Gaze Estimation and Interaction in Real-World Environments

A dissertation submitted towards the degree
Doctor of Engineering
(Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Xucong Zhang, M.Sc.

Saarbrücken
June 2018

Day of Colloquium	26th of September, 2018
Dean of the Faculty	Univ.-Prof. Dr. Sebastian Hack Saarland University, Germany

Examination Committee

Chair	Prof. Dr. Antonio Krüger
Reviewer, Advisor	Prof. Dr. Andreas Bulling
Reviewer	Prof. Dr. Bernt Schiele
Reviewer	Prof. Dr. Elisabeth André
Reviewer	Prof. Dr. Yoichi Sato
Academic Assistant	Dr. Michael Xuelin Huang

ABSTRACT

Human eye gaze has been widely used in human-computer interaction, as it is a promising modality for natural, fast, pervasive, and non-verbal interaction between humans and computers. Gaze information can either be used as explicit input to interactive systems or analysed over time to recognise user activity and cognitive states.

As the foundation of gaze-related interactions, gaze estimation has been a hot research topic in recent decades. Dominant traditional gaze estimation methods require dedicated hardware including high-resolution cameras and additional infrared light sources to detect eye features, such as the pupil centre and iris boundary. Such dedicated hardware results in extra costs that can be very expensive, constrained head movement due to narrow field of view of the camera, limited distance between user and camera to capture high-resolution images, and sensitivity to illumination conditions as sunlight will interfere with infrared light sources. Consequently, most gaze-based interaction researchers conducted their studies in settings in which lighting, user position, etc. can be controlled.

In contrast, appearance-based gaze estimation methods directly regress from eye images to gaze targets without eye feature detection, and therefore have great potential to work with ordinary devices, such as a cellphone, tablet, laptop, or public display. However, these methods require a large amount of domain-specific training data to cover the significant variability in eye appearance. Unfortunately, most of the previous gaze estimation datasets were collected under laboratory conditions, and thus we cannot properly study the capabilities of appearance-based methods for interactive applications in practice.

In this thesis, we focus on developing appearance-based gaze estimation methods and corresponding attentive user interfaces with a single webcam for challenging real-world environments. First, we collect a large-scale gaze estimation dataset, *MPIIGaze*, the first of its kind, outside of controlled laboratory conditions. This dataset enables us, for the first time, to conduct cross-dataset evaluation as training and test on different datasets to study the generalisation capabilities of appearance-based methods in real-world settings. To fully utilise the large amount of data, we the propose *GazeNet* method, which uses convolutional neural networks (CNN) for the first time for the gaze estimation task, and achieves significant improvement over previous state-of-the-art methods.

Second, we propose an appearance-based method that, in stark contrast to a long-standing tradition in gaze estimation, only takes the full face image as input. The face patch has been used in recent work as part of the input. However, it is still an open question whether there is additional information in the other regions of the face, and how to efficiently encode this. We investigate the single full-face approach for 2D and 3D appearance-based gaze estimation and improve the model learning capacity via a novel spatial weights mechanism that efficiently encodes the information of different regions from the face.

Third, we study data normalisation for the first time in a principled way, and propose a modification that yields significant performance improvements. Data normalisation was used successfully in previous works to cancel out the variability in head pose for appearance-based gaze estimation by mapping input images and gaze labels to a normalized space. However, the role and importance of data normalisation remained unclear. Based on visualizations and principled evaluations on both simulated and real data, we first demonstrate the importance of data normalisation for appearance-based gaze estimation. We then propose a modification to the original data normalisation formulation, which performs significantly better.

Fourth, we contribute an unsupervised detector for human-human and human-object eye contact. Eye contact is one of the most important non-verbal social cues and a tool for understanding human social behaviour and cognition. However, existing eye contact detection methods require either specific hardware or human-annotated labels for training samples. We present a novel method for eye contact detection that combines our single webcam gaze estimation method with unsupervised gaze target discovery. The unsupervised learning strategy enables our method to improve detection accuracy the longer it is deployed. Since it requires no prior knowledge except for assuming the target object is closest to the camera, our method can be used for the arbitrary object without any calibration.

Finally, we study personal gaze estimation with multiple personal devices, such as mobile phones, tablets, and laptops. Current gaze estimation methods require a so-called explicit calibration in which users have to iteratively fixate on predefined locations on the device screen, which is both tedious and time-consuming given that it has to be performed on each device separately. Training a generic gaze estimator across devices could be a solution. However, this task poses significant difficulties in handling too much unnecessary ambiguity, due to device-specific camera and screen properties such as image quality and screen resolution. We propose the first method to resolve these issues, with shared feature layers that extract device-independent image information, and encoder/decoder branches that adapt the shared features to device-specific properties. In addition, we demonstrate how our approach can be combined with implicit calibration as personal eye gaze data collection while users naturally interact with their devices, which makes our method a highly practical solution.

ZUSAMMENFASSUNG

Der Blick des menschlichen Auges wird in Mensch-Computer-Interaktionen verbreitet eingesetzt, da dies eine vielversprechende Möglichkeit für natürliche, schnelle, allgegenwärtige und nonverbale Interaktion zwischen Mensch und Computer ist. Blickinformationen können entweder als explizite Eingabe in interaktive Systeme verwendet oder im Laufe der Zeit analysiert werden, um Benutzeraktivität und kognitive Zustände zu erkennen.

Als Grundlage von blickbezogenen Interaktionen ist die Blickschätzung in den letzten Jahrzehnten ein wichtiges Forschungsthema geworden. Die dominierenden traditionellen Blickschätzungsmethoden erfordern spezialisierte Hardware, wie hochauflösende Kameras und zusätzliche Infrarotlichtquellen zur Erkennung von Augenmerkmalen wie Pupillenmitte und Irisgrenze. Eine solche spezialisierte Hardware verursacht zusätzliche Kosten, eine Einschränkung der Kopfbewegungen durch das enge Sichtfeld der Kamera, einen geringen Abstand zwischen Benutzer und Kamera zur Erfassung von hochauflösenden Bildern, und sie reagiert empfindlich auf Beleuchtungsbedingungen, da Sonnenlicht Infrarotlichtquellen stört. Folglich wurden die meisten der blickbasierten Interaktionsstudien in einer Umgebung durchgeführt, in der Beleuchtung, Position des Benutzers etc. kontrolliert werden konnten.

Im Gegensatz dazu schließen Erscheinungsbild-basierte Verfahren zur Blickschätzung direkt von Aufnahmen des Auges auf Blickziele ohne Erkennung von Augenmerkmalen und haben daher großes Potential für die Arbeit mit gewöhnlichen Geräten wie Mobiltelefon, Tablet, Laptop und öffentlichen Displays. Diese Methoden erfordern jedoch eine große Menge an domänenspezifischen Trainingsdaten, um die erheblichen Varianzmöglichkeiten des Erscheinungsbildes des Auges abdecken zu können. Leider wurden die meisten Blickschätzungsdatensätze unter Laborbedingungen gesammelt, daher können wir die Tauglichkeit von Erscheinungsbild-basierten Methoden für interaktive Anwendungen in der Praxis nicht angemessen untersuchen.

In dieser Arbeit konzentrieren wir uns auf die Entwicklung Erscheinungsbild-basierter Methoden zur Blickschätzung und entsprechender "attentive user interfaces" (die Aufmerksamkeit des Benutzers einbeziehende Benutzerschnittstellen) mit nur einer Webcam für anspruchsvolle natürliche Umgebungen. Zunächst sammeln wir einen umfangreichen Datensatz zur Blickschätzung, MPIIGaze, der erste, der außerhalb von kontrollierten Laborbedingungen erstellt wurde. Dieser Datensatz ermöglicht uns erstmals eine datenbankübergreifende Auswertung als Training und Test verschiedener Datensätze, um die Tauglichkeit Erscheinungsbild-basierter Methoden in einer natürlichen Umgebung einschätzen zu können. Um die große Datenmenge in vollem Umfang zu nutzen, schlagen wir die GazeNet-Methode vor; diese verwendet erstmals faltende neuronale Netze (CNN) zur Blickschätzung und

erreicht eine signifikante Verbesserung gegenüber bisherigen Methoden auf dem neuesten Stand der Technik.

Zweitens schlagen wir eine Erscheinungsbild-basierte Methode vor, die im Gegensatz zur langjährigen Tradition in der Blickschätzung nur eine vollständige Aufnahme des Gesichtes als Eingabe verwendet. Gesichtsaufnahmen wurde in aktuellen Arbeiten als Teil der Eingabe zur Kompensation von Kopfhaltungen benutzt. Es ist jedoch noch offen, ob es weitere Informationen in anderen Gesichtsbereichen gibt, und wie man sie effizient kodiert. Wir untersuchen die ‚Single-Full-Face' Herangehensweise für 2D- und 3D- Erscheinungsbild-basierte Blickschätzung und verbessern die Lernfähigkeit des Modells durch einen neuen Ansatz mit räumlicher Gewichtung, der Informationen verschiedener Gesichtsbereiche effizient kodiert.

Drittens untersuchen wir die Datennormalisierung erstmals grundsätzlich und schlagen eine Modifizierung vor, die zu signifikanten Leistungsverbesserungen führt. Die Datennormalisierung wurde in bisherigen Arbeiten erfolgreich eingesetzt, um die Varianzmöglichkeiten von Kopfhaltungen bei der Erscheinungsbild-basierten Blickschätzung auszugleichen, indem Eingabebilder und Blicklabels einem normalisierten Raum zugeordnet wurden. Die Rolle und Bedeutung der Datennormalisierung blieb jedoch unklar. Basierend auf Visualisierungen und grundsätzlicher Auswertung sowohl simulierter als auch natürlicher Daten zeigen wir zunächst die Bedeutung der Datennormalisierung für die Erscheinungsbild-basierte Blickschätzung. Dann schlagen wir eine Modifizierung der ursprünglichen Datennormalisierungsformel vor, die wesentlich besser funktioniert.

Viertens stellen wir einen unüberwachten Detektor für Augenkontakte zwischen Mensch und Mensch und zwischen Mensch und Objekt vor. Augenkontakt ist eines der bedeutendsten nonverbalen sozialen Signale und ein Werkzeug zum Verständnis des sozialen Verhaltens und der Wahrnehmung. Die vorhandenen Augenkontakt-Detektionsmethoden erfordern jedoch entweder spezifische Hardware oder von Menschen annotierte Labels für Trainingsbeispiele. Wir präsentieren eine neuartige Methode zur Augenkontaktdetektion, die unsere Methode zur Blickschätzung mit nur einer Webcam kombiniert mit der unüberwachten Entdeckung von Blickzielen. Die unüberwachte Lernstrategie ermöglicht eine Verbesserung der Detektionsgenauigkeit unserer Methode, je länger sie eingesetzt wird. Da sie keine vorherigen Kenntnisse erfordert - abgesehen von der Annahme, daß das Zielobjekt der Kamera am nächsten ist - kann unsere Methode für jedes beliebige Objekt ohne Kalibrierung verwendet werden.

Abschließend untersuchen wir die persönliche Blickschätzung mit mehreren persönlichen Geräten wie Handy, Tablet und Laptop. Aktuelle Methoden zur Blickschätzung erfordern eine sogenannte explizite Kalibrierung, bei der Benutzer wiederholt vorgegebene Punkte auf dem Bildschirm des Gerätes fixieren müssen, was sowohl mühsam als auch zeitaufwendig ist, da dies an jedem Gerät separat durchgeführt werden muß. Eine intuitive Lösung wäre es, einen generischen Blickschätzer geräteübergreifend zu trainieren. Diese Aufgabe wird jedoch durch unnötige Vieldeutigkeit aufgrund der gerätespezifischen Kamera- und Bildschirmeigenschaften wie Bildqualität und Bildschirmauflösung sehr erschwert. Wir schlagen

vor, daß die erste Methode diese Probleme durch gemeinsame Merkmalsschichten, die geräteunabhängige Bildinformationen extrahieren, löst; Kodierer / Dekodierer passen die gemeinsamen Merkmale den gerätespezifischen Eigenschaften an. Darüber hinaus zeigen wir, wie unser Ansatz mit einer impliziten Kalibrierung als persönliche Blickdatensammlung kombiniert werden kann, während die Benutzer natürlich mit ihren Geräten interagieren, was unsere Methode zu einer sehr praktischen Lösung macht.

CONTENTS

INTRODUCTION 1

Eyes are a remarkable organ for non-verbal communication between humans, which makes eye gaze an important interactive method to sense active attention or passive intention of users. Active attention sensing through eye gaze could replace mouse clicking or finger touch as methods of input to the system (Hutchinson *et al.*, 1989; Kristensson and Vertanen, 2012; Majaranta and Räihä, 2002). Using eye gaze as input is essential for large displays where a mouse is not available (Zhang *et al.*, 2014a), as well as small displays where finger touch would occlude actual targets, such as a mobile phone (Biedert *et al.*, 2012) or smart watch (Esteves *et al.*, 2015). Passive intention can also be measured through eye gaze without explicit action from users. This is crucial for attentive user interfaces to perform interactions according to current user attentional focus and capacity (Bulling, 2016), and also analysis of the effectiveness of visual content (Schrammel *et al.*, 2011). The demand for eye gaze on the part of human-computer interactive systems leads to plentiful studies on gaze estimation methods in the past decades (Hansen and Ji, 2010).

Most current gaze-based interactive works are using commercial eye trackers with the implementation of traditional gaze estimation methods (Guo and Agichtein, 2010; Huang *et al.*, 2012; Zhang *et al.*, 2014a). Unfortunately, traditional, dominant gaze estimation methods require dedicated hardware including high-resolution cameras and infrared light sources (Hennessey *et al.*, 2006; Cristina and Camilleri, 2016). This is caused by explicit eye feature detection, such as the pupil, iris, eye corners, and corneal-reflection detection required by these methods. These detected eye features are used in 3D eyeball model fitting for model-based methods or treated as feature vectors for feature-based methods (Hansen and Ji, 2010). In the ideal case, commercial eye trackers could reach 0.5 to one degree gaze estimation errors under the conditions of indoor setting, constrained head movement, optimal distance between camera and user, and person-specific calibration (Feit *et al.*, 2017). In practice, these additional devices can cause difficulties for positioning at angles and setting up with limited space, such as being in bed, using a bathroom, or travelling in a car (Zhang *et al.*, 2017a).

Just a few interactive systems can work with the a single webcam, with appearance-based methods directly learning a mapping from input eye images to gaze directions (Zhang *et al.*, 2017a). Since there is no explicit eye feature detection, this group of methods can work with low-quality images as captured by RGB cameras readily integrated into billions of portable devices, as well as, increasingly, public interactive and non-interactive displays. Appearance-based methods can achieve a reasonable accuracy of around one to two degrees with evaluations conducted in controlled laboratory settings with fixed head pose using a chicest and good illumination conditions (Tan *et al.*, 2002; Sewell and Komogortsev, 2010). Recent works were proposed to improve the ability of appearance-based methods to handle variant head poses (Lu *et al.*, 2012, 2014a), which were further studied with large-scale datasets and learning-based methods (Sugano *et al.*, 2014; Funes Mora and Odobez, 2014). However, head poses of participants in previous works were performed under instruction rather than by natural movement, and more importantly, illumination conditions were not considered as one of the factors. The lack of data with sufficient variations creates a gap between laboratory study and practical use. Thus, it is not clear how appearance-based gaze estimation methods would perform in challenging real-world settings.

For most gaze estimation methods, domain-specific knowledge is essential for gaze estimation performance, and personal calibration is the typical way to access such knowledge. Commercial gaze trackers usually perform the 9-point personal calibration procedure, i.e. users have to iteratively fixate on predefined locations on the device screen. This procedure requires close concentration that becomes tedious and even annoying. Even worse, users have to repeat the calibration procedure until the system shows good tracking quality, and such frequent re-calibration as three to ten times per day is required (Feit *et al.*, 2017). For appearance-based methods, the learning-based regression models require quite a few personal samples to train, which makes the personal calibration impractical. This has promoted studies on person-independent gaze estimation, i.e. training a generic model on data from multiple devices, which can be directly applied to an unknown user (Sugano *et al.*, 2014; Funes Mora and Odobez, 2014). Nonetheless, person-independent gaze estimation still assumes training and test scenarios come from different environments and devices. It is still a mystery how far we are from gaze estimation that is independent of the user, environment, or camera.

In this thesis, we aim to develop appearance-based gaze estimation methods and associated attentive user interfaces that can work with a single webcam in real-world settings without explicit personal calibration. In the first part of the thesis, we establish the research foundations of unconstrained gaze estimation, i.e. gaze estimation from a monocular RGB camera without assumptions regarding user, environment or device. We take four steps toward this research goal. First, we present a new dataset, *MPIIGaze*, which includes eye gaze data collected from participants' laptops during their daily use through long-term recording. Without constraints on when and where the data collection occurs, we were able to acquire a large amount of data in real-world settings with natural head movement and variant illumination conditions. Second, taking our MPIIGaze as target data under real-world environments, we empirically conduct cross-dataset and cross-personal evaluations to study gaze estimation research in this challenging setting. Although our proposed *GazeNet* method achieves the best performance, it is still a difficult task and open for future research, is gaze estimation in real-world settings across users, environments, and devices in general. This starts the research trend toward taking appearance-based gaze estimation out of the laboratory and being prepared for real-world applications. We further study key challenges including target gaze range, illumination conditions, and facial appearance variation for this challenging task. Third, to extend the input dimensionality to handle high appearance variability, we are first to propose gaze estimation with the single full-face patch as sole input, instead of only eye regions. Our model includes a novel spatial weights mechanism to encode information of different regions of the face for efficient model learning. This is one of the important directions for unconstrained gaze estimation, as rich information hidden in the face patch could provide far more clues than eye images, especially for the low-quality input images from a webcam. Last, during our research on appearance-based gaze estimation, we found that data normalisation (Sugano *et al.*, 2014) is a helpful preprocessing step for handling significant variability in the head pose. However, the importance of rotation and translation/scaling of data normalisation remains unclear and its impact on the gaze estimation performance has not yet been quantified. We first explain the variability caused by head poses and how data normalisation can cancel out some of this variability. Then we demonstrate the importance of data normalisation with extensive experiments on both synthetic and real data.

The second part of the thesis focuses on developing applications of attentive user interfaces based on our gaze estimation methods. The first application we study is eye contact detection, which includes both human-object eye contact and human-human eye contact. To enable the eye contact detector to work with arbitrary objects, we assume that the target object is visually salient and is the closest to the camera. We cluster gaze targets from daily interactions, and pick the one that is nearest to the camera as a positive cluster and others as negative clusters. The samples belonging to these clusters are used for eye contact detector training. Our method can work with arbitrary cameras and objects without the need for tedious and time-consuming manual data annotation. For the second application, we exploit the fact that users usually have multiple devices for gaze estimation to build up a personal gaze estimator with multiple devices. We propose a multi-device CNN gaze estimation method, which includes shared feature extraction layers for the generic gaze estimation task and encoder/decoder branches for device-specific properties, such as screen size and captured image quality. Detailed evaluations on a new dataset of interactions with five common devices demonstrate the significant potential of multi-device training.

1.1 CONTRIBUTIONS OF THE THESIS

The core contribution of the thesis is enabling gaze-based attentive user interfaces in real-world settings, which includes two research areas: *gaze estimation in real-world settings* and *attentive user interfaces*. In the following, we detail the challenges involved in these tasks, as well as the contributions this thesis makes to address them.

1.1.1 Gaze estimation in real-world settings

The first target of the thesis is performing appearance-based gaze estimation with a single webcam in real-world settings without personal calibration, which is essential to gaze-based attentive user interfaces.

1.1.1.1 *Data from real-world settings*

Challenges. Most the previous appearance-based gaze estimation works evaluated their methods with self-collected data, usually collected in controlled settings with a fixed head pose and very good illumination conditions (Tan *et al.*, 2002; Lu *et al.*, 2014b). Recent works on gaze estimation datasets released data that covers variant head poses (Sugano *et al.*, 2014; Funes Mora and Odobez, 2014). However, their collection procedures were artificially designed, as participants' heads moved according to instructions, and data collections were done in indoor laboratory settings. For a single webcam without additional light sources, the illumination condition is an important factor for unconstrained gaze estimation, since it can cause very different image appearances. Besides, head pose movement is another important factor that needs to be considered for evaluating gaze estimation methods in natural human-computer interactive applications. For example, extreme head poses should not be considered equally with frontal head poses since they will not appear frequently during normal interactions.

Contributions. To bring gaze estimation with a single webcam into real-world settings, we introduce the MPIIGaze dataset, which was collected during ordinary laptop daily use though long-term recording as described in Chapter 3. We installed data collection software in participants' laptops, which popped up every 10 minutes and asked for 20 samples. There were 15 participants who collected the data over different time periods ranging from 9 days to 3 months, which resulted in a total of 213,659 images. There was no constraint on where and when the participant should collect data, so that our eye-gaze data covers natural head poses under variant illumination conditions. This dataset is significant since it makes it possible to evaluate appearance-based gaze estimation, for the first time, with data from real-world environments.

1.1.1.2 *Domain-independent gaze estimation*

Challenges. Personal appearance, environments, and devices are three domains for appearance-based gaze estimation. The majority of gaze estimation methods require personal calibration to acquire training data of the same user, under consistent illumination conditions, and with the same camera and display setup (Tan *et al.*, 2002; Sewell and Komogortsev, 2010; Lu *et al.*, 2014b). Person-independent gaze estimation has been proposed recently to train the model across different users (Sugano *et al.*, 2014; Funes Mora and Odobez, 2014). These methods were tested across personal appearances, but not evaluated across different datasets/domains, to properly study their generalisation capabilities. The comparison between person-independent and domain-independent settings has not yet been studied and it is not clear how significant the differences could be. This limitation not only bears the risk of significant dataset bias, but also impedes further progress towards unconstrained gaze estimation.

Contributions. We introduce the task of unconstrained gaze estimation, i.e. gaze estimation from a monocular RGB camera without assumptions regarding the user, environment, or device. In Chapter 3, we make an important step toward unconstrained gaze estimation with cross-dataset and cross-personal evaluations and propose a deep neural network gaze estimation model called *GazeNet*. The cross-dataset evaluation conducts experiments for which all the methods were trained and tested on two different datasets, respectively. It has practical value for showing the performance we can achieve by applying a gaze estimator pre-trained on one data domain to real-world settings. This evaluation is done across all three domains: personal appearance, environments, and devices. To discuss the comparison with person-independent gaze estimation, we then perform a cross-person evaluation using a leave-one-person-out approach within our MPIIGaze dataset. With the environment- and device-specific prior knowledge, performances of gaze estimation methods in this cross-person evaluation are much better than in cross-dataset evaluation. The gap between cross-dataset and cross-person evaluations demonstrates the fundamental difficulty of the unconstrained gaze estimation task compared to the previous person-independent evaluation scheme.

1.1.1.3 *Input space*

Challenges. For gaze estimation research in the past decades, it has been commonly agreed that the eye regions are the only parts of the face image that should be taken as input. This holds for traditional gaze estimation methods with explicit eye feature detection, since an accurate iris boundary is theoretically sufficient to estimate human gaze. Since it is unrealistic to achieve accurate eye feature detection with a single webcam, appearance-based gaze estimation methods estimate gaze from whole input eye images directly. However, eye regions as input space are limited in that previous works have had to struggle with handling additional factors, such as head pose (Lu *et al.*, 2014a, 2015). Although a face patch has been used as one part of the input (Krafka *et al.*, 2016), it is still an open question whether a (more efficient and elegant) face-only approach can work, and which facial regions are the most important for such a full-face appearance-based method.

