Eye-Tracking Technologies in Mobile Devices Using Edge Computing: A Systematic Review

NISHAN GUNAWARDENA, Western Sydney University, Australia
JEEWANI ANUPAMA GINIGE, Western Sydney University, Australia
BAHMAN JAVADI, Western Sydney University, Australia

Eye-tracking provides invaluable insight into the cognitive activities underlying a wide range of human behaviours. Identifying cognitive activities provide valuable perceptions of human learning patterns and signs of cognitive diseases like Alzheimer's, Parkinson's, autism. Also, mobile devices have changed the way that we experience daily life and become a pervasive part. This systematic review provides a detailed analysis of mobile device eye-tracking technology reported in 36 studies published in high ranked scientific journals from 2010 to 2020 (September), along with several reports from grey literature. The review provides in-depth analysis on algorithms, additional apparatus, calibration methods, computational systems, and metrics applied to measure the performance of the proposed solutions. Also, the review presents a comprehensive classification of mobile device eye-tracking applications used across various domains such as healthcare, education, road safety, news and human authentication. We have outlined the shortcomings identified in the literature and the limitations of the current mobile device eye-tracking technologies, such as using the front-facing mobile camera. Further, we have proposed an edge computing driven eye tracking solution to achieve the real-time eye tracking experience. Based on the findings, the paper outlines various research gaps and future opportunities that are expected to be of significant value for improving the work in the eye-tracking domain.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing theory, concepts and paradigms**; • **Computer systems organization** → *Distributed architectures.*

Additional Key Words and Phrases: eye tracking, edge computing, systematic review, mobile devices, mobile human computer interaction

Just Accepted

1 INTRODUCTION

Humans use a "saccade and fixate" strategy when observing the world, where information gathered during stable fixations and saccades is used to quickly shift the gaze direction [50]. Eye-tracking is the process of observing and recording eye movements such as fixations, saccades and pupil dilations. It provides information on where the subject is looking, how long the subject looks at a particular area, how often he/she looks at the particular objects and the types of movements made by eyes. An eye tracker is a tool used to measure eye position and eye movements. Eye-tracking provides an opportunity to find the reactions of viewers to a specific visual stimulus.

In 1823, Charles Bell was the first person to describe the effect of eye movements related to visual orientation and is considered as the founder of eye-tracking [54]. Bell explained the relationship between the eyes and the neuron system. Yarbus, in 1967 used a unique suction device called "cap" to conduct a series of experiments on

Authors' addresses: Nishan Gunawardena, 19937931@student.westernsydney.edu.au, Western Sydney University, Victoria Rd, Rydalmere, NSW, Australia, 2116; Jeewani Anupama Ginige, Western Sydney University, Australia, j.ginige@westernsydney.edu.au; Bahman Javadi, b.javadi@westernsydney.edu.au, Western Sydney University, Australia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
0360-0300/2022/5-ART1 $15.00
https://doi.org/10.1145/3546938

eye movements and different perception methods [96]. The apparatus used in Yarbus's research contained a round suction part, a canal and a small mirror. Each cap was fixed to the anesthetized sclera, making it more arduous for the subjects to perform their eye movements. Before the 21st century, eye-tracking studies required significant effort and money to develop an eye tracker. Over the past 20 years, with the development of technology, artificial intelligence, and machine learning techniques, increased research interest in eye-tracking has increased. Improvement of technology has made more user-friendly eye trackers for both researchers and participants.

Modern eye trackers track the gaze direction using near-infrared technology and a high-resolution optical sensor (camera), which monitors the pupil centre and where light reflects from the cornea. This is known as Pupil Center Corneal Reflection (PCCR) [40]. There is a complex mathematical algorithm behind this PCCR. Over the last two decades, eye tracking has proven the ability to identify the human cognitive load and attention behaviours in different fields, including marketing citeWedel2006, driving [44, 77], medical diagnosis [52], and surgery [29, 74].

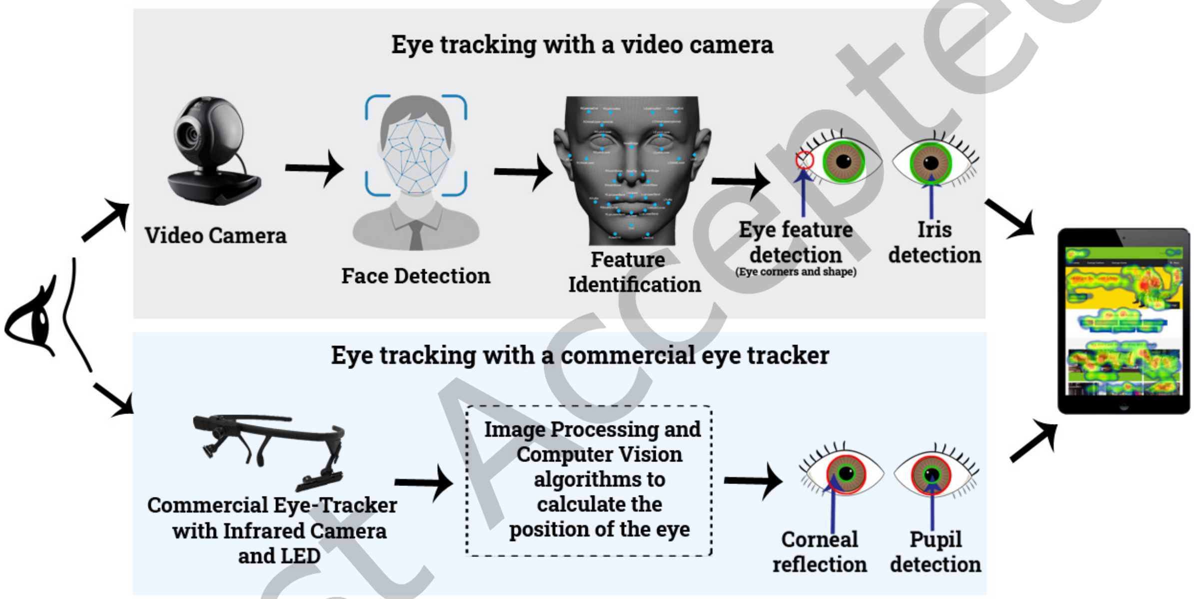


Fig. 1. How eye tracking technology works with commercial eye trackers and video cameras.

Commercial eye tracking devices cost tens of thousands of dollars, and these studies have been performed in lab settings. During the last decade, researchers focused on developing eye-tracking technologies with high accuracy and minimal cost. A video camera is a standard feature on most smart devices including, desktop and laptop and mobile devices. Eye-tracking with a standard video camera is the cheapest and the best solution. Available eye-tracking systems developed with video cameras use the positional relationship between a reference point (eye corner) and a moving point (iris). These systems cannot use the exact pupil location to track the gaze direction as standard video camera only detects light in the visible spectrum. Also, low lighting may result in less accuracy, as the algorithm fails to differentiate the iris with the face's background. Figure 1 illustrates how eye-tracking technology works with both commercial eye trackers and video cameras.

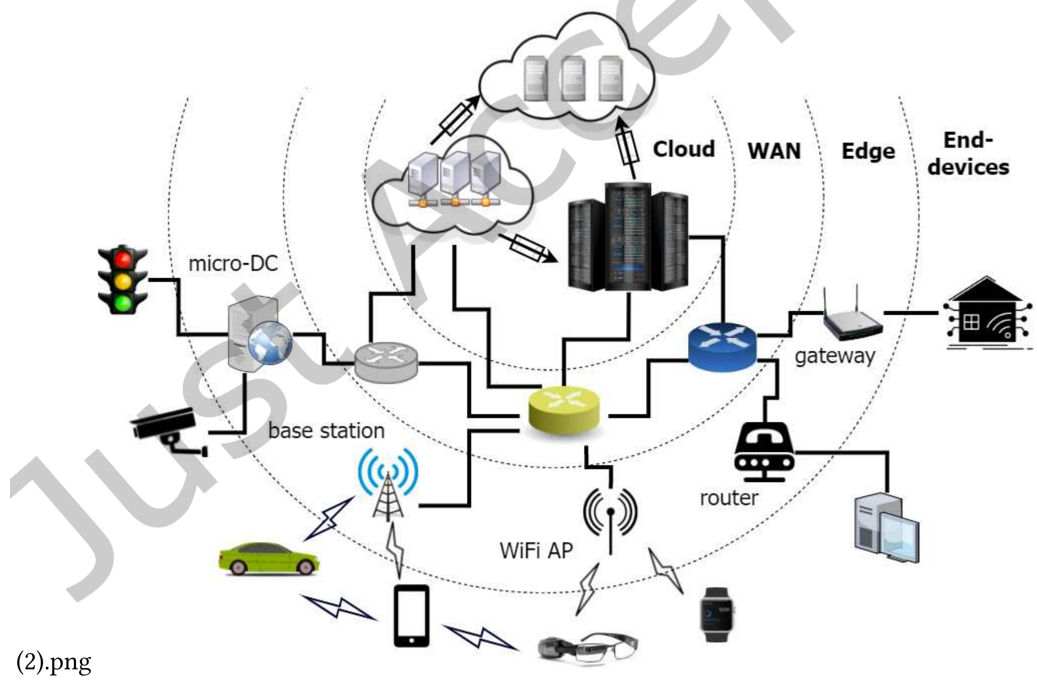
During the last decade, smartphones and tablets have become indispensable pervasive devices globally. According to the CISCO forecast, the number of global mobile devices will grow from 10.6 billion to 13.1 billion

Just Accepted

in the next three years [3]. The invention of mobile applications is one of the most significant developments in technology. Currently, the availability of mobile apps is highly increasing, and it produces a significant change in the way humans experience mobile computing. With this evolution of mobile user interaction, researchers have been interested to understand human focus areas on screens based on various content, using eye-tracking techniques. Quantifying human attention on the mobile device is invaluable in human-computer interaction.

The explosive development of internet-connected devices and real-time applications drive towards edge-computing systems [36]. Edge computing extends the concept of content delivery networks (CDN) by leveraging cloud computing infrastructure. The initial goal of edge computing was to address the network latency for data travelling long distances. Edge computing brings computation closer to the devices where data is gathered rather than depending on a cloud or central data centre, which is thousands of miles away. This increases the system's and application's efficiency and decreases traffic flow to the central server.

As shown in Figure 2, an edge node can be a nearby end-device connected via device-to-device (D2D) communications, a server attached to an access point such as WiFi, router, and base station, a network gateway, or even a micro data centre accessible to nearby devices. Edge nodes can range from a credit-card-sized computer to a micro data centre with multiple server racks. The most important feature emphasized by edge computing is physical proximity to information-generation sources. The physical proximity of computing and information-generation sources, as opposed to the traditional cloud-based computing paradigm, offers several advantages, including low latency, energy efficiency, privacy protection, reduced bandwidth consumption, on-premises computing, and context awareness [53, 79].



(2).png

Fig. 2. Edge computing architecture

In 1997, Noble et al. first proved the value of edge computing to mobile computing by demonstrating how speech recognition could be implemented with acceptable performance on a resource-limited mobile device [63].

Just Accepted

There are numerous cases of edge computing use. Smart cities are metropolitan zones that collect data on traffic conditions and utility usage using various sensors. Edge computing allows responsible organizations to respond to changing situations in near real time as a result of these sensors producing vast amounts of data [83]. Another use of edge computing is cloud gaming, in which some aspects of a game run in the cloud while the rendered video is sent to lightweight clients running on mobile phones and virtual reality glasses [2].

The capacity to process and store data more quickly is the primary benefit of edge computing. Edge computing provides many advantages over traditional cloud computing. For example, it increases network performance by reducing latency. Additionally, it is easy to implement security protocols on edge computing architecture. Further, it offers more excellent reliability by processing data closer to the source, and the bandwidth demand may not be tested. Edge computing filters out sensitive information and transfers only non-sensitive data for further processing to ensure tight security and compliance. Edge computing eliminates the need for connected devices to wait for a centralized service provider, and service availability is more significant than cloud computing. Furthermore, even if a single node is unavailable, users should access a service without interruption, and mistakes should be easy to detect and recover from.

Real-time eye tracking is vital for hands-free mobile computer interaction. Eye-tracking algorithms developed with built-in mobile device cameras need to process each frame as quickly as possible to obtain the eye feature vectors in real-time. Due to lack of computational power in mobile devices, eye-tracking applications implemented with the off-the-shelf camera only work offline [5]. By processing data closer to the mobile device, edge computing can significantly reduce the latency and achieve a real-time eye tracking experience. Moreover, when using edge computing, temporary disruptions in internet connectivity will not affect the real-time mobile device eye-tracking applications. Processing a considerable number of frames in a real-time application may introduce more data redundancy. Edge computing helps to reduce data redundancy by storing data temporarily. These benefits create the opportunity to apply edge computing into mobile device eye-tracking applications developed with built-in cameras.

Edge computing [30] was introduced due to the exponential growth of the Internet of Things (IoT) devices that are connected to the internet. These IoT devices either receive information from the cloud or deliver data back to the cloud and generate massive amounts of data. Instead of relying on the cloud, edge computing moves the processing power closer to the source of the data, and once the data is processed, only the relevant data will be sent back to the cloud. Edge computing reduces cloud computing's latency while also conserving bandwidth. Real-time applications can obtain the majority of the benefits of edge computing. It will enhance users' Quality of Experience (QoE). Therefore, applying edge computing into mobile device eye-tracking applications will be valuable.

1.1 Motivation

Eye-tracking is an emerging technology to identify users' cognition and intention [24]. Quantifying human attention on the mobile device can be valuable in human-computer interaction. Also, Eye tracking helps to determine where users are looking and for how long they fixate their gaze in a particular location. Although eye-tracking has a very long history, it is not a pervasive technology that everyone can use in their day to day lives. With the rapid advancement of mobile technology, there is a solid demand to investigate the influence of eye-tracking applications developed with mobile devices.

There have not yet been any systematic reviews of eye-tracking technologies developed on mobile devices. The primary goal of this work is to understand the concepts behind eye-tracking on mobile devices and how edge computing can be utilized in eye-tracking applications. Therefore, we investigate how eye tracking approaches have taken advantage of mobile human-computer interaction and edge computing. We will provide complete

Just Accepted

coverage of techniques, methods, data and open issues related to mobile eye-tracking technology and edge computing.

1.2 Contribution

This systematic review provides a detailed summary of available applications across different domains and guides researchers to identify promising research avenues and opportunities where future research is needed. The full extent of this survey's contribution is listed below.

