom right, and bottom left corners, respectively.

E. Regression models

The regression model used for real-time gaze tracking is of critical importance for the performance of the gaze tracking pipeline. Three regression methods were evaluated after establishing the best performing landmark detector and eye path extraction method: (a) Ridge regression, similar to the one used in Webgazer [19], (b) Random Forest regression, and (c) a simple Feed-forward Neural Network. The random forest architecture was set up with 100 trees and a depth of 2. The neural network was set up with a hidden layer of 100 neurons. The ReLU activation function was used for the hidden layer, whereas a linear activation function was used for the output layer. Training was conducted using the Adam optimiser. The Ridge regression method was used as the baseline for our pipeline. These hyper parameters were chosen based on preliminary experimentation.

Given that the aim of this work is to evaluate real-time gaze trackers that can be calibrated and run in web browsers in the background, we opted to evaluate a simple neural network and did not consider some recent works, e.g. [14], [21], [41], [55], that rely on convolutional neural networks that are more complex to train and run and require more data to train. The reason for keeping the examined models reserved to these simple regression techniques is our focus on individualised gaze tracking as opposed to generalised. Many more advanced techniques and complex network architectures are used in generalised approaches and see great success, but the specific use case that is being reviewed in this study requires that the model be trained on a single participant's data and the training time is short enough that the model can be used almost immediately. Even simple convolutional neural networks (CNN) require lengthier training time when performed on commercial and easy accessibly hardware. Furthermore, most networks that are more complex will require more data than a single individual looking at 13 points will provide and so only a small selection of models fit the criteria to be tested.

In must be noted that in order to keep the input size of the regression models consistent across all samples in the dataset, square and angled eye patches were resized to 20×20 pixels, whereas rectangular eye patches were resized to 20×10 pixels.

TABLE II

GAZE TRACKING PERFORMANCE RESULTS FOR THE EXAMINED LANDMARK DETECTORS AND EYE-PATCH EXTRACTION METHODS.

	Method	Acc. X (↓)	Acc. Y (↓)	Accuracy (↓)
Landmark	Dlib 68 point	3.51cm	2.70cm	4.86cm ± 3.92cm
	Facemesh	**3.06cm**	**2.15cm**	**4.11cm ± 2.85cm**
	Clmtrackr	3.39cm	2.33cm	4.56cm ± 3.37cm
Eye patch	Square	3.25cm	2.18cm	4.30cm ± 2.97cm
	Rectangular	3.19cm	3.05cm	4.91cm ± 3.56cm
	Angled	**2.75cm**	**2.17cm**	**3.89cm ± 2.49cm**

IV. RESULTS

The aforementioned experiments were conducted on the data acquired from the 15 participants of this study. To simulate a realistic individualised gaze tracking scenario, we opted not to do a random split of our data into a training and a test set, as using a random train/test split would lead to the models being trained on video frames between frames used for calibration which will artificially boost the trackers performance. Instead, all the data from Phase 1 of the data acquisition were used to train each gaze tracking pipeline, whereas all the data from Phase 2 were used to evaluate them, simulating a more realistic scenario of "real-world" gaze tracking. The performance of each examined pipeline was evaluated in terms of the mean overall on-screen accuracy, the mean on-screen accuracy in the horizontal direction (X), and the mean on-screen accuracy in the vertical direction (Y). Detailed results are reported in Table II and III.

A. Landmark Detectors

As shown in Table II, when the examined landmark detectors were evaluated under the suggested "realistic" protocol, then Facemesh [23] performed the best, reaching an overall on screen accuracy of 4.11 ± 2.85cm, compared to 4.56 ± 3.37cm and 4.86 ± 3.92cm for the Clmtrackr and the Dlib 68 point, respectively.Additionally the level of consistency with the results seems to be far more favourable for Facemesh, with the standard deviation being 2.85cm compared to 3.37cm for Clmtrackr and 3.92cm for Dlib 68 point. Even with the accuracy being superior for Facemesh, with an almost 10% improvement in overall accuracy over Dlib 68 points and Clmtrackr, the greatest improvement is in the standard deviation, where a 27% and a 15% reduction in the standard deviation, respectively, indicates a far more consistent and stable gaze tracker.

The causes of these fall heavily on the way that eye patches are generated. It is likely that Facemesh's eye patches are more consistent, meaning that when the head falls in a slightly different position, the eye patches that are generated are very close to those that the model was trained with. There are two factors that are required for stability for eye patches:

- **Consecutive (short-term) stability**: The change from one patch to the immediate next frame's patch should be small.