Contributions. In Chapter 4, we provide a detailed analysis of the potential of the full-face approach for 2D and 3D appearance-based gaze estimation. We present a gaze estimation method that only takes the single face patch as input, without any explicit eye region localisation or head pose estimation. Since different regions of the face should play different roles for the gaze estimation task, we propose a *spatial weights* mechanism to efficiently encode information about different regions of the full face into a standard CNN architecture. Our method achieves significant improvement compared to existing state-of-the-art eye-only (Zhang *et al.*, 2015a) and multi-region (Krafka *et al.*, 2016) approaches. We extend the input space of gaze estimation from eye regions to the full face: such an extension can benefit the appearance-based gaze estimation task especially with data-driven learning-based methods.

1.1.1.4 *Data normalisation*

Challenges. Device-specific gaze estimation can directly predict 2D gaze targets on a screen that does not need cross-device data normalisation (Krafka *et al.*, 2016). For unconstrained gaze estimation, there is a need to unify data from different devices to a normalised 3D camera coordinate system. During such conversion from the original camera coordinate system to the normalised camera coordinate system, variability in head pose and user-camera distance poses significant challenges for training generic gaze estimators. Data normalisation was proposed to cancel out this geometric variability by mapping input images and gaze labels to a normalized space (Sugano *et al.*, 2014). Although used successfully in prior works (Zhang *et al.*, 2015a), the role and importance of data normalisation remain unclear, its impact on the gaze estimation performance has not yet been quantified, and it is not precise enough when dealing with 2D images.

Contributions. In Chapter 5, we explore the importance of data normalisation for unconstrained gaze estimation. We explain the variability caused by different distances between camera and eye and discuss how data normalisation can cancel out some of this variability in principle. We then propose a modification to the original data normalisation formulation to consider only rotation without scaling for 2D images process. To demonstrate the effectiveness of data normalisation on gaze estimation performance, we conduct experiments with both synthetic and real data. With evaluations on synthesised eye images for different head poses, we show the benefit of applying data normalisation in principle. Afterwards, we evaluate within- and cross-dataset settings for gaze estimation and quantify the advantages of data normalisation with respect to performance. Our modified data normalisation shows clear advantages over the original version for all the above experiments.

1.1.1.5 *Data generation*

Challenges. To cover high data variability in unconstrained gaze estimation, data-driven supervised learning methods require time-consuming data collection and accurate ground truth annotation. Capturing input image appearances under all situations is an impossible task; also, the annotation process can be expensive and tedious, while there is no guarantee that human-provided labels will be correct. Recently, a learning-by-synthesis technique has been employed to generate large amounts of training data with computer graphics (Sugano *et al.*, 2014). Due to the dynamic shape changes the eye region undergoes with facial motion and eyeball rotation, and the complex structure of the eyeball, current eye image synthesis is only rendering eye meshes with low-resolution multiple real images, without consideration of modelling illumination changes.

Contributions. We present a novel method for rendering realistic eye-region images with dynamic and controllable 3D eye-region models (Wood *et al.*, 2015). We combine 3D head scan geometry with our own portable eyeball model, which can handle the continuous changes in appearance that the face and eyes undergo during eye movement. We realistically illuminate our eye model using image-based lighting, a technique where high dynamic range (HDR) panoramic images are used to provide light in a scene (Debevec, 2006). As a result, our model can generate realistic eye images for arbitrary head poses, gaze directions and illumination conditions. We show that gaze estimation models can perform better than training with real data by training with large-scale synthesised eye images. Since it is convenient to control data synthesis, we further show that a targeted dataset having the same head pose and gaze ranges as a test scenario can result in significant performance improvements. Among the collected gaze data from real environments, data synthesis is another potential direction to increase model capacity significantly, which was confirmed by works following ours (Wood *et al.*, 2016a; Shrivastava *et al.*, 2017).

1.1.2 Gaze-related attentive user interface

Next, we develop new gaze-related attentive user interfaces in real-world settings. We particularly focus on enabling systems to work with a single webcam with the help of our gaze estimation methods. According to the specific scenario, we propose interactive methods for applications with consideration of the accuracy of gaze estimation methods, target users, and hardware.

1.1.2.1 *Eye contact detection*

Challenges. As an important cue of human attention, eye contact has a wide range of applications in interactive systems (Kleinke, 1986). Here, eye contact detection includes human-object eye contact detection as well as human-human eye contact detection. The former concerns about whether the user is looking at a target object, and the latter whether there is a second person looking at the user's face. Robust detection of eye contact across different users, gaze targets, camera positions, and illumination conditions is notoriously challenging. Although state-of-the-art appearance-based methods have improved in terms of robustness, gaze estimation accuracy is still not sufficient to detect eye contact on small objects in challenging real-world settings. Also, current gaze estimation methods require adaptation via camera-screen calibration as prior geometry information. While gaze estimation can be seen as a regression task to infer arbitrary gaze directions, eye contact detection is a binary classification task to output whether the user is looking at a target or not. However, existing eye contact detection methods require either dedicated hardware (Shell *et al.*, 2004) or high-resolution input images (Smith *et al.*, 2013), and most of them simplify the task as judging whether there is eye contact on an observing camera instead of a true object/face. Specific eye contact detectors can be trained, while on-site data collection is necessary to adapt models to the specific user and relationship between camera and display.

Contributions. In Chapter 6, we address the eye contact detection task by approaching eye contact detection without personal calibration through discovering gaze target distributions. We exploit the fact that visual attention tends to be biased towards the centre of objects and faces, and that the fixation distribution consequently has a centre-surround structure around gaze targets. Thus there will be clear gaze target clusters centred with individual salient objects in front of users. For deployment, we place one camera close to the target object. We pick the cluster nearest to the camera as a positive sample cluster, and the others as negative sample clusters. These samples labels are used to train a specific eye contact detector for the target object. In this way, we transform arbitrary cameras into eye contact sensors that perform accurately when the target is visually salient and the closest to the camera. Unlike previous works, our method can work with arbitrary objects without on-site data collection.

1.1.2.2 *Personal gaze estimation with multiple devices*

Challenges. Gaze estimation methods require person- and device-specific calibration data to achieve practically useful accuracy. This requires an explicit calibration in which users have to iteratively fixate on predefined locations on the device screen. This calibration data is then used to train a person-specific gaze estimator. For most of the previous gaze estimation methods, such personal calibration has to be performed on each device separately, even though they belong to the same user. The key challenge here is diverse frontal cameras and screen sizes of these devices, such as a cellphone, tablet, or laptop. Training a generic gaze estimator across different devices poses a significant difficulty, since the model has to handle the device-specific variations.

Contributions. As described in Chapter 7, we exploit the fact that the personal devices will be mainly used only by one user. Therefore, the data collected from these devices could be used for person-specific gaze estimator training. We design a multi-CNN model with shared feature extraction layers and device-specific encoder/decoder branches. The shared feature extraction layers encode device-independent image information indicative for the gaze estimation task; the encoders and decoders adapt these shared features to the device-specific camera and screen properties, such as image quality and screen resolution. This approach is scalable, as it can use data from an arbitrary number and type of devices a user might own. In addition, we demonstrate how our approach can be combined with implicit calibration into a highly practical solution for person-specific gaze estimation. The implicit calibration collects calibration data during ordinary interaction without additional user effort, which also suffers from unreliable ground-truth gaze labels and low input frequency. Multi-device person-specific gaze estimation can alleviate these issues by using data from other personal devices, and by sharing the learned person-specific features across all devices.

1.1.2.3 *Attention map from multiple users with public display*

Challenges. Passively monitoring attention of multiple users on public display is a specific scenario, in which users can look at the display from arbitrary distances and angles, and also while moving. In this case, ranges of head pose and gaze directions become very large, and it would be unrealistic to hope for any user calibration is. While our works in appearance-based methods promise gaze estimation in real-world settings without personal calibration (Zhang *et al.*, 2015a; Wood *et al.*, 2015), how to transfer a gaze estimator trained in one setting, for example a laptop, to another setting, such as a public display, remains unsolved for interaction research.

Contributions. We present a novel method for estimating audience attention on public displays (Sugano *et al.*, 2016). In this work, we propose two ways to address the limited gaze estimation accuracy of a state-of-the-art appearance-based gaze estimation method for public displays. First, we train a mapping function on top of the gaze estimator to compensate for errors caused by differences in camera angles and display position between training and deployment. This is device-wise error compensation, with calibration data gained by showing visual stimuli on screen. In addition, our method aggregates gaze estimates from different users to compute overall attention distribution, even if these estimates are inaccurate and thus unreliable on their own. This is user-wise error compensation, by assuming users would look at a similar salient region of the screen for the same content. Our method can generate spatio-temporal heatmaps of audience attention, which can be used by content providers to analyse whether the audience is paying attention to the intended on-screen locations.

1.2 OUTLINE OF THE THESIS

In this section we summarise each chapter of the thesis. In addition, we also indicate the respective publications and connections with other previous works.

Chapter 2: Related work. In this chapter, we review the related works on gaze estimation, as well as attentive user interface research. We analyse the relations of previous and subsequent works to the research presented in this thesis.

Chapter 3: Real-World Dataset and Deep Gaze Estimation. This chapter presents our MPIIGaze dataset and exclusive evaluations for the task of unconstrained gaze estimation. We collect the MPIIGaze dataset such that it includes natural head poses under variant illumination conditions. We perform cross-dataset, cross-person evaluations with state-of-the-art gaze estimation methods on three current datasets, including our own MPIIGaze. We study key challenges including target gaze range, illumination conditions, and facial appearance variation. These are important steps toward unconstrained gaze estimation.

The content of this chapter corresponds to the CVPR 2015 publication "Appearance-Based Gaze Estimation in the Wild" (Zhang *et al.*, 2015a) and the TPAMI 2018 publication: "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation" (Zhang *et al.*, 2018c). Xucong Zhang was the lead author of both papers.

Chapter 4: Full-Face Appearance-Based Gaze Estimation. In this chapter, we study full face-patch appearance-based gaze estimation, taking a single face-patch as sole input. We assume there is rich information hidden in the other face regions besides the eyes, and taking a full face-patch as input can enable the model to access the full capacity of the input space. To make the model learning more efficient, we propose a novel spatial weights mechanism encoding information about different regions of the full face into a standard CNN architecture. Our experiments show that simply taking full face patch as input instead of eye images can result in better performance, and our proposed spatial weights further achieve significant improvement over previous state-of-the-art methods.

The content of this chapter corresponds to the CVPRW 2017 publication "It is Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation" (Zhang *et al.*, 2017d). Xucong Zhang was the lead author of this paper.

Chapter 5: Data Normalisation. In this chapter, we visualise and discuss the importance of data normalisation for appearance-based gaze estimation. We explain the variability caused by different distances between camera and eye and discuss how data normalisation can cancel out some of this variability. We then demonstrate the importance of data normalisation for appearance-based gaze estimation with extensive experiments on both synthetic and real data.

The content of this chapter corresponds to the ETRA 2018 publication "Revisiting Data Normalisation for Appearance-Based Gaze Estimation" (Zhang *et al.*, 2018b). Xucong Zhang was the lead author of this paper.

Chapter 6: Unsupervised Eye Contact Detection. This chapter presents our gaze-based interactive application on eye contact detection. We exploit the fact that visual attention tends to have a centre-surround structure around gaze targets toward the target object; thus, we could cluster the gaze targets into several clusters associated with individual objects. By assuming the target object is visually salient and the closest to the camera, our method can automatically acquire training data during deployment, and adaptively learns an eye contact detector.

The content of this chapter corresponds to the UIST 2017 publication "Everyday Eye Contact Detection Using Unsupervised Gaze Target Discovery" (Zhang *et al.*, 2017b), which received a best paper honourable mention award. Xucong Zhang was the lead author of this paper.

Chapter 7: Person-Specific Gaze Estimators with Multiple Devices. In this chapter, we are the first to propose a solution to personal gaze estimation with multiple devices. Our method could use the data from different personal devices to train a gaze estimation model for the specific user. This is due to the device-specific encoders and decoders used to handle device-specific properties, and the shared feature extraction layers encode device-independent image information indicative of the gaze estimation task itself. Our evaluations on the newly collected dataset demonstrate the effectiveness and significant potential of multi-device person-specific gaze estimation.

The content of this chapter corresponds to the CHI 2018 publication "Training Person-Specific Gaze Estimators from Interactions with Multiple Devices" (Zhang *et al.*, 2018a). Xucong Zhang was the lead author.

Chapter 8: Conclusions and Future Prospects. In this chapter we summarise the thesis and discuss possible future research directions for gaze estimation and attentive user interfaces.

PUBLICATIONS

[8] *MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation.*
Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling.
IEEE Trans. on Pattern Analysis and Machine Intelligence (**PAMI**), 2018.

[7] *Revisiting Data Normalization for Appearance-Based Gaze Estimation.*
Xucong Zhang, Yusuke Sugano, and Andreas Bulling.
in Proc. ACM Symp. on Eye Tracking Research and Applications (**ETRA**), 2018.

[6] *Training Person-Specific Gaze Estimators from Interactions with Multiple Devices.*
Xucong Zhang, Yusuke Sugano, and Andreas Bulling.
In Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (**CHI**), 2018.

[5] *Everyday Eye Contact Detection Using Unsupervised Gaze Target Discovery.*
Xucong Zhang, Yusuke Sugano, and Andreas Bulling.
In Proc. ACM Symp. on User Interface Software and Technology (**UIST**), 2017.

[4] *It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation.*
Xucong Zhang, Yusuke Sugano, and Andreas Bulling.
In 1nd International Workshop on Deep Affective Learning and Context Modeling in conjuction with **CVPR**, 2017.

[3] *AggreGaze: Collective Estimation of Audience Attention on Public Displays.*
Yusuke Sugano, Xucong Zhang, and Andreas Bulling.
In Proc. ACM Symp. on User Interface Software and Technology (**UIST**), 2016.

[2] *Rendering of Eyes for Eye-Shape Registration and Gaze Estimation.*
Erroll Wood, Tadas Baltrusaitis, Xucong Zhang Yusuke Sugano, Peter Robinson, and Andreas Bulling.
In Proc. IEEE Int. Conf. on Computer Vision (**ICCV**), 2015.

[1] *Appearance-Based Gaze Estimation in the Wild.*
Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2015.

RELATED WORK 2

Gaze estimation research has a long history and attentive user interfaces have profited from recent development of commercial eye trackers. In this chapter we give an overview of related work, focusing on the directions explored in this thesis.

This chapter is organised as follows. We first make a tour of gaze estimation methods and reveal their limitations in Section 2.1. We then review previous works on attentive user interfaces, specifically on eye contact detection and personal gaze estimation, in Section 2.2.

2.1 GAZE ESTIMATION

In this section, we give an overview of gaze estimation methods research. We first introduce traditional gaze estimation as the main implementation of current commercial eye trackers in Section 2.1.1. Then, we present the recent progress of appearance-based gaze estimation in Section 2.1.2; this has great potential for the unconstrained gaze estimation task in Section 2.1.3, related to the content in Chapter 3. We discuss previous works on gaze estimation datasets in Section 2.1.4, as a comparison with our MPIIGaze dataset discussed in Chapter 3. We clarify significant differences between 2D and 3D gaze estimation and how data normalisation unifies different kinds of data for the unconstrained gaze estimation task in Section 2.1.5, which relates to topics in Chapter 4. We introduce gaze estimation methods with more than just eye regions as input and spatial encoding techniques in CNN research in Section 2.1.6, which inspired our invention of the spatial weights mechanism covered in Chapter 4. Last, we give a brief overview of data normalisation for appearance-based gaze estimation on handling variability in head poses in Section 2.1.7, as referred to in Chapter 5.

2.1.1 Traditional gaze estimation

We can categorise the traditional gaze estimation methods into two groups: model-based and feature-based (Hansen and Ji, 2010). Model-based methods use a geometric 3D eyeball model as the basis for estimating gaze, and the main research content regards how to detect eye features to estimate parameters of this eyeball model according to input 2D images. The iris contour and pupil centre are usually chosen as eye features, and early works usually require dedicated hardware such as high-resolution camera and additional infrared light sources for accurate detection (Hansen and Pece, 2005; Ishikawa *et al.*, 2004). In contrast, recent works made efforts to reduce the hardware requirement to a single camera. Chen and Ji (2008) extended the 3D model to include not only eye feature but also facial feature points which can be tracked with a single camera. Yamazoe *et al.* (2008) used iris segmentation instead of contour detection to reduce the demanding requirement for pixel-level accuracy. As one of the core challenges, Valenti *et al.* (2012) handle the pupil detection task with additional information from the head pose.

Feature-based methods take detected eye feature locations as input for gaze target regression. Depending on the different types of features, there are two main groups of methods: corneal-reflection and pupil centre-eye corner (PC-EC) vector. Corneal-reflection methods rely on eye features detected using reflections of an external infrared light source on the outermost layer of the eye, the cornea. These methods usually take the positions of the corneal-reflection point and pupil centre as feature vectors. Early works on corneal reflection-based methods were limited to stationary settings (Merchant *et al.*, 1974; Morimoto *et al.*, 2002; Shih and Liu, 2004; Yoo and Chung, 2005; Hennessey *et al.*, 2006; Guestrin and Eizenman, 2006) but were later extended to handle arbitrary head poses using multiple light sources or cameras (Zhu and Ji, 2005; Zhu *et al.*, 2006). PC-EC vector methods do not use an infrared light source; they take the eye corners as the replacement, and the distances between pupil centre and two eye corners usually are normalized (Sesma *et al.*, 2012; Bengoechea *et al.*, 2014). Since there is no need for corneal reflection, these methods can work with a single camera.

Although the traditional methods have recently been applied to more practical application scenarios (Jianfeng and Shigang, 2014; Funes Mora and Odobez, 2014; Sun *et al.*, 2014; Wood and Bulling, 2014; Cristina and Camilleri, 2016; Wang and Ji, 2017), their gaze estimation accuracy is still lower, since they depend on accurate eye feature detections, for which high-resolution images and homogeneous illumination are required. These requirements have largely prevented these methods from being widely used in real-world settings or on commodity devices.

2.1.2 Appearance-based gaze estimation

Appearance-based gaze estimation methods do not rely on eye feature point detection, but directly regress from eye images to gaze targets. Because they do not rely on any explicit shape extraction stage, appearance-based methods can handle low-resolution images and long distances. The earliest work on appearance-based gaze estimation took eye images as input for a neural network for a 100-classification task where each class represents one gaze direction (Baluja and Pomerleau, 1994). Tan *et al.* (2002) is the first to propose appearance-based gaze estimation as the title of their paper, which treated calibration samples as points in a manifold and tried to find weights to interpolate the test sample. This work defined the basic form of appearance-based gaze estimation as feature extraction and regression mapping from feature to gaze target. The following works continued to use raw pixel intensities as their input features for Gaussian process regression (Williams *et al.*, 2006; Liang *et al.*, 2013), neural networks (Sewell and Komogortsev, 2010), and adaptive linear regression (Lu *et al.*, 2014b). Due to the high dimensionality of the input eye images, these early appearance-based gaze estimation methods assumed a fixed head pose to reduce the appearance variations.

More recent works started to allow for free 3D head movement and developed different ways to handle ambiguous image appearances caused by it. Lu *et al.* (2014a) learned the gaze bias caused by head pose as compensation the final gaze estimation results. The same authors proposed to generate synthetic eye images for the arbitrary head pose with additional calibration samples (Lu *et al.*, 2012). The Kinect sensor also was used for the appearance-based gaze estimation task, since the depth information is very useful for estimating the head pose (Choi *et al.*, 2013; Gao *et al.*, 2014). Funes Mora and Odobez (2012) used the Kinect as a capturing sensor and eliminated head pose by rotating the face to be frontal. This face frontalisation idea was later extended to normal RGB cameras although accurate facial landmark detection is needed (Jeni and Cohn, 2016).

Despite the progress on appearance-based gaze estimation, these methods require more person-specific training data than traditional approaches to cover the significant variability in eye appearance caused by free head motion. They were therefore mainly evaluated for a specific domain or person. An open research challenge in gaze estimation is to learn gaze estimators that do not make any assumptions regarding the user, environment, or camera.

2.1.3 Unconstrained gaze estimation

As we discussed in Sections 2.1.1 and 2.1.2, the need to collect person-specific training data represents a fundamental limitation for both model-based and appearance-based methods. To reduce the burden on the user, several previous works used interaction events, such as mouse clicks or key presses, as a proxy for users' on-screen gaze position (Sugano *et al.*, 2008; Huang *et al.*, 2014). Alternatively, visual saliency maps (Chen and Ji, 2011; Sugano *et al.*, 2013) or pre-recorded human gaze patterns of the presented visual stimuli (Alnajar *et al.*, 2013) were used as probabilistic training data to learn the gaze estimation function. However, the need to acquire user input fundamentally limits the applicability of these approaches to interactive settings.

Other methods aimed to learn gaze estimators that generalise to arbitrary persons without requiring additional input. A large body of works focused on cross-person evaluations in which the model is trained and tested on data from different groups of participants. For example, Schneider *et al.* (2014) performed a cross-person evaluation on the Columbia dataset (Smith *et al.*, 2013) with 21 gaze points for one frontal head pose of 56 participants. Funes Mora and Odobez (2013) followed a similar approach, but only evaluated on five participants. To reduce data collection and annotation effort, Sugano *et al.* (2014) presented a clustered random forest method that was trained on a large number of synthetic eye images. The images were synthesised from a smaller number of real images captured using a multi-camera setup and controlled lighting in a laboratory. Later works evaluated person-independent gaze estimation methods on the same dataset (Yu *et al.*, 2016; Jeni and Cohn, 2016). Krafka *et al.*

(2016) presented a method for person-independent gaze estimation that achieved 1.71 cm estimation error on an iPhone and 2.53 cm on an iPad screen. However, the method assumed a fixed camera-screen relationship and therefore cannot be used for cross-dataset gaze estimation. Deng and Zhu (2017) recently evaluated their methods in the same way as cross-person evaluation with their dataset including 200 subjects.

Despite significant advances in person-independent gaze estimation, all previous works assumed training and test data to come from the same dataset. We were first to study the practically more relevant, but also significantly more challenging, task of unconstrained gaze estimation via cross-dataset evaluation (Zhang *et al.*, 2015a), which is described in Chapter 3. We introduced a method based on a multimodal deep convolutional neural network that outperformed all state-of-the-art methods by a large margin. More recently, we proposed another method that, in contrast to a long-standing line of work in computer vision, only takes the full face image as input, resulting again in significant performance improvements for both 2D and 3D gaze estimation (Zhang *et al.*, 2017d), with details given in Chapter 4. In later works, Wood et al. demonstrated that large-scale methods for unconstrained gaze estimation could benefit from parallel advances in computer graphics techniques for eye region modelling. These models were used to synthesise large amounts of highly realistic eye region images, thereby significantly reducing both data collection and annotation efforts (Wood *et al.*, 2015). The following model is fully morphable (Wood *et al.*, 2016a) and can synthesise large numbers of images in a few hours on commodity hardware (Wood *et al.*, 2016b). The latest model synthesises realistic images with adversarial training (Shrivastava *et al.*, 2017) or generative models (Wang *et al.*, 2018).