- We systematically present a thorough review of the various characteristics of data and algorithms used in mobile eye-tracking applications and research.
- We provide a quick rundown of how target visual stimuli, calibration methods, evaluation metrics, and eye metrics have been used in the literature.
- We present how various computational systems are used in mobile eye-tracking applications and how edge computing and mobile device eye tracking can be integrated to provide a real-time eye-tracking experience.
- Finally, we identify several unexplored research avenues for developing adaptable mobile device eye tracking solutions.

The remainder of this systematic review is organized as follows. Section 2 presents the methodology of the systematic review, including the record identification method and the record selection criteria. Different eye-tracking algorithms that are used to develop gaze tracking with the use of the built-in camera of the mobile devices are described in section 3. Various data sources, their characteristics and calibration methods used in mobile device eye tracking are explained in section 4. Furthermore, computational infrastructures are outlined in section 5. In contrast, a comprehensive classification of mobile device eye-tracking applications across various domains is presented in section 6, and the shortcomings identified in the reported literature in section 7. Finally, the article concludes in section 8.

2 METHODOLOGY

This systematic literature review follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines [60] to select the relevant articles and present the findings.

2.1 Research Questions

This study addresses the following research questions:

- RQ1: What are different eye-tracking approaches/algorithms utilized in the mobile device eye-tracking studies?
- RQ2: How eye-tracking data, calibration methods and different metrics have been used in various mobile device eye-tracking studies?
- RQ3: Which types of mobile devices and computational systems have been used in the literature?
- RQ4: How well cloud computing and edge computing have been applied to extend the computational capabilities of mobile device eye-tracking applications.
- RQ5: What kinds of applications have been developed with mobile device eye-tracking technology?

Different kinds of eye-tracking approaches, such as commercial eye-trackers and video camera-based eye trackers, have been applied to analyze human attention on mobile devices in the literature. Also, it is essential to evaluate the screen contents as human attention may vary from static (images, texts) to dynamic (videos, games) content. RQ1 will help to identify these eye-tracking methods along with different visual contents used in previous studies.

The lack of movableness and high prices of commercial eye trackers prevent the eye-tracking from becoming a pervasive technology. With the increasing number of mobile devices, researchers tend to develop eye-tracking

Just Accepted

algorithms using the mobile device's front-facing camera. Due to the unavailability of infrared cameras in mobile devices, different computer vision and machine learning algorithms need to be applied for accurate gaze tracking. Also, current eye-tracking algorithms use various calibration procedures to achieve greater accuracy in gaze estimation. Therefore, RQ2 is expected to offer an overview of different eye-tracking algorithms and calibration methods reported in the literature.

The development of mobile device eye-tracking applications has been applied to various domains, including education, health, marketing, sports and games. The RQ3 will provide an application-oriented classification of mobile device eye-tracking applications. The answer to this research question will be based on findings and limitations identified in the literature, as well as on various further research opportunities for each mobile device eye-tracking application.

2.2 Record Identification

Relevant papers were selected by searching the online databases, including Scopus, Embase, Pubmed, Medline, Engineering Village, PsyInfo, and ACM. Out of these databases, Scopus was used as the primary database for the review, with more than 34,000 peer-reviewed journals in engineering, health sciences, social sciences, psychology, and economics. Relevant "grey literature" was also identified through Google scholar.

IBM introduced the first smartphone in 1994, and the first smartphone with a front-facing camera was released in 2003 by Motorola. However, the revolution of the smartphone began at the end of 2009 [47]. Considering this factor, papers from 2010 to 2020 were selected for this review. The final search was conducted on 11th September 2020. We first derived the main terms and synonyms for those from our research questions listed above in our search strategy. Based on these identified terms and synonyms, the search query was developed using logical operations as shown below:

("Eye tracking" OR "Gaze tracking" OR "Eye movements") AND ("Smartphone" OR "Tablets") AND ("Attention" OR "Front Camera")

Table 1 illustrated the number of papers selected from each database and record selection strategy is explained in the following section.

Table 1. Summary of the papers selected through each electronic database

Database	Paper count	After duplicates removed	From 2010 to 2020	Only Articles/ Conference Papers	Selected Papers
Scopus	91	91	85	75	22
Engineering Village	35	2	2	2	1
Embase	15	15	15	0	0
PsyInfo	7	0	0	0	0
Pubmed	6	1	1	1	0
Medline	5	0	0	0	0
ACM	2	1	1	1	1
Gray Literature	35	30	26	26	12
Total	196	140	130	105	36

2.3 Record Selection

The initial search returned 196 records, which were reduced to 140 by removing the duplicates. Then we eliminated all the records published before 2010 and selected only journal articles and conference papers. The full texts of 105

Just Accepted

publications were screened by examining the title, abstract and keywords based on the following inclusion and exclusion criteria to determine the eligibility. Finally, we selected 36 publications to be included in this systematic review.This record selection strategy is depicted in the flowchart presented in Figure 3.

Inclusion criteria:

- Article was an original and empirical research article
- Article was in the area of eye-tracking and Human Attention
- Mobile device eye-tracking technology was used in the research
- Smartphone or Tablet devices were used in the study

Exclusion criteria:

- Article was a review or commentary article
- Article was published before 2010
- Study investigated the usability of a system
- Participants looked at more than one device at the time of the experiment

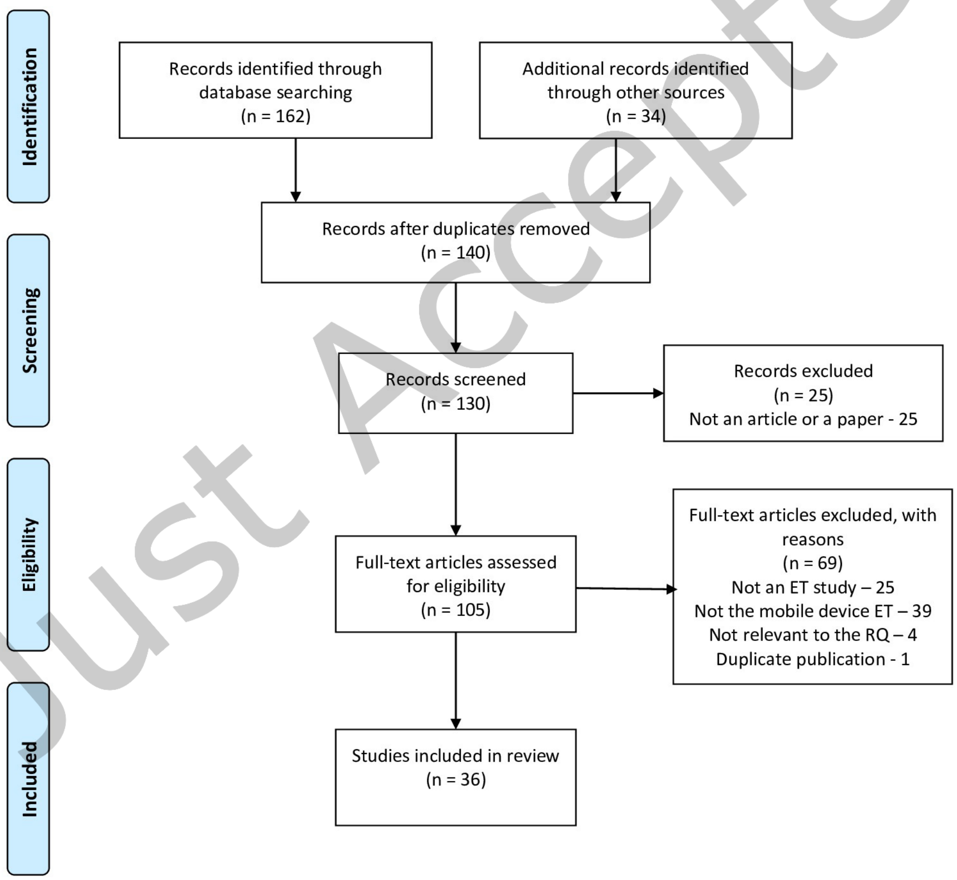


Fig. 3. PRISMA flow chart of the search strategy

Just Accepted

2.4 Detailed Analysis Method

In the first reading round, the content was divided into domains, findings, limitations, algorithms, dataset, devices used, computational architecture, evaluation metrics, tools and future directions, and the summary was extracted into a table. The table was analyzed, and the similarities between each column were identified. Domain was used to identify different areas of mobile device eye-tracking applications developed in the literature. Findings, limitations and future directions helped to clarify the open issues and future research opportunities. Various computer vision and machine learning algorithms were divided into four steps- face detection, eye detection, iris detection and gaze angle calculation. Datasets were classified based on the data type and participant details, including age, gender and ethnicity.

During the second round of reading, devices were further simplified under mobile devices and eye-tracking devices. Mobile devices were used to perform eye-tracking experiments, and eye-tracking devices can be either commercial eye-trackers or eye-tracking methods developed using built-in cameras. Further, the size of the mobile device screens was added to the table for further evaluation. Computational architectures used in mobile device eye-tracking settings were separated into processing units and computational platforms. Finally, all the information required for the reporting was extracted from the publications.

3 EYE-TRACKING APPROACHES

This section will answer RQ1 by providing an in-depth analysis of eye-tracking algorithms. Overall, the eye-tracking approaches used by previous researchers can be divided into two types. The first type involves using external commercial eye-tracking apparatus such as eye-tracking glasses (e.g., Tobii Pro Glasses 2, Pupil Labs Core, SMI Eye Tracking Glasses) and screen-based eye trackers (e.g., Tobii Pro X3-120, Gazepoint GP3 HD, Smart Eye AI-X). In total, 16 articles used this approach. The second type involves the use of a built-in camera of the mobile device. Nineteen papers utilized this method.

3.1 Use of external apparatus for eye-tracking in mobile devices

Commercial eye-trackers with video cameras were used to take high-resolution images of the user's eye, and infrared cameras were used to create a pattern of near-infrared light on the eyes in fourteen of the papers studied. Normal lights sources are not capable of providing a good contrast when compared to infrared light. In commercial eye trackers, an infrared light source is directed towards the pupil, and the reflections that are caused due to the infrared light are tracked with an infrared camera. Additionally, infrared light sources do not cause any distraction as it is not visible to the human eye. Commercial eye trackers use unique machine learning, image processing and mathematical algorithms to determine the gaze direction. However, these existing proprietary eye-tracking algorithms cannot be revealed.

Two types of commercial eye trackers can be found in the literature. They include (i) screen-based eye trackers and (ii) eye-tracking glasses. Screen-based eye trackers are attached to the screen and require the user to sit in front of the screen. These eye trackers track the eyes only within certain limits, and users cannot freely move their heads. The most common types of commercial eye-trackers that are used in the retrieved literature are Tobii X2-60 eye tracker [19, 20, 78], and Tobii x120 eye tracker [1, 7, 88]. Both Tobii X2-60 eye tracker and Tobii x120 eye tracker are screen-based eye trackers. Eye-tracking glasses are fitted close to the eyes, and users can move their heads as freely as they would like. SMI Eye-Tracking Glasses have been used in three reported studies [17, 64, 69]. There are many commercial eye-trackers available, and Figure 4 illustrates screen-based and glass wear commercial eye-trackers.

Existing commercial eye-trackers are suffering from high cost, inaccuracy under real-world conditions, and limited user mobility. There is a significant variation in the eye-tracker prices from $500 to $10000. The price of the eye-tracker will always depend on the capabilities and accuracy of the device. Further, these eye-trackers failed

Just Accepted

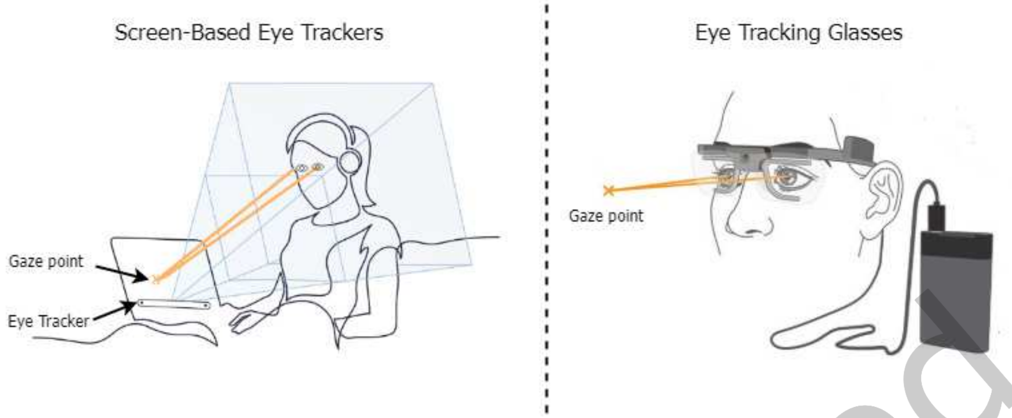


Fig. 4. Two types of commercial eye trackers used in the literature (i) Screen-based eye trackers and (ii) Eye tracking glasses

to provide more accurate gaze directions in some real-world conditions as they required a tedious calibration process every time. Also, commercial eye-trackers available today need to have other equipment like a battery pack and a processor, typically connected through a wired cable. Therefore, it is highly inconvenient for users to use these eye-trackers in mobile daily-life situations. These factors prevent eye-tracking from becoming a ubiquitous technology that should be accessible to everyone.

3.2 Use of mobile built-in camera for eye-tracking

The most common type of mobile user attention recognition method requires special-purpose eye-tracking equipment. The increasing number of mobile devices and the rapid development in computer vision technologies allow researchers to develop eye-tracking algorithms using built-in cameras in mobile devices. The development of such an algorithm can be used as human attention recognition in situ without having any additional equipment. In the retrieved literature, we have identified 22 papers that tried to develop such an algorithm using the front-facing camera or the back camera of the devices.

Most of the studies (21 out of 22) have used the front-facing camera of the mobile device, while only one study [99] has used the back camera for eye-tracking. According to Zhang et al. [99], there is a significant problem while using the back camera for eye-tracking as it requires an extra person or equipment to hold the device. However, when it comes to eye-tracking via the mobile device's front camera, it can be applied to many applications as the front camera is on the same plane as the screen. However, it introduces a significant variation in the head pose, appearance and background. To overcome these issues, researchers developed many robust algorithms using different techniques, like machine learning [6, 62], mathematical modelling [41] and image processing algorithms [71]. However, these algorithms were developed based on face detection, eye detection, iris or pupil detection and gaze angle calculation. The following subsections will describe each step in more detail, and Figure 5 depicts different algorithms used in each step to implement a mobile device eye-tracking system in the selected studies.