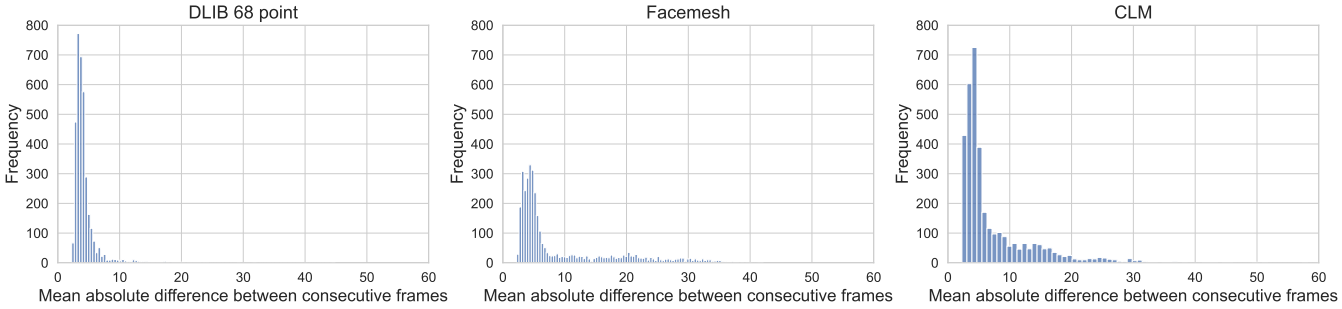


Fig. 6. Consecutive (short-term) stability of the examined landmark detectors.

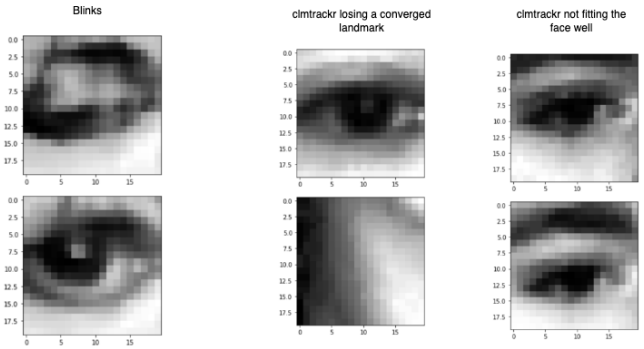


Fig. 7. Blinks and other faults affecting consecutive (short-term) stability.

- **Long-term stability**: The eye patches that are generated with a large time apart should be similar.

Consecutive (short-term) stability can be evaluated by computing the difference between consecutive eye patch frames. Although not purely a method of checking stability, we argue that computing the difference by subtracting one frame from the next is a good indicator. The distribution of the differences can then be plotted (Figure 6), in order to examine the consecutive stability of the landmark detectors. From Figure 6, it is evident that the Dlib 68 point landmark detector exhibits the best consecutive stability among the three examined methods. More severe cases that may affect consecutive (short-term) stability for Clmtrackr are shown in Figure 7. All of the large differences found in Dlib 68 point's and the Facemesh's landmarks were blinks, which cannot be helped. Although Facemesh has the largest mean difference between frames, visibly the consecutive frames seemed to capture the eye well and did not seem to move. However, small amounts of jitter is probably what is being detected. Clmtrackr had moments where the converged facemask was completely lost, or the mask did not fit well to the face in the picture.

Long-term stability was evaluated in a similar manner by computing the mean difference between eye patch frames from the calibration (Phase 1) and the test data (Phase 2), when participants looked at the same point on the screen some minutes apart. The distribution of these differences is shown in Figure 8. Facemesh produces eye patches that have better long term stability, thus explaining the superiority of Facemesh

TABLE III

GAZE TRACKING PERFORMANCE RESULTS FOR THE EXAMINED REGRESSION MODELS

	Method	Acc. X (↓)	Acc. Y (↓)	Accuracy (↓)	Fit Time
Realistic	Ridge Regression	3.25cm	2.18cm	4.30cm ± 2.97cm	**0.66s**
	Random Forests	2.85cm	3.09cm	4.63cm ± 2.74cm	2.13s
	Neural Network	**1.74cm**	**1.63cm**	**2.66cm ± 1.84cm**	2.12s
Random	Ridge Regression	2.09cm	1.53cm	2.87cm ± 2.08cm	**0.65s**
	Random Forests	2.39cm	1.88cm	3.31cm ± 2.25cm	1.77s
	Neural Network	**1.05cm**	**0.94cm**	**1.58cm ± 1.32cm**	1.50s

in terms of overall on screen accuracy under the examined realistic protocol, where gaze trackers were trained with data from Phase 1 and tested on data from Phase 2 of the data acquisition.

B. Eye patches

Results under the examined realistic protocol for eye-patch extraction are shown in Table II. In this case, rectangular eye patches perform the worst, with the best performance in terms of all metrics achieved by using the proposed angled eye patches, reaching an overall on screen accuracy of 3.89 ± 2.49cm, an accuracy in the horizontal (X) direction of 2.75cm, and an accuracy in the vertical (Y) direction of 2.17cm. This is most likely because angled eye patches are large enough to be robust, ensuring that nothing important is cut out. When considering the results from Table II, it becomes clear that angled eye patches are superior, outperforming rectangular and square eye patches for all metrics. They also exhibit less standard deviation, indicating better stability. We believe that further investigation into angled eye patches is required, as we hypothesise that they may help more with slight head movements and so investigations into the impact of head movement may further favour the angled eye patch approaches.