2.1.4 Dataset

To fulfil the need for data-driven methods, several gaze estimation datasets have been published in recent years (see Table 3.1 in Chapter 3 for an overview). Early datasets were severely limited with respect to variability in head poses, on-screen gaze targets, illumination conditions, number of images, face and facial landmark annotations, collection duration per participant, and annotations of 3D gaze directions and head poses (McMurrough *et al.*, 2012; Villanueva *et al.*, 2013; Weidenbacher *et al.*, 2007; Smith *et al.*, 2013).

More recent datasets are larger and cover the head pose and gaze ranges continuously. The OMEG dataset includes 200 image sequences from 50 people with fixed and free head movement but discrete visual targets (He *et al.*, 2015). TabletGaze includes 16 videos recorded from 51 people looking at different points on a tablet screen (Huang *et al.*, 2017). The EYEDIAP dataset contains 94 video sequences of 16 participants who looked at three different targets (discrete and continuous markers displayed on a monitor, and floating physical targets) under both static and free head motion conditions (Funes Mora *et al.*, 2014). The UT Multiview dataset also contains dense gaze samples of 50 participants and 3D reconstructions of eye regions that can be used to synthesise images for arbitrary head poses and gaze targets (Sugano *et al.*, 2014). However, all of these datasets were still recorded under controlled laboratory settings and therefore only include a few illumination conditions. While the recent GazeCapture dataset (Krafka *et al.*, 2016) includes a large number of participants, the limited number of images and similar illumination conditions per participant make it less interesting for unconstrained gaze estimation. Even more importantly, the lack of 3D annotations limits its use to within-dataset evaluations. Several large-scale datasets were published for visual saliency prediction, such as the crowd-sourced iSUN dataset (Xu *et al.*, 2015b), but their focus is on bottom-up saliency prediction, and input face or eye images are not available.

To fulfil unconstrained gaze estimation, a dataset with varying illumination conditions, head poses, gaze directions, and personal appearance is needed. In Chapter 3, we present the MPIIGaze dataset, which contains a large number of images from different participants, covering several months of their daily life.

2.1.5 2D vs. 3D Gaze Estimation

Appearance-based gaze estimation methods can be further categorised depending on whether the regression target is in 2D or 3D. Early works assumed a fixed head pose of the target person (Baluja and Pomerleau, 1994; Tan *et al.*, 2002; Valenti *et al.*, 2012; Williams *et al.*, 2006), and consequently focused on the 2D gaze estimation task where the estimator is trained to output on-screen gaze locations. While more recent methods use 3D head pose (Lu *et al.*, 2015; Sugano *et al.*, 2015) or size and location of the face bounding box (Krafka *et al.*, 2016) to allow for free head movement, they still formulate the task as a direct mapping to 2D on-screen gaze locations. The underlying assumption behind these 2D approaches is that the target screen plane is fixed in the camera coordinate system. Therefore they do not allow for free camera movement after training, which can be a practical limitation, especially for learning-based person-independent estimators.

In contrast, in 3D gaze estimation, the estimator is trained to output 3D gaze directions in the camera coordinate system (Mora and Odobez, 2016; Lu *et al.*, 2014a, 2015; Funes Mora and Odobez, 2013; Wood *et al.*, 2015; Zhang *et al.*, 2015a). The 3D formulation is closely related to pose- and person-independent training approaches and the most important technical challenge is how to efficiently train estimators without requiring too much training data. To facilitate model training, Sugano *et al.* (2014) proposed a data normalisation technique to restrict the appearance variation into a single, normalised training space. Although it required additional technical components, such as 3D head pose estimation, 3D methods have a technical advantage in that they can estimate gaze locations for any target object and camera setup.

As covered in Chapter 4, we are first to perform both 2D and 3D gaze estimation tasks with the same dataset to compare their performances. We also discuss the importance of data normalisation in our work and proposed a modified version, which achieves better performance than the original data normalisation (Zhang *et al.*, 2018b).

2.1.6 Multi-region gaze estimation and spatial encoding

Multi-region gaze estimation. Most previous works used a single eye image as input to the regressor; only a few considered alternative approaches, such as using two images, one of each eye (Huang *et al.*, 2017), or a single image covering both eyes (He *et al.*, 2015; Rikert and Jones, 1998). Krafka *et al.* (2016) recently presented a multi-region 2D gaze estimation method that took individual eye images, the face image, and a face grid as input. Their results suggested that adding the face image can be beneficial for appearance-based gaze estimation.

As discussed in Chapter 4, our work is first to explore the potential of using information on the full face appearance-based gaze estimation. Pushing this idea forward, we further propose the first method that learns a gaze estimator only from the full face image in a truly end-to-end manner.

Spatial encoding. Convolutional neural networks were successful not only for classification (Krizhevsky *et al.*, 2012) but also regression (Simonyan and Zisserman, 2015), including gaze estimation (Zhang *et al.*, 2015a). Several previous works encoded spatial information more efficiently, for example by cropping sub-regions of the image (Girshick, 2015; Jaderberg *et al.*, 2015) or treating different regions on the image equally (He *et al.*, 2014). Tompson *et al.* (2015) used a spatial dropout before the fully connected layer to avoid overfitting during training, but the dropout extended to the entire feature maps instead of one unit.

In Chapter 4, we propose a spatial weights mechanism that encodes the weights for the different regions of the full face, suppresses noise and enhances the contribution from low activation regions. This novel mechanism is mainly inspired by the SpatialDropout in Tompson *et al.* (2015) but has different function and output.

2.1.7 Data normalisation

Sugano et al. proposed a *data normalisation* process to transform eye images and gaze directions into a normalized space to facilitate the synthesis of eye images from a 3D face mesh with arbitrary head poses (Sugano *et al.*, 2014). These synthesised images were then used for gaze estimation. Their basic idea was to rotate and translate the camera to a fixed distance from the eye and to adjust the gaze direction accordingly. Given that images in that normalized space shared the same intrinsic and extrinsic camera parameters, the gaze estimator could be trained and tested in this normalized space. That original data normalisation was successfully used in several subsequent works and was key to facilitate cross-dataset evaluations of appearance-based gaze estimation methods (Zhang *et al.*, 2015a; Wood *et al.*, 2015; Zhang *et al.*, 2017d; Shrivastava *et al.*, 2017). Later works demonstrated that such data normalisation could also be used to adapt gaze estimators trained in one setting to new settings, for example to estimate audience attention on public displays (Sugano *et al.*, 2016); to detect eye contact independent of the target object type and size, camera position, or user (Zhang *et al.*, 2017c; Müller *et al.*, 2018);

or to train person-specific gaze estimators from user interactions across multiple devices (Zhang *et al.*, 2018a). Although data normalisation was successfully used in different prior works, it was mainly used to align the training and test data, and its advantage of making the learning-based approach more efficient has not yet been discussed. Also, a principled comparison of gaze estimation performance with and without data normalisation is still missing from the current literature.

In Chapter 5, we visualise and discuss the importance of data normalisation for appearance-based gaze estimation, and demonstrate the effectiveness of data normalisation with extensive experiments on both synthetic and real data.

2.2 ATTENTIVE USER INTERFACES

Attentive user interfaces are user interface that aim to support users' attentional capacities (Vertegaal and Shell, 2008). One type of attentive user interfaces is gaze-contingent displays that present information at users' focus of attention, achieving maximum information throughput (Bulling, 2016). Eye contact is one of the most efficient ways for an interactive system to detect users' visual attention. Several works demonstrated that when issuing spoken commands, users do indeed look at the individual devices that execute the associated tasks (Maglio *et al.*, 2000a,b; Oh *et al.*, 2002). This means that eye contact sensing can be used to open and close communication channels between users and remote devices, which is a principle known as *Look-to-Talk*. Attentive user interfaces take such information as input to optimise interactions (Horvitz *et al.*, 2003; Vertegaal and Shell, 2008). This requires estimating the users' attention on different objects, but robust gaze estimation, and thus eye contact detection, remains a challenging task, in particular for arbitrary targets in real-world environments. This is the reason that previous works used head orientation as a proxy to detect if the user was looking at an object (Brudy *et al.*, 2014; Smith *et al.*, 2008).

In terms of devices, cameras are being integrated into an ever-increasing number of personal devices, such as mobile phones and laptops. At the same time, gaze estimation methods can be broadly differentiated into model-based and learning-based approaches. Learning-based gaze estimation methods work with ordinary cameras under variable lighting conditions (Zhang *et al.*, 2015a; Mora and Odobez, 2016). Recently, a number of works have explored means to train one generic gaze estimator that can be directly applied to any device and user (Sugano *et al.*, 2014; Zhang *et al.*, 2015a; Krafka *et al.*, 2016; Wood *et al.*, 2015). Taken together, these advances promise to finally enable attentive user interface (Bulling, 2016), eye-based user modelling (Seifert *et al.*, 2017; Huang *et al.*, 2016b), and gaze interaction (Kristensson and Vertanen, 2012; Sugano *et al.*, 2016; Zhang *et al.*, 2017b) on devices that we all use in everyday life. Despite all of these advances in learning-based gaze estimation in recent years, the performance heavily relies on the amount and quality of the training data – which is tedious and time-consuming to collect and annotate. Also, cross-device, cross-person gaze estimation still only achieves a relatively low accuracy of around $7 \sim 10°$ (Zhang *et al.*, 2015a; Shrivastava *et al.*, 2017), and person-specific training data is necessary for good performance of about 3° (Sugano *et al.*, 2014).

In the following, we present the related works corresponding to the two applications in this thesis: eye contact detection in Section 2.2.1 for Chapter 6 and personal gaze estimation with multiple devices in Section 2.2.2 for Chapter 7.

2.2.1 Eye contact detection

Directly using the obtained gaze estimates to detect eye contact on a given target object is challenging for arbitrary camera-target configurations, variable face appearances, and real-world environments. Several previous works investigated dedicated eye contact detection devices and methods. Selker *et al.* (2001) proposed a glasses-mounted eye fixation detector which can also transmit the user ID to the object of interest.

Dedicated eye contact sensors that consisted of a camera and infrared LEDs were proposed that used the light reflection on the eyeball to determine whether the user was looking at the camera (Dickie *et al.*, 2004b; Shell *et al.*, 2004, 2003b; Vertegaal *et al.*, 2002). These approaches were later extended to a wearable setting with a head-mounted eye camera that determined eye contact by observing reflections from infrared LED tags attached to the target objects (Smith *et al.*, 2005). While these device-based approaches can potentially enable robust eye contact detection, the need for target augmentation using dedicated eye contact sensors fundamentally limits their use.

Other works explored learning-based eye contact detection. For example, the GazeLocking method (Smith *et al.*, 2013) followed a classification approach to determine eye contact with a camera. Ye *et al.* (2015) proposed a supervised learning-based approach for eye contact detection from a second-person perspective using wearable cameras. In contrast, Recasens *et al.* (2015, 2016) considered a scenario in which both the person and target objects are present in the image or video, and proposed a CNN-based model to predict the eye contact target. These methods, in essence, share the same limitations as image-based gaze estimation methods, and high performance cannot be achieved without user- or environment-specific training. Another common limitation of these methods is that they assume prior knowledge about the size and location of the target object.

Our unsupervised approach covered in Chapter 6 collects on-site training data for the specific camera-target configuration. Compared to previous works, our method can leverage the increasing number of off-the-shelf cameras readily available – such as those integrated with laptops, placed in the environment, or worn on the body. Our method has no requirement for prior knowledge or human annotation, by assuming the target object is visually salient and the closest to the camera.

2.2.2 User- or device-specific adaptation

Traditional methods for learning-based gaze estimation assumed both user- and device-specific training data (Tan *et al.*, 2002; Williams *et al.*, 2006; Lu *et al.*, 2014a). While they could achieve better performance, it is usually quite impractical to assume large amounts of training data from each target user and device. In the context of learning-based gaze estimation, some methods focused on the cross-person device-specific training task, as we discussed in Section 2.1.3. From a practical point of view, however, a large amount of device-specific training data is still a major requirement for most application scenarios. Sugano *et al.* (2016) proposed an alternative method that combined aggregation of gaze data from multiple users on a public display with an on-site training data collection.

Another challenge of learning-based gaze estimation methods is how to reduce the cost of collecting the required amount of training data. Several previous works investigated the use of saliency maps (Chen and Ji, 2011; Sugano *et al.*, 2013; Sugano and Bulling, 2015) or predicting fixations on images using a regression CNN (Wang *et al.*, 2016). Others proposed to leverage the correlation between gaze and user interactions. Huang *et al.* (2012) explored multiple cursor activities for gaze prediction. Sugano *et al.* (2015) used mouse-clicks to incrementally update the gaze estimator. Papoutsaki *et al.* (2016) developed a browser-based eye tracker that learned from mouse-clicks and mouse movements. Huang *et al.* (2016a) further investigated the temporal and spatial alignments between key presses, mouse-clicks and gaze signals for user-specific gaze learning. Such interaction-based implicit calibration complements the idea of cross-device person-specific gaze estimation, and the most important goal of this work is to investigate our method together with a more realistic assumption of implicit data collection.

In Chapter 7, we focus on the *multi-device person-specific* training task that has not been explored in the gaze estimation literature so far. We further explore training with gaze locations derived from natural interactions, such as mouse or touch input.

Multi-domain learning. Multi-task learning has been researched in the machine learning literature for decades, such as for natural language processing (Collobert and Weston, 2008), speech recognition (Deng *et al.*, 2013), facial landmark detection (Zhang *et al.*, 2014b), or facial expression recognition (Chen *et al.*, 2013). Kaiser *et al.* (2017) used a single model for multiple unrelated tasks by incorporating an encoder and a decoder for each task. While multi-task learning is about solving different tasks using a single model with a shared feature representation, multi-domain learning follows the same approach but with the goal of improving performance on multiple data *domains*. Nam and Han (2016) recently proposed a multi-domain CNN for visual tracking composed of shared layers and multiple branches of domain-specific layers. The multi-device person-specific gaze estimation can also be interpreted as a multi-domain learning task, and therefore the underlying architecture of our method is inspired by recent multi-domain neural networks.

As covered in Chapter 7, we are first to investigate the practical feasibility of a multi-domain approach in the context of multi-device personal gaze estimation. We propose the multi-device CNN to leverage data from other personal devices, and share the learned person-specific features across all devices.

Part I

APPEARANCE-BASED GAZE ESTIMATION IN REAL-WORLD SETTING

Compared to traditional gaze estimation methods, appearance-based gaze estimation methods can work with a single webcam since it directly learns mapping from input image appearances to gaze targets without explicit eye feature detection. These methods have great potential to work with low-quality images as captured by RGB cameras readily integrated into billions of modern devices. As appearance-based gaze estimation is still at an early stage of development, most of the previous works were mainly evaluated for a specific domain or person. An open research challenge in gaze estimation is to learn gaze estimators that do not make any assumptions regarding the user, environment, or camera. In order to address significant variability in eye appearance caused by personal appearances, head poses, gaze directions, illumination conditions etc., there is a need to improve gaze estimation in terms of both dataset and method. In this part, we provide new gaze estimation dataset MPIIGaze as well as novel appearance-based gaze estimation methods.

In Chapter 3 we present our MPIIGaze dataset for gaze estimation task, the first time of its kind, collected in real-world settings. This dataset enables us, for the first time, to conduct the cross-dataset evaluation as training and test on different datasets to study the generalisation capabilities of appearance-based methods in real-world settings. Our proposed CNN-based GazeNet gaze estimation method achieves significant improvements over previous state-of-the-art methods. We then explore the full-face gaze estimation with novel spatial weights mechanism in Chapter 4 in stark contrast to a long-standing tradition in gaze estimation. Through quantitative and qualitative evaluations we show that the proposed techniques facilitate the learning of estimators that are robust to significant variation in illumination conditions as well as head pose and gaze directions available in current datasets. In Chapter 5 we explain and demonstrate the importance of data normalization for appearance-based gaze estimation, and propose a modification to the original data normalization. We then show the effectiveness of data normalisation with extensive experiments on both synthetic and real data.

3 MPIIGAZE: REAL-WORLD DATASET AND DEEP APPEARANCE-BASED GAZE ESTIMATION

LEARNING-BASED methods are believed to work well for unconstrained gaze estimation, i.e. gaze estimation from a monocular RGB camera without assumptions regarding user, environment, or camera. However, current gaze datasets were collected under laboratory conditions and methods were not evaluated across multiple datasets. In this chapter, we make three contributions towards addressing these limitations. First, we present the MPIIGaze dataset, which contains 213,659 full face images and corresponding ground-truth gaze positions collected from 15 users during everyday laptop use over several months. An experience sampling approach ensured continuous gaze and head poses and realistic variation in eye appearance and illumination. To facilitate cross-dataset evaluations, 37,667 images were manually annotated with eye corners, mouth corners, and pupil centres. Second, we present an extensive evaluation of state-of-the-art gaze estimation methods on three current datasets, including MPIIGaze. We study key challenges including target gaze range, illumination conditions, and facial appearance variation. We show that image resolution and the use of both eyes affect gaze estimation performance, while head pose and pupil centre information are less informative. Finally, we propose GazeNet, the first deep appearance-based gaze estimation method. GazeNet improves on the state of the art by 22% (from a mean error of 13.9 degrees to 10.8 degrees) for the most challenging cross-dataset evaluation.

3.1 INTRODUCTION

Gaze estimation is well established as a research topic in computer vision because of its relevance for several applications, such as gaze-based human-computer interaction (Majaranta and Bulling, 2014) or visual attention analysis (Sugano *et al.*, 2016; Sattar *et al.*, 2015). Most recent learning-based methods leverage large amounts of both real and synthetic training data (Funes Mora and Odobez, 2013; Schneider *et al.*, 2014; Sugano *et al.*, 2014; Wood *et al.*, 2016b) for person-independent gaze estimation. They have thus brought us one step closer to the grand vision of *unconstrained gaze estimation*: the 3D gaze estimation in everyday environments and without any assumptions regarding users' facial appearance, geometric properties of the environment and camera, or image formation properties of the camera itself. Unconstrained gaze estimation using monocular RGB cameras is particularly promising given the proliferation of such cameras in portable devices (Wood and Bulling, 2014) and public displays (Zhang *et al.*, 2013).

While learning-based methods have demonstrated their potential for person-

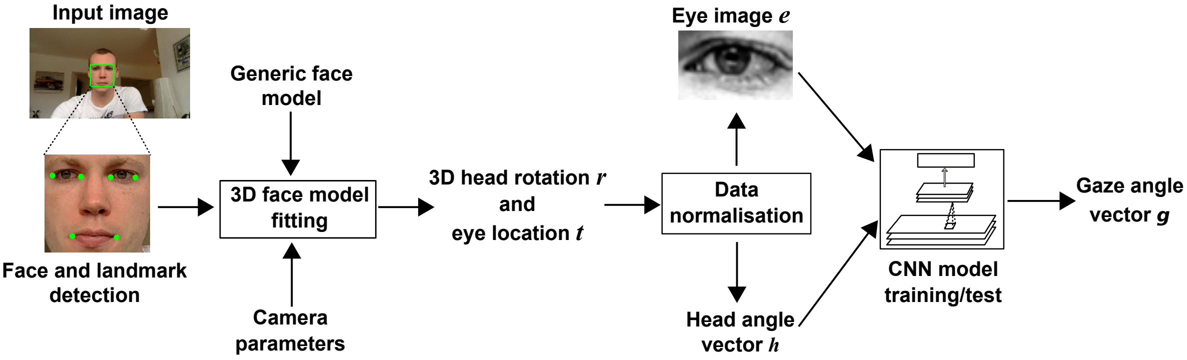


Figure 3.1: Overview of GazeNet– appearance-based gaze estimation using a deep convolutional neural network (CNN).

independent gaze estimation, methods have not been evaluated across different datasets to properly study their generalisation capabilities. In addition, current datasets have been collected under controlled laboratory conditions that are characterised by limited variability in appearance and illumination and the assumption of accurate head pose estimates. These limitations not only bear the risk of significant dataset bias – an important problem also identified in other areas in computer vision, such as object recognition (Torralba and Efros, 2011) or salient object detection (Li *et al.*, 2014). They also impede further progress towards unconstrained gaze estimation, given that it currently remains unclear how state-of-the-art methods perform on real-world images and across multiple datasets.

This chapter aims to shed light on these questions and make the next step towards unconstrained gaze estimation. To facilitate cross-dataset evaluations, we first introduce the MPIIGaze dataset, which contains 213,659 images that we collected from 15 laptop users over several months in their daily life (see Figure 3.2). To ensure frequent sampling during this time period, we opted for an experience sampling approach in which participants were regularly triggered to look at random on-screen positions on their laptop. This way, MPIIGaze not only offers an unprecedented realism in eye appearance and illumination variation but also in personal appearance – properties not available in any existing dataset. Methods for unconstrained gaze estimation have to handle significantly different 3D geometries between user, environment, and camera. To study the importance of such geometry information, we ground-truth annotated 37,667 images with six facial landmarks (eye and mouth corners) and pupil centres. These annotations make the dataset also interesting for closely related computer vision tasks, such as pupil detection. The full dataset including annotations is available at `https://www.mpi-inf.mpg.de/MPIIGaze`.

Second, we conducted an extensive evaluation of several state-of-the-art methods on three current datasets: MPIIGaze, EYEDIAP (Funes Mora *et al.*, 2014), and UT Multiview (Sugano *et al.*, 2014). We include a recent learning-by-synthesis approach that trains the model with synthetic data and fine-tunes it on real data (Wood *et al.*, 2015). We first demonstrate the significant performance gap between previous within- and cross-dataset evaluation conditions. We then analyse various challenges

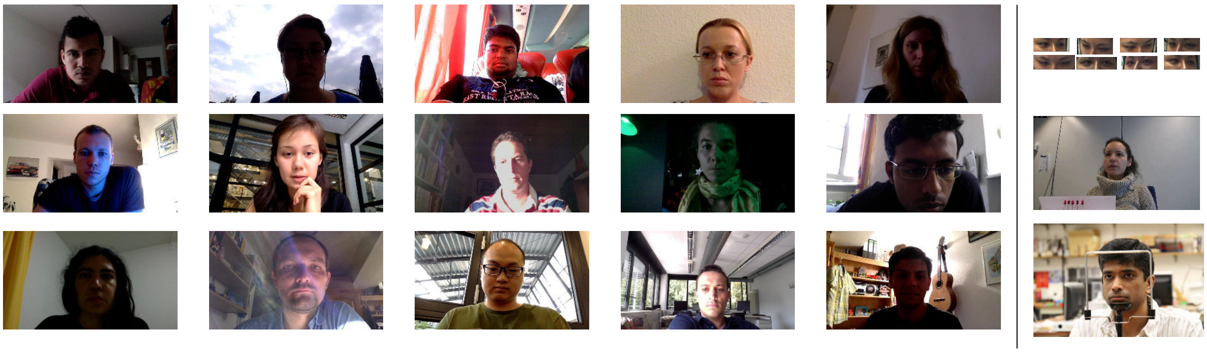


Fig. 3.2: Sample images from the MPIIGaze dataset showing the considerable variability in terms of place and time of recording, eye appearance, and illumination (particularly directional light and shadows). For comparison, the last column shows sample images from other current publicly available datasets (cf. Table 3.1): UT Multiview (Sugano *et al.*, 2014) (top), EYEDIAP (Funes Mora *et al.*, 2014) (middle), and Columbia (Smith *et al.*, 2013) (bottom).

associated with the unconstrained gaze estimation task, including gaze range, illumination conditions, and personal differences. Our experiments show these three factors are responsible for 25%, 35% and 40% performance gap respectively, when extending or restricting the coverage of training data. These analyses reveal that, although largely neglected in previous research, illumination conditions represent an important source of error, comparable to differences in personal appearance.