3.2.1 *Face Detection.*

Data captured from built-in cameras in mobile devices contain a wide range of various head poses while some of them have only a fragment of the face [**?**]. Researchers have utilized different face detection algorithms to overcome this challenge in mobile device eye-tracking applications. Face detection can be beneficial to identify any faces in the frame captured from the mobile device camera. Once the face detection algorithm identifies a face, a face-localization is performed to determine the position of the face. If the algorithm fails to detect any face, it is considered as there is no eye contact. There are a significant number of

Just Accepted

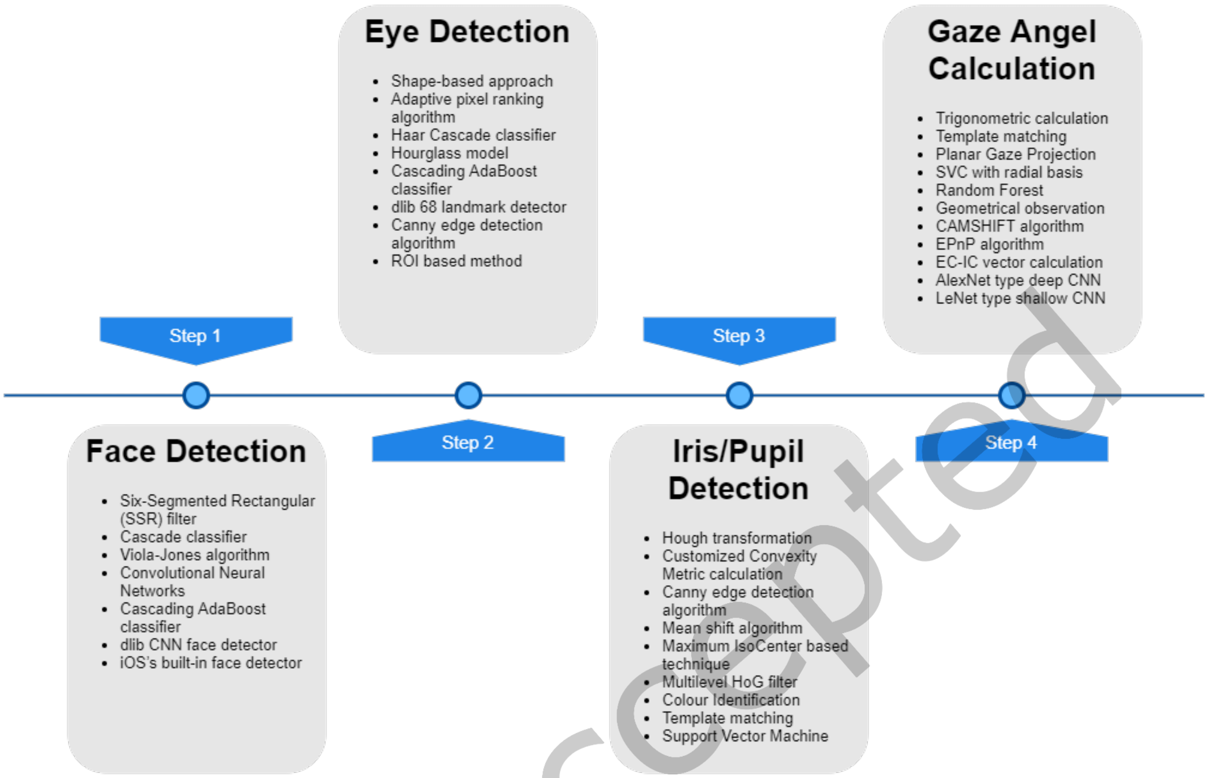


Fig. 5. Different algorithms used for each step of the mobile device eye-tracking systems developed on built-in camera

face detection algorithms, such as Viola-Jones face detector, Support Vector Machine based face detection and Neural Network-based face detection presented in the literature [35, 39, 66, 75, 91]. There is only one study [99] which uses the iOS's built-in face detector with the back camera to detect the faces.

Among these studies, [86] and [46] have used the Viola-Jones method [90] which is a cascade classifier based on Haar features to detect the face in each frame. Further, [101] and [48] report that they have used Cascading AdaBoost classifier and Cascading classifier, respectively, for the face detection. In 2001, Paul Viola and Michael Jones developed this algorithm as an object recognition framework. Viola-Jones algorithm consists of two steps, namely training and detection. Adaptive Boosting (AdaBoost) algorithm is combined with the training process to identify the best features in the feature vector. Cascading is another technique for increasing the model's speed and accuracy. Because there is less data to process, the image is converted to a grayscale image before the Viola-Jones algorithm detects the face. It then recognizes the face in the grayscale image and then locates it in the coloured image. Alfred Haar, the inventor of Haar wavelets, is honored with the name Haar-like features. Horizontal edge features and vertical line features are used to detect the face among the three types of Haar-like features (edge features, line features, and four-sided features). Horizontal edge feature and vertical line feature, for example, are used to describe the brows and nose, respectively. The AdaBoost learning algorithm and Haar-like features were used [23], the Viola-Jones algorithm implements a powerful face detection mechanism.

Sawettanusorn et al. [76] proposed a real-time face detection algorithm based on a Six-Segmented Rectangular (SSR) filter in 2003, which achieved 92% detection accuracy at 30 frames per second. Song et al. [81] have used

Just Accepted

the SSR filter to identify the face position. Assuming that the nose is brighter than the eye areas and the eye area is relatively darker than the cheekbone area, a six segmented rectangle is used to scan the input image. If all these conditions have been satisfied, the rectangle's centre is considered a candidate between the eyes.

Researchers are now focusing on building more robust deep neural networks to achieve better accuracy in near real-time as they deal with massive amounts of data. A convolutional neural network (CNN) is a deep neural network approach that has been shown to be effective in image recognition and classification [28, 56]. A CNN is made up of an input layer, hidden layers, and an output layer. Hidden layers include convolutional layers, pooling layers, and fully connected layers. CNN can detect the critical features without the need for human interpretation. Bâce et al. [5]used three multi-task CNNs to detect faces in the frame captured by the mobile device's front-facing camera. They also stated that when multiple faces are present, the algorithm selects only the face with the largest bounding box, assuming that only one user is using the system at a time.

3.2.2 Eye Detection . Once the face is detected through one of the algorithms mentioned above, most researchers have used an eye detection algorithm to locate the position of the eyes. The most common eye detection algorithm is the Haar Cascade Classifier used in 12 studies out of 21, which used a built-in camera to develop mobile device eye-tracking algorithms [5, 31, 37, 38, 46, 48, 49, 55, 71, 86, 92, 94]. Paul Viola and Michael Jones proposed object detection using Haar feature-based cascade classifiers in 2001 [90]. The area around the eyes is darker than the cheeks, bridge of the nose, and forehead. As a result, a set of two adjacent rectangles above the eye and the cheek region is a common Haar feature for face detection. These rectangles' positions are defined in relation to a detection window that acts as a bounding box for the target object. In addition, Zhu et al. citezhu2018mobiet used the Cascading AdaBoost classifier to detect the eyes, which provides an effective learning algorithm and strong bounds on generalization performance [90].

Some studies [6, 99] have used "dlib landmark detector", a pre-built model to detect the face landmarks including mouth, eyes, nose, and jaw. dlib landmark detector can be used to locate 68 coordinates in the facial structure. Dlib has already a pre-built face detection model, which [6] have used in their face detection step. Further, Song et al. [81] used a shape-based method to detect the eye. Considering the shape of the eye as a global feature and a shape-based algorithm detects an object with a distinct shape. Edges, boundaries, contours of an eye can be used to detect a shape of an eye. Otoom et. al. [67] performed Canny edge detection algorithm [10] prior to shape-based eye detection. Canny edge detection algorithm consists of five different steps, namely, (i) noise reduction, (ii) intensity gradients finding, (iii) non-maximum suppression and (iv) hysteresis thresholding.

To overcome the problem of non-uniform luminance distribution in eye detection, Jiang et al. citejiang2019vals used an algorithm called adaptive pixel ranking. The spatial relevance of the pixels is considered in an adaptive pixel ranking algorithm to preserve the structure information. Then, to overcome the non-uniform luminance distribution and glow spots caused by the cornea's reflection, use an iterative pixel clustering process. To identify the 2D locations of facial landmarks, including the eyes, Bâce et al. [5]used a state-of-the-art hourglass model [18], which is a fully connected convolutional network. All the above-mentioned eye detection methods resulted in a bounding rectangle area covering the user's eyes. The results of this step are fed into the next step of iris/pupil detection.

3.2.3 Iris/Pupil Detection . The pupil is the dark-coloured circle in the centre of the eye, surrounded by a circle called the iris. The iris is set within the sclera, white in colour, providing a contrasting background to all iris colours such as black, brown, blue, amber, and green. In implementing a more precise eye-tracking algorithm, the pupil needs to be detected. As it is hard to detect the exact location of the pupil, most studies have used iris detection instead of pupil detection. Due to the rounded shape of the iris/pupil, researchers tend to develop their iris/pupil detection algorithm using the Hough transformation [67, 81] which is a feature extraction method used in digital image processing. These studies have used the extended Hough transformation to identify the positions of the pupil/iris.

Just Accepted

Apart from the Hough transformation, Otoom et al. [67] have used the integral projection algorithm [12] to determine the position of the lateral and medial canthus of the eye. Huang et al. [38] have used a multi-level Histogram of Oriented Gradients (mHoG) [15] feature extraction method with the Linear Discriminant Analysis to reduce the feature dimensionality. The multi-level Histogram of Oriented Gradients counts the number of occurrences of gradient orientation in a selected region of interest.

Image processing algorithms, like Canny edge detection algorithm [10], mean-shift algorithm [25] and color Identification algorithms play a key-role in iris/pupil detection. Using the Canny edge detection algorithm significantly reduces the amount of data that need to process, and boundaries are enough to detect the pupil/iris using shape-based methods. On the other hand, the mean shift algorithm is a clustering algorithm that does not require to specify the number of clusters, and the algorithm itself determines it. Due to its robustness to outliers and model freeness, Wang et al. [92] applied Maximum IsoCenter based technique to estimate the centre of the pupil. Due to undesirable lighting conditions, Wang et al. found it difficult to detect the pupil centre accurately. Pupil and Iris have the darkest area in the rectangle area of the user's eyes. Identifying colour values of a given range of pixels would be very straightforward to distinguish the pupil/iris from other eye features [46, 55].

After finding, the inner and outer canthus, Jiang et al. [41] utilized a customized convexity metric to identify the centre of the pupil. The customized convexity metric was developed using an analytic geometry that transforms the intersection point between the directional vector towards the iris centre and the face plane, from the front-camera frame to the face-frame. This method allowed the researchers to obtain the coordinates of the iris centre. Studies [94] and [6] used template matching and support vector machine for iris/pupil detection in their studies respectively.

3.2.4 Gaze Angle Calculation. The eyeball is represented as a sphere with the iris resting on the surface. The eye's optical axis is a line that runs through the centre of the eyeball and the centre of the iris. The anatomical axis of the eye is a line that passes through the centre of the eyeball and is perpendicular to the screen plane [81]. In eye-tracking studies, the angle calculation between these two axes is referred to as the gaze angle calculation. The calculation of gaze angle is the most critical step in any eye-tracking algorithm.

According to the studies, there are several methods for calculating gaze angle. For example, [67, 81] calculated the gaze angle by multiplying the vertical (angle between the optical axis and the anatomical axis in the gaze-vertical plane) and horizontal (angle between the optical axis and the anatomical axis in the gaze-vertical plane) angles (optical axis and the anatomical axis in the gaze-horizontal plane). The vertical angle determines whether the user looks left, right, or in the middle of the screen. The horizontal angle determines whether the user is looking up, down, or in the middle of the screen. According to the previous studies, this is the most precise way of calculating the gaze angle. Jiang et al. used the Planar Gaze Projection Model, which is more equivalent to the pinhole camera model, to calculate the gaze angle in their study of visual attention on mobile devices [41]. Apart from these algorithms, [55] and [94] developed eye-tracking algorithms based on geometrical observations and trigonometric calculations respectively.

However, several studies have not calculated the exact angle between these two axes. Without calculating the gaze angle, these studies predict the gaze direction using different algorithms like template matching [31, 46, 59, 86, 99], convolutional neural networks [6, 62, 87], and random forests [38]. The study [92] used a Support Vector Classifier (SVC) with a radial basis function to detect the gaze direction. Pion et al. [71] used a colour-based object tracking algorithm known as CAMSHIFT to detect the gaze position and to track objects by matching the probability density functions of two consecutive frames. B âce et. al. [5] and Zhu et. al. [101] used EPnP algorithm with Levenberg-Marquardt optimization and EC-IC vector calculation respectively.

In this section, we have discussed two eye-tracking approaches which were used in mobile-human interaction. At first, we have outlined various commercial eye tracker and their limitations in mobile application settings. Subsequently, different steps of eye-tracking algorithms developed on mobile device in-build cameras have been

Just Accepted

addressed in this section. These steps include face detection, eye detection, iris/pupil detection and gaze angle calculation.

4 EYE-TRACKING DATA, CALIBRATION METHODS AND METRICS

In this section, we have answered RQ2. Table 2 illustrated various characteristics of data, including source, scale and data type. The remaining sub-sections explain how target visual stimuli, calibration methods, evaluation metrics and eye metrics have been used in the literature.

4.1 Eye-Tracking Data

Mobile device eye-tracking applications were developed based on different datasets. Most of the reported studies (29) have been implemented by a collection of Private datasets are those that are accessible only to the dataset's owner and are not publicly accessible. Two studies have been conducted by collecting data which are now publicly available (GazeCapture dataset - [45], TabletGaze dataset -[37]). GazeCapture dataset contains data from 1474 participants, including 2.5 million frames, and the TabletGaze dataset consists of 816 gaze video sequences from 51 subjects. Krafka et al. [45] and Valliappan et al. [87] used the GazeCapture dataset to train the CNN and TabletGaze dataset to validate the implemented eye-tracking algorithm. Mobile Face Video dataset(MFV) [21], Understanding Face-Eye Visibility dataset (UFEV) [**?**], Multi-Pie dataset [27] and Menpo dataset [27] are another four datasets which are publicly available. MFV dataset consists of 750 face videos from 50 people, captured using the front-facing camera, while UFEV includes 25,726 images from 11 participants. Multi-Pie and Menpo datasets include synthesized images of the eye, where the Multi-Pie dataset consists of 755,370 images from 337 subjects, and the Menpo dataset includes more than 16,000 images.