C. Models

The baseline landmark detector and eye patch extraction method (Facemesh, Basic Square) were then used as the basis for evaluating the three regression methods for the final step of the pipeline. It is evident from Table III that, under the realistic protocol, the neural network-based regression achieved the

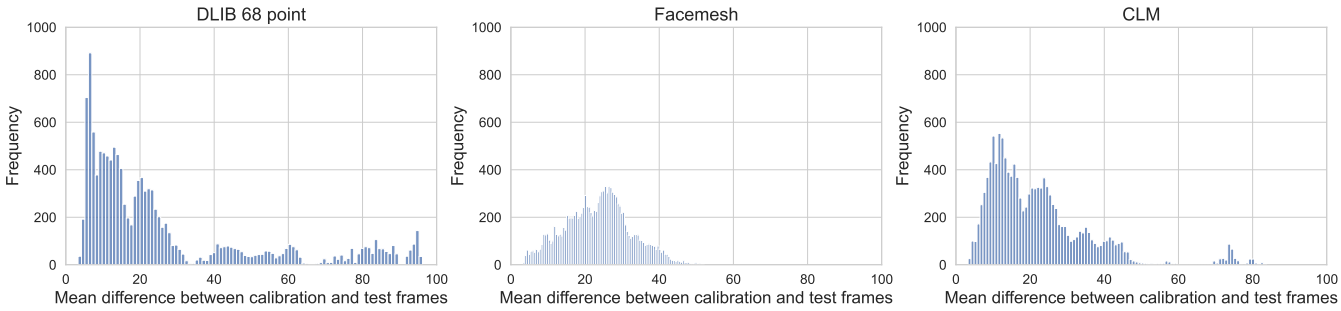


Fig. 8. Long-term stability of the examined landmark detectors.

best overall on-screen accuracy (2.66 ± 1.84cm), as well as the best accuracy in the horizontal (X) direction (1.74cm) and the best accuracy in the vertical (Y) direction (1.63cm). Ridge regression achieved the second best performance for all examined metrics with an overall accuracy of 4.30 ± 2.97cm, whereas random forests regression performed the worst, achieving an overall accuracy of 4.63 ± 2.74cm.

In addition, to further support our evaluation protocol selection, we also evaluated the three regression approaches by randomly splitting the data from Phase 1 of the data acquisition into 80% for training the models and 20% for testing. Results for the random train/test split are also provided in Table III. It is evident from the decrease in accuracy from the random split protocol to the realistic protocol that there is a degradation of the accuracy the more time has passed since the initial training data was captured. It is also noteworthy that the better results demonstrated by the random split protocol are artificial and do not hold when simulating "real world" conditions which should deter researchers from evaluating their trackers using this data splitting method.

Average fit times for each gaze tracking model are also shown in Table III. Small fit times are essential for the examined use case because it demonstrates the amount of time after calibration data is received that an individual would have to wait in order to have a real time gaze tracker operating. For the realistic protocol, training took 2.12s for the neural network model and 2.13s for the random forest model. The fastest model to train was the ridge regression model at 0.66s, however it also provided the worst gaze tracking performance. Nevertheless, the 2.12s required to train the best performing neural network model are acceptable for the examined use case, where after performing the calibration, it would take only approximately 2s until the gaze tracker was operational.

D. Comparison to other individualised gaze trackers

The experimental results from Table II and III indicate that an individualised gaze tracker using the Facemesh landmark detection method, the angled eye patches, and the neural network regression model would perform the best. We evaluated our best performing gaze tracker against the original Webgazer [19] approach, as well as against Webgazer's modern iteration, in terms of binocular accuracy. All methods were trained using the same training data and evaluated on the same test data, under the realistic protocol. Results in

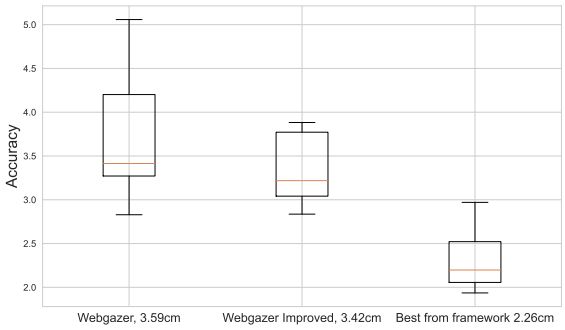


Fig. 9. Comparison with state-of-the-art individualised gaze trackers in terms of binocular accuracy.

Figure 9 show that our proposed best approach achieved the best on-screen binocular accuracy (2.26cm), with Webgazer's modern iteration achieving the second best (3.42cm), and the original Webgazer achieving the worst (3.59cm). From Figure 9, it is notable that the current iteration of Webgazer that utilises Facemesh for landmark detection not only achieves a better binocular accuracy than its original version but also achieves considerably less variance and greater consistency. The proposed individualised real-time webcam-based gaze tracker also exhibits low standard deviation, while providing a 0.90cm improvement in average binocular accuracy over the second best performing gaze tracker. In addition, it achieved an average gaze tracking execution time of 0.00897 s per frame on the MacBook Pro (15 inch, 2012) laptop used for data collection.