Finally, we propose GazeNet, the first deep appearance-based gaze estimation method based on a 16-layer VGG deep convolutional neural network. GazeNet outperforms the state of the art by 22% on MPIIGaze and 8% on EYEDIAP for the most difficult cross-dataset evaluation. Our evaluations represent the first account of the state of the art in cross-dataset gaze estimation and, as such, provide valuable insights for future research on this important but so far under-investigated computer vision task.

	Participants	Head poses	Gaze targets	Illuminations	Face anno.	Amount of data	3D anno.
(Villanueva *et al.*, 2013)	103	1	12	1	1,236	1,236	No
(Huang *et al.*, 2017)	51	**C**	35	1	none	1,428 min	No
(Krafka *et al.*, 2016)	1,474	**C**	**C**	**D**	none	2,445,504	No
(Smith *et al.*, 2013)	56	5	21	1	none	5,880	Yes
(McMurrough *et al.*, 2012)	20	1	16	1	none	97 min	Yes
(Weidenbacher *et al.*, 2007)	20	19	2-9	1	2,220	2,220	Yes
(He *et al.*, 2015)	50	3 + **C**	10	1	unknown	333 min	Yes
(Funes Mora *et al.*, 2014)	16	**C**	**C**	2	none	237 min	Yes
(Sugano *et al.*, 2014)	50	8 + synthesised	160	1	64,000	64,000	Yes
MPIIGaze (ours)	**15**	**C**	**C**	**D**	**37,667**	**213,659**	**Yes**

C: continuous
D: daily life

Table 3.1: Overview of publicly available appearance-based gaze estimation datasets showing the number of participants, head poses and on-screen gaze targets (discrete or continuous), illumination conditions, images with annotated face and facial landmarks, amount of data (number of images or duration of video), as well as the availability of 3D annotations of gaze directions and head poses. Datasets suitable for cross-dataset evaluation (i.e. that have 3D annotations) are listed below the double line.

3.2 MPIIGAZE DATASET

To be able to evaluate methods for unconstrained gaze estimation, a dataset with varying illumination conditions, head poses, gaze directions, and personal appearance was needed. To fill this gap, we collected the MPIIGaze dataset that contains a large number of images from different participants, covering several months of their daily life (see Figure 3.2 for sample images from our dataset). The long-term recording resulted in a dataset that is one order of magnitude larger and significantly more variable than existing datasets (cf. Table 3.1). All images in the dataset come with 3D annotations of gaze target and detected eye/head positions, which is required for cross-dataset training and evaluation. Our dataset also provides manual facial landmark annotations on a subset of images, which enables a principled evaluation of gaze estimation performance and makes the dataset useful for other face-related tasks, such as eye or pupil detection.

3.2.1 Collection Procedure

We designed our data collection procedure with two main objectives in mind: 1) to record images of participants outside of controlled laboratory conditions, i.e during their daily routine, and 2) to record participants over several months to cover a wider range of recording locations and times, illuminations, and eye appearances. We opted for recording images on laptops not only because they are suited for long-term daily recordings but also because they are an important platform for eye tracking applications (Majaranta and Bulling, 2014). Laptops are personal devices, therefore typically remaining with a single user, and they are used throughout the day and over long periods of time. Although head pose and gaze range are a bit limited compared to the fully unconstrained case due to the screen size, they have a strong advantage in that the data recording can be carried out in a mobile setup. They also come with high-resolution front-facing cameras and their large screen size allows us to cover a wide range of gaze directions. We further opted to use an experience sampling approach to ensure images were collected regularly throughout the data collection period (Larson and Csikszentmihalyi, 1983).

We implemented custom software running as a background service on participants' laptops, and opted to use the well-established moving dot stimulus(Kassner *et al.*, 2014), to collect ground-truth annotations. Every 10 minutes the software automatically asked participants to look at a random sequence of 20 on-screen positions (a recording session), visualised as a grey circle shrinking in size and with a white dot in the middle. Participants were asked to fixate on these dots and confirm each by pressing the spacebar exactly once when the circle was about to disappear. If they missed this small time window of about 500 ms, the software asked them to record the same on-screen location again right after the failure. While we cannot completely eliminate the possibility of bad ground truth, this approach ensured that participants had to concentrate and look carefully at each point during the recording.

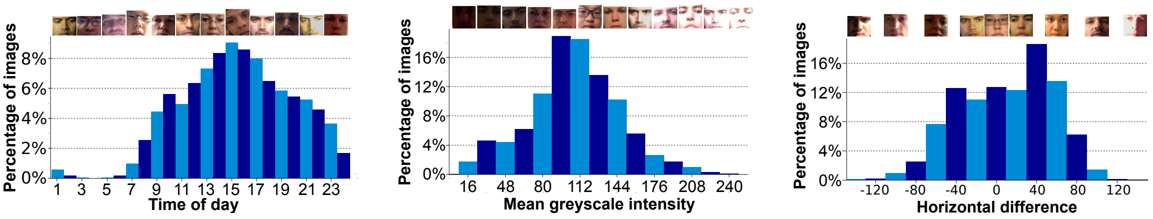


Figure 3.3: Key characteristics of our dataset. Percentage of images collected at different times of day (left), having different mean grey-scale intensities within the face region (middle), and having horizontally different mean grey-scale intensities between the left to right half of the face region (right). Representative sample images are shown at the top.

Otherwise, participants were not constrained in any way, in particular as to how and where they should use their laptops. Because our dataset covers different laptop models with varying screen size and resolution, on-screen gaze positions were converted to 3D positions in the camera coordinate system. We obtained the intrinsic parameters from each camera beforehand using the camera calibration procedure from OpenCV (Bradski, 2000). The 3D position of the screen plane in the camera coordinate system was estimated using a mirror-based calibration method in which the calibration pattern was shown on the screen and reflected to the camera using a mirror (Rodrigues *et al.*, 2010). Both calibrations are required for evaluating gaze estimation methods across different devices. 3D positions of the six facial landmarks were recorded from all participants using an external stereo camera prior to the data collection, which could be used to build the 3D face model.

3.2.2 Dataset Characteristics

We collected a total of 213,659 images from 15 participants (six female, five with glasses) aged between 21 and 35 years. 10 participants had brown, 4 green, and one grey iris colour. Participants collected the data over different time periods ranging from 9 days to 3 months. The number of images collected for each participant varied from 1,498 to 34,745. Note that we only included images in which a face could be detected (see Section 4.1). Figure 3.3 (left) shows a histogram of times of the recording sessions. Although there is a natural bias towards working hours, the figure shows the high variation in recording times. Consequently, our dataset also covers significant variation in illumination. To visualise the different illumination conditions, Figure 3.3 (bottom) shows a histogram of mean grey-scale intensities inside the face region. Figure 3.3 (right) further shows a histogram of the mean intensity differences from the right side to the left side of the face region, indicative of strong directional light for a substantial number of images.

The 2D histograms in Figure 3.4 visualise the distributions of head and gaze angles $\boldsymbol{h}, \boldsymbol{g}$ in the normalised space, colour-coded from blue (minimum) to red

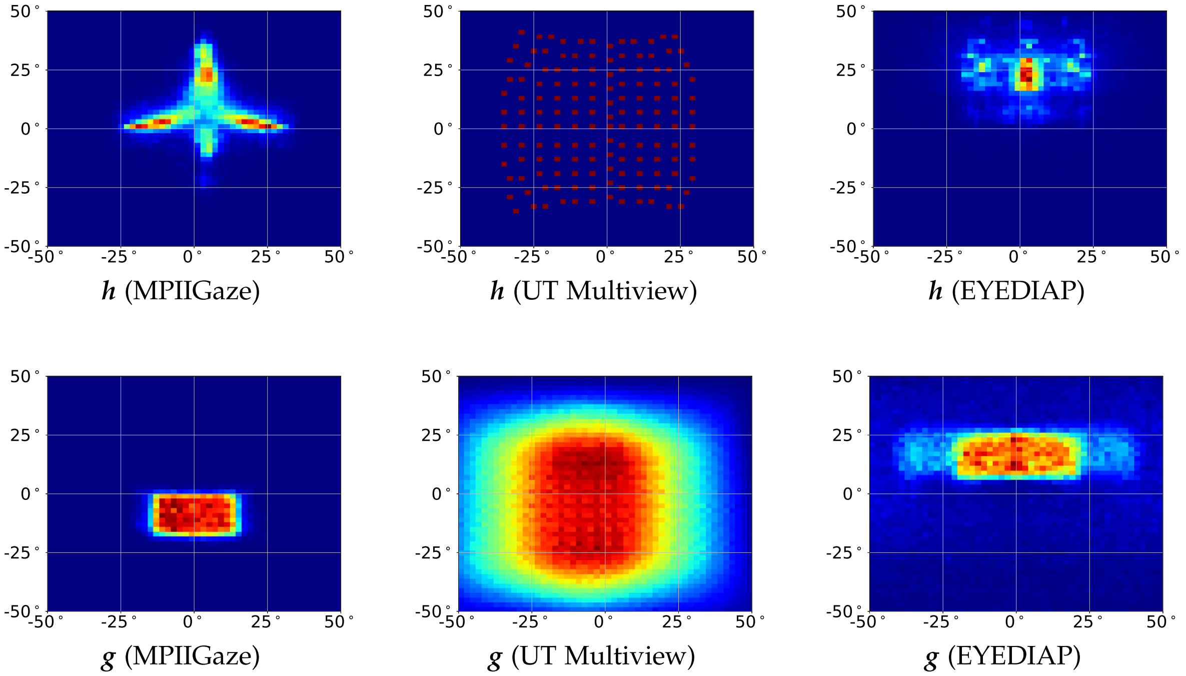
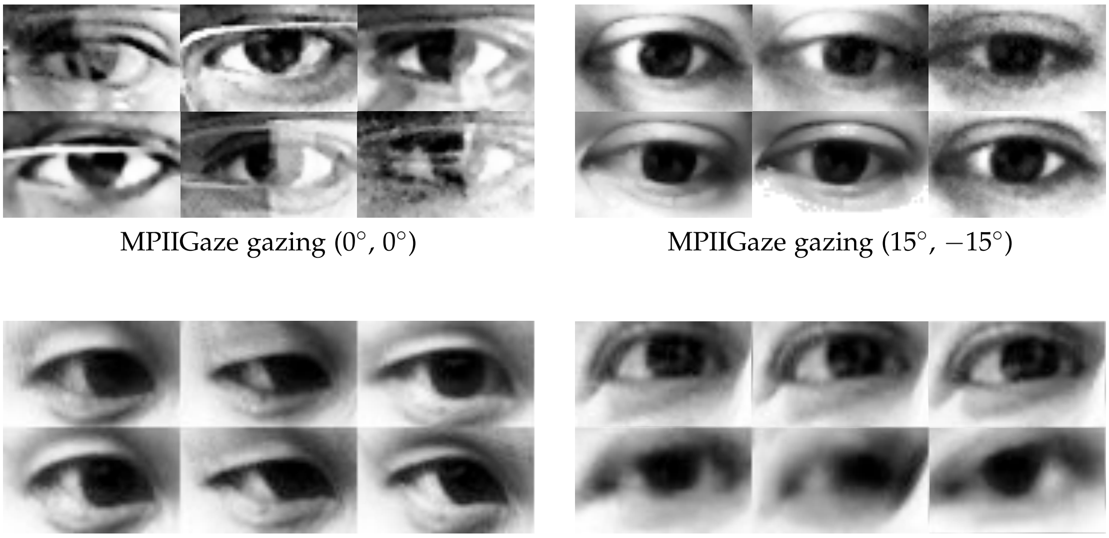


Figure 3.4: Distributions of head angle (***h***) and gaze angle (***g***) in degrees for MPIIGaze, UT Multiview, and the screen target sequences in EYEDIAP (cf. Table 3.1).

(maximum), for MPIIGaze in comparison with two other recent datasets, EYEDIAP (all screen target sequences) (Funes Mora *et al.*, 2014) and UT Multiview (Sugano *et al.*, 2014) (see Section 3.3.2 for a description of the normalisation procedure). The UT Multiview dataset (see Figure 3.4b and Figure 3.4e) was only recorded under a single controlled lighting condition, but provides good coverage of the gaze and pose spaces. For the EYEDIAP dataset, Figure 3.4c and Figure 3.4f show distributions of 2D screen targets that are comparable to our setting, yet gaze angle distributions do not overlap, due to different camera and gaze target plane setups (see Figure 3.4a and Figure 3.4d). For our MPIIGaze dataset, gaze directions tend to be below the horizontal axis in the camera coordinate system because the laptop-integrated cameras were positioned above the screen, and the recording setup biased the head pose to a near-frontal pose. The gaze angles in our dataset are in the range of [-1.5, 20] degrees in the vertical and [-18, +18] degrees in the horizontal direction.

Finally, Figure 3.5 shows sample eye images from each dataset after normalisation. Each group of images was randomly selected from a single person for roughly the same gaze directions. Compared to the UT Multiview and EYEDIAP datasets (see Figure 3.5c and 3.5d), MPIIGaze contains larger appearance variations even inside the eye region (see Figure 3.5b), particularly for participants wearing glasses (see Figure 3.5a).



MPIIGaze gazing (0°, 0°)

MPIIGaze gazing (15°, −15°)

UT Multiview gazing (30°, 5°)

EYEDIAP gazing (25°, 15°)

Figure 3.5: Sample images from a single person for roughly the same gaze directions from MPIIGaze with (a) and without (b) glasses, UT Multiview (c), and EYEDIAP (d).

3.2.3 Facial Landmark Annotation

We manually annotated a subset of images with facial landmarks to be able to evaluate the impact of face alignment errors on gaze estimation performance. To this end, we annotated the evaluation subset used in Zhang *et al.* (2015a) that consists of a randomly-selected 1,500 left eye and 1,500 right eye images of all 15 participants. Because eye images could be selected from the same face, this subset contains a total of 37,667 face images.

The annotation was conducted in a semi-automatic manner. We first applied a state-of-the-art facial landmark detection method (Baltrušaitis *et al.*, 2014), yielding six facial landmarks per face image: the four eye and two mouth corners. We then showed these landmarks to two experienced human annotators and asked them to flag those images that contained incorrect landmark locations or wrong face detections (see Figure 3.6b). 5,630 out of 37,667 images were flagged for manual annotation in this process. Subsequently, landmark locations for all of these images were manually corrected by the same annotators. Since automatic pupil centre localisation remains challenging (Tonsen *et al.*, 2016), we cropped the eye images using the manually-annotated facial landmarks and asked the annotators to annotate the pupil centres (see Figure 3.6c).

Figure 3.7 shows the detection error for facial landmarks and pupil centres when compared to the manual annotation. We calculated the error as the average root-mean-square (RMS) distances between the detected and annotated landmarks per face image. As can be seen from the figure, 85% of the images had no error in the detected facial landmarks. 0.98% of the images had normalised RMS error less

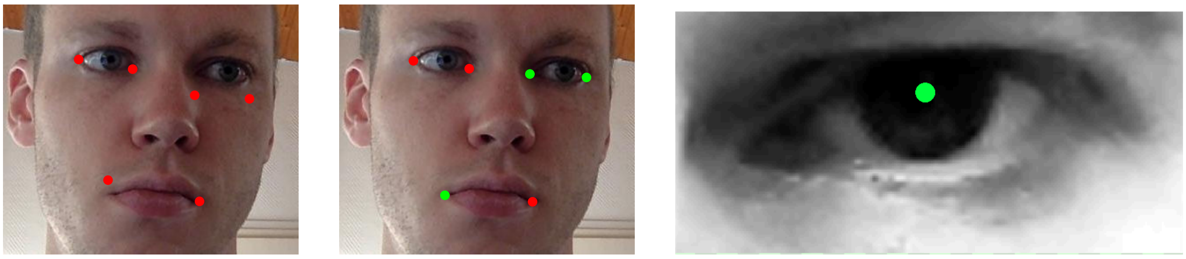


Figure 3.6: We manually annotated 37,667 images with seven facial landmarks: the corners of the left and right eye, the mouth corners, and the pupil centres. We used a semi-automatic annotation approach: (a) Landmarks were first detected automatically (in red) and, (b) if needed, corrected manually post-hoc (in green). We also manually annotated the pupil centre without any detection (c). Note that this is only for completeness and we do not use the pupil centre as input for our method later.

than 0.3. This error roughly corresponds to the size of one eye and indicates that in these cases the face detection method failed to correctly detect the target face. For the pupil centre (red line), the error for each eye image is the RMS between the detected and annotated pupil centre normalised by the distance between both eyes. A normalised RMS error of 0.01 roughly corresponds to the size of the pupil, and 80% of the images had lower pupil detection performance.

3.3 GAZENET

Prior work performed person-independent gaze estimation using 2D regression in the screen coordinate system (Huang *et al.*, 2017; Krafka *et al.*, 2016). Because this requires a fixed position of the camera relative to the screen, these methods are limited to the specific device configuration, i.e. do not directly generalise to other devices. The recent success of deep learning combined with the availability of large-scale datasets, such as MPIIGaze, opens up promising new directions towards unconstrained gaze estimation that was not previously possible. In particular, large-scale methods promise to learn gaze estimators that can handle the significant variability in domain properties as well as user appearance. Figure 3.1 shows an overview of our GazeNet method based on a multimodal convolutional neural network (CNN). We first use state-of-the-art face detection (King, 2009) and facial landmark detection (Baltrušaitis *et al.*, 2014) methods to locate landmarks in the input image obtained from the calibrated monocular RGB camera. We then fit a generic 3D facial shape model to estimate 3D poses of the detected faces and apply the space normalisation technique proposed in Sugano *et al.* (2014) to crop and warp the head pose and eye images to the normalised training space. A CNN is finally used to learn a mapping from the head poses and eye images to 3D gaze directions

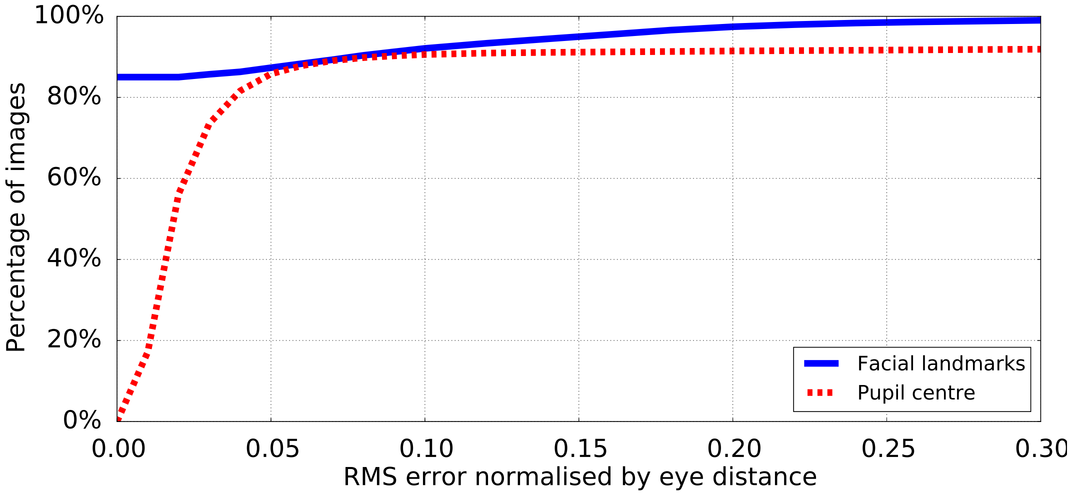


Figure 3.7: Percentage of images for different error levels in the detection of facial landmarks (blue solid line) and pupil centres (red dashed line). The x-axis shows the root-mean-square (RMS) distance between the detected and annotated landmarks, normalised by the distance between both eyes.

in the camera coordinate system.

3.3.1 Face Alignment and 3D Head Pose Estimation

Our method first detects the user's face in the image with a HOG-based method (King, 2009). We assume a single face in the images and take the largest bounding box if the detector returned multiple face proposals. We discard all images in which the detector fails to find any face, which happened in about 5% of all cases. Afterwards, we use a continuous conditional neural fields (CCNF) model framework to detect facial landmarks (Baltrušaitis *et al.*, 2014).

While previous works assumed accurate head poses, we use a generic mean facial shape model $\boldsymbol{F}$ for the 3D pose estimation to evaluate the whole gaze estimation pipeline in a practical setting. The generic mean facial shape $\boldsymbol{F}$ is built as the averaged shape across all the participants, which could also be derived from any other 3D face models. We use the same definition of the face model and head coordinate system as Sugano *et al.* (2014). The face model $\boldsymbol{F}$ consists of 3D positions of six facial landmarks (eye and mouth corners, cf. Figure 3.1). As shown in Figure 3.8, the right-handed head coordinate system is defined according to the triangle connecting three midpoints of the eyes and mouth. The x-axis is defined as the line connecting midpoints of the two eyes in the direction from the right eye to the left eye, and the y-axis is defined to be perpendicular to the x-axis inside the triangle plane in the direction from the eye to the mouth. The z-axis is hence perpendicular to the triangle, and pointing backwards from the face. Obtaining the 3D rotation matrix $\boldsymbol{R}_r$ and translation vector $\boldsymbol{t}_r$ of the face model from the detected 2D facial landmarks $\boldsymbol{p}$ is a classical *Perspective-n-Point*, problem which is estimating the 3D

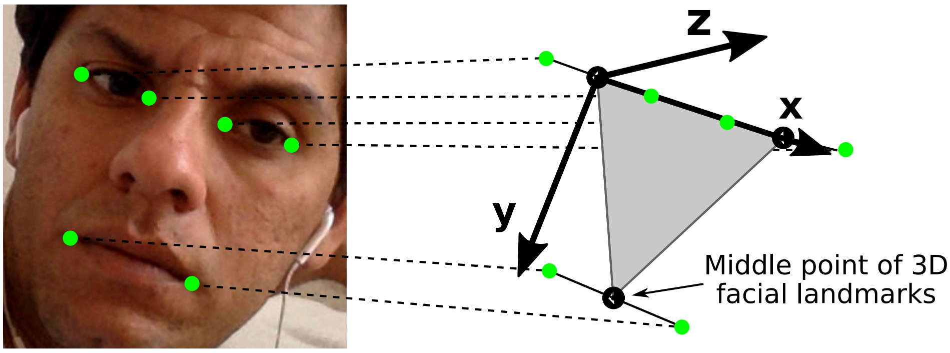


Figure 3.8: Definition of the head coordinate system defined based on the triangle connecting three midpoints of the eyes and mouth. The x-axis goes through the midpoints of both while the y-axis is perpendicular to the x-axis inside the triangle plane. The z-axis is perpendicular to this triangle plane.

pose of an object given its 3D model and the corresponding 2D projections in the image. We fit $\boldsymbol{F}$ to detected facial landmarks by estimating the initial solution using the EPnP algorithm (Lepetit *et al.*, 2009) and further refine the pose by minimising the Levenberg-Marquardt distance.