Table 2 outlines participant details, including the number of participants, demographic information (age, gender, ethnicity), and the number of bespectacled persons who participated in the respective studies. In the table, private datasets used in the literature are labeled with the author's name, while public datasets are labeled with the dataset's name. Back et al. [5], Krafka et al. [45] and Valiappan et al. [87] used the GazeCapture dataset, which contains data of more than 1400 subjects, and it is the only existing large-scale public dataset. When considering modern machine learning techniques such as deep neural networks require a large number of data to train. 34 out of the 36 selected papers have reported the number of participants for each experiment. Nevertheless, two studies [59, 69] failed to report the participant details.

In eye-tracking studies, ethnicity is a significant factor that must be taken into account. The narrow eyes of Asian people cause the eye-tracker to lose the glint, according to Blignaut and Wium [8]. In comparison to African and Caucasian participants, Asian participants' eye-tracking accuracy and precision are extremely low. According to Blignaut and Wium, some African and Caucasian people have narrow clefts in their eyes, but not all Asians do. However, only eight studies have stated the ethnic details of the participants, making it difficult to assess the quality of eye-tracking in terms of ethnicity [4, 7, 37, 38, 55, 58, 64, 88].

Dahlberg [14] showed that wearing glasses causes 20% error in results than when not wearing glasses as eyeglasses may introduce extra reflections, block eye features and create erroneous glints. Therefore, it is vital to consider the eye-tracking accuracy while wearing glasses or contact lenses. Six studies have reported the number of bespectacled persons who participated in the experiments [37, 38, 49, 55, 81, 99].

The entire eye-tracking algorithm will depend on the type of data captured by the camera. Two types of data have been used in the retrieved literature, and it is essential to compare these two types of data descriptively. All the explored studies except one have used the video recordings of the eye (VR) as the primary input for the eye-tracking algorithm. These video data consist of time-synchronized mobile screen recordings and eye gaze data. Nambi et al. [62] have used synthesized images of the eye (SI) as the primary input for their eye-tracking

Just Accepted

Table 2. Summary of different datasets used in selected mobile device eye tracking papers. (VR - Video recording of eye movements, SI - Synthesized Images of eye, MFV - Mobile Face Video dataset, UFEV - Understanding Face and Eye Visibility)

Reference/Dataset	Data type	Participants count	Male	Female	Age	Ethnicity	Glass Wear
De Haan et al. [17]	VR	122	26	96	22.87		
Dunaway et al. [20]	VR	115	28	87	20.2		
Shayan et al. [78]	VR	76	42	34	11.825		
Ohm et al. [64]	VR	69	37	32	24.02	Germans	0
Mcewen et al. [58]	VR	30	13	17	7	Canadians	
Paletta et al. [68]	VR	25					
Dachuan et al. [49]	VR	21					14
Cauwenberge et al. [88]	VR	21	7	14	22	Belgians	
Domachowski et al. [19]	VR	20	8	12	26.25		
Song et al. [81]	VR	20	3	17			5
Lu et al. [51]t	VR	18	15	3	1825		
Biedert et al. [7]	VR	18	13	5	1828	Germans	
Mariakakis et al. [55]	VR	17	9	8	26.7	Caucasians and Asians	5
Li et al. [48]	VR	16					
Pino et al. [71]t	VR	16					0
Tonomura et al. [84]	VR	13					
Wang et al. [92]	VR	12	8	4	2233		
Zhang et al. [99]	VR	12	5	7	29		1
Jiang et al. [41]	VR	10					
Otoom et al. [67]	VR	9	6	3	1822		
Wood et al. [94]	VR	8	7	1	2027		
Aşçı et al. [4]	VR	8			1828	Turks	
Al-Zeer et al. [1]	VR	7	2	5	7.3		
Kwiatkowska et al. [46]	VR	6					
Seongwon et al. [31]	VR	5					
Vaitukaitis et al. [86]	VR	5	4	1	2125		
Zhu et al. [101]	VR	5			2328		
Miluzzo et al. [59]	VR						
Paletta et al. [69]	VR						
GazeCapture [45, 87]	VR	1474					
GazeCapture, MFV, UFEV [5]	VR	1534					
Menpo, MultiPie-face, Unity Eyes [62]	SI	10					
TabletGaze [37, 38]	VR	51	39	12	2040	Caucasians and Asians	26
UFEV, MFV [6]	VR	60					

Just Accepted

algorithm. Park et al. [**?**] have demonstrated that accurate gaze estimation based on images is a challenging task compared to video recordings as images provide less information compared to videos.

4.2 Target Visual Stimuli

Visual stimuli or mobile screen content can be divided into static and dynamic based on the type of visual stimuli. Static visual stimuli present image type data which are fixed and do not change rapidly. Static visual stimuli types, such as images [46, 68, 78, 84], static web pages [7, 71], mobile key-board [51, 59, 92, 99], google search result [19], news articles [17, 20, 48, 88] and static objects [38, 94], have been used in the mobile device eye-tracking studies represented in this corpus. Dynamic visual stimuli represent frequently changing visual data, such as videos and moving objects on the mobile screen. Song et al. [81] have evaluated their mobile authentication method using different visual stimuli, including an illusion-image, simple-dot, rotating circle and a fruit row which represent both static and dynamic visual stimuli. None of these studies has considered the human attention in dynamic content of the visual stimuli (e.g. video clips, video games).

4.3 Calibration methods

Calibration is the process that measures the geometric properties of the user's eyes as the basis for an accurate and completely personalized viewpoint calculation. Most of the available eye-tracking algorithms use different calibration methods to build an anatomical 3D eye model using these geometric properties to calculate the gaze data. During the calibration process, the user is asked to look at a point, a video or other graphical feature which moves across the screen. This section describes different calibration procedures used in the reported studies. Table 3 shows different calibration methods used in the selected studies.

Table 3. Summary of different calibration methods used in selected mobile device eye tracking papers.

Reference	Calibration method
[45]	13-point calibration
[64]	3-point calibration
[58]	4-point calibration
[68]	5-point calibration
[20, 101]	9-point calibration
[99]	Six gesture calibration

Point-based calibration is the most common type of calibration in eye-tracking experiments where the user has to look at specific points (calibration dots) on the screen. During the calibration period, images of the eyes are collected and evaluated to build the 3D eye model. In 2014, Harezlak et al. [33] presented numerous ways to simplify the calibration process in eye-tracking studies. They concluded that the number of points and the layout of the stimuli affect the accuracy of a calibration model. Krafka et al. [45] have used a 13-point calibration procedure on their eye-tracking algorithm. With a larger number of calibration points, the accuracy of gaze tracking improves [33]. However, users prefer fewer calibration points or no calibration as it is considered a tedious procedure. Also, the number of calibration points will depend on the size of the screen. A few other studies have also used point-to-point calibration such as 3-point calibration [64], 4-point calibration [58], 5-point calibration [68], and 9-point calibration [20, 101].

The user is asked to look up, down, left, right, centre, and close both eyes in a six-gesture calibration procedure by Zhang et al. [99]. They extract an image of each eye during the calibration process, normalize it, and save the eye-gesture template. By template matching, these templates are used to determine the gaze direction. Twenty-four studies did not mention calibration at all. Among the findings, the study [94] is the only study that has reported that they have not performed calibration prior to the eye-tracking experiment.

4.4 Evaluation Metrics

Without experiments, researchers are unable to evaluate the effectiveness of these eye-tracking algorithms. This section reviews evaluation metrics with the results obtained from the mobile device eye-tracking experiments. The

Just Accepted

purpose of Table 4 is to identify the most popular evaluation metrics that have been used with the eye-tracking systems developed with the mobile device off-the-shelf camera. Further, Table 4 includes the datasets and the algorithms used in four different steps described in section 3. We could not compare results derived from the evaluation metrics as the datasets, the size of the data, and the evaluation metrics varied according to studies.

Accuracy is the average difference between the measured gaze position and the actual stimuli position. Precision is the ability to reproduce the exact gaze point measurement. According to the literature, it can be observed that accuracy and precision are the most common type of evaluation metric used in the reported literature [49, 62, 67, 71, 81, 92, 99]. One of the reasons for this might be that those metrics are easy to understand and simple to formulate and compare results with other studies. Accuracy is defined as the average difference between the measured gaze position and actual stimuli position. In several studies, the accuracy is presented as a percentage value [59, 71, 92, 99] and two studies have given the exact difference in degrees(°) [41, 101]. Miluzzo et al. have used motion analysis and template matching to implement their eye-tracking algorithm and achieved an accuracy between 76.73-99.42%. The upper limit of their accuracy is the highest accuracy obtained in the literature. Wang et al. [92] achieved the lowest accuracy of 46% as they tried to infer passwords using eye-tracking in three trials.

On the other hand, precision is the ability of the eye-tracking algorithm to reproduce the same gaze direction reliably. Nambi et al. [62] have used the precision as their evaluation metric and achieved 98%. The accuracy and precision of these studies are highly dependent on the properties of the participants and stimuli, illumination, data collection procedure and proposed algorithm.

Matthews Correlation Coefficient (MCC) is a quality measure in binary class classification that was introduced by Brina Mathews in 1975 and is widely used in machine learning studies [57]. When considering the balance ratio of the four confusion matrix categories, the MCC score provides a more informative result than accuracy and precision (true positives, true negatives, false positives, false negatives). The higher the value of MCC, the better the performance of the algorithm. Bâce et. al. have utilized MCC score, True Negative Rate (TNR) to demonstrate the results of their two studies [5] and [6]. For both studies, they have used MFV and UFEV datasets and achieved significantly different MCC scores. In study [5], they have obtained 0.86 while testing and training on the MFV dataset and 0.76 on the UFEV dataset. However, in [6], they have achieved fewer MCC scores while performing the cross dataset evaluation. The MCC score is 0.124 while training on MFV and evaluating on UFEV. Also, training on UFEV and testing on MFV get the MCC score of 0.484. Further, they have stated that identifying non-eye contacts is a major problem in eye contact detection [5]. They used TNR as the performance indicator to identify such cases. Kwiatkowska et al. [46] also used the TNR to measure the performance of non-eye contacts, and they have achieved the best TNR of 90%. Other matrices like True Positive Rate (TPR) [46, 48] and False Positive Rate(FPR) [86] have been also used in the literature.

Performance evaluation with error metrics helps to find how close the predictions were to expected values. Studies like [38, 45, 55, 81, 94] have used different error metrics like Error, Root-mean-square error (RMSE), Mean Error (ME), and Dot error (DE). The error was reported by Mariakakis et al. [55]as the difference between the number of lines excerpted and the line prediction in their algorithm, while Krafka et al. [45]reported the error as the average Euclidean distance (in centimetres) from the true fixation location. Krafka et al. calculated a metric called dot error, which gives the average Euclidean distance of all the frames corresponding to a gaze point at a specific location, rather than processing a single frame for a given fixation. They claim that in real-world eye tracking applications, dot error is lower than error because it uses temporal averaging. Huang et al. [38] reported that the accuracy of their gaze estimation algorithm is a ME while Wood et al. [94] used RMSE. Both ME and RMSE express average model prediction error, and both metrics can range from 0 to infinity. RMSE is more useful when large errors are particularly undesirable as it gives high weights to large errors.

Just Accepted

Table 4. Evaluation metrics utilized in mobile device eye tracking studies developed with built-in cameras

Ref.	Year	Dataset	Face Detection	Eye Detection	Pupil Detection	Gaze Direction	Results
[59]	2010	PD		[t]62cmMotion analysis		[t]62cmTM	Acc = 76.73-99.42%
							BDA = 67-84%
							ACU = 65.4%
							AMC = 56.51%
							CT = 100 msec
							BC3 = 40%
[86]	2012	PD	VJ			TM	Sen = 28.3%
							FPR = 17.6%
							Acc = 60%
[31]	2012	PD		Haar		TM	Acc = 90%
							BC = 2.15%
[71]	2012	PD		Haar		CAMSHIFT	Acc = 53%
[94]	2014	PD		Haar	TM	Trigonometric	RMSE = 25.8
[49]	2015	PD	Qualcomm			PCA	Acc = 97.3%
[55]	2015	PD	Qualcomm		CI	Geo	Error = 0.3
[81]	2016	PD	[t]52cmSSR	[t]52cmShape-based approach	[t]52cmHough	[t]52cmSpatial angles calculation	BAM = 88%
							AUC = 96.74%
							EER = 10.61%
							ACU = 45%
							AMC = 41.1MB
[38]	2016	TG		Haar	mHoG	Random Forest	ME = 3.17cm
[45]	2016	GC/ TG				CNN	Error = 2.53cm
							DE = 2.38cm
[87]	2021	GC/ PD				CNN	Error = 1.92cm
[99]	2017	PD	iOS's face detector	dlib landmark		TM	Acc = 86%
[48]	2018	PD	Cascade classifier	Haar/ ROI	Canny	Harris	TPR = 91%
[92]	2018	PD		Haar	Iso and MS	SVC	Acc = 46%
[46]	2018	PD	VJ	Haar	[t]32cmCI	TM	TPR = 84%
							TNR = 90%
							Acc = 86.9%
[101]	2018	PD	CAC	CAC		EC-IC vector	Acc = 2.34 - 4.69°
[67]	2018	PD		Canny	Hough	angle calculation	Acc = 91.3%
[62]	2019	PD	PnP, RANSAC	ROI		CNN	Pre = 98%
[41]	2019	PD		APR	Customized convexity metric	Planar Gaze Projection Model	Acc = 1.3°
[5]	2019	MFV	CNN	[t]42cmHourglass model		[t]42cmEPnP, LM and Kalman filter	TNR = 88%
		MFV					MCC= 0.86
		UFEV					TNR = 74%
		UFEV					MCC= 0.76

Just Accepted

Table 4, Evaluation metrics utilized in mobile device eye tracking studies developed with built-in cameras cont.