V. Discussion

Our experimental study of landmark detectors, eye path extraction methods, and regression models for real-time webcam-based individualised gaze tracking led to the selection of a well-performing gaze tracking approach that outperforms existing methods. Although many of these techniques have been explored in the past, comparative studies have rarely used the same data for training and evaluation across so many different factors. These comparisons were performed in isolation but there is much more that could be discussed from the perspective of having different combinations of factors.

For the landmark detection algorithms it is fairly clear that there have been significant improvements in the time since CLMtracker was developed. Both the Dlib 68 point landmark detection algorithm and the neural network-based Facemesh demonstrate sizeable improvements in maintaining the important feature, eye stability, over CLMtracker. It is worth noting that facemesh gives an estimate of distance which may be important for improving the stability of gaze predictions as the head moves. Prior works, e.g. [20], explored rectangular eye patches as an alternative to the commonly used square eye patches. However, this work showed that rectangular eye patches did not lead to an improvement in accuracy compared to square patches. Furthermore, this work introduced angled eye patches in an attempt to maintain consistency of the eye patches even when the head moved. Results showed that angled eye patches performed consistently better than square and rectangular eye patches.

The results of the experimental comparison against state-of-the-art individualised gaze trackers, such as the original and the modern iteration of Webgazer [19], indicate rather promising performance improvements. A sizeable improvement can be seen, which given promising works demonstrating the use of Webgazer in cognitive behavioural studies amongst other things, is rather compelling. A 2.26cm on-screen accuracy on a desktop computer, such as the one achieved by our proposed approach, also showcases that individualised trackers are able to compete with generalised trackers, such as iTracker [21] that boasts a 2.5cm accuracy on the closest device to a desktop. This is not even taking into account the fact that multiple studies show that bigger screens, such as tablets, have a worse accuracy than smaller screens, e.g. mobile phones, and the results from this study refer to a large laptop screen that is considerably larger than the screens found in tablets.

In recent years, there have been many advancements in more complex neural networks for gaze tracking. These models are trained on many participants and are referred to in this work as "generalised models" because of the fact that they are trained on multiple individuals. More complex models require vastly more training data and considerably longer times to train. Works such as [56], reporting an accuracy of 4.09° and [57], reporting an accuracy of 4.1° use different evaluation strategies, making results difficult to compare. Cheng et al. [57] use an evaluation set reserved from each participant to evaluate and [56] use a leave-one-out evaluation strategy, meaning the model is tested on a person it has never seen before. These differences highlight the need for clear terminology. Comparisons between gaze trackers are heavily dependent on their use cases, as stated by [19].

The lack of consistency within evaluation of generalised methodologies leaves ambiguity in terms of performance. This is the case when comparing methodologies with similar use cases. Individualised methods have different use cases and in many cases can be easier to use, as generalised ones require importing pre-trained models (which are not always public) and can require specific landmark detectors, such as the iPhone's in-built one, limiting them to specific and some times out-of-date hardware. To compare our pipeline to some of these approaches, we adopt the EyeDIAP dataset [32], which focuses on predicting vectors in a range of 3D scenarios. EyeDIAP has been used in many works to benchmark their eye trackers. This dataset is not the ideal dataset for our proposed tracker, but it does contain a 2D component. The main factor that makes it less ideal for our tracker is that the HD camera used to capture the dataset was positioned off to the side and underneath the screen, meaning that the eye images extracted are at an unusual angle and full of shadows. This can be demonstrated in Figure 10, which shows an eye image extracted for EyeDIAP and one for our dataset. In contrast, an ideal dataset for a webcam tracker has a camera positioned above the screen which is being viewed.

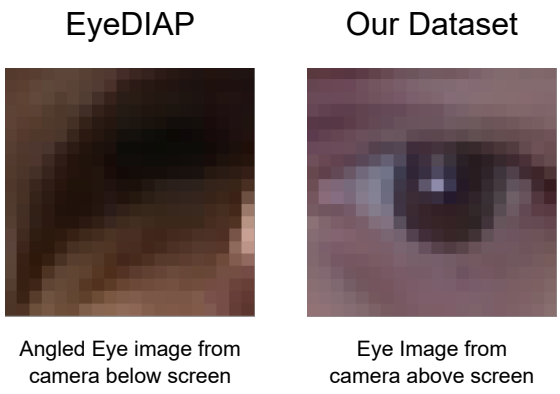


Fig. 10. Examples of eye images from our dataset and from EyeDIAP.

TABLE IV
GAZE TRACKING PERFORMANCE ON THE EYEDIAP [32] DATASET

Approach	Training	Method	Accuracy (↓)
Generalised	Pre-trained	iTracker [21]	10.13cm
		AFF-Net [32]	9.25cm
		GazeNet [58]	8.51cm
	Leave-One-Subject-Out on EyeDIAP	Gaze360 [59]	6.37cm
		FullFace [60]	7.70cm
		CA-Net [57]	6.30cm
		RT-Gene [54]	7.19cm
Individualised	On EyeDIAP	Webgazer [19]	9.24cm
		Our Approach	**6.17cm**