3.3.2 Eye Image Normalisation

Given that our key interest is in cross-dataset evaluation, we normalise the image and head pose space as introduced in Sugano *et al.* (2014). Fundamentally speaking, object pose has six degrees of freedom, and in the simplest case the gaze estimator has to handle eye appearance changes in this 6D space. However, if we assume that the eye region is planar, arbitrary scaling and rotation of the camera can be compensated for by its corresponding perspective image warping. Therefore, the appearance variation that needs to be handled inside the appearance-based estimation function has only two degrees of freedom. The task of pose-independent appearance-based gaze estimation is to learn the mapping between gaze directions and eye appearances, which cannot be compensated for by virtually rotating and scaling the camera.

The detailed procedure for the eye image normalisation is shown in Figure 3.9. Given the head rotation matrix $\boldsymbol{R}_r$ and the eye position in the camera coordinate system $\boldsymbol{e}_r = \boldsymbol{t}_r + \boldsymbol{e}_h$ where $\boldsymbol{e}_h$ is the position of the midpoint of the two eye corners defined in the head coordinate system (Figure 3.9 (a)), we need to compute the conversion matrix $\boldsymbol{M} = \boldsymbol{S}\boldsymbol{R}$ for normalisation. As illustrated in Figure 3.9 (b), $\boldsymbol{R}$ is the inverse of the rotation matrix that rotates the camera so that the the camera looks at $\boldsymbol{e}_r$ (i.e., the eye position is located along the z-axis of the rotated camera), the x-axis of the head coordinate system is perpendicular to the y-axis of the camera coordinate system. The scaling matrix $\boldsymbol{S} = \mathrm{diag}(1, 1, d_n/\|\boldsymbol{e}_r\|)$ (Figure 3.9 (c)) is then defined so that the eye position $\boldsymbol{e}_r$ is located at a distance d_n from the origin of the scaled camera coordinate system.

$\boldsymbol{M}$ describes a 3D scaling and rotation that brings the eye centre to a fixed position

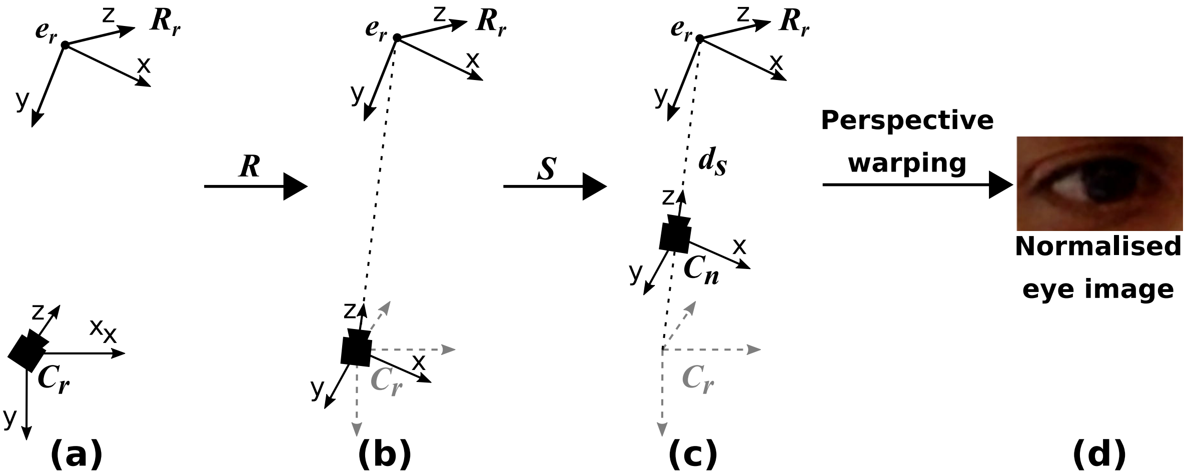


Figure 3.9: Procedure for eye image normalisation. (a) Starting from the head pose coordinate system centred at one of the eye centres $\boldsymbol{e}_r$ (top) and the camera coordinate system (bottom); (b) the camera coordinate system is rotated with $\boldsymbol{R}$; (c) the head pose coordinate system is scaled with matrix $\boldsymbol{S}$; (d) the normalised eye image is cropped from the input image by the image transformation matrix corresponding to these rotations and scaling.

in the (normalised) camera coordinate system, and is used for interconversion of 3D positions between the original and the normalised camera coordinate system. If we denote the original camera projection matrix obtained from camera calibration as $\boldsymbol{C}_r$ and the normalised camera projection matrix as $\boldsymbol{C}_n$, the same conversion can be applied to the original image pixels via perspective warping using the image transformation matrix $\boldsymbol{W} = \boldsymbol{C}_n \boldsymbol{M} \boldsymbol{C}_r^{-1}$ (Figure 3.9 (d)). $\boldsymbol{C}_n = [f_x, 0, c_x; 0, f_y, c_y; 0, 0, 1]$, where f and c indicate the focal length and principal point of the normalised camera, which are arbitrary parameters of the normalised space. The whole normalisation process is applied to both right and left eyes in the same manner, with $\boldsymbol{e}_r$ defined according to the corresponding eye position.

This yields a set of an eye image $\boldsymbol{I}$, a head rotation matrix $\boldsymbol{R}_n = \boldsymbol{M}\boldsymbol{R}_r$, and a gaze angle vector $\boldsymbol{g}_n = \boldsymbol{M}\boldsymbol{g}_r$ in the normalised space. $\boldsymbol{g}_r$ is the 3D gaze vector originating from $\boldsymbol{e}_r$ in the original camera coordinate system. The normalised head rotation matrix $\boldsymbol{R}_n$ is then converted to a three-dimensional rotation angle vector $\boldsymbol{h}_n$. Since rotation around the z-axis is always zero after normalisation, $\boldsymbol{h}_n$ can be represented as a two-dimensional rotation vector (horizontal and vertical orientations) $\boldsymbol{h}$. $\boldsymbol{g}_n$ is also represented as a two-dimensional rotation vector $\boldsymbol{g}$ assuming a unit length. We define d_n to be 600 mm and focal length f_x and f_y of the normalised camera projection matrix $\boldsymbol{C}_n$ to be 960, so that it is compatible with the UT Multiview dataset (Sugano *et al.*, 2014). The resolution of the normalised eye image is set to $\boldsymbol{I}$ in 60×36 pixels, and thus c_x and c_y are set to 30 and 18, respectively. Eye images $\boldsymbol{I}$ are converted to grey scale and histogram-equalised after normalisation to make the normalised eye images compatible between different datasets, facilitating cross-dataset evaluations.

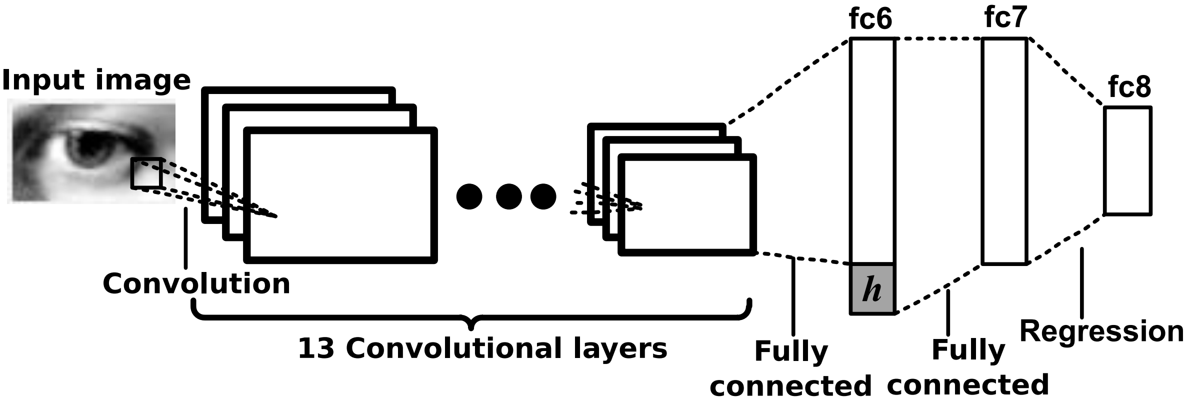


Figure 3.10: Architecture of the proposed GazeNet. The head angle $\boldsymbol{h}$ is injected into the first fully connected layer. The 13 convolutional layers are inherited from a 16-layer VGG network (Simonyan and Zisserman, 2015).

3.3.3 GazeNet Architecture

The task for the CNN is to learn a mapping from the input features (2D head angle $\boldsymbol{h}$ and eye image $\boldsymbol{e}$) to gaze angles $\boldsymbol{g}$ in the normalised space. In the unconstrained setting, the distance to the target gaze plane can vary. The above formulation thus has the advantage that training data does not have to consider the angle of convergence between both eyes. As pointed out in Sugano *et al.* (2014), the difference between the left and right eyes is irrelevant in the person-independent evaluation scenario: By flipping eye images horizontally and mirroring $\boldsymbol{h}$ and $\boldsymbol{g}$ around the horizontal direction, both eyes can be handled using a single regression function.

Our method is based on the 16-layer VGGNet architecture (Simonyan and Zisserman, 2015) that includes 13 convolutional layers, two fully connected layers, and one classification layer with five max pooling layers in between. Following prior work on face (Baltrušaitis *et al.*, 2014; Chen *et al.*, 2014) and gaze (Sugano *et al.*, 2015; Lu *et al.*, 2014b) analysis, we use a grey-scale single channel image as input with a resolution of 60 × 36 pixels. We changed the stride of the first and second pooling layer from two to one to reflect the smaller input resolution. The output of the network is a 2D gaze angle vector $\hat{\boldsymbol{g}}$ consisting of two gaze angles, yaw $\hat{g}_{\phi}$ and pitch $\hat{g}_{\theta}$. We extended the vanilla VGGNet architecture into a multimodal model to also take advantage of head pose information (Ngiam *et al.*, 2011). To this end we injected head pose information $\boldsymbol{h}$ into the first fully connected layer (fc6) (see Figure 3.10). As a loss function we used the sum of the individual L_2 losses measuring the distance between the predicted $\hat{\boldsymbol{g}}$ and true gaze angle vector $\boldsymbol{g}$.

3.4 EXPERIMENTS

We first evaluated GazeNet for cross-dataset and cross-person evaluation. We then explored key challenges in unconstrained gaze estimation including differences in gaze ranges, illumination conditions, and personal appearance. Finally, we studied other closely related topics, such as the influence of image resolution, the use of both eyes, and the use of head pose and pupil centre information on gaze estimation performance. GazeNet was implemented using the Caffe library (Jia *et al.*, 2014). We used the weights of the 16-layer VGGNet (Simonyan and Zisserman, 2015) pre-trained on ImageNet for all our evaluations, and fine-tuned the whole network in 15,000 iterations with a batch size of 256 on the training set. We used the Adam solver (Kingma and Ba, 2015) with the two momentum values set to $\beta_1 = 0.9$ and $\beta_2 = 0.95$. An initial learning rate of 0.00001 was used and multiplied by 0.1 after every 5,000 iterations.

Baseline Methods

We further evaluated the following baseline methods:

- **MnistNet**: The four-layer (two convolutional and two fully connected layers) MnistNet architecture (LeCun *et al.*, 1998) has been used as the first CNN-based method for appearance-based gaze estimation (Zhang *et al.*, 2015a). We used the implementation provided by Jia *et al.* (2014) and trained weights from scratch. The learning rate was set to be 0.1 and the loss was also changed to the Euclidean distance between estimated and ground-truth gaze directions.

- **Random Forests (RF)**: Random forests were recently demonstrated to outperform existing methods for person-independent appearance-based gaze estimation (Sugano *et al.*, 2014). We used the implementation provided by the authors, and the same parameters as in Sugano *et al.* (2014), and we resized input eye images to 15×9 according to the implementation in Sugano *et al.* (2014), which has been optimised.

- ***k*-Nearest Neighbours (kNN)**: As shown in Sugano *et al.* (2014), a simple kNN regression estimator can perform well in scenarios that offer a large amount of dense training images. We used the same kNN implementation and also incorporated a training images clustering in head angle space.

- **Adaptive Linear Regression (ALR)**: Because it was originally designed for a person-specific and sparse set of training images (Lu *et al.*, 2014b), ALR does not scale well to large datasets. We therefore used the same approximation as in Funes Mora and Odobez (2013), i.e. we selected five training persons for each test person with lowest interpolation weights. We further selected random subsets of images from the neigbours of the test image in head pose space. We used the same image resolution as for RF.

- **Support Vector Regression (SVR)**: Schneider et al. used SVR with a polynomial kernel under a fixed head pose (Schneider *et al.*, 2014). We used a linear

SVR (Fan *et al.*, 2008) for scalability given the large amount of training data. We also used a concatenated vector of HOG and LBP features (6×4 blocks, 2×2 cells for HOG) as suggested in Schneider *et al.* (2014). However, we did not use manifold alignment since it does not support pose-independent training.

- **Shape-based approach (EyeTab)**: In addition to the appearance-based methods, we evaluated one state-of-the-art shape-based method that estimates gaze by fitting a limbus model (a fixed-diameter disc) to detected iris edges (Wood and Bulling, 2014). We used the implementation provided by the authors.

Datasets

As in Zhang *et al.* (2015a), in all experiments that follow, we used a random subset of the full dataset consisting of 1,500 left eye images and 1,500 right eye images from each participant. Because one participant only offered 1,448 face images, we randomly oversampled data of that participant to 3,000 eye images. From now on we refer to this subset as *MPIIGaze*, while we call the same subset with manual facial landmark annotations *MPIIGaze+*. To evaluate the generalisation capabilities of the proposed method, in addition to MPIIGaze, we used all screen target sequences with both VGA and HD videos of the EYEDIAP dataset for testing (Funes Mora *et al.*, 2014). We did not use the floating target sequences in the EYEDIAP dataset since they contain many extreme gaze directions that are not covered by UT Multiview. We further used the SynthesEyes dataset (Wood *et al.*, 2015) that contains 11,382 eye samples from 10 virtual participants.

Evaluation Procedure

For cross-dataset evaluation, each method was trained on UT Multiview or SynthesEyes, and tested on MPIIGaze, MPIIGaze+ or EYEDIAP. We used the UT Multiview dataset as the training set for each method because it covers the largest area in head and gaze angle spaces compared to EYEDIAP and our MPIIGaze datasets (see Figure 3.4). Note that SynthesEyes has the same head and gaze angle ranges as UT Multiview dataset. For cross-person evaluation, we performed a leave-one-person-out cross-validation for all participants on MPIIGaze+.

3.4.1 Performance Evaluation

We first report the performance evaluation for the cross-dataset setting, for which all the methods were trained and tested on two different datasets respectively, followed by the cross-person evaluation setting, for which all methods were evaluated with leave-one-person-out cross-validation.

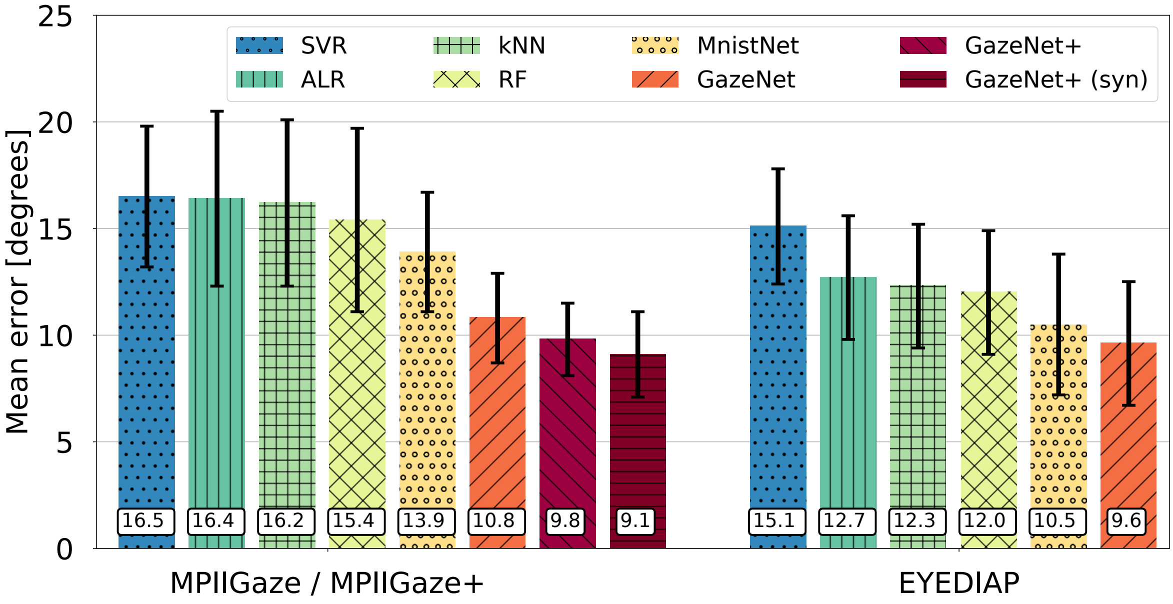


Figure 3.11: Gaze estimation error for cross-dataset evaluation with training on 64,000 eye images in UT Multiview and testing on 45,000 eye images of MPIIGaze or MPIIGaze+ (left) and EYEDIAP (right). Bars show mean error across all participants; error bars indicate standard deviations.

3.4.1.1 *Cross-Dataset Evaluation*

Figure 3.11 shows the mean angular errors of the different methods when trained on UT Multiview dataset and tested on both MPIIGaze, or MPIIGaze+, and EYEDIAP datasets. Bars correspond to mean error across all participants in each dataset, and error bars indicate standard deviations across persons. As can be seen from the figure, our GazeNet shows the lowest error on both datasets (10.8 degrees on MPIIGaze, 9.6 degrees on EYEDIAP). This represents a significant performance gain of 22% (3.1 degrees) on MPIIGaze and 8% on EYEDIAP (0.9 degrees), $p < 0.01$ using a paired Wilcoxon signed rank test (Wilcoxon, 1945), over the state-of-the-art method (Zhang *et al.*, 2015a). Performance on MPIIGaze and MPIIGaze+ is generally worse than on the EYEDIAP dataset, which indicates the fundamental difficulty of the in-the-wild setting covered by our dataset. We also evaluated performance on the different sequences of EYEDIAP (not shown in the figure). Our method achieved 10.0 degrees on the HD sequences and 9.2 degrees on the VGA sequences. This difference is most likely caused by differences in camera angles and image quality. The shape-based EyeTab method performs poorly on MPIIGaze (47.1 degrees mean error and 7% misdetection rate), which shows the advantage of appearance-based approaches in this challenging cross-dataset setting.

The input image size for some baselines, like RF, kNN and ALR, has been optimized to be 15×9 pixels, which was lower than the 60×36 pixels used in our method. To make the comparison complete, we also evaluated our GazeNet with 15×9 pixels input images and achieved 11.4 degrees gaze estimation error on

MPIIGaze, thereby still outperforming the other baseline methods.

Compared to GazeNet, GazeNet+ uses the manually annotated facial landmark locations MPIIGaze+ instead of the detected ones. In this case the mean error is reduced from 10.8 degrees to 9.8 degrees, which indicates that the face detection and landmark alignment accuracy is still a dominant error factor in practice. Furthermore, GazeNet+ (syn) implements the strategy proposed in Wood *et al.* (2015). That is, we first trained the model with synthetic data and then fine-tuned it on the UT Multiview dataset. This approach further reduced the gaze estimation error to 9.1 degrees. For comparison, the naive predictor that always outputs the average gaze direction of all training eye images in UT Multiview (not shown in the figure) achieves an estimation error of 34.2 degrees on MPIIGaze and 42.4 degrees on EYEDIAP.

While GazeNet achieved significant performance improvements for this challenging generalisation task, the results underline the difficulty of unconstrained gaze estimation. They also reveal a critical limitation of previous laboratory-based datasets such as UT Multiview with respect to variation in eye appearance, compared to MPIIGaze, which was collected in the real world. The learning-by-synthesis approach presented in Wood *et al.* (2015) is promising given that it allows the synthesis of variable eye appearance and illumination conditions. This confirms the importance of the training data and indicates that future efforts should focus on addressing the gaze estimation task both in terms of training data as well as methodology to bridge the gap to the within-dataset scenario.

3.4.1.2 *Cross-Person Evaluation*

Although results of the previous cross-dataset evaluation showed the advantage of our GazeNet, they still fall short of the cross-person performance reported in Sugano *et al.* (2014). To discuss the challenges of person-independent gaze estimation within MPIIGaze, we performed a cross-person evaluation using a leave-one-person-out approach. Figure 3.12 shows the mean angular errors of this cross-person evaluation. Since the model-based EyeTab method has been shown to perform poorly in our setting, we opted to instead show a learning-based result using the detected pupil (iris centre) positions. More specifically, we used the pupil positions detected using (Wood and Bulling, 2014) in the normalised eye image space as a feature for kNN regression, and performed the leave-one-person-out evaluation.

As can be seen from the figure, all methods performed better than in the cross-dataset evaluation, which indicates the importance of domain-specific training data for appearance-based gaze estimation methods. Although the performance gain is smaller in this setting, our GazeNet still significantly (13%) outperformed the second-best MnistNet with 5.5 degrees mean error ($p < 0.01$, paired Wilcoxon signed rank test). While the pupil position-based approach worked better than the original EyeTab method, its performance was still worse than the different appearance-based methods. In this case there is dataset-specific prior knowledge about gaze distribution, and the mean prediction error that always outputs the

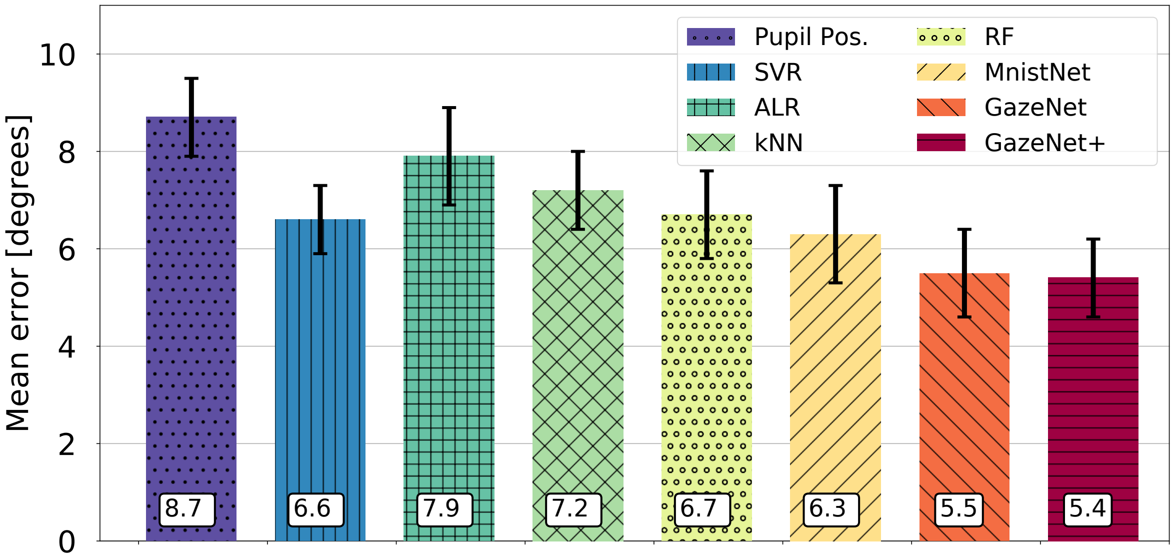


Figure 3.12: Gaze estimation error on MPIIGaze and MPIIGaze+ for cross-person evaluation using a leave-one-person-out approach. Bars show the mean error across participants; error bars indicate standard deviations; numbers on the bottom are the mean estimation error in degrees. GazeNet+ refers to the result for MPIIGaze+.

average gaze direction of all training images becomes 13.9 degrees. Because the noise in facial landmark detections is included in the training set, there was no noticeable improvement when testing our GazeNet on MPIIGaze+ (shown as GazeNet+ in Figure 3.12). It contradicts the observation with the previous cross-dataset evaluation that testing on MPIIGaze+ can bring one degree of improvement compared to MPIIGaze with detected facial landmarks (from 10.8 to 9.8 degrees).