Ref.	Year	Dataset	Face Detection	Eye Detection	Pupil Detection	Gaze Direction	Metric and Results
[6]	2020	UFEV	[22cmdlib CNN face detector	[22cmdlib landmark	SVM	CNN	MCC = 0.124
		MFV					MCC = 0.484

Acronyms used in the table 4: MCC=Matthews Correlation Coefficient,Acc=Accuracy,ME=Mean Error,TPR=True Positive Rate,TNR=True Negative Rate,DE=Dot error,BC=Battery Consumption per hour,BC3=Battery Consumption after 3 hours,RMSE=Root-mean square error,TPR=True positive rate,FPR=False positive rate,Pre=Precision,Sen=Sensitivity,Err=Error,AMC=Average memory consumption,CT=computation time for one frame,ACU=Average CPU usage,BDA=Blink Detection Accuracy,BAM=Balanced Accuracy metric,AUC=Area under the curve,EER=Equal error rate,PCA=Principle Component Analysis,TM=Template Matching,SSR=Six-Segmented Rectangular filter,PD=Private dataset,GC=GazeCapture dataset,TG=TabletGaze dataset,Haar=Haar Cascade Classifier,ROI=Region of Interest,CNN=Convolutional Neural Network,Canny=Canny edge detection,CAC=Cascading AdaBoost classifier,Qualcomm=Qualcomm Snapdragon SDK,VJ=Viola-Jones algorithm,Hough=Hough transformation,SVC=Support Vector Classifier,MS=mean shift algorithm,Iso=Maximum IsoCenter based technique,CI=Colour Identification,Geo=Geometrical observati,APR=Adaptive pixel ranking algorithm,mHoG=multilevel HoG filter,Harris=Harris corner detection,dlib landmark=dlib 68 landmark detector,LM=Levenberg-Marquardt optimization,PnP=Perspective n Point,RANSAC=Random sample consensus,EPnP=Efficient Perspective-n-Point.

It is also important to consider the battery and power consumption of proposed algorithms in the literature. Song et al. [81] used Average memory consumption (AMC) and Average CPU usage (ACU) to evaluate the impact of resource consumption. These metrics have been obtained from the meminfo and cpuinfo command of the Android Debug Bridge shell. Further, they reported that 41.1MB in AMC and 45% in ACU have been utilized in their best authentication method. Miluzzo et al. [59] have reported that 65.4% in ACU, 56.51% in AMC and 40% in battery consumption after three hours have been utilized in their EyePhone application, and it took only 100msec to process a single frame. Designing a mobile application with minimum battery and power consumption is vital, as higher power consumption reduces the usability and usefulness of the application.

4.5 Eye Metrics

Eye movements can be utilized as an innovative way to determine situation awareness and human cognitive ability. The literature describes two types of eye metrics. These types include fixation metrics and saccadic metrics calculated based on fixation and saccadic data, respectively. Table 5 depicts different eye metrics used in the reported literature with its' definitions.

Table 5. Various eye metrics utilized in the mobile device eye tracking studies

Eye Metrics	Definition	Study
Dwell time	Total time spent looking within an AOI	[17, 69]
Fixation Count	Total count of fixation points in a given AOI	[20, 58, 69, 71]
Fixation Duration	Average time for fixations	[1, 19, 20, 58]
Antisaccade count	Number of rapid movement of the eye ball to the opposite direction of the moving object	[68]
Prosaccade count	Number of rapid movement of the eye ball to the same direction of the moving object	[68]

The gaze point indicates the exact location where the eyes are looking at. A fixation point is a combination of several gaze points which are very close in time and space. Number of fixations (fixation count), mean fixation

Just Accepted

duration, time to first fixation, hit ratio and fixation rate are some fixation based metrics used in eye-tracking studies. The number of fixations is the total count of fixation points in a given area of interest (AOI). The dwell time, or the total amount of time spent looking within an AOI, determines the number of fixations or fixation count [85]. When humans inspect a scene, they generate a considerable variability in fixation duration, which is the average time for fixations. Time to first fixation is the measure to identify how long it takes a respondent to notice a particular object at first. Hit ratio is the percentage of users whose fixation overlaps within a particular AOI. Fixation rate is the number of fixed gaze points per second. Most of the studies have used either the number of fixations [58, 64] or fixation duration [1, 19, 20, 58, 88]. Apart from the above-mentioned fixation metrics, Krafka et al. [45] have used the average Euclidean distance between the object in the mobile device screen and the true fixation point.

A saccade is a rapid movement of the eyeball between two fixation points, and Holmqvist et al. [34] have considered it as an important attribute to obtain information about viewing behaviour. During a saccadic eye movement, human brain does not process any visual information which become effectively blind, a phenomenon is known as saccadic suppression [9]. The most common saccadic eye metrics are number of saccades, saccade duration (Mean saccade duration), saccade amplitude, and saccadic velocity. Similar to fixation metrics, the number of saccades indicates the total count of saccades in a given AOI and saccade duration specifies the average time for saccades. Saccade amplitude implies the angular distance between two fixation points, and saccadic velocity linearly depends on saccade amplitude. Paletta et al. [68] have used anti-saccade tasks with mobile device eye-tracking to diagnose dementia patients. Subjects were asked to make a saccade in the opposite direction of the moving object during the anti-saccade task.

Eye-tracking data, target visual stimuli, calibration methods, and evaluation metrics significantly impact mobile device eye-tracking studies. We have provided a detailed analysis of the data used in the literature. Also, we have identified the absence of mobile device eye-tracking studies on dynamic visual content like videos and games. Available calibration methods make eye-tracking a challenging process. We have reported these calibration methods in this section. Further, we analysed various evaluation and eye metrics used in the literature.

5 COMPUTATIONAL SYSTEMS FOR MOBILE DEVICE EYE TRACKING

This section describes different types of devices and computational systems that have been used in the retrieved literature. Table 6 displays a summary of different eye-tracking devices and computational systems used by each study. For simplicity, this section is divided into three sub-sections. The different target devices and computational systems that have been used in the literature will be described with their respective screen sizes in sub-section 5.1. The second sub-section will describe how cloud computing can be applied in mobile device eye-tracking studies. Finally, sub-section 5.3 explains how edge computing can be combined with eye tracking to achieve the real-time eye tracking experience. Therefore, this section will answer RQ3 and RQ4.

5.1 Mobile Devices and Computational Systems

In this study, we have considered mobile device eye-tracking applications developed on both smartphones and tablets. We found a wide variety of mobile devices used in the literature, and most of these devices are either android or iOS devices. More than half of the studies have used android phones for their experiments [31, 46, 48, 49, 81, 86]. Researchers have gained lots of advantages when developing their applications on android devices. Android application development can take place in any operating system. For iOS and windows, application development requires Mac OS and Windows OS, respectively. However, Android-based application development is more complex and slower due to the wide range of operating systems compared to other devices. Further, developers need to consider the wide range of screen sizes that are available in android devices.

Just Accepted

Table 6. Different eye tracking devices and computational systems used by each study

Reference	Eye-Tracking device	Mobile Device	Screen Size (in)	Processing Unit	Platform
[99]	Back camera	iphone or ipad	-	CPU	Mobile
[4]	Eye tracking glasses	iPhone 4S	3.5	CPU	Desktop
[58]	FaceLab 5 eye tracker	iPad & LeapFrog LeapPad 2	5	CPU	Desktop
[55]	Front-facing camera	Sony Xperia Z smartphone	5	CPU	Mobile
[101]	Front-facing camera	Android Smartphone	5.5	CPU	Mobile
[92]	Front-facing camera	Huawei Honor V8, Oppo R11, and Samsung Galaxy S5	5.7 & 5.5 & 5.1	CPU	Mobile
[46, 48, 49]	Front-facing camera	Android Smartphone	-	CPU	Mobile
[37, 38]	Front-facing camera	Samsung Galaxy Tab S 10.5 tablet	10.5	CPU	Mobile
[31]	Front-facing camera	Android Tablet	-	CPU	Mobile
[71]	Front-facing camera	iPhone	-	CPU	Mobile
[81]	Front-facing camera	Nexus 4 smartphone	4.7	CPU	Mobile
[59]	Front-facing camera	Nokia N810 smartphone	4.13	CPU	Mobile
[86]	Front-facing camera	Samsung Galaxy S smartphone	4	CPU	Mobile
[41]	Front-facing camera	Smartphone	-	CPU	Mobile
[5, 6, 45, 87]	Front-facing camera	Smartphone and tablets	-	CPU	Mobile
[94]	Front-facing camera	Tablet	11	CPU	Mobile
[67]	front-facing camera and Eye Tribe eye tracker	smartphone	-	CPU	Mobile Edge
[62]	front-facing camera and Tobii Pro Glasses 2 eye tracker	Lenovo ZUK Z2 Android smartphone	5	CPU & GPU	Mobile
[51]	Pupil-Labs eye tracker	Samsung Galaxy Tab SM-T800 tablet	10.5	CPU	Desktop
[69]	SMI Eye-Tracking glasses	Smartphone	-	CPU	Mobile
[64]	SMI Eye-Tracking glasses	Nexus 4 smartphone	4.7	CPU	Desktop
[17]	SMI Eye-Tracking Glasses	Tablet	-	CPU	Desktop
[78]	Tobii X2-60 Eye tracker	iPad	-	CPU	Desktop
[20]	Tobii X2-60 Eye tracker	iPad 2 and iPhone 6s	9.7 & 4.7	CPU	Desktop
[19]	Tobii X2-60 Eye tracker	Nexus 4 smartphone	4.7	CPU	Desktop
[84]	Tobii Eye Tracker 4C	iPad	-	CPU	Desktop
[68]	Tobii EyeX eye tracker	Tablet	-	CPU	Desktop
[1, 88]	Tobii x120 eye tracker	iPad	-	CPU	Desktop
[7]	Tobii x120 eye tracker	Nexus one smartphone	3.7	CPU	Desktop

In addition to the mobile device operating systems, the size of the mobile device screen has a huge impact on the mobile device eye-tracking experiments as the accuracy of the eye-tracking algorithm will always depend on the size of the display. Many eye-tracking studies using smaller screens have resulted in low accuracy compared to those based on bigger screens. Wibirama et al. [93] have used eye-tracking to evaluate the effect of mobile device screen size. They have stated that users get higher fixation duration on smaller screen sizes when playing

Just Accepted

mobile games. In the extracted literature, only a few studies have reported the screen size. Wood et al. [94] have used the largest tablet device with 11 inches in screen size for their experiments. Table 6 shows different screen sizes used by these studies.

Another important factor to consider is the eye-tracking algorithm processing unit. All the studies that have used commercial eye-trackers have used a Central processing unit (CPU) integrated with the device to process the gaze data. All these commercial eye-trackers consist of software and hardware. Software packages process gaze data which are captured from hardware. All the studies that used off-the-shelf cameras to capture the user's gaze data have used CPUs for data processing. Only one study [62] has used both CPU and Graphical processing unit (GPU) as the processing unit for their experiments. According to their study, gaze estimation takes 45 ms per frame on the CPU and only 16 ms on the low-end Adreno 530 GPU on the smartphone.

Eye-tracking data captured from the built-in camera or the commercial eye tracker have been processed in desktop or mobile environments. All the eye-tracking algorithms developed with mobile device built-in cameras have used the same mobile environment to process the eye-tracking data. However, low resource mobile devices are not capable enough to process a significant number of frames in real-time [71]. Nevertheless, most commercial eye trackers need to be connected to a desktop environment to store and process the eye-tracking data. Processing eye-tracking data on low-performance local devices may reduce the usability of the application. The following sections describe different computing architectures and how they can be used in mobile device eye-tracking applications to achieve a better user experience.

5.2 Cloud computing in Eye-Tracking Systems

There are many challenges like a massive number of mobile users, huge data aggregation with local computations, limited research laboratories that prevent mobile device eye tracking from becoming a ubiquitous technology. Future eye-tracking applications will be merged with augmented reality systems, and they need high-performance computational platforms to process the eye-tracking data. Cloud computing can be utilized to solve these problems. Cloud computing allows instant access to a massive amount of eye-tracking data to many distributed mobile users and provides higher computational capacity. Researchers can use cloud services to develop their eye-tracking algorithms within a shorter period.

Dao [16] has leveraged cloud computing systems for real-time eye-tracking analysis. In that study, he has developed three applications for fixation detection, heat map visualization, and eye-tracking data classification. Dao's results show that cloud computing has many advantages compared to single PCs regarding running time, data aggregation and delay time. Therefore, combining eye tracking with cloud computing will allow efficient processing, evaluation and data storage. Designers and developers need to focus on design principles and hardware requirements when integrating eye-tracking applications with cloud computing, For example, running multiple tasks in parallel may help to leverage cloud computing.

According to Dao [16], cloud computing should not be used (i) when the computation requires less time compared to data upload time, (ii) when the number of nodes required for analysis is not too large and (iii) when the cloud requires many communication messages. There is no value in using cloud service when there is a considerable amount of data to be uploaded into the cloud, and computation tasks do not require much time. Also, the response time gets decreased when many communication messages are happening between cloud and eye-tracking applications. The limitations of cloud computing can be solved through emerging edge computing architecture, and the following section will explain how edge computing can be utilized in eye-tracking studies.

5.3 Edge Computing in Eye-Tracking Systems

Edge computing processes data closer to data sources, such as IoT devices or local edge servers. Edge computing provides efficient processing and storage by reducing the amount of data transferred to the cloud. Edge computing

Just Accepted

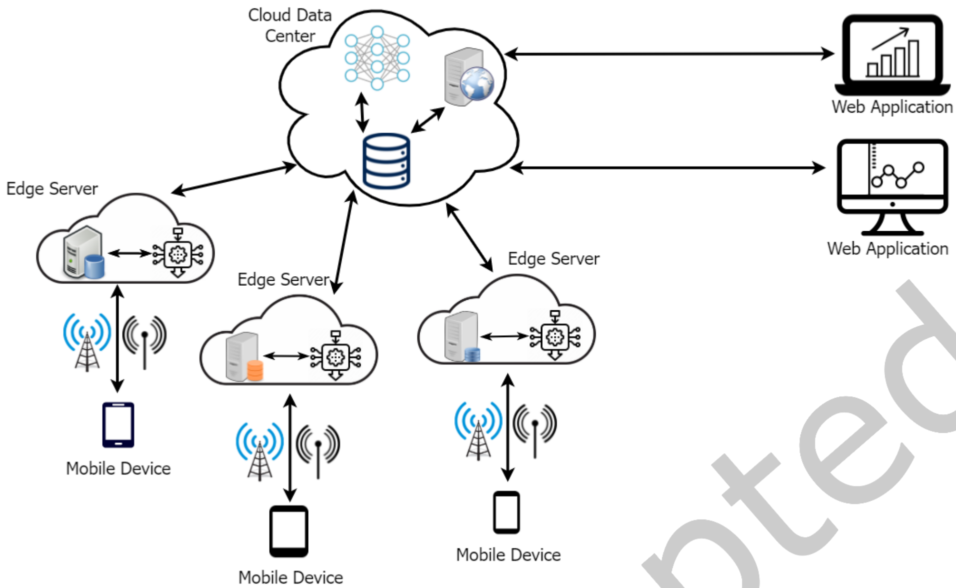


Fig. 6. Proposed mobile edge computing architecture for real-time eye tracking

brings computation closer to the devices where data is gathered rather than depending on a cloud or central data centre, which is thousands of miles away. This makes the system and application more efficient and reduces the flow of traffic to the central server.