After being trained and evaluated on EyeDIAP, our individualised approach was compared against three pre-trained generalised trackers ([21], [32], [58]), four generalised trackers trained and evaluated on EyeDIAP using a leave-one-subject-out cross validation strategy ([54], [57], [59], [60]), and one individualised tracker trained on EyeDIAP (Webgazer [19]). Results in terms of pixel values were converted to cm for all trackers to ease comparisons. Despite the challenge of the EyeDIAP dataset not being ideal for our use case, it is evident from Table IV that our approach outperforms the compared methods. Furthermore, it is clear from Table IV that the generalised methods trained on EyeDIAP perform much better than the pre-trained ones. This is unsurprising given the fact that these methods are trained on the EyeDIAP dataset, meaning that factors such as specific lighting, distances from the screen and unusual angle are captured into the gaze

tracking model. Results also show that Webgazer performs worse than most of the generalised approaches, except for AFF-Net [32] and iTracker [21]. In addition, the previously discussed, and shown in Figure 10, difference between how EyeDIAP's eye images were captured compared to our dataset can account for the difference in accuracy of our method in the EyeDIAP dataset compared to our dataset (6.17cm vs 2.26cm, repsectively).

It must be highlighted that the practical use case of our approach is the building of disposable gaze tracking models that rely on a short training phase to achieve the best results for a specific individual. This use case can be highly beneficial and practical for researchers across various domains. As shown in Table IV, relying on pre-trained generalised models leads to reduced accuracy. This issue could be mitigated through further calibration, as demonstrated by the generalised models trained on EyeDIAP. However, this approach would negate the primary advantage of being calibration-less, ultimately requiring a comparable amount of time and effort to our approach. Other limitations of pre-trained models could also restrict their practical use in research fields such as advertising and healthcare. These limitations include the need to import pre-trained weights, which may not always be accessible, long training times if additional training is required, and dependence on the specific landmark detector used in the training set.

Established webcam-based individualised trackers like Webgazer have been evaluated for use cases in cognitive research [35], evaluated against infrared [61] trackers, and evaluated for use with children [62]. Our tracker improves upon Webgazer and demonstrates notable improvements over all compared trackers, thus it could provide many of these research areas with a tracker that could achieve a sizeable boost in accuracy and alleviate some of the limitations [61]. Although many datasets prove to be challenging to use with this style of gaze tracker, our comparison on EyeDIAP should instil confidence on our proposed approach. Additionally, the simplicity of the proposed tracker and the absence of extensive training steps make it easy for other researchers to reproduce similar experiments and verify the results.

The impacts of the gaze tracking pipeline's component analysis could have many implications for generalised gaze tracking pipelines, as many of these approaches use the same landmark detector, eye patch extraction and eye gaze model as individualised approaches. Models such as CA-Net [57], iTracker [21] and RT-gene [54] all require eye patches to be extracted and in these cases similar conclusions could potentially be drawn as to the effectiveness of Facemesh in this work. Other models such as FullFace [60] and Gaze360 [59] aim to have an end-to-end pipeline which makes them less dependant on the same components. In practice, Gaze360 requires face detection to locate subjects in the images and FullFace requires a landmark detector to perform a face patch extraction to pass into the end-to-end network. Additional components are required in almost every gaze tracking pipeline. The results from our component analysis demonstrate how components, such as landmark detection, are an important factor and how the variability of landmark detection or eye patch extraction techniques and other less conventional techniques directly impact the accuracy and reproducibility of the results. As a result, it would be strongly encouraged for gaze tracking researchers to consider all the components as part of their solution rather than just the eye-gaze model. In addition, the variability of end results from different components should be a cause for concern for using extracted eye patches in pre-existing datasets like GazeCapture [21] or MPIIGaze [50]. This can lead to dependence on eye patches that might be tied to out-of-date landmark detectors on commercial hardware, making their declared accuracy not reproducible.

Individualised gaze tracking using the examined pipeline creates simple easy-to-use gaze trackers with better accuracy and therefore makes the effort to improve the components the key to true modular improvements. Given the achieved accuracy, it is likely that improving the architecture already defined and generating compelling results is a potential way to reach generalised gaze tracking success. Although this study was on individualised trackers, it is noteworthy that generalised trackers utilise much of the same components. This work clearly demonstrates that the components that most of these approaches use could potentially see sizeable improvements by changing the techniques that they use for extracting eye patches and finding landmarks.

Our work explores fairly lightweight easily computed models. These gaze tracking techniques have their place, but due to the need to provide training data, they require calibration. Even with the improvement in long-term stability presented in this paper, these techniques are still limited to maintain the initial relationship of where the head and eye was in relation to the screen. Whilst these improvements are a step forward, more notable improvements should tackle the issues that limit these approaches getting wide spread use. These limitations are:

- **Need to be calibrated:** Every time a person wants to use these techniques, they have to provide lengthy calibration over the screen, where the individual must look at different spots and it is vital that the participant adheres to look at the stimuli provided.
- **Long-term gaze stability:** Over time, due to lighting variations and head movement, the models almost always lose accuracy, as shown in this study (random split vs. realistic protocol). This happens because the relationship learnt by the model is how the eye patch relates to an on screen location and as soon as the head moves, this is lost.
- **Generalised models:** The models that have been presented work exclusively for the individual they are trained for. In order to get wide spread use, a model needs to work with anyone and not require calibration.