3.4.2 Key Challenges

The previous results showed a performance gap between cross-dataset and cross-person evaluation settings. To better understand this gap, we additionally studied several key challenges. In all analyses that follow, we used GazeNet+ in combination with MPIIGaze+ to minimise error in face detection and alignment.

3.4.2.1 *Differences in Gaze Ranges*

As discussed in Zhang *et al.* (2015a) and Wood *et al.* (2015), one of the most important challenges for unconstrained gaze estimation is differences in gaze ranges between the training and testing domains. Although handling the different gaze angles has been researched by combining geometric and appearance-based methods (Mora and Odobez, 2016), it is still challenging for appearance-based gaze estimation methods. The first bar in Figure 3.13 (*UT*) corresponds to the cross-dataset evaluation using the UT Multiview dataset for training and MPIIGaze+ for testing. In this case,

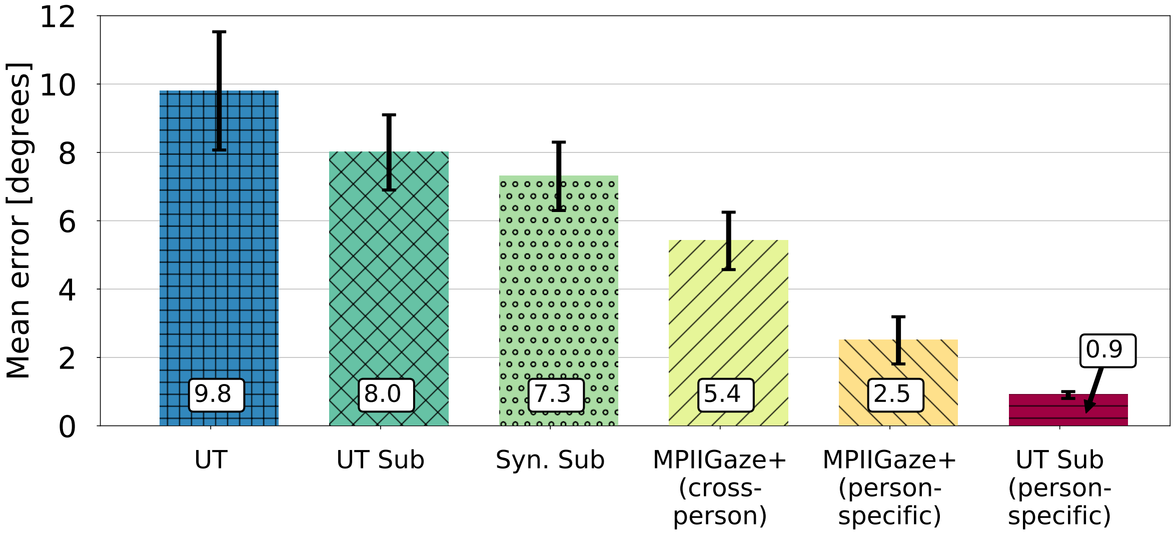


Figure 3.13: Gaze estimation error on MPIIGaze+ using GazeNet+ for different training strategies and evaluation settings. Bars show the mean error across participants; error bars indicate standard deviations; numbers on the bottom are the mean estimation error in degrees. From left to right: 1) training on UT Multiview, 2) training on UT Multiview subset, 3) training on synthetic images targeted to the gaze and head pose ranges, 4) training on MPIIGaze+ with cross-person evaluation, 5) training on *MPIIGaze+* with person-specific evaluation, and 6) training on UT Multiview subset with person-specific evaluation.

as illustrated in Figure 3.4, the training data covers wider gaze ranges than the testing data. The second bar (*UT Sub*) corresponds to the performance of the model trained on a subset of the UT Multiview dataset that consists of 3,000 eye images per participant selected so as to have the same head pose and gaze angle distributions as MPIIGaze+. If the training dataset is tailored to the target domain and the specific gaze range, we achieve about 18% improvement in performance (from 9.8 to 8.0 degrees).

The top of Figure 3.14 shows the gaze estimation errors in horizontal gaze direction with training on UT Multiview, UT Multiview subset, and MPIIGaze+, and testing on MPIIGaze+. The dots correspond to the average error for that particular gaze direction, while the line is the result of a quadratic polynomial curve fitting. The lines correspond to the *UT*, *UT Sub* and *MPIIGaze+ (cross-person)* bars in Figure 3.13. As can be seen from the figure, for the model trained on UT Multiview subset, gaze estimation error increased for images that were close to the edge of the gaze range. In contrast, the model trained on the whole UT Multiview showed more robust performance across the full gaze direction range. The most likely reason for this difference is given by Figure 3.14, which showns the percentage of images for the horizontal gaze directions for the training samples of MPIIGaze+ and UT Multiview. As can be seen from the figure, while *UT Sub* and MPIIGaze+ have the same gaze direction distribution, UT Multiview and MPIIGaze+ differ substantially.

This finding demonstrates the fundamental shortcoming of previous works that only focused on cross-person evaluations and thereby implicitly or explicitly assumed a single, and thus restricted, gaze range. As such, this finding highlights the importance not only of cross-dataset evaluations, but also of developing methods that are robust to (potentially very) different gaze ranges found in different settings.

3.4.2.2 *Differences in Illumination Conditions*

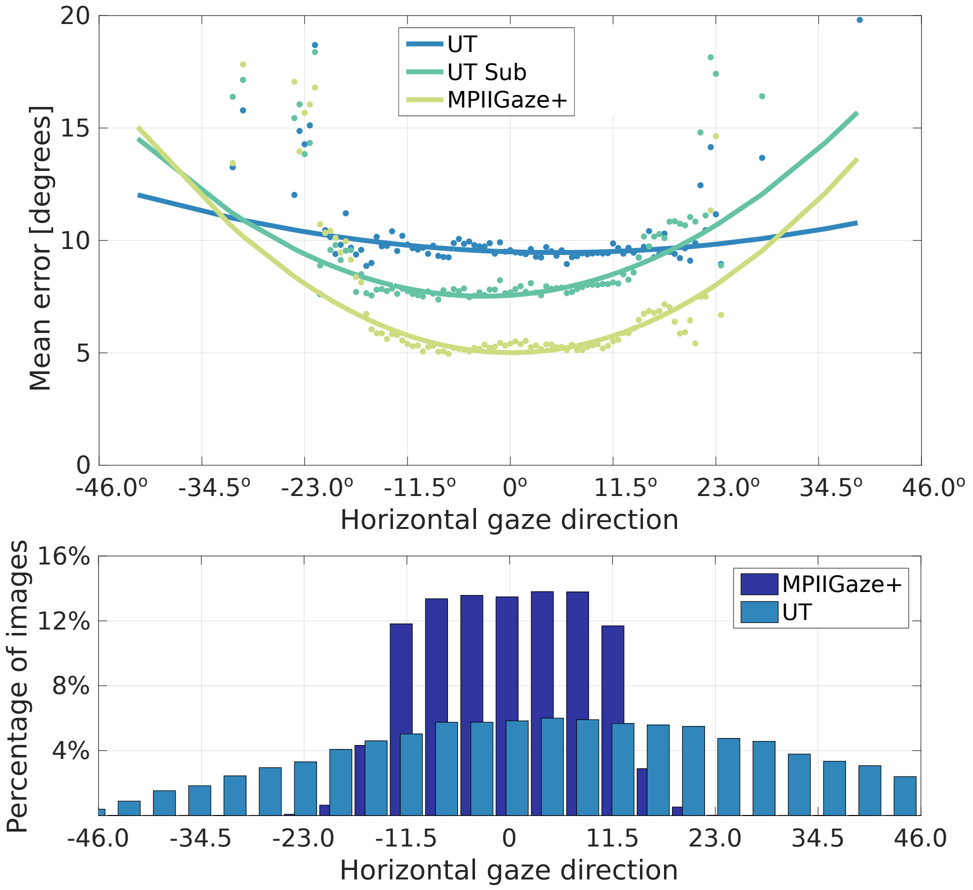


Figure 3.14: Top: Gaze estimation error on MPIIGaze+ for the model trained with UT Multiview, UT Multiview subset, and MPIIGaze+ for different horizontal gaze directions. Bottom: Percentage of images for the horizontal gaze directions of MPIIGaze+ and UT.

Illumination conditions are another important factor in unconstrained gaze estimation and have been the main motivation for using fully synthetic training data that can cover a wider range of different illuminations (Wood *et al.*, 2015). The third bar in Figure 3.13 (*Syn Sub*) corresponds to the same fine-tuned model as *GazeNet+ (syn)* in Figure 3.11, but with the gaze range restricted to the same head pose and gaze angle distributions as MPIIGaze+. The fourth bar in Figure 3.13

(*MPIIGaze+ (cross-person)*) shows the results of within-dataset cross-person evaluation on MPIIGaze+. For the second to the fourth bar in Figure 3.13, the training data has nearly the same head angle and gaze direction range. The only difference is in the variation in illumination conditions in the training data. While the use of synthetic training data results in improved performance (from 8.0 degrees to 7.3 degrees), there is still a large gap between cross-dataset and cross-person settings.

This tendency is further illustrated in Figure 3.15, in which we evaluated gaze estimation error with respect to lighting directions with our GazeNet. Similar to Figure 3.3, we plotted the mean gaze estimation error according to the mean intensity difference between the left and right face region. The different colours represent the models trained with UT Multiview subset, synthetic subset and MPIIGaze+. They also correspond to *UT Sub*, *Syn. Sub* and *MPIIGaze+ (cross-person)* in Figure 3.13. The dots are averaged error for horizontal difference in the mean intensity in the face region, and lines are with quadratic polynomial curve fitting. Similar to Figure 3.14, the bottom of Figure 3.15 shows the percentage of images for mean greyscale intensity difference between the left and right half of the face region. We cannot show the distribution for *UT Sub* and *Syn. Sub* since their face images are not available. Compared to the model trained solely on the UT subset, the model with synthetic data shows better performance across different lighting conditions. While there still remains an overall performance gap from the domain-specific performance, the effect of synthetic data is more visible in the area with extreme lighting directions.

3.4.2.3 *Differences in Personal Appearance*

To further study the unconstrained gaze estimation task, we then evaluated person-specific gaze estimation performance, i.e. where training and testing data come from the same person. The results of this evaluation on MPIIGaze+ are shown as the second last bar (*MPIIGaze+ (person-specific)*) in Figure 3.13. Since there are 3,000 eye images for each participant in MPIIGaze+, we picked the first 2,500 eye images for training and the rest for testing. Similarly, the last bar (*UT Sub (p.s.)*) in Figure 3.13 shows the person-specific evaluation within the UT subset, also with 2,500 eye images for training and 500 eye images for testing. The performance gap between *MPIIGaze+ (cross-person)* and *MPIIGaze+ (person-specific)* illustrates the fundamental difficulty of person-independent gaze estimation. The difference between *MPIIGaze+* (person-specific) and *UT Sub (p.s.)* also shows, however, that in-the-wild settings are challenging even for the person-specific case.

Figure 3.16 shows the estimation error of each participant in both cross-dataset (trained on the UT Multiview) and person-specific (leave-one-person-out training on MPIIGaze+) settings with our GazeNet. Bars correspond to mean error for each participant and the error bars indicate standard deviations. Example faces from each participant are shown at the bottom. As the figure shows, for the cross-dataset evaluation the worst performance was achieved for participants wearing glasses (P5, P8, and P10). This is because the UT Multiview dataset does not include training images covering this case, although glasses can cause noise in the eye appearance as

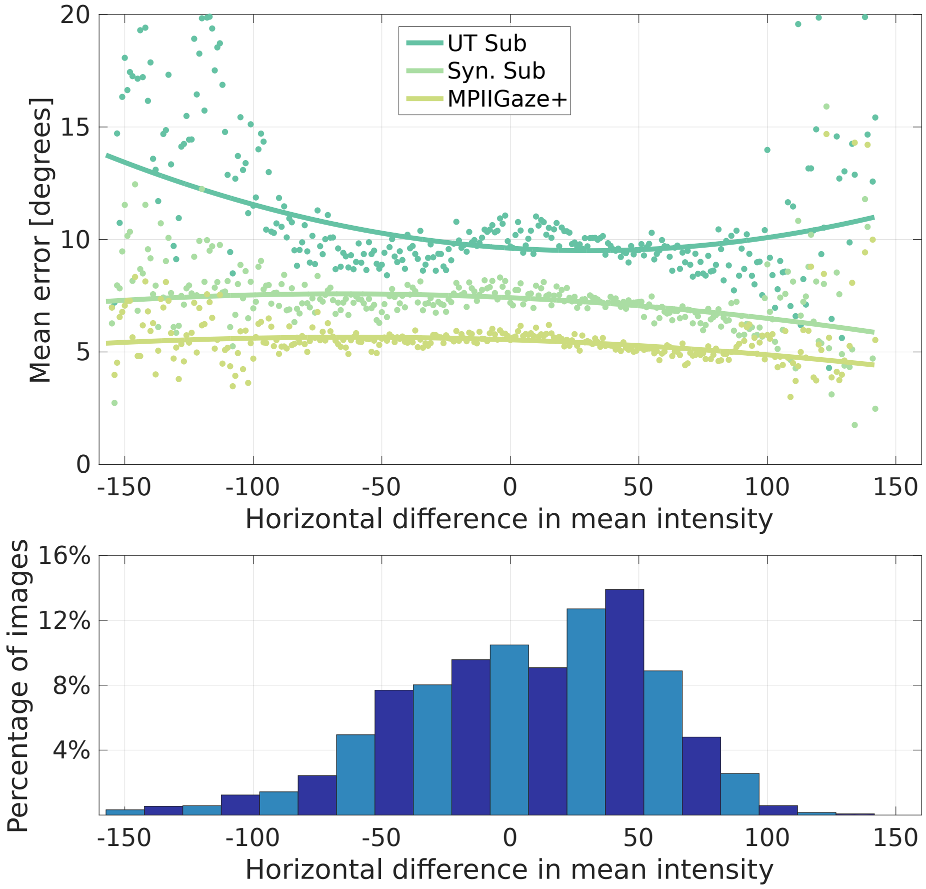


Figure 3.15: Top: Gaze estimation error on MPIIGaze+ across mean greyscale intensity differences between the left and right half of the face region for models trained on UT subset, SynthesEyes subset, and MPIIGaze+. Bottom: Corresponding percentage of images for all mean greyscale intensity differences.

shown in Figure 3.5a. For the person-specific evaluation, glasses are not the biggest error source, given that corresponding images are available in the training set. It can also be seen that the performance differences between participants are smaller in the person-specific evaluation. This indicates a clear need for developing new methods that can robustly handle differences in personal appearance for unconstrained gaze estimation.

3.4.3 Further Analyses

Following the previous evaluations of unconstrained gaze estimation performance and key challenges, we now provide further analyses on closely related topics, specifically the influence of image resolution, the use of both eyes, and the use of head pose and pupil centre information on gaze estimation performance.

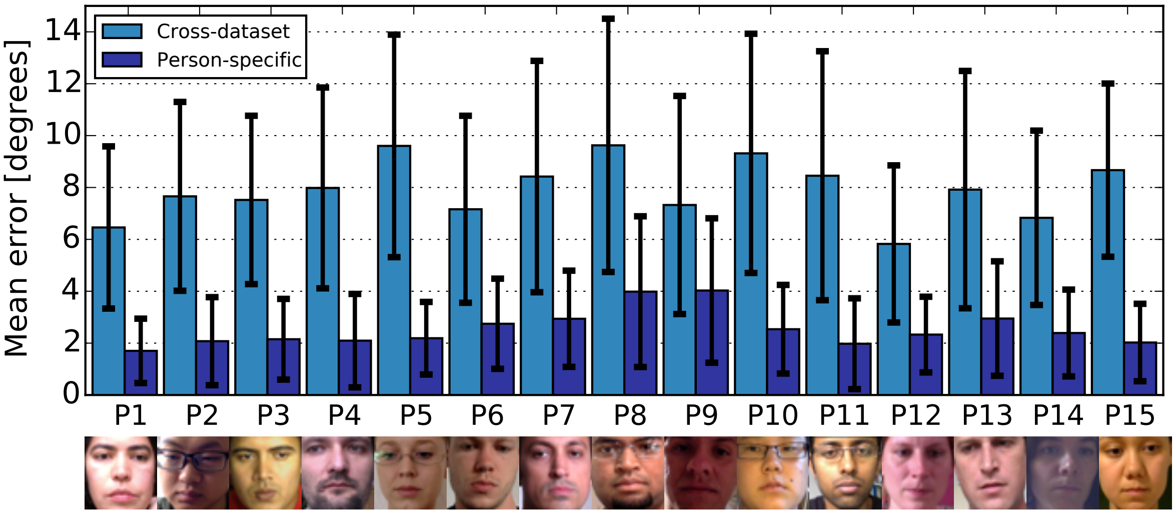


Figure 3.16: Gaze estimation error for each participant for two evaluation schemes: *cross-dataset*, where the model was trained on UT Multiview and tested on MPIIGaze+, and *person-specific*, where the model was trained and tested on the same person from MPIIGaze+. Sample images are shown at the bottom.

3.4.3.1 *Image Resolution*

We first explored the influence of image resolution on gaze estimation performance, since it is conceivable that this represents a challenge for unconstrained gaze estimation. To this end, we evaluated the performance for the cross-dataset evaluation setting (trained on UT Multiview and tested on MPIIGaze+) for different training and testing resolutions with our GazeNet. Starting from the default input resolution 60×36 in our model, we reduced the size to 30×18, 15×9 and 8×5. We always resized the test images according to the training resolution with bicubic interpolation. During training, we modified the stride of the first convolutional and max pooling layers of our GazeNet accordingly so that the input became the same starting from the second convolutional layer, regardless of the original image input resolution. Figure 3.17a summarises the results of this evaluation with resolutions of training images along the x-axis, and resolutions of testing images on the y-axis. In general, if the test images have higher resolution than the training images, higher resolution results in better performance. Performance becomes significantly worse if the test images are smaller than the training images.

Figure 3.17b shows the mean error of these models trained on one image resolution and tested across all testing resolutions, with the error bar denoting the standard deviation across all images. For the reason discussed above, the overall performance of the highest-resolution model is worse than that of the second 30×18 model. This shows that higher resolution does not always mean better performance for unconstrained gaze estimation.

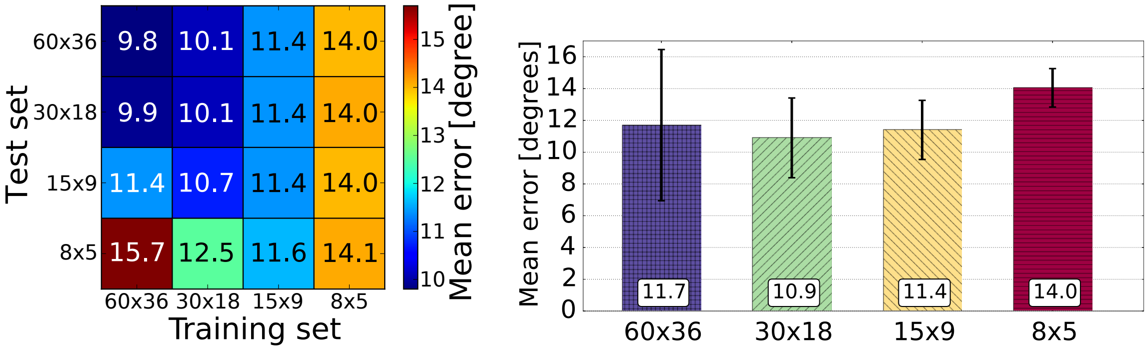


Figure 3.17: Gaze estimation error of the models trained on UT Multiview and tested on MPIIGaze+ for different image resolutions. Test images were resized to the resolution of the training images. (a) Combinations of different training and test set resolutions with cell numbers indicating the average error in degrees. (b) The mean estimation error for the models trained with certain image resolutions across all images. Bars show the mean error across participants in degrees; error bars indicate standard deviations.

3.4.3.2 *Use of Both Eyes*

Previous methods typically used a single eye image as input. However, it is reasonable to assume that for some cases, such as strong directional lighting, performance can be improved by using information from both eyes. To study this in more detail, we selected all images from MPIIGaze+ with two annotated eyes. We then evaluated different means of merging information from both eyes. The gaze estimation error when averaging across both eye images using the model trained on the UT Multiview dataset is 9.8 degrees with a standard deviation of 2.1 degrees. The best-case performance, i.e. always selecting the eye showing lower gaze estimation error, is 8.4 degrees with a standard deviation of 1.9 degrees. The gap between these two bars illustrates the limitations of the single eye-based estimation approach.

One approach to integrate estimation results from both eyes is to geometrically merge 3D gaze vectors after the appearance-based estimation pipeline. Given two 3D gaze vectors from both eyes, we thus further computed the mean gaze vector originating from the centre of both eyes. Ground-truth gaze vectors were also defined from the same origin, and the mean error across all faces using this approach was 7.2 degrees (standard deviation 1.4 degrees). It can be seen that even such a simple late fusion approach improves the estimation performance, indicating the potential of more sophisticated methods for fusing information from both eyes.

3.4.3.3 *Use of Head Pose Information*

To handle arbitrary head poses in the gaze estimation task, 3D head pose information has been used for the data normalisation as described in Sec. 3.3.2. After normalisation, 2D head angle vectors $\boldsymbol{h}$ were injected into the network as an additional geometry feature. The left side of Figure 3.18 shows a comparison between different architectures of the multi-modal CNN on the UT Multiview dataset. We followed the same three-fold cross-validation setting as in Sugano *et al.* (2014). The best performance reported in Sugano *et al.* (2014) is 6.5 degrees mean estimation error achieved by the head pose-clustered Random Forest. However, when the same clustering architecture is applied to the MnistNet (*Clustered MnistNet*), the performance became worse than for the model without clustering. In addition, our GazeNet (*Clustered GazeNet*) did not show any noticeable difference with the clustering structure. This indicates the higher learning flexibility of the CNN, which contributed to the large performance gain in the estimation task. The role of the additional head pose feature is also different in the two CNN architectures. While the MnistNet architecture achieved better performance with the help of the head pose feature, the effect of the head pose feature became marginal in the case of the GazeNet. Even though deeper networks like GazeNet can in general achieve better performance, achieving better performance with shallower networks is still important in some practical use cases where tehre is limited computational power, such as on mobile devices.

The right side of Figure 3.18 shows a comparison of models with and without the head pose feature in the cross-dataset setting (trained on UT and tested on MPIIGaze+). The effect of the additional head pose feature is marginal in this case, but this is likely because the head pose variation in the MPIIGaze dataset is already limited to near-frontal cases. We performed an additional experiment to compare the gaze estimation performance when using the head pose estimated from the personal and the generic 3D face model. We achieved 9.8 degrees and 9.7 degrees for the cross-dataset evaluation, respectively, suggesting that the generic face model is sufficiently accurate for the gaze estimation task.