Multi-access edge computing or Mobile Edge Computing (MEC) is an extensive mobile and edge computing architecture. According to European Telecommunications Standards Institute (ETSI), MEC is a platform that provides IT and cloud-computing capabilities within the Radio Access Network in 4G and 5G, in close proximity to mobile users [26]. MEC architecture provides computation and data storage close to low energy, and low resource mobile devices [97]. Like edge computing, MEC is located at the edge of the network close to mobile devices.

In the recent past, researchers have achieved higher accuracy in eye-tracking algorithms developed using mobile device built-in cameras [5, 37, 45, 87]. A massive amount of data has been used to develop more robust deep neural networks to obtain the precise gaze direction [11]. Deep neural networks with millions of data will take many hours to train a single machine learning model. A combination of edge computing and artificial intelligence is called edge intelligence [95]. Edge intelligence processes and analyses data locally to reduce the bandwidth and latency. However, edge intelligence with limited computational power in edge nodes will not train and analyze a tremendous amount of data.

According to the literature, only one study has combined edge computing with eye-tracking [67]. This study has used an application server to process the eye-tracking data. Otoom and Alzubaidi [67] were able to reduce the computational requirement of the mobile application by using edge computing. However, they have not utilized cloud computing in their application. High performing cloud computing can be used to perform all the training of the mobile device eye-tracking application.

Figure 6 depicts the proposed mobile edge computing architecture for real-time eye tracking. This architecture consists of a cloud data centre, edge servers, mobile devices and a web application. Deep learning requires

Just Accepted

computationally intensive training and lots of computational power. The cloud data centre contains a machine learning module that trains a deep neural network with more powerful resources. The use of GPU processing power allows deep learning models to scale efficiently and at lower costs by allowing large datasets to be easily ingested and managed for training algorithms. Further, it contains data storage to keep the training data and a webserver to respond to the web application requests. The web application is an interface for technical users to access processed eye-tracking information such as gaze heatmaps and different eye metrics.

The edge server retrieves the trained deep neural network model from the cloud data storage to make the inference at the edge server, and the prediction results will be returned to the device. As the DNN model is hosted on an edge server, porting the application to various mobile platforms is simple. On the other hand, the main disadvantage is that the inference performance is highly dependent on the network bandwidth available between the device and the edge server. The edge server also contains a database server, which is used to store the inference results, along with the data transmitted from mobile devices on the network. The mobile devices are responsible for input data collection from the front camera and sending these data to the edge servers. End users will use the mobile application, and the data will be transmitted to the edge server using either a wireless access point or a base station.

This section described different target devices with their respective screen sizes and computational systems used in the literature. Also, this included how cloud Computing can be applied in eye-tracking studies and how edge computing can be combined with eye tracking to achieve the real-time eye tracking experience.

6 EYE-TRACKING APPLICATIONS ON MOBILE DEVICES

This section explains different types of mobile device eye-tracking applications, domain and limitations of the retrieved literature to answer the RQ5. In the last decade, the development of mobile device eye-tracking applications has been applied to a wide range of domains and applications. Figure 7 depicts different kinds of applications that can be developed with the mobile device eye-tracking technology.

6.1 Mobile user interaction

Mobile user interaction related applications represent a large portion of all available applications in the literature (17 out of the 36 research items). These applications provide hands-free human device interaction, such as mobile application control and peripheral vision-based keyboards. However, 7 out of the 17 mobile user interaction related applications have been developed to detect the most accurate gaze tracking algorithm in different environments. Huang et al. [38] have proposed an algorithm for automatic gaze estimation using multi-level HoG feature and Random Forests regression, and they have achieved a mean error of 3.17cm on the tablet screen. The proposed algorithm of the Wood et al. [94] has estimated gaze with an accuracy of 7 degrees at 12 frames per second. Krafka et al. [45] and Valliappan et al. [87] have utilized a CNN model to achieve a prediction error of 1.71cm and 2.53cm on mobile phones and tablets, respectively, without calibration. With the calibration, the prediction error is reduced to 1.34cm and 2.12cm on mobile phones and tablets, respectively. Until today, Valliappan's CNN based algorithm has achieved the highest accuracy compared to other mobile device eye-tracking studies.

Application controlling is another significant usage in the Mobile user interaction domain. Miluzzo et al. [59] have suggested that gaze direction accuracy degrades if the distance between the eye and mobile device is more significant than 18-20 cm. They have implemented a blink detection algorithm that achieved the highest accuracy of 84%. Kwiatkowska et al. [46] have developed an application controlling mobile device eye-tracking system to conclude that it is possible to control the functions of a mobile phone using only eye movement. Jiang et al. [41], and Paletta et al. [69] have used mobile device eye-tracking to identify human visual attention. Jiang et al. [41] have developed a visual attention localization application using a mobile device eye-tracking called I-VALS. The proposed application only requires the user to gaze at the intended object and hold up the mobile device.

Just Accepted

Then the algorithm will obtain the user's gaze direction and the object's direction from the camera to compute the object's location. Surprisingly, they have obtained the lowest error of 1.3° when users move both the head and eyes and the highest error while users move only the eye (3°). They have proven that both the face pose estimation and iris centre localization are more accurate when the pose is changed, or the length of the eye movement is small and when the head is moved freely. Paletta et al. [69] have developed an application using the SMI Eye Tracking Glasses to map the human visual attention onto the display coordinates, and they have achieved a mean localization error on the display of around 1.5± 0.9mm.

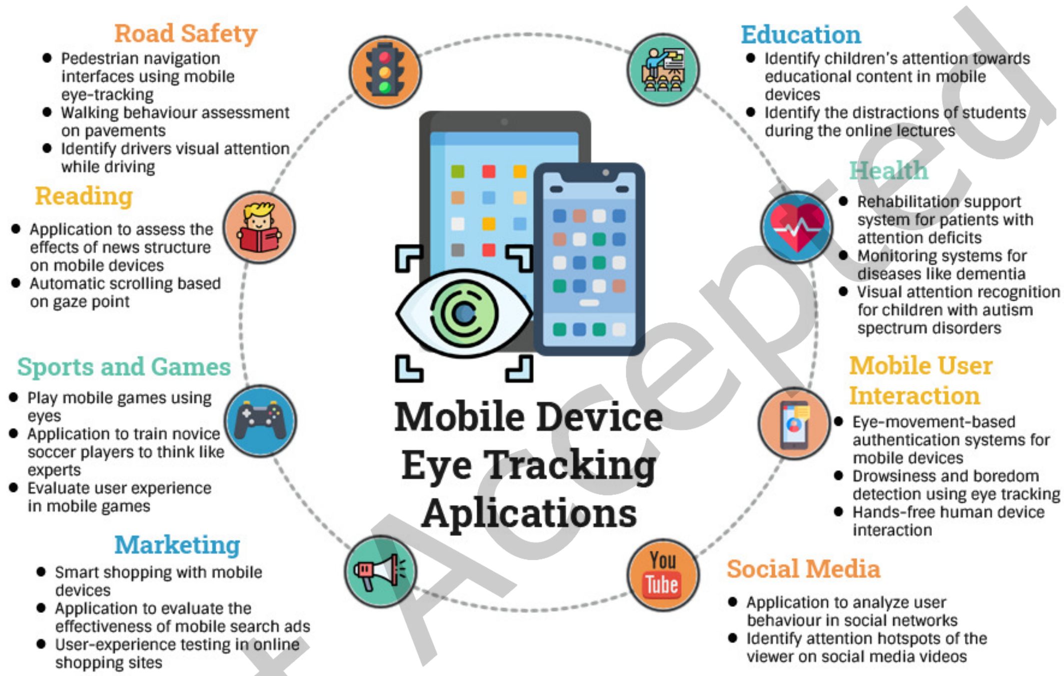


Fig. 7. Different kinds of applications that can be used in mobile device eye-tracking

Just Accepted

User satisfaction is the most important factor to be considered when designing mobile device applications. Regarding accuracy, speed, and inclusive touch interfaces, both the layout of the applications and handedness affect the usability of a mobile device touchscreen. In 2014, Açşi et. al. [4] investigated the different aspects of left or right-handed user experience using a mobile point-of-sale application. They have used mobile device eye-tracking to observe the participants and concluded that the limitation of the touchscreen mobile devices would affect the user experience. Mariakakis et al. [55] have developed an algorithm to track the user's reading pattern and guide them to the last read line of the text when they return from another task. The algorithm proposed by Mariakakis at. al. has improved reading speed by 8% in the presence of distractions. Furthemore, by developing an eye-tracking-based mobile device typing mode, Lu et al. [51] has improved human mobile computer interaction quality.

Widespread mobile devices increase mobile security threats, and traditional user authentication systems may be vulnerable to keylogger attacks, shoulder surfing attacks, and smudge attacks. These attacks infer the user authentication methods using the user's finger movement or tapping input. Dachuan et al. [49] have developed an eye-tracking based user authentication system for smartphones that utilizes the front face camera of the device to capture the user's eye movements. This method needs the user to track moving objects on the screen, and it prevents many attacks like shoulder surfing attacks and smudge attacks. Song et al. [81] have proposed a mobile user authentication system that utilizes signal processing and pattern matching techniques on the data captured from the front-facing camera of the device. They have used different visual stimuli designs such as a rotating circle (RC), an illusion image and a simple dot. They have proved that RC achieves the best accuracy of 88.73% (STD – 3.04%) among these designs. This study has considered the application's memory consumption and, according to the results, the average memory consumption is 41.1MB for RC and 34.9MB for the other two. Inputing a password on the number-only soft keyboard requires that the user's eye movements correspond to the keystrokes typed. A keystroke inference framework has been presented by Want et al. [92] for inferring sensitive inputs on a smartphone using video recordings of the victim's eye patterns collected by the smartphone's front-facing camera. For a single key number, the proposed approach has obtained 77.43% detection accuracy and an 83.33% inference rate. A number-only soft keyboard on a mobile device provides a new route for attackers to extrapolate password keystrokes.

A fingerprint authentication system is a secure method of verifying human identities, especially in employee time attendance systems and secure login access to electronic systems. With the global outbreak of COVID-19, there is a rising question about how to use fingerprint authentication, which has a significantly higher possibility of virus transmission [65]. A human finger contains lots of fat that leave an invisible mark where it touches, and this stain can be made visible in many ways [82]. Therefore, fingerprint-based authentication systems are vulnerable and were violated several years ago. Quintana et al. [**?**] have proposed an authentication method using human eye movements and achieved an acceptance rate of 80%. This indicates that we can use mobile device eye-tracking technology as a human authentication method instead of fingerprint authentication systems.

6.2 Education

Three applications developed through mobile device eye-tracking and related to the news domain have been identified in the literature. The first application was developed in 2015 by Van et al. [88] to structure the news on tablets, and it increases the attention for news and understanding of news, compared to a linear news structure. They have proven that developing news story structure using mobile device eye-tracking leads to a significantly higher Knowledge Structure Density score than the linear news structure. Dunaway et al. [20] have used eye-tracking to capture the human attention for news when reading on a mobile device and proved that mobile applications related news readings have considerably higher attention compared to mobile browser-related news readings. De Haan et al. [17] have conducted a mobile-device eye-tracking study to obtain accurate information on how people use data visualization on e-newspapers.

6.3 Health

Health care applications can be utilized in improving the physical and mental well-being of the people, and they can provide continuous and comprehensive care. The mental health or mental well-being of the community is important for the entire society to thrive. Previous studies have mentioned a significant improvement in mental health monitoring and related treatments when eye-trackers [89, 98]. According to the reported literature, three applications have been developed to cater to different mental health-related problems such as augmentative and alternative communication (AAC) for children with autism spectrum disorders [1], a diagnostic tool for patients with dementia [68], and a rehabilitation support system for patients with attention-deficits [84].

Autism is a neuro-developmental disorder that affects the social interaction and communication of children. Al-Zeer et al. [1] have used eye-tracking data on a mobile device application to identify the relationship between

Just Accepted

fixations and what autistic children are thinking about. They have used a commercial eye tracker to record the user's eye-tracking data and proved that an autistic child reaches the highest fixation duration compared to a healthy child. According to the World Alzheimer Report 2020, over 50 million people are living with dementia, and it is one of the major problems among older people worldwide [22]. Paletta et al. [68] have developed an Alzheimer's Disease diagnostic tool using mobile device eye-tracking and showed that gaze data could be used to classify dementia and healthy participants. Tonomura et al. [84] have implemented a rehabilitation system for patients with attention deficits. Attention disorders can be defined as the inability of a human to control attentional capture and disengagement. These three applications provide a good insight into the mobile device eye-tracking applications that have been developed in the health care domain.

6.4 Traffic

Driver's attention recognition systems and pedestrian navigation support systems are applications developed in the domain of traffic. Visual attention is one of the major factors for traffic crashes [13]. As eye movements and visual attention are closely related, eye movement recordings can identify the driver's visual attention. Nambi et al. [62] have developed a smartphone-based visual attention recognition system for the driver license test. The front camera of the windshield-mounted smartphone is used to capture the driver's gaze direction. Their solution has resulted in higher accuracy when compared to visual attention recognition systems developed using head pose estimation, and it supports 8-10 fps to run driver gaze detection on a smartphone.

Navigation is a task that requires a significant spatial cognition[61]. Ohm et al. [64] have implemented two android prototypes to evaluate the indoor pedestrian navigation systems. One prototype includes a map interface, and the other includes a sketch-like graph interface. According to their findings, participants spent 2.13 seconds on each screen of the map interface compared to 1.03 seconds on the graph UI. When using the graph-like interface, participants spent about 20 seconds less time looking at the screen, with an average of 80.7 seconds for the map interface. Li et al. [48] have proposed a smartphone-based system to detect the walking behaviours of pedestrians. This application utilizes the front-facing camera of the device to track the pedestrian's gaze direction, and if the pedestrian is staring at the screens, the application may provide feedback to the pedestrian to pay attention to the road conditions. The proposed system has a true positive rate of 91%, which eliminates the risk of pedestrians staring at their smartphones while walking.