VI. Conclusion

This work explored the problem of webcam-based real-time individualised gaze tracking for use cases like gaze tracking in the background while users perform everyday tasks like web browsing. To this end, gaze tracking data were collected from 15 individuals and were used in an experimental framework

to evaluate the contribution and effects of various steps in the gaze tracking pipeline on the overall performance. Our evaluation on three different landmark detectors, three eye patch extraction techniques, and three final regression models, showed that a pipeline using a 13-point calibration pattern, the Facemesh landmark detector, the angled eye patches and a feed-forward neural network for regression achieved the best performance, reaching a gaze tracking accuracy of 2.26cm and outperforming the compared methods under the examined use case. In addition, the low fit time required to train the gaze tracking model, as well as the small number of training samples required, further demonstrate the suitability of the proposed approach for individualised webcam-based real-time gaze-tracking under "real-world" use cases.

Nevertheless, it is the belief of the authors that this technology is still limited by the factors discussed in this work, such as need for calibration, long term gaze stability, and requirement for generalised models. Future work will aim to address some of these issues, as this should enable more widespread adoption of webcam-based gaze tracking, thus unlocking the potential of this technology for industries such as user experience, accessibility of mobile devices and advertising to facilitate a greater understanding of where people (users) are looking. In addition to improvements to the models, there are a number of practical issues that will be evaluated in future work: (a) Real world fixations test: The common practice for evaluating gaze trackers is through fixation tests, however it would be beneficial to evaluate gaze trackers by having fixation pop ups while someone is doing a task such as browsing the web. This should give a more practical sense of how the gaze tracker would perform, but would require a new dataset. (b) Mobile phones: The entire pipeline of this work runs on javascript code that can work on any browser, from desktop to mobile, and so evaluation on a mobile dataset would be a useful exploration for a world that is majority mobile browsing. (c) Clicking-Looking relationship: The known relationship that users look where they click could be exploited by adding such gaze data to the training data. The examined models have low training times, thus every time a user clicks when browsing, the data could be added and the models retrained. This kind of model may have practical applications and could potentially overcome some of the limitations of head movement by constantly retraining and boosting the accuracy. (d) Varying environmental conditions: In this work environmental conditions, such as lighting, distance from screen, webcam position, screen size, etc., were kept static in order to reduce variability in our experimental evaluation. An evaluation with varying environmental conditions would be useful for assessing their impact on gaze tracking performance.

REFERENCES

[1] G. Zammarchi and C. Conversano, "Application of eye tracking technology in medicine: A bibliometric analysis," *Vision*, vol. 5, p. 56, 11 2021.

[2] N. Modi and J. Singh, "Understanding online consumer behavior at e-commerce portals using eye-gaze tracking," *International Journal of Human–Computer Interaction*, vol. 39, no. 4, pp. 721–742, 2023.

[3] Z. Pan, J. Zhu, J. Zhang, W. Li, G. Jia, W. Luo, J. Peng, and M. Li, "An eye-gaze-controlled needle deployment robot: Design, modeling, and experimental evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024.

[4] X. Zhang and S.-M. Yuan, "An eye tracking analysis for video advertising: Relationship between advertisement elements and effectiveness," *IEEE Access*, vol. 6, pp. 10 699–10 707, 02 2018.

[5] A. Konovalov and I. Krajbich, "Gaze data reveal distinct choice processes underlying model-based and model-free reinforcement learning," *Nature Communications*, vol. 7, no. 1, p. 12438, Aug 2016.

[6] E. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 1124 – 1133, 07 2006.

[7] C.-S. Hwang, H.-H. Weng, L.-F. Wang, C.-H. Tsai, and H.-T. Chang, "An eye-tracking assistive device improves the quality of life for als patients and reduces the caregivers' burden," *Journal of Motor Behavior*, vol. 46, no. 4, pp. 233–238, 2014.

[8] K. Semmelmann and S. Weigelt, "Online webcam-based eye tracking in cognitive science: A first look," *Behavior Research Methods*, vol. 50, no. 2, pp. 451–465, 2018.

[9] K. Wisiecka, K. Krejtz, I. Krejtz, D. Sromek, A. Cellary, B. Lewandowska, and A. Duchowski, "Comparison of webcam and remote eye tracking," in *2022 Symposium on Eye Tracking Research and Applications*, ser. ETRA '22, New York, NY, USA, 2022, pp. 1–7.

[10] Y. Wang, S. Lu, and D. Harter, "Multi-sensor eye-tracking systems and tools for capturing student attention and understanding engagement in learning: A review," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 22 402–22 413, 2021.

[11] G. Lio, R. Fadda, G. Doneddu, J. Duhamel, and A. Sirigu, "Digit-tracking as a new tactile interface for visual perception analysis," *Nature Communications*, vol. 10, no. 1, p. 5392, Nov 2019.

[12] J. Gutiérrez, Z. Che, G. Zhai, and P. Le Callet, "Saliency4asd: Challenge, dataset and tools for visual attention modeling for autism spectrum disorder," *Signal Processing: Image Communication*, vol. 92, p. 116092, 2021.