3.4.3.4 *Use of Pupil Centres*

In GazeNet, we do not use pupil centre information as input. Although intuitively, eye shape features, such as pupil centres, can be a strong cue for gaze estimation, the model- or shape-based baseline performed relatively poorly for both the cross-dataset and cross-person evaluations. We therefore finally evaluated the performance of GazeNet when using the pupil centre as an additional feature for cross-person evaluation on MPIIGaze+. We detected the pupil centre location inside the normalised eye images using (Wood and Bulling, 2014) and concatenated the pupil location to the geometry feature vector (head angle $\boldsymbol{h}$). While there was an improvement between the models without and with the pupil centre feature, the improvement was relatively small (from 5.4 to 5.2 degrees). Performance improved more when using the manually annotated pupil centres, but still not significantly (5.0 degrees).

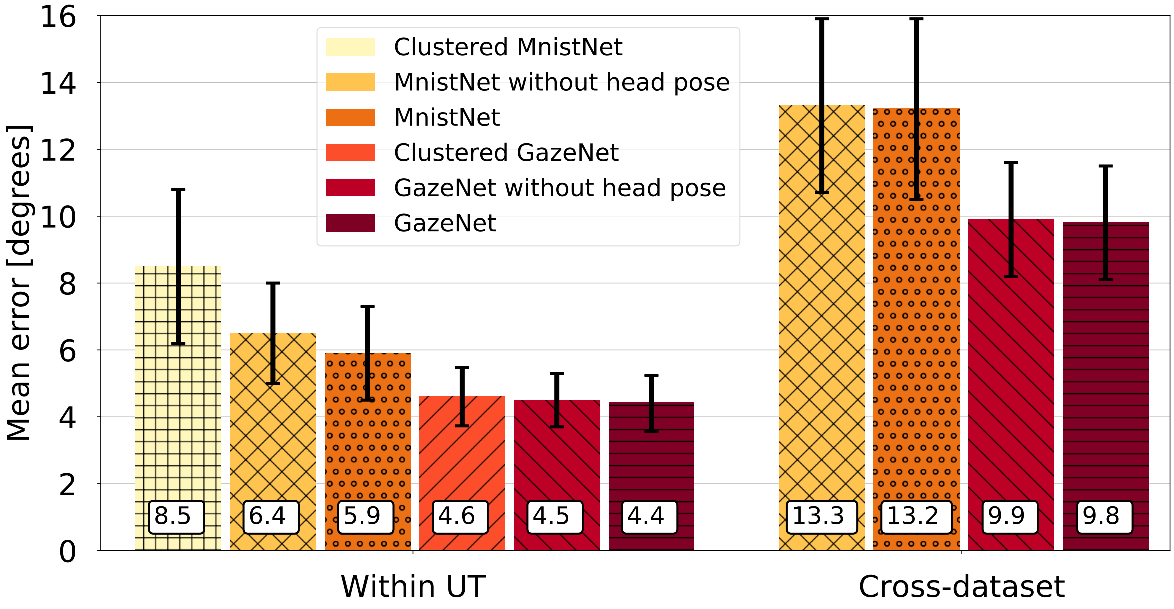


Figure 3.18: Gaze estimation error when using the pose-clustered structure (*Clustered MnistNet* and *Clustered GazeNet*), without head angle vectors $\boldsymbol{h}$ (*MnistNet without head pose* and *GazeNet without head pose*) for within-UT and cross-dataset (trained on UT, tested on MPIIGaze+) settings. Bars show the mean error across participants; error bars indicate standard deviations; numbers on the bottom are the mean estimation error in degrees.

3.5 DISCUSSION

This chapter made an important step towards unconstrained gaze estimation, i.e. gaze estimation from a single monocular RGB camera without assumptions regarding users' facial appearance, geometric properties of the environment or the camera and user therein. Unconstrained gaze estimation represents the practically most relevant but also most challenging gaze estimation task. Unconstrained gaze estimation is, for example, required for second-person gaze estimation from egocentric cameras or by a mobile robot. Through cross-dataset evaluation on our new MPIIGaze dataset, we demonstrated the fundamental difficulty of this task compared to the commonly used person-independent, yet still domain-specific, evaluation scheme. Specifically, gaze estimation performance dropped by up to 69% (from a gaze estimation error of 5.4 to 9.1 degrees) for the cross-dataset evaluation, as can be seen by comparing Figure 3.11 and Figure 3.12. The proposed GazeNet significantly outperformed the state of the art for both evaluation settings and in particular when pre-trained on synthetic data (see Figure 3.11). The 3.1 degrees improvement that we achieved in the cross-dataset evaluation corresponds to around 2.9 cm on the laptop screen after backprojection. Performance on MPIIGaze was generally worse than on EYEDIAP, which highlights the difficulty but also the importance of developing and evaluating gaze estimators on images collected in real-world environments.

We further explored key challenges of this task, including differences in gaze ranges, illumination conditions, and personal appearance. Previous works either implicitly or explicitly side stepped these challenges by restricting the gaze or head pose range (McMurrough *et al.*, 2012; Ponz *et al.*, 2012), studying a fixed illumination condition (He *et al.*, 2015; Funes Mora *et al.*, 2014; Sugano *et al.*, 2014), or by only recording for short amounts of time and thereby limiting variations in personal appearance (Smith *et al.*, 2013; Weidenbacher *et al.*, 2007). Several recent works also did not study 3D gaze estimation but, instead, simplified the task to regression from eye images to 2D on-screen coordinates (Huang *et al.*, 2017; Krafka *et al.*, 2016). While the 3D gaze estimation task generalises across hardware and geometric settings and thus facilities full comparison with other methods, the 2D task depends on the camera-screen relationship. Our evaluations demonstrated the fundamental shortcomings of such simplifications. They also showed that the development of 3D gaze estimation methods that properly handle all of these challenges, while important, remains largely unexplored. The ultimate goal of unconstrained gaze estimation is to obtain a generic estimator that can be distributed as a pre-trained library. While it is challenging to learn estimators that are robust and accurate across multiple domains, an intermediate solution might be to develop methods that adapt using domain-specific data automatically collected during deployment (Sugano *et al.*, 2016; Zhang *et al.*, 2017c).

The head angle vector plays different roles for the cross- and within-dataset evaluations. It is important to note that a 3D formulation is always required for unconstrained gaze estimation without restricting the focal length of the camera or the pose of the gaze target plane. 3D geometry, including the head pose, therefore has to be handled properly for unconstrained gaze estimation – a challenge still open at the moment. In this chapter we additionally explored the use of the head angle vector as a separate input to the CNN architecture as described in Zhang *et al.* (2015a). As shown in Figure 3.18, while head pose information does result in a performance improvement for the shallower MnistNet architecture used in Zhang *et al.* (2015a), it does not significantly improve the performance of GazeNet.

The state-of-the-art shape-based method (Wood and Bulling, 2014) performed poorly in the cross-dataset evaluation, achieving only 47.1 degrees mean error. Similarly, adding the detected pupil centres as additional input to the CNN resulted in only a small performance improvement (see Section 3.4.3.4). While using eye shape and pupil centre features is typically considered to be a promising approach, both findings suggest that its usefulness may be limited for unconstrained gaze estimation, particularly on images collected in real-world settings – leaving aside the challenge of detecting these features robustly and accurately on such images in the first place.

3.6 CONCLUSION

In this chapter we made a case for unconstrained gaze estimation – a task that, despite its scientific and practical importance, has been simplified in several ways in prior work. To address some of these simplifications, we presented the new MPIIGaze dataset that we collected over several months in everyday life and that therefore covers significant variation in eye appearance and illumination. The dataset also offers manually annotated facial landmarks for a large subset of images and is therefore well-suited for cross-dataset evaluations. Through extensive evaluation of several state-of-the-art appearance- and model-based gaze estimation methods, we demonstrated both the critical need for and challenges of developing new methods for unconstrained gaze estimation. Finally, we proposed an appearance-based method based on a deep convolutional neural network that improves performance by 22% for the most challenging cross-dataset evaluation on MPIIGaze. Taken together, our evaluations provide a detailed account of the state of the art in appearance-based gaze estimation and guide future research on this important computer vision task.

4 FULL-FACE APPEARANCE-BASED GAZE ESTIMATION

Eye gaze is an important non-verbal cue for human affect analysis. Recent gaze estimation work indicated that information from the full face region can benefit performance. Pushing this idea further, we propose an appearance-based method that, in contrast to a long-standing line of work in computer vision, only takes the full face image as input. Our method encodes the face image using a convolutional neural network with spatial weights applied on the feature maps to flexibly suppress or enhance information in different facial regions. Through extensive evaluation, we show that our full-face method significantly outperforms the state of the art for both 2D and 3D gaze estimation, achieving improvements of up to 14.3% on MPIIGaze and 27.7% on EYEDIAP for person-independent 3D gaze estimation. We further show that this improvement is consistent across different illumination conditions and gaze directions and particularly pronounced for the most challenging extreme head poses.

4.1 INTRODUCTION

A large number of works in computer vision have studied the problem of estimating human eye gaze (Hansen and Ji, 2010) given its importance for different applications, such as human-robot interaction (Mutlu *et al.*, 2009), affective computing (D'Mello *et al.*, 2012), and social signal processing (Vinciarelli *et al.*, 2008). While early methods typically required settings in which lighting conditions or head pose could be controlled (Lu *et al.*, 2014b; Baluja and Pomerleau, 1994; Tan *et al.*, 2002; Williams *et al.*, 2006), latest appearance-based methods using convolutional neural networks (CNN) have paved the way for gaze estimation in everyday settings that are characterised by significant amount of lighting and appearance variation (Zhang *et al.*, 2015a). Despite these advances, previous appearance-based methods have only used image information encoded from one or both eyes.

Recent results by Krafka et al. indicated that a multi-region CNN architecture that takes both eye and face images as input can benefit gaze estimation performance (Krafka *et al.*, 2016). While, intuitively, human gaze is closely linked to eyeball pose and eye images should therefore be sufficient to estimate gaze direction, it is indeed conceivable that especially machine learning-based methods can leverage additional information from other facial regions. These regions could, for example, encode head pose or illumination-specific information across larger image areas than those available in the eye region. However, it is still an open question whether a (more efficient and elegant) face-only approach can work, which facial regions are most important for such a full-face appearance-based method, and whether current

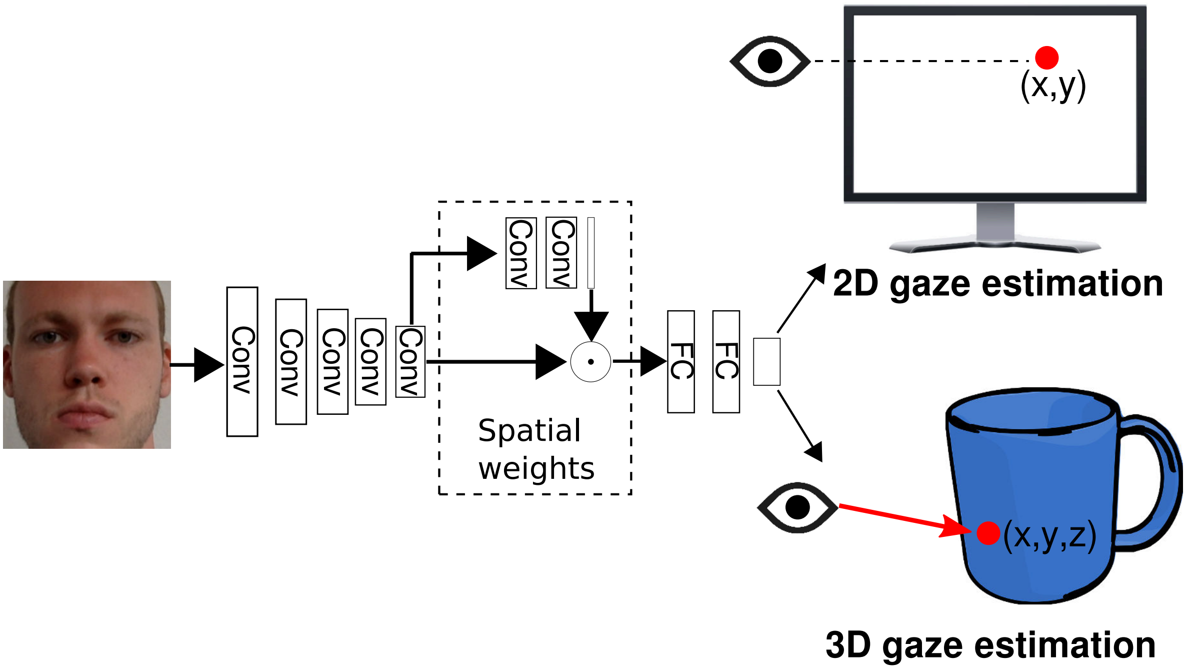


Figure 4.1: Overview of the proposed full face appearance-based gaze estimation pipeline. Our method only takes the face image as input and performs 2D and 3D gaze estimation using a convolutional neural network with spatial weights applied on the feature maps.

deep architectures can encode the information in these regions. In addition, the gaze estimation task in Krafka *et al.* (2016) was limited to a simple 2D screen mapping and the potential of the full-face approach for 3D gaze estimation thus remains unclear.

The goal of this chapter is to shed light on these questions by providing a detailed analysis of the potential of the full-face approach for 2D and 3D appearance-based gaze estimation (see Figure 4.1). The specific contributions of this chapter are two-fold. First, we propose a full-face CNN architecture for gaze estimation that, in stark contrast to a long-standing tradition in gaze estimation, takes the full face image as input and directly regresses to 2D or 3D gaze estimates. We quantitatively compare our full-face method with existing eye-only (Zhang *et al.*, 2015a) and multi-region (Krafka *et al.*, 2016) methods and show that it can achieve a person-independent 3D gaze estimation accuracy of 4.8° on the challenging MPIIGaze dataset, thereby improving by 14.3% over the state of the art. Second, we propose a *spatial weights* mechanism to efficiently encode information about different regions of the full face into a standard CNN architecture. The mechanism learns spatial weights on the activation maps of the convolutional layers, reflecting that the information contained in different facial regions. Through further quantitative and qualitative evaluations we show that the proposed spatial weights network facilitates the learning of estimators that are robust to significant variation in illumination conditions as well as head pose and gaze directions available in current datasets.

4.2 GAZE ESTIMATION TASKS

Before detailing our model architecture for full-face appearance-based gaze estimation, we first formulate and discuss two different gaze estimation tasks: 2D and 3D gaze estimation. A key contribution of this chapter is to investigate full-face appearance-based gaze estimation for both tasks. This not only leads to a generic model architecture but also provides valuable insights into the difference and benefits gained from full-face information for both task formulations.

Although the 3D task formulation poses additional technical challenges to properly handle the complex 3D geometry, it can be applied to different device and setups without assuming a fixed camera-screen relationship. This formulation therefore is the most general and practically most relevant. If the application scenario can afford a fixed screen position, the 2D formulation is technically less demanding and therefore expected to show better accuracy.

4.2.1 2D Gaze Estimation

As the most straightforward strategy, the 2D gaze estimation task is formulated as a regression from the input image $\boldsymbol{I}$ to a 2-dimensional on-screen gaze location $\boldsymbol{p}$ as $\boldsymbol{p} = f(\boldsymbol{I})$, where f is the regression function. Usually $\boldsymbol{p}$ is directly defined in the coordinate system of the target screen (Lu *et al.*, 2014b; Sugano *et al.*, 2015; Tan *et al.*, 2002; Valenti *et al.*, 2012) or, more generally, a virtual plane defined in the camera coordinate system (Krafka *et al.*, 2016). Since the relationship between eye appearance and gaze location depends on the position of the head, the regression function usually requires 3D head poses (Valenti *et al.*, 2012) or face bounding box locations (Huang *et al.*, 2017; Krafka *et al.*, 2016) in addition to eye and face images.

It is important to note that, in addition to the fixed target plane, another important assumption in this formulation is that the input image $\boldsymbol{I}$ is always taken from the same camera with fixed intrinsic parameters. Although no prior work explicitly discussed this issue, trained regression functions cannot be directly applied to different cameras without proper treatment of the difference in projection models.

4.2.2 3D Gaze Estimation

In contrast, the 3D gaze estimation task is formulated as a regression from the input image $\boldsymbol{I}$ to a 3D gaze vector $\boldsymbol{g} = f(\boldsymbol{I})$. Similarly as for the 2D case, the regression function f typically takes the 3D head pose as an additional input. The gaze vector $\boldsymbol{g}$ is usually defined as a unit vector originating from a 3D reference point $\boldsymbol{x}$ such as the center of the eye (Mora and Odobez, 2016; Lu *et al.*, 2014a, 2015; Wood *et al.*, 2015; Zhang *et al.*, 2015a). By assuming a calibrated camera and with information on the 3D pose of the target plane, the 3D gaze vector $\boldsymbol{g}$ can be converted by projecting gaze location $\boldsymbol{p}$ into the camera coordinate system. The gaze location $\boldsymbol{p}$ as in the 2D case can be obtained by intersecting the 3D gaze vector $\boldsymbol{g}$ with the target plane.

Image Normalization. To both handle different camera parameters and address the task of cross-person training efficiently, Sugano et al. proposed a data normalization procedure for 3D appearance-based gaze estimation (Sugano *et al.*, 2014). The basic idea is to apply a perspective warp to the input image so that the estimation can be performed in a normalized space with fixed camera parameters and reference point location. Given the input image $\boldsymbol{I}$ and the location of the reference point $\boldsymbol{x}$, the task is to compute the conversion matrix $\boldsymbol{M} = \boldsymbol{SR}$.

$\boldsymbol{R}$ is the inverse of the rotation matrix that rotates the camera so that it looks at the reference point and so that the x-axes of both the camera and head coordinate systems become parallel. The scaling matrix $\boldsymbol{S}$ is defined so that the reference point is located at a distance d_s from the origin of the normalized camera coordinate system.

The conversion matrix $\boldsymbol{M}$ rotates and scales any 3D points in the input camera coordinate system to the normalized coordinate system, and the same conversion can be applied to the input image $\boldsymbol{I}$ via perspective warping using the image transformation matrix $\boldsymbol{W} = \boldsymbol{C}_s\boldsymbol{M}\boldsymbol{C}_r^{-1}$. $\boldsymbol{C}_r$ is the projection matrix corresponding to the input image obtained from a camera calibration, and $\boldsymbol{C}_s$ is another predefined parameter that defines the camera projection matrix in the normalized space.

During training, all training images $\boldsymbol{I}$ with ground-truth gaze vectors $\boldsymbol{g}$ are normalized to or directly synthesized (Sugano *et al.*, 2014; Wood *et al.*, 2015) in the training space, which is defined by d_s and $\boldsymbol{C}_s$. Ground-truth gaze vectors are also normalized as $\hat{\boldsymbol{g}} = \boldsymbol{Mg}$, while in practice they are further converted to an angular representation (horizontal and vertical gaze direction) assuming a unit length. At test time, test images are normalized in the same manner and their corresponding gaze vectors in the normalized space are estimated via regression function trained in the normalized space. Estimated gaze vectors are then transformed back to the input camera coordinates by $\boldsymbol{g} = \boldsymbol{M}^{-1}\hat{\boldsymbol{g}}$.

4.3 FULL-FACE GAZE ESTIMATION WITH A SPATIAL WEIGHTS CNN

For both the 2D and 3D gaze estimation case, the core challenge is to learn the regression function f. While a large body of work has only considered the use of the eye region for this task, we instead aim to explore the potential of extracting information from the full face.

Our hypothesis is that other regions of the face beyond the eyes contain valuable information for gaze estimation.

As shown in Figure 4.2, to this end we propose a CNN with spatial weights (spatial weights CNN) for full-face appearance-based 2D and 3D gaze estimation. To efficiently use the information from full-face images, we propose to use additional layers that learn spatial weights for the activation of the last convolutional layer. The motivation behind this spatial weighting is two-fold. First, there could be some image regions that do not contribute to the gaze estimation task such as background regions, and activations from such regions have to be suppressed for better performance. Second, more importantly, compared to the eye region that is expected to always

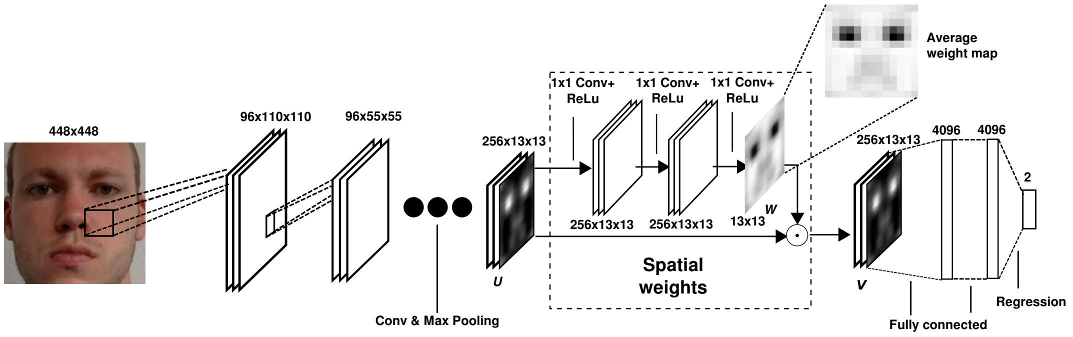


Figure 4.2: Spatial weights CNN for full-face appearance-based gaze estimation. The input image is passed through multiple convolutional layers to generate a feature tensor $\boldsymbol{U}$. The proposed spatial weights mechanism takes $\boldsymbol{U}$ as input to generate the weight map $\boldsymbol{W}$, which is applied to $\boldsymbol{U}$ using element-wise multiplication. The output feature tensor $\boldsymbol{V}$ is fed into the following fully connected layers to – depending on the task – output the final 2D or 3D gaze estimate.

contribute to the gaze estimation performance, activations from other facial regions are expected to subtle. The role of facial appearance is also depending on various input-dependent conditions such as head pose, gaze direction and illumination, and thus have to be properly enhanced according to the input image appearance. Although, theoretically, such differences can be learned by a normal network, we opted to introduce a mechanism that forces the network more explicitly to learn and understand that different regions of the face can have different importance for estimating gaze for a given test sample. To implement this stronger supervision, we used the concept of the three 1×1 convolutional layers plus rectified linear unit layers from Tompson *et al.* (2015) as a basis and adapted it to our full face gaze estimation task. Specifically, instead of generating multiple heatmaps (one to localise each body joint) we only generated a single heatmap encoding the importance across the whole face image. We then performed an element-wise multiplication of this weight map with the feature map of the previous convolutional layer. An example weight map is shown in Figure 4.2, averaged from all samples from the MPIIGaze dataset.

4.3.1 Spatial Weights Mechanism

The proposed spatial weights mechanism includes three additional convolutional layers with filter size 1×1 followed by a rectified linear unit layer (see Figure 4.2). Given activation tensor $\boldsymbol{U}$ of size $N \times H \times W$ as input from the convolutional layer, where N is the number of feature channels and H and W are height and width of the output, the spatial weights mechanism generates a $H \times W$ spatial weight matrix $\boldsymbol{W}$. Weighted activation maps are obtained from element-wise multiplication of $\boldsymbol{W}$ with the original activation $\boldsymbol{U}$ with

$$\boldsymbol{V}_c = \boldsymbol{W} \odot \boldsymbol{U}_c, \tag{4.1}$$

where $\boldsymbol{U}_c$ is the c-th channel of $\boldsymbol{U}$, and $\boldsymbol{V}_c$ corresponds to the weighted activation map of the same channel. These maps are stacked to form the weighted activation tensor $\boldsymbol{V}$, and are fed into the next layer. Different from the spatial dropout (Tompson *et al.*, 2015), the spatial weights mechanism weights the information continuously and keeps the information from different regions. The same weights are applied to all feature channels, and thus the estimated weights directly correspond to the facial region in the input image.

During training, the filter weights of the first two convolutional layers are initialized randomly from a Gaussian distribution with 0 mean and 0.01, and a constant bias of 0.1. The filter weights of the last convolutional layers are initialized randomly from a Gaussian distribution with 0 mean and 0.001 variance, and a constant bias of 1.

Gradients with respect to $\boldsymbol{U}$ and $\boldsymbol{W}$ are

$$\frac{\partial V}{\partial \boldsymbol{U}} = \partial \boldsymbol{W}, \tag{4.2}$$

and

$$\frac{\partial V}{\partial \boldsymbol{W}} = \frac{1}{N} \sum_{c}^{N} \partial \boldsymbol{U}_c. \tag{4.3}$$

The gradient with respect to $\boldsymbol{W}$ is normalised by the total number of the feature maps N, since the weight map $\boldsymbol{W}$ affects all the feature maps in $\boldsymbol{U}$ equally.

4.3.2 Implementation Details

As the baseline CNN architecture we used AlexNet (Krizhevsky *et al.*, 2012) that consists of five convolutional layers and two fully connected layers. We trained an additional linear regression layer on top of the last fully connected layer to predict the $\boldsymbol{p}$ in screen coordinates for 2D gaze estimation or normalized gaze vectors $\hat{\boldsymbol{g}}$ for the 3D gaze estimation task. We used the pre-training result on the LSVRC-2010 ImageNet training set (Krizhevsky *et al.*, 2012) to initialize the five convolution layers, and fine-tuned the whole network on the MPIIGaze dataset (Zhang *et al.*, 2015a). The input image size of our networks was 448×448 pixels, which results in an activation $\boldsymbol{U}$ of size $256 \times 13 \times 13$ after the pooling layer of the 5-th convolutional layers.

For 2D gaze estimation, input face images were cropped according to the six facial landmark locations (four eye corners and two mouth corners). While in practice this is assumed to be done with face alignment methods such as Baltrušaitis *et al.* (2014), in the following experiments we used dataset-provided landmark locations. The centroid of the six landmarks was used as the center of the face, and a rectangle with a width of 1.5 times the maximum distance between landmarks was used as the face bounding box. The loss function was the $\ell 1$ distance between the predicted and ground-truth gaze positions in the target screen coordinate system.

For 3D gaze estimation, the reference point $\boldsymbol{x}$ was defined as the center of 3D locations of the same six facial landmarks. We fit the generic 3D face model provided with MPIIGaze to the landmark locations to estimate the 3D head pose. During image normalization, we defined d_s and $\boldsymbol{C}_s$ so that the input face image size became 448×448 pixels. In preliminary experiments we noticed that the additional head pose feature proposed by Zhang *et al.* (2015a) did not improve the performance in the full-face case. In this chapter we therefore only used image features. The loss function was the $\ell 1$ distance between the predicted and ground-truth gaze angle vectors in the normalized space.

4.4 EVALUATION

To evaluate our architecture for the 2D and 3D gaze estimation tasks, we conducted experiments on two current gaze datasets: MPIIGaze (Zhang *et al.*, 2015a) and EYEDIAP (Funes Mora *et al.*, 2014). For the MPIIGaze dataset, we performed a leave-one-person-out cross-validation on all 15 participants. In order to eliminate the error caused by face alignment, we manually annotated the six facial landmarks for data normalization and image cropping. In the original evaluation, there were 1,500 left and 1,500 right eye samples randomly taken from each participant. For a direct comparison, we obtained face images corresponding to the same evaluation set and flipped the face images when they came from the right eye. Our face patch-based setting took the middle point of face (the center of all six landmarks) as the origin of gaze direction.

For the EYEDIAP dataset, we used the screen target session for evaluation and sampled one image per 15 frames from four VGA videos of each participant. We used head pose and eye centres annotations provided by the dataset for image normalization, and reference points were set to the midpoint of the two eye centres. The eye images were cropped by the same way as MPIIGaze dataset. We randomly separated the 14 participants into 5 groups and performed 5-fold cross-validation.

We compared our full-face gaze estimation method with two state-of-the-art baselines: A single eye method (Zhang *et al.*, 2015a) that only uses information encoded from one eye as well as a multi-region method (Krafka *et al.*, 2016) that takes eye images, the face image, and a face grid as input.

Single Eye. One of the baseline methods is the state-of-the-art single eye appearance-based gaze estimation method (Zhang *et al.*, 2015a), which originally used the LeNet (Jia *et al.*, 2014; LeCun *et al.*, 1998) architecture. For a fair comparison, we instead used the AlexNet architecture as our proposed model (see Section 4.3.2). Eye images were cropped by taking the center of the eye corners as the center and with the width of 1.5 times of the distance between corners, and resized to 60×36 pixels as proposed in Zhang *et al.* (2015a). In this case, each individual eye became the input to the model, and the reference point $\boldsymbol{x}$ was set to the middle of inner and outer eye corners.

iTracker. Since neither code nor models were available, we re-implemented the iTracker architecture (Krafka *et al.*, 2016) according to the description provided in the chapter. Face images were cropped in the same manner as our proposed method and resized to 224×224 pixels. Eye images were cropped by taking the middle point of

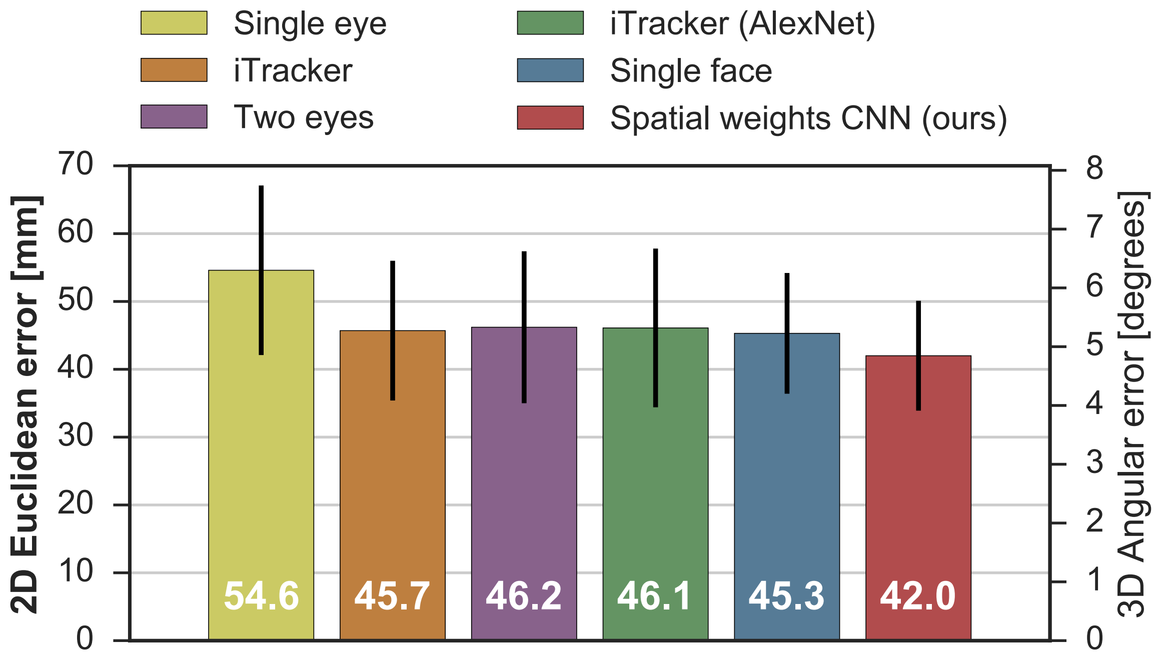


Figure 4.3: Error for 2D gaze estimation on the MPIIGaze dataset in millimetres (Euclidean error) and degrees (angular error). The face grid was used as additional input. Error bars indicate standard deviations.

the inner and outer eye corners as the image center and with the width of 1.7 times of the distance between the corners, and resized to 224 × 224 pixels. For the 2D gaze estimation task, we also used the face grid feature (Krafka *et al.*, 2016) with a size of 25 × 25 pixels. The face grid encodes the face size and location inside the original image. For a fair comparison with our proposed architecture, we also evaluated the model using the same AlexNet CNN architecture as *iTracker (AlexNet)*. To validate the effect of the face input, we also tested the iTracker (AlexNet) architecture only taking two eye images as *Two eyes* model.

4.4.1 2D Gaze Estimation

Figure 4.3 summarises the results for the 2D gaze estimation task. Each row corresponds to one method, and if not noted otherwise, the face grid feature was used in addition to the image input. The left axis shows the Euclidean error between estimated and ground-truth gaze positions in the screen coordinate system in millimetres. The right axis shows the corresponding angular error that was approximately calculated from the camera and monitor calibration information provided by the dataset and the same reference position for the 3D gaze estimation task.

As can be seen from Figure 4.3, all methods that take full-face information as input significantly outperformed the single eye baseline. The single face image

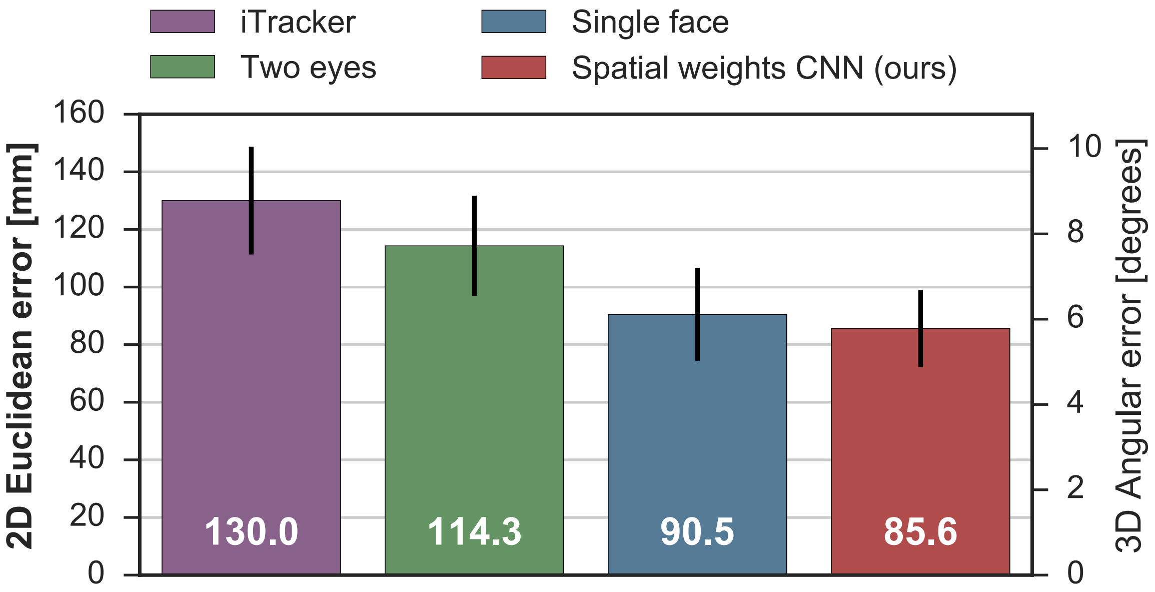


Figure 4.4: Error for 2D gaze estimation on the EYEDIAP dataset in millimetres (Euclidean error) and degrees (angular error). Error bars indicate standard deviations.

model achieved a competitive result to the iTracker and the iTracker (AlexNet) models. Performance was further improved by incorporating the proposed spatial weights network. The proposed spatial weights network achieved a statistically significant 7.2% performance improvement (paired t-test: $p < 0.01$) over the second best single face model. These findings are in general mirrored for the EYEDIAP dataset shown in Figure 4.4, while the overall performance is worse most likely due to the lower resolution and the limited amount of training images. Although the iTracker architecture performs worse than the two eyes model, our proposed model still performed the best.

4.4.2 3D Gaze Estimation

Figure 4.5 summarises the results for the 3D gaze estimation task. The left axis shows the angular error that was directly calculated from the estimated and ground-truth 3D gaze vectors. The right axis shows the corresponding Euclidean error that was approximated by intersecting the estimated 3D gaze vector with the screen plane. Compared to the 2D gaze estimation task, the performance gap between iTracker and the single face model is larger (0.7 degrees). Since the AlexNet-based iTracker model could achieve similar performance as the single face model, the performance drop seems to be partly due to its network architecture. Our proposed model achieved a significant performance improvement of 14.3% (paired t-test: $p < 0.01$) over iTracker, and a performance consistent with the 2D case.

As shown in Figure 4.6, the proposed model also achieved the best performance

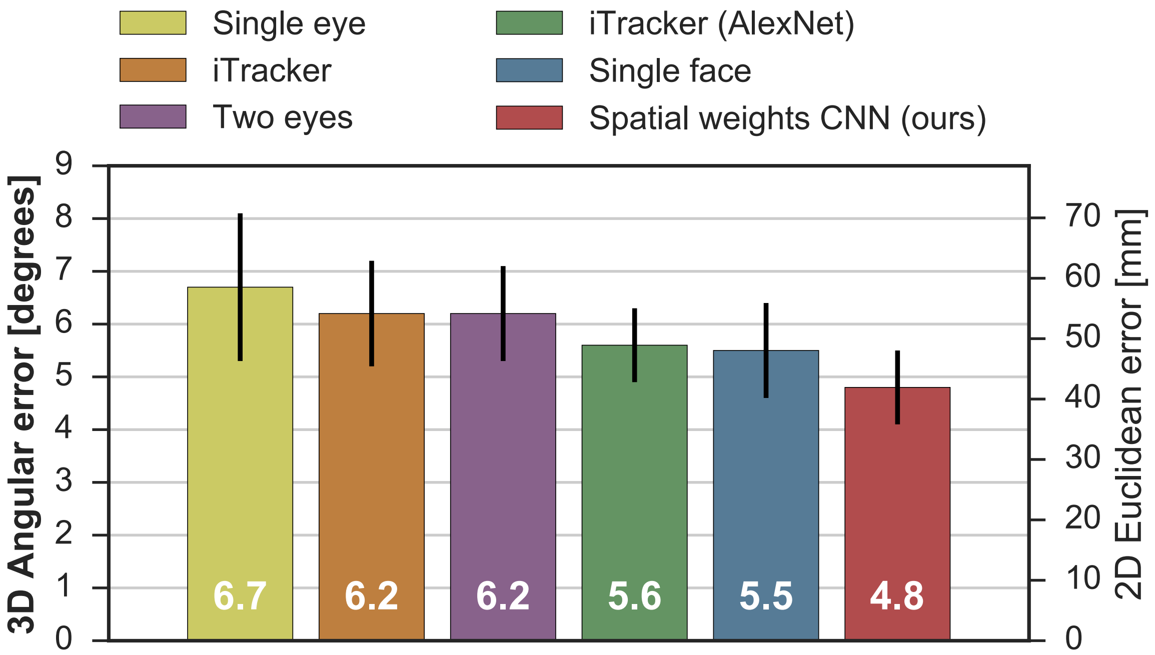


Figure 4.5: Error for 3D gaze estimation on the MPIIGaze dataset in degrees (angular error) and millimetres (Euclidean error). Error bars indicate standard deviations.

for the 3D gaze estimation task on the EYEDIAP dataset.

4.4.3 Head Pose and Facial Appearance

One natural hypothesis about why full-face input can help the gaze estimation task is that it brings head pose information which can be a prior for inferring gaze direction. In this section, we provide more insights on this hypothesis by comparing performance using face images *without* eye regions with a simple head pose-based baseline. More specifically, using the MPIIGaze dataset, we created face images where both eye regions were blocked with a gray box according to the facial landmark annotation. We compared the estimation performance using eye-blocked face images with: 1) a naive estimator directly treating the head pose as gaze direction, and 2) a linear regression function trained to output gaze directions from head pose input.

Angular error of these methods for the 3D estimation task are shown in Figure 4.7. While the error using eye-blocked face images was larger than the original single face architecture (5.5 degrees), the performance was better than baseline head pose-based estimators. This indicates, somewhat surprisingly, that the impact of taking full-face input is larger than head pose information, and the facial appearance itself is beneficial for inferring gaze direction.

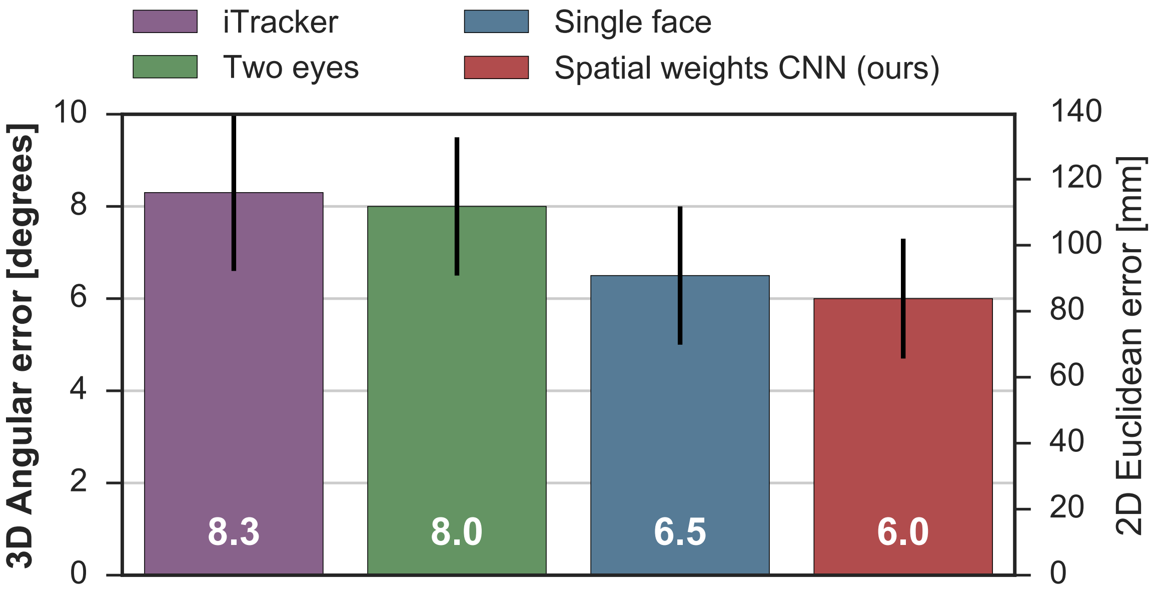


Figure 4.6: Error for 3D gaze estimation on the EYEDIAP dataset in degrees (angular error) and millimetres (Euclidean error). Error bars indicate standard deviations.

4.4.4 Importance of Different Facial Regions

To further analyse how different facial regions contribute to the overall performance, we generated region importance maps of the full-face model with respect to different factors for 3D gaze estimation. As proposed in (Zeiler and Fergus, 2014), region importance maps were generated by evaluating estimation error after masking parts of the input image. Specifically, given the 448 × 448 input face image, we used a grey-coloured mask with a size of 64 × 64 pixels and moved this mask over the whole image in a sliding window fashion with a 32 pixel stride. The per-image region importance maps were obtained by smoothing the obtained 64 × 64 error distribution with a box filter. The larger the resulting drop in gaze estimation accuracy the higher the importance of that region of the face. Individual face images and their importance maps were then aligned by warping the whole image using three facial landmark locations (centres of both eye corners and mouth corners). Finally, mean face patches and mean region importance maps were computed by averaging over all images. To illustrate the effect of the face image input, we compare these region importance maps with a quantitative performance comparison between two eyes (*Baseline*) and our proposed full-face model (*Ours*).

Illumination Conditions. The original MPIIGaze paper characterised the dataset with respect to different illumination conditions as well as gaze ranges (Zhang *et al.*, 2015a). We therefore first explored whether and which facial regions encode information on these illumination conditions. As in the original paper, we used the difference in mean intensity values of the right and left half of the face as a

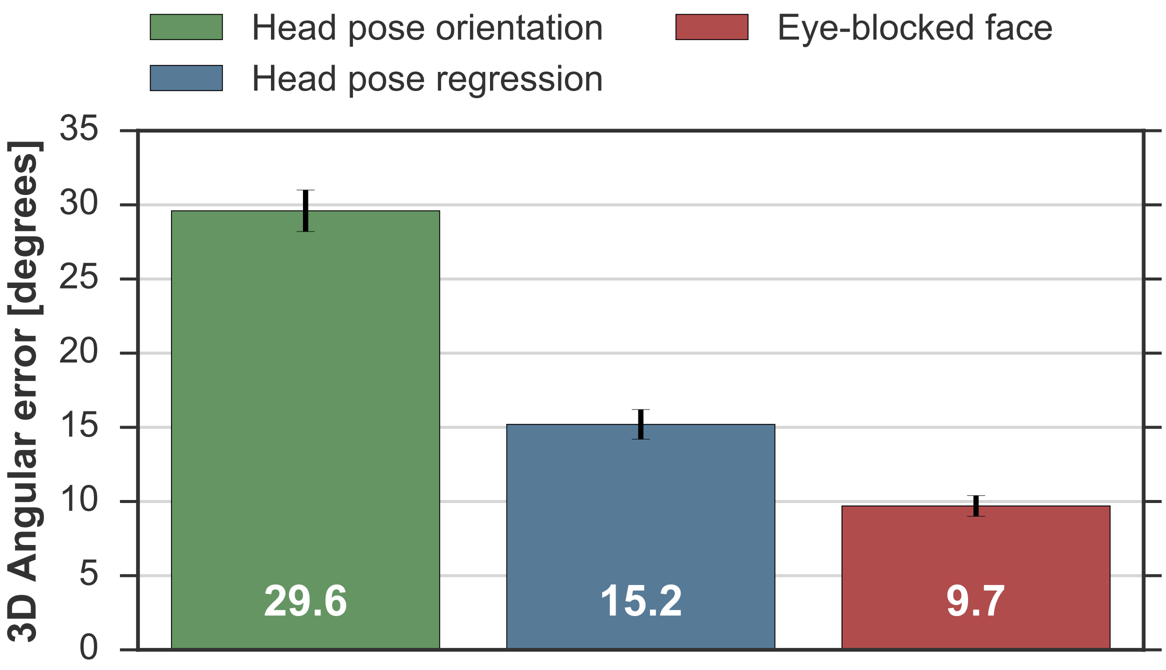


Figure 4.7: Gaze estimation error from the different models related to head pose. The numbers are angular error for 3D gaze estimation in degrees. Error bars indicate standard deviations.

proxy to infer directional light. We clustered all $15 \times 3,000$ images according to the illumination difference using *k*-means clustering, and computed the mean face image and mean importance map for each cluster. Figure 4.8 shows resulting sample region importance maps with respect to illumination conditions. As can be seen from the figure, under strong directional lighting (leftmost and rightmost example), more widespread regions around the eyes are required on the brighter side of the face. The proposed method consistently performed better than the two eye model over all lighting conditions.

Gaze Directions. Another factor that potentially influences the importance of different facial regions is the gaze direction. We therefore clustered images according to gaze direction in the same manner as before. The top two rows of Figure 4.9 show the corresponding region importance maps depending on horizontal gaze direction while