This section outlined various mobile device eye-tracking applications, domain and limitations of the retrieved literature. Application-oriented classification has been presented in this section, including domain, application and key findings of all the studies which we have selected for this systematic literature review.

7 OPEN ISSUES AND FUTURE DIRECTIONS

From the literature analysis, a number of insights can be obtained regarding the limitations of the mobile device eye-tracking technology and its usability across different domains. As described in section 6, mobile device eye-tracking is adopted in many research fields. It provides limitless opportunities for exploration, and there are a number of issues and challenges that can arise. This section describes the limitations of mobile device eye-tracking, and we are opening new avenues of productive areas for future directions.

- Limitations in commercial eye trackers

Commercial eye trackers prevent mobile device eye tracking from becoming a ubiquitous technology. There is a significant trade-off between the cost and sensitivity of available commercial eye trackers. The most sensitive eye-tracking apparatus costs thousands of dollars and requires considerable competence to manage it properly. The costliness of modern eye-trackers is one of the major issues that we have identified in the literature. As mentioned earlier, there are two types of commercial eye trackers- (i) Screen-based eye trackers and (ii) Eye-tracking glasses. Screen-based eye trackers are attached to the screen and require the user to sit in front of the

Just Accepted

screen. These eye trackers track the eyes only within certain limits, and users cannot freely move their heads. This limited range of head movement is called "headbox". Eye-tracking glasses are fitted close to the eyes, and users are allowed to move their heads freely. However, eye-tracking glasses cannot track the eyes in all situations as they cannot be utilized with prescription eyeglasses. These are the main limitations associated with commercial eye trackers. To overcome these limitations, the research community is trying to develop an eye-tracking technology available to anyone. Researchers have developed eye-tracking systems using built-in cameras on mobile devices without using any external device.

- Unavailability of full face

Capturing the user's entire face while identifying the gaze direction using the built-in cameras is one of the key challenges. According to Khamis et al. [43], the full face is only visible around 30% of the time. Bâce et al. [6] have identified that it is vital to capture the entire face of the user from the front camera of the device while detecting the gaze direction. Even though both eyes are visible, their face detection algorithm fails with partially available faces. Therefore, it is important to note that eyes are not adequate to calculate the gaze angle as mobile devices are held and being looked at in different ways. Another challenge is determining how the head is tilted relative to the camera since the appearance-based gaze estimation algorithms require accurate head pose estimation to train a neural network or data normalization.

- Low quality video capturing and variable lighting conditions

Another drawback of using a built-in camera in the mobile device is the quality of the captured video. For instance, low-resolution cameras and poor lighting conditions may result in low-quality videos [92]. Gaze angle calculation can be divided into model-based or appearance-based [32]. Model-based methods can be further subdivided into corneal-reflection-based methods and shape-based methods. Shape-based methods use eye shapes, such as pupil and iris edges, for gaze estimation. Low-quality video capturing and variable lighting conditions may affect the accuracy of the shape-based gaze estimation. Several studies have used controlled lighting conditions to overcome this issue [55, 67, 86]. Appearance-based methods directly use the eye as an input to train a machine learning model, and Appearance-based methods may work on low-resolution images. Krafka et al. [45], Valliappan et al. [87] and Bâce et al. [5] have used CNNs with a significant number of data to build an appearance-based gaze estimation system using the front camera of the mobile device. However, these studies have proven that appearance-based methods fail to predict the accurate gaze direction with poor lighting conditions. Therefore, significant research needs to be carried out to calculate the precise gaze direction in low-resolution videos and poor lighting conditions.

- Limited battery and computational power

Due to the limited capacity in mobile device battery and other hardware components like CPU and memory, it is crucial to use the optimum resources on mobile devices when developing applications [72]. Eye-tracking approaches, like appearance-based methods, require running many complex algorithms (neural networks), and it costs many computing resources, including CPU, memory and battery power of the mobile devices. Another limitation in the literature is the unavailability of a real-time gaze estimation algorithm. The shortage of computational power in mobile devices is challenging to obtain the real-time eye tracking experience. Utilizing the mobile edge computing architecture and optimizing the device's computational power and memory usage by offloading the computation and data storage to the edge of the network can be considered a solution. According to the literature, there is a considerable research gap in applying mobile edge computing with mobile device eye-tracking applications.

- The impact of telecommunication on edge computing

Edge computing significantly reduces bandwidth consumption, enabling data centres to conserve bandwidth capacity while avoiding costly cloud storage feature upgrades. The most significant advantage of 5G/6G over 4G

Just Accepted

is the increased bandwidth. The availability of 5G/6G networks will aid in overcoming the main disadvantage of using edge computing with mobile device eye tracking, which is that inference performance is dependent on network bandwidth between the device and the edge server. Therefore, 5G/6G technology enables eye tracking users to access streaming services without interruption. While there are numerous advantages to edge computing, telecom can negatively affect the migration of mobile device eye-tracking to edge computing. Edge devices may require additional hardware and software to perform optimally. While reducing the time where data spends in transit mitigates some security risks, it also introduces new and more complex security challenges. Because eye tracking applications for mobile devices collect and process sensitive data, security concerns must be addressed. For instance, if you store or process data on non-controlled end-user devices, it is difficult to ensure that those devices are free of vulnerabilities that attackers could exploit. Even if you use a cloud-edge model that gives you control over the edge infrastructure, adding to the amount of infrastructure you must manage increases your attack surface.

- Attention recognition for dynamic visual stimuli

Human attention may change with the type of visual stimuli presented to the user in the mobile device. For instance, dynamic visual stimuli, like videos and mobile games, require higher attention than static visual stimuli such as web pages and images. However, we have not identified a single study that has attempted to differentiate human attention with different dynamic objects in visual stimuli. Therefore, future studies need to consider how human attention is changed with dynamic visual stimuli.

- Human cognitive state evaluation with eye metrics

Eye-tracking is a process of determining eye movements, but not human thoughts. To evaluate human thoughts, we need to consider eye metrics. Significantly, the user's cognitive state or psychological behaviour can be identified using eye metrics. Only a few studies in the reported literature have used these eye metrics to analyze the user's cognitive behaviour. The research community needs to find out different eye metrics that can assess the user's cognitive state in mobile device eye-tracking applications.

- Calibration free eye tracking

The accuracy of the available eye-tracking algorithms depends on the calibration procedure that they follow. Calibration is a tedious process where the users are required to perform it whenever they are using the eye tracker. During this calibration process, the eye-tracking algorithm builds an anatomical 3D eye model using the geometric properties of the eye, such as the radius of the cornea, the position of the pupil, and the position of the user's head. 3D eye model will be further used to determine the gaze direction. As we mentioned in the section 4.3, there are different ways of calibration, and the point-based calibration procedure is the most common type. A few studies [80, 100] that have used calibration-free gaze tracking with desktop applications. The main advantage of using a calibration-free eye tracking method is that the user is independent of the cumbersome calibration procedure. However, significant research needs to be conducted to build a more accurate calibration-free mobile device eye-tracking algorithm.

8 CONCLUSIONS

Over the last ten years, mobile technology has improved drastically, and without any doubt, the last ten years can be named the "decade of the smartphone". Also, eye tracking is a forthcoming method of collecting data on visual attention and the cognitive process of humans. This article has detailed different mobile device eye-tracking methods, including commercial eye trackers and eye tracking with the mobile device's front-facing camera. Further, this article has provided an in-depth analysis of various eye-tracking approaches used in mobile device eye-tracking studies. We found that majority of the reported studies have used the mobile device's built-in front camera to track the human gaze direction.

Just Accepted

We discussed different visual stimuli used in mobile eye-tracking applications. Most eye-tracking studies have used a static visual stimulus to capture human attention. However, none of these reported studies has used video type visual stimuli on a mobile device to capture human attention using the in-built front camera of the mobile device. We hope, limited computational capabilities in mobile devices are restricted to use with video type content. As a solution, researchers are focusing on edge computing, a distributed computing paradigm that brings computation and data storage closer to the sources of data. We conclude that more research is needed to achieve an acceptable quality of user experience with mobile device eye-tracking and edge computing.

Finally, we overview existing mobile device eye tracking applications in various domains. The lack of movability, expensiveness and tedious calibration process are the major limitations in commercial eye trackers, which prevent the eye-tracking from becoming a pervasive technology that everyone can use in their daily life. Moreover, different lighting conditions and partial face visibility make it very challenging for eye-tracking studies developed on inbuilt cameras of mobile devices to achieve reasonable accuracy. The unavailability of real-time mobile device eye-tracking application and the significant amount of computational resource usage in mobile device applications are other key issue that researchers need to consider in the future. We believe that mobile edge computing might be a better solution for these limitations.

REFERENCES

[1] Shaden Al-Zeer, Amjad Al-Ghanim, and Lyan Al-Wakeel. 2014. Visual attention in interaction with Arabic augmentative and alternative communication apps. In *2014 3rd International Conference on User Science and Engineering (i-USEr)*. IEEE, 210–215.

[2] Bhojan Anand and Pan Wenren. 2017. CloudHide: Towards Latency Hiding Techniques for Thin-client Cloud Gaming. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. 144–152.

[3] Cisco Annual and Internet Report. 2018. *White paper Cisco public.* Technical Report. https://www.cisco.com/c/en/us/solutions/collateral/executiveperspectives/annual-internet-report/white-paper-c11-741490.html.

[4] Sinan Aşçı and Kerem Rızvanoğlu. 2014. Left vs. right-handed UX: A comparative user study on a mobile application with left and right-handed users. In *International Conference of Design, User Experience, and Usability*. Springer, 173–183.

[5] Mihai Bâce, Sander Staal, and Andreas Bulling. 2019. Accurate and Robust Eye Contact Detection During Everyday Mobile Device Interactions. *arXiv preprint arXiv:1907.11115* (2019).

[6] Mihai Bâce, Sander Staal, and Andreas Bulling. 2020. How far are we from quantifying visual attention in mobile HCI? *IEEE Pervasive Computing* 19, 2 (2020), 46–55.

[7] Ralf Biedert, Andreas Dengel, Georg Buscher, and Arman Vartan. 2012. Reading and estimating gaze on smart phones. In *Proceedings of the symposium on eye tracking research and applications*. 385–388.

[8] Pieter Blignaut and Daniël Wium. 2014. Eye-tracking data quality as affected by ethnicity and experimental design. *Behavior Research Methods* 46, 1 (apr 2014), 67–80. https://doi.org/10.3758/s13428-013-0343-0

[9] Bruce Bridgeman, Derek Hendry, and Lawrence Stark. 1975. Failure to detect displacement of the visual world during saccadic eye movements. *Vision research* 15, 6 (1975), 719–722.

[10] John Canny. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698.

[11] Chensi Cao, Feng Liu, Hai Tan, Deshou Song, Wenjie Shu, Weizhong Li, Yiming Zhou, Xiaochen Bo, and Zhi Xie. 2018. Deep learning and its applications in biomedicine. *Genomics, proteomics & bioinformatics* 16, 1 (2018), 17–32.

[12] CB Chen, XL Wu, and Wei Huang. 2009. Fast integral projection algorithm. *Journal of Chinese Computer Systems* 4 (2009), 041.

[13] David Crundall, Claire Shenton, and Geoffrey Underwood. 2004. Eye movements during intentional car following. *Perception* 33, 8 (2004), 975–986.

[14] Joakim Dahlberg and Kenneth Bodin. 2010. *Eye Tracking With Eye Glasses.* Technical Report.

[15] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Vol. 1. IEEE, 886–893.

[16] Thanh-Chung Dao. 2014. Real-time Eye tracking Analysis from Large scale Dataset using Cloud computing. *School of Computing, University of Eastern Finland* (2014).

[17] Yael De Haan, Sanne Kruikemeier, Sophie Lecheler, Gerard Smit, and Renee van der Nat. 2018. When does an infographic say more than a thousand words? Audience evaluations of news visualizations. *Journalism Studies* 19, 9 (2018), 1293–1312.

[18] Jiankang Deng, Yuxiang Zhou, Shiyang Cheng, and Stefanos Zaferiou. 2018. Cascade multi-view hourglass model for robust 3d face alignment. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 399–403.

Just Accepted

[19] Alexa Domachowski, Joachim Griesbaum, and Ben Heuwing. 2016. Perception and effectiveness of search advertising on smartphones. *Proceedings of the Association for Information Science and Technology* 53, 1 (2016), 1–10.

[20] Johanna Dunaway, Kathleen Searles, Mingxiao Sui, and Newly Paul. 2018. News attention in a mobile era. *Journal of Computer-Mediated Communication* 23, 2 (2018), 107–124.

[21] Mohammed E. Fathy, Vishal M. Patel, and Rama Chellappa. 2015. Face-based Active Authentication on mobile devices. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Vol. 2015-Augus. Institute of Electrical and Electronics Engineers Inc., 1687–1691. https://doi.org/10.1109/ICASSP.2015.7178258

[22] Richard Fleming, John Zeisel, and Kirsty Bennett. 2020. World Alzheimer Report 2020 – Design Dignity Dementia: dementia-related design and the built environment, Volume 1. , 248 pages. https://www.alzint.org/resource/world-alzheimer-report-2020/

[23] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.

[24] Alexandra Frischen, Andrew P Bayliss, and Steven P Tipper. 2007. Gaze cueing of attention: visual attention, social cognition, and individual differences. *Psychological bulletin* 133, 4 (2007), 694.

[25] Keinosuke Fukunaga and Larry Hostetler. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory* 21, 1 (1975), 32–40.

[26] Fabio Giust, Gianluca Verin, Kiril Antevski, Joey Chou, Yonggang Fang, Walter Featherstone, Francisco Fontes, Danny Frydman, Alice Li, Antonio Manzalini, et al. 2018. MEC deployments in 4G and evolution towards 5G. *ETSI White Paper* 24 (2018), 1–24.

[27] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. 2008. Multi-PIE. In *2008 8th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2008.* https://doi.org/10.1109/AFGR.2008.4813399

[28] KANNP Gunawardena, RN Rajapakse, and ND Kodikara. 2017. Applying convolutional neural networks for pre-detection of alzheimer's disease from structural MRI data. In *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 1–7.