[13] T. Wen, A. Cheng, C. Andreason, J. Zahiri, Y. Xiao, R. Xu, B. Bao, E. Courchesne, C. Barnes, S. Arias, and K. Pierce, "Large scale validation of an early-age eye-tracking biomarker of an autism spectrum disorder subtype," *Scientific Reports*, vol. 12, p. 4253, 03 2022.

[14] N. Modi and J. Singh, "Real-time camera-based eye gaze tracking using convolutional neural network: a case study on social media website," *Virtual Reality*, vol. 26, no. 4, pp. 1489–1506, Dec 2022.

[15] M. Strobl, F. Lipsmeier, L. Demenescu, C. Gossens, M. Lindemann, and M. De Vos, "Look me in the eye: Evaluating the accuracy of smartphone-based eye tracking for potential application in autism spectrum disorder research," *BioMedical Engineering OnLine*, vol. 18, 05 2019.

[16] Z. Chang, J. M. Di Martino, R. Aiello, J. Baker, K. Carpenter, S. Compton, N. Davis, B. Eichner, S. Espinosa, J. Flowers, L. Franz, A. Harris, J. Howard, S. Perochon, E. M. Perrin, P. R. Krishnappa Babu, M. Spanos, C. Sullivan, B. K. Walter, S. H. Kollins, G. Dawson, and G. Sapiro, "Computational Methods to Measure Patterns of Gaze in Toddlers With Autism Spectrum Disorder," *JAMA Pediatrics*, vol. 175, no. 8, pp. 827–836, 08 2021.

[17] S. Perochon, J. Martino, K. Carpenter, S. Compton, N. Davis, B. Eichner, S. Espinosa, L. Franz, P. R. Krishnappa Babu, G. Sapiro, and G. Dawson, "Early detection of autism using digital behavioral phenotyping," *Nature Medicine*, vol. 29, 10 2023.

[18] W. Jones, C. Klaiman, S. Richardson, C. Aoki, C. Smith, M. Minjarez, R. Bernier, E. Pedapati, S. Bishop, W. Ence, A. Wainer, J. Moriuchi, S.-W. Tay, and A. Klin, "Eye-tracking-based measurement of social visual engagement compared with expert clinical diagnosis of autism," *JAMA*, vol. 330, pp. 854–865, 09 2023.

[19] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, "Webgazer: Scalable webcam eye tracking using user interactions," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 3839–3845.

[20] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," *CoRR*, vol. abs/1504.06755, 2015.

[21] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2176–2184.

[22] A. Øygard, "ClmTrackr," 2017, accessed 4 October 2023. [Online]. Available: https://www.auduno.com/clmtrackr

[23] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," arXiv:1907.06724, 2019.

[24] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[25] J. Adler, "Mobile device gaze estimation with deep learning: Using siamese neural networks," MSc thesis, KTH, School of Electrical Engineering and Computer Science, 2019.

[26] J. Lemley, A. Kar, A. Drimbarean, and P. Corcoran, "Convolutional neural network implementation for eye-gaze estimation on low-quality consumer imaging systems," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 179–187, 2019.

[27] A. A. Akinyelu and P. Blignaut, "Convolutional Neural Network-Based Technique for Gaze Estimation on Mobile Devices," *Frontiers in Artificial Intelligence*, vol. 4, p. 796825, 2021.

[28] T. Guo, Y. Liu, H. Zhang, X. Liu, Y. Kwak, B. I. Yoo, J.-J. Han, and C. Choi, "A generalized and robust method towards practical gaze estimation on smart phone," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 1131–1139.

[29] Y. Erel, C. E. Potter, S. Jaffe-Dax, C. Lew-Williams, and A. H. Bermano, "iCatcher: A neural network approach for automated coding of young children's eye movements," *Infancy*, vol. 27, no. 4, pp. 765–779, 2022.

[30] H. Balim, S. Park, X. Wang, X. Zhang, and O. Hilliges, "EFE: End-to-end frame-to-gaze estimation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2023, pp. 2688–2697.

[31] J. He, K. Pham, N. Valliappan, P. Xu, C. Roberts, D. Lagun, and V. Navalpakkam, "On-device few-shot personalization for real-time gaze estimation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 1149–1158.

[32] K. A. Funes Mora, F. Monay, and J.-M. Odobez, "Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14, 2014, p. 255–258.

[33] N. Valliappan, N. Dai, E. Steinberg, J. He, K. Rogers, V. Ramachandran, P. Xu, M. Shojaeizadeh, L. Guo, K. Kohlhoff, and V. Navalpakkam, "Accelerating eye movement research via accurate and affordable smartphone eye tracking," *Nature communications*, vol. 11, p. 4553, 09 2020.

[34] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, "Webgazer," 2024, accessed 4 March 2024. [Online]. Available: https://webgazer.cs.brown.edu/

[35] D. Adiani, C. Qu, T. Gass, S. Gurram, D. LeMay, A. Bhusal, M. Sarkar, and N. Sarkar, "Evaluation of webcam-based eye tracking for a job interview training platform: Preliminary results," in *Artificial Intelligence in HCI*, H. Degen and S. Ntoa, Eds. Cham: Springer International Publishing, 2022, pp. 337–352.