[29] Nishan Gunawardena, Michael Matscheko, Bernhard Anzengruber, Alois Ferscha, Martin Schobesberger, Andreas Shamiyeh, Bettina Klugsberger, and Peter Solleder. 2019. Assessing surgeons' skill level in laparoscopic cholecystectomy using eye metrics. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications - ETRA '19*. ACM Press, New York, New York, USA, 1–8. https://doi.org/10.1145/3314111.3319832

[30] Eric Hamilton. 2019. What is Edge Computing: The Network Edge Explained. *cloudwards. net. Retrieved* (2019), 05–14.

[31] Seongwon Han, Sungwon Yang, Jihyoung Kim, and Mario Gerla. 2012. EyeGuardian: a framework of eye tracking and blink detection for mobile device users. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. 1–6.

[32] Dan Witzner Hansen and Qiang Ji. 2009. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence* 32, 3 (2009), 478–500.

[33] Katarzyna Harezlak, Pawel Kasprowski, and Mateusz Stasch. 2014. Towards accurate eye tracker calibration -methods and procedures. In *Procedia Computer Science*, Vol. 35. Elsevier B.V., 1073–1081. https://doi.org/10.1016/j.procs.2014.08.194

[34] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. 2011. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford.

[35] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K Jain. 2002. Face detection in color images. *IEEE transactions on pattern analysis and machine intelligence* 24, 5 (2002), 696–706.

[36] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. 2015. Mobile edge computing—A key technology towards 5G. *ETSI white paper* 11, 11 (2015), 1–16.

[37] Qiong Huang, A. Veeraraghavan, and A. Sabharwal. 2015. TabletGaze: Unconstrained Appearance-based Gaze Estimation in Mobile Tablets. *arXiv: Computer Vision and Pattern Recognition* (2015).

[38] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. 2016. TabletGaze: Unconstrained Appearance-based Gaze Estimation in Mobile Tablets. arXiv:1508.01244 [cs.CV]

[39] Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz. 2001. Robust face detection using the hausdorff distance. In *International conference on audio-and video-based biometric person authentication*. Springer, 90–95.

[40] Chi Jian-nan, Zhang Peng-yi, Zheng Si-yi, Zhang Chuang, and Huang Ying. 2009. Key techniques of eye gaze tracking based on pupil corneal reflection. In *2009 WRI Global Congress on Intelligent Systems*, Vol. 2. IEEE, 133–138.

[41] Zhiping Jiang, Kun Zhao, Rui Li, and Jizhong Zhao. 2019. i-VALS: Visual Attention Localization for Mobile Service Computing. *IEEE Access* 7 (2019), 45166–45181.

[73]]Khamis Mohamed Khamis, Anita Baier, Niels Henze, Florian Alt, and Andreas Bulling. [n. d.]. Understanding Face and Eye Visibility in Front-Facing Cameras of Smartphones used in the Wild. ([n. d.]). https://doi.org/10.1145/3173574.3173854

[43] Mohamed Khamis, Anita Baier, Niels Henze, Florian Alt, and Andreas Bulling. 2018. Understanding face and eye visibility in front-facing cameras of smartphones used in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[44] Panos Konstantopoulos, Peter Chapman, and David Crundall. 2010. Driver's visual attention as a function of driving experience and visibility. Using a driving simulator to explore driversâ€™ eye movements in day, night and rain driving. *Accident Analysis and*

Just Accepted

Prevention 42 (2010), 825–832. https://doi.org/10.1016/j.aap.2009.09.022

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

[46] A. Kwiatkowska and D. Sawicki. 2018. Eye Tracking as a Method of Controlling Applications on Mobile Devices. In *ICETE*.

[47] Xun Li, Pablo J Ortiz, Jeffrey Browne, Diana Franklin, John Y Oliver, Roland Geyer, Yuanyuan Zhou, and Frederic T Chong. 2010. Smartphone evolution and reuse: Establishing a more sustainable model. In *2010 39th International Conference on Parallel Processing Workshops*. IEEE, 476–484.

[48] Yantao Li, Fengtao Xue, Xinqi Fan, Zehui Qu, and Gang Zhou. 2018. Pedestrian walking safety system based on smartphone built-in sensors. *IET Communications* 12, 6 (2018), 751–758.

[49] Dachuan Liu, Bo Dong, Xing Gao, and Haining Wang. 2015. Exploiting eye tracking for smartphone authentication. In *International Conference on Applied Cryptography and Network Security*. Springer, 457–477.

[50] SP Liversedge, Iain D Gilchrist, and S Everling. 2012. *The Oxford Handbook of Eye Movements*. Oxford University Press, United Kingdom. https://doi.org/10.1093/oxfordhb/9780199539789.001.0001

[51] Yiqin Lu, Chun Yu, Shuyi Fan, Xiaojun Bi, and Yuanchun Shi. 2019. Typing on Split Keyboards with Peripheral Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[52] David Manning, Susan Ethell, Tim Donovan, and Trevor Crawford. 2006. How do radiologists do it? The influence of experience and training on searching for chest nodules. *Radiography* 12, 2 (2006), 134–142. https://doi.org/10.1016/j.radi.2005.02.003

[53] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. 2017. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2322–2358.

[54] ME Marcove. 1938. On the motions of the eye in illustration of the uses of the muscles and nerves of the orbit. *American Journal of Ophthalmology* 21, 4 (1938), 456.

[55] Alexander Mariakakis, Mayank Goel, Md Tanvir Islam Aumi, Shwetak N Patel, and Jacob O Wobbrock. 2015. SwitchBack: Using focus and saccade tracking to guide users' attention for mobile task resumption. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2953–2962.

[56] Fabio Martinelli, Fiammetta Marulli, and Francesco Mercaldo. 2017. Evaluating convolutional neural network for effective mobile malware detection. *Procedia computer science* 112 (2017), 2372–2381.

[57] Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405, 2 (1975), 442–451.

[58] Rhonda N McEwen and Adam K Dubé. 2015. Engaging or distracting: Children's tablet computer use in education. *Journal of Educational Technology & Society* 18, 4 (2015), 9.

[59] Emiliano Miluzzo, Tianyu Wang, and Andrew T Campbell. 2010. EyePhone: activating mobile phones with your eyes. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*. 15–20.

[60] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, Prisma Group, et al. 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS med* 6, 7 (2009), e1000097.

[61] Daniel R Montello and Martin Raubal. 2013. Functions and applications of spatial cognition. (2013).

[62] Akshay Uttama Nambi, Ishit Mehta, Anurag Ghosh, Vijay Lingam, and Venkata N Padmanabhan. 2019. ALT: towards automating driver license testing using smartphones. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 29–42.

[63] Brian D Noble, Mahadev Satyanarayanan, Dushyanth Narayanan, James Eric Tilton, Jason Flinn, and Kevin R Walker. 1997. Agile application-aware adaptation for mobility. *ACM SIGOPS Operating Systems Review* 31, 5 (1997), 276–287.

[64] Christina Ohm, Manuel Müller, and Bernd Ludwig. 2017. Evaluating indoor pedestrian navigation interfaces using mobile eye tracking. *Spatial Cognition & Computation* 17, 1-2 (2017), 89–120.

[65] Kenneth Okereafor, Iniobong Ekong, Ini Okon Markson, and Kingsley Enwere. 2020. Fingerprint Biometric System Hygiene and the Risk of COVID-19 Transmission. *JMIR Biomedical Engineering* 5, 1 (sep 2020), e19623. https://doi.org/10.2196/19623

[66] Edgar Osuna, Robert Freund, and Federico Girosit. 1997. Training support vector machines: an application to face detection. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE, 130–136.

[67] Mwaffaq Otoom and Mohammad A Alzubaidi. 2018. Ambient intelligence framework for real-time speech-to-sign translation. *Assistive Technology* 30, 3 (2018), 119–132.

[68] Lucas Paletta, Maria Fellner, Martin Pszeida, Alexander Lerch, Christian Kemp, Lara Pittino, Josef Steiner, Mariella Panagl, and Manuela Künstner. 2018. Playful multimodal training for persons with dementia with executive function based decision support. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. 237–240.

[69] Lucas Paletta, Helmut Neuschmied, Michael Schwarz, Gerald Lodron, Martin Pszeida, Stefan Ladstätter, and Patrick Luley. 2014. Smartphone eye tracking toolbox: accurate gaze recovery on mobile displays. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 367–68.

[73]]Park Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. [n. d.]. *Towards End-to-end Video-based Eye-Tracking*. Technical Report. arXiv:2007.13120v1 https://ait.ethz.ch/projects/2020/EVE

Just Accepted

[71] Carmelo Pino and Isaak Kavasidis. 2012. Improving mobile device interaction by eye tracking analysis. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 1199–1202.

[72] Pijush Kanti Dutta Pramanik, Nilanjan Sinhababu, Bulbul Mukherjee, Sanjeevikumar Padmanaban, Aranyak Maity, Bijoy Kumar Upadhyaya, Jens Bo Holm-Nielsen, and Prasenjit Choudhury. 2019. Power Consumption Analysis, Measurement, Management, and Issues: A State-of-the-Art Review of Smartphone Battery and Energy Usage. *IEEE Access* 7 (2019), 182113–182172.

[73]]Quintana-Nevarez Carlos-Alberto Quintana-Nevárez, Nev Nevárez, Francisco L Opez-Orozco, and Rogelio Florencia-Jú Arez. [n. d.]. *Biometric authentication based on eye movements by using scan-path comparison algorithms*. Technical Report.

[74] Lee Richstone, Michael J. Schwartz, Casey Seideman, Jeffrey Cadeddu, Sandra Marshall, and Louis R. Kavoussi. 2010. Eye metrics as an objective assessment of surgical skill. *Annals of Surgery* 252, 1 (2010), 177–182. https://doi.org/10.1097/SLA.0b013e3181e464fb

[75] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. 1998. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence* 20, 1 (1998), 23–38.

[76] Oraya Sawettanusorn, Y. Senda, S. Kawato, N. Tetsutani, and H. Yamauchi. 2003. Real-time face detection using six-segmented rectangular filter.

[77] Maximilian Schwalm, Andreas Keinath, and Hubert D Zimmer. 2008. Pupillometry as a method for measuring mental workload within a simulated driving task. *Human Factors for assistance and automation* January (2008), 1–13.

[78] Shakila Shayan, Dor Abrahamson, Arthur Bakker, Carolien ACG Duijzer, and Marieke van der Schaaf. 2017. Eye-tracking the emergence of attentional anchors in a mathematics learning tablet activity. In *Eye-tracking technology applications in educational research*. IGI Global, 166–194.

[79] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE internet of things journal* 3, 5 (2016), 637–646.

[80] Sheng-Wen Shih, Yu-Te Wu, and Jin Liu. 2000. A calibration-free gaze tracking technique. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Vol. 4. IEEE, 201–204.

[81] Chen Song, Aosen Wang, Kui Ren, and Wenyao Xu. 2016. Eyeveri: A secure and usable approach for smartphone user authentication. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[82] Antti Stén, Antti Kaseva, and Teemupekka Virtanen. 2003. Fooling fingerprint scanners-biometric vulnerabilities of the precise biometrics 100 SC scanner. In *Proceedings of 4th Australian Information Warfare and IT Security Conference*, Vol. 2003. 333–340.

[83] Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. 2017. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine* 55, 3 (2017), 38–43.

[84] Ryuta Tonomura, Tadamitsu Tadamitsu, Atsushi Manji, Naoyuki Kubota, and Takenori Obo. 2018. Rehabilitation Support System for Attentional Deficits Based on Trail-Making Test. In *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, 1127–1132.

[85] Tom Tullis and Bill Albert. 2013. Chapter 7 - Behavioral and Physiological Metrics. In *Measuring the User Experience (Second Edition)* (second edition ed.), Tom Tullis and Bill Albert (Eds.). Morgan Kaufmann, Boston, 163 – 186. https://doi.org/10.1016/B978-0-12-415781-1.00007-8

[86] Vytautas Vaitukaitis and Andreas Bulling. 2012. Eye gesture recognition on portable devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. 711–714.

[87] Nachiappan Valliappan, Na Dai, Ethan Steinberg, Junfeng He, Kantwon Rogers, Venky Ramachandran, Pingmei Xu, Mina Shojaeizadeh, Li Guo, Kai Kohlhoff, et al. 2020. Accelerating eye movement research via accurate and affordable smartphone eye tracking. *Nature communications* 11, 1 (2020), 1–12.

[88] Anna Van Cauwenberge, Leen d'Haenens, and Hans Beentjes. 2015. How to take advantage of tablet computers: Effects of news structure on recall and comprehension. *Communications* 40, 4 (2015), 425–446.

[89] Mélodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2011. Wearable eye tracking for mental health monitoring. (2011). https://doi.org/10.1016/j.comcom.2011.11.002

[90] Paul Viola and Michael Jones. 2001. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Technical Report.

[91] Paul Viola and Michael J Jones. 2004. Robust real-time face detection. *International journal of computer vision* 57, 2 (2004), 137–154.

[92] Yao Wang, Wandong Cai, Tao Gu, Wei Shao, Ibrahim Khalil, and Xianghua Xu. 2018. GazeRevealer: Inferring password using smartphone front camera. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 254–263.

[93] Sunu Wibirama and Hanung A Nugroho. 2017. Towards understanding addiction factors of mobile devices: An eye tracking study on effect of screen size. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2454–2457.

[94] Erroll Wood and Andreas Bulling. 2014. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 207–210.

[95] Dianlei Xu, Tong Li, Yong Li, Xiang Su, Sasu Tarkoma, Tao Jiang, Jon Crowcroft, and Pan Hui. 2020. Edge Intelligence: Architectures, Challenges, and Applications. *arXiv e-prints* (2020), arXiv–2003.

Just Accepted

[96] Alfred L Yarbus. 1967. Eye movements during perception of complex objects. In *Eye movements and vision*. Springer, 171–211.

[97] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. 2019. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture* 98 (2019), 289–330.

[98] Antonia Ypsilanti, Anna Robson, Lambros Lazuras, Philip A. Powell, and Paul G. Overton. 2020. Self-disgust, loneliness and mental health outcomes in older adults: An eye-tracking study. *Journal of Affective Disorders* 266 (apr 2020), 646–654. https://doi.org/10.1016/j.jad.2020.01.166

[99] Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. 2017. Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2878–2889.

[100] Yanxia Zhang, Andreas Bulling, and Hans Gellersen. 2014. Pupil-canthi-ratio: a calibration-free method for tracking horizontal gaze direction. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. 129–132.

[101] Anjie Zhu, Qianjing Wei, Yilin Hu, Zhangwei Zhang, and Shiwei Cheng. 2018. MobiET: a new approach to eye tracking for mobile device. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 862–869.

Just Accepted