[36] H. Skovsgaard, J. S. Agustin, S. A. Johansen, J. P. Hansen, and M. Tall, "Evaluation of a remote webcam-based eye tracker," in *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, ser. NGCA '11, New York, NY, USA, 2011, pp. 1–4.

[37] M. F. Ansari, P. Kasprowski, and M. Obetkal, "Gaze tracking using an unmodified web camera and convolutional neural network," *Applied Sciences*, vol. 11, no. 19, 2021.

[38] W. A. Blakey, S. Katsigiannis, N. Hajimirza, and N. Ramzan, "Defining gaze tracking metrics by observing a growing divide between 2d and 3d tracking," *Electronic Imaging*, vol. 32, no. 11, pp. 129–1–129–1, 2020.

[39] A. Kar and P. Corcoran, "Performance evaluation strategies for eye gaze estimation systems with quantitative metrics and visualizations," *Sensors*, vol. 18, no. 9, 2018.

[40] A. Kar, S. Bazrafkan, C. C. Ostache, and P. Corcoran, "Eye-gaze systems - an analysis of error sources and potential accuracy in consumer electronics use cases," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, 2016, pp. 319–320.

[41] Y. Cheng, H. Wang, Y. Bao, and F. Lu, "Appearance-based gaze estimation with deep learning: A review and benchmark," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 46, no. 12, pp. 7509–7528, Dec. 2024.

[42] B. V. Ehinger, K. Groß, I. Ibs, and P. König, "A new comprehensive eye-tracking test battery concurrently evaluating the pupil labs glasses and the EyeLink 1000," *PeerJ*, vol. 7, p. e7086, Jul. 2019.

[43] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, vol. 91, no. 2, p. 200–215, 2011.

[44] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[45] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Computer Vision — ECCV'98*, H. Burkhardt and B. Neumann, Eds. Springer Berlin Heidelberg, 1998, pp. 484–498.

[46] D. Cristinacce and T. Cootes, "A comparison of shape constrained facial feature detectors," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, 2004, pp. 375–380.

[47] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 679–692.

[48] A. Papoutsaki, A. Gokaslan, J. Tompkin, Y. He, and J. Huang, "The eye of the typer: A benchmark and analysis of gaze behavior during typing," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, 2018, pp. 1–9.

[49] K. Harezlak, P. Kasprowski, and M. Stasch, "Towards accurate eye tracker calibration – methods and procedures," *Procedia Computer Science*, vol. 35, pp. 1073–1081, 2014, knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.

[50] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4511–4520.

[51] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13, 2013, p. 271–280.

[52] S. Porta, B. Bossavit, R. Cabeza, A. Larumbe-Bergera, G. Garde, and A. Villanueva, "U2Eyes: A binocular dataset for eye tracking and gaze estimation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3660–3664.

[53] M. Tonsen, J. Steil, Y. Sugano, and A. Bulling, "Invisibleeye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, sep 2017.

[54] T. Fischer, H. J. Chang, and Y. Demiris, "RT-GENE: Real-time eye gaze estimation in natural environments," in *15th European Conference on Computer Vision (ECCV 2018)*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 339–357.

[55] J. He, K. Pham, N. Valliappan, P. Xu, C. Roberts, D. Lagun, and V. Navalpakkam, "On-device few-shot personalization for real-time gaze estimation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 1149–1158.

[56] M. L R D and P. Biswas, "Appearance-based gaze estimation using attention and difference mechanism," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3137–3146.

[57] Y. Cheng, S. Huang, F. Wang, C. Qian, and F. Lu, "A coarse-to-fine adaptive network for appearance-based gaze estimation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 10 623–10 630, 04 2020.

[58] R. Zemblys, D. C. Niehorster, and K. Holmqvist, "gazeNet: End-to-end eye-movement event detection with deep neural networks," *Behavior research methods*, vol. 51, p. 840–864, 2019.

[59] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6911–6920.

[60] X. Zhang, Y. Sugano, M. Fritz, and A. 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 (CVPRW)*. IEEE, Jul. 2017.

[61] M. S. Slim, M. Kandel, A. Yacovone, and J. Snedeker, "Webcams as windows to the mind? a direct comparison between in-lab and web-based eye-tracking methods," *Open Mind*, vol. 8, pp. 1369–1424, 11 2024.

[62] A. Steffan, L. Zimmer, N. Arias-Trejo, M. Bohn, R. Dal Ben, M. Flores-Coronado, L. Franchin, I. Garbisch, C. Grosse Wiesmann, J. K. Hamlin, N. Havron, J. Hay, T. Hermansen, K. Jakobsen, S. Kalinke, E.-S. Ko, L. Kulke, J. Mayor, M. Meristo, and T. Schuwerk, "Validation of an open source, remote web-based eye-tracking method (webgazer) for research in early childhood," *Infancy*, vol. 29, 10 2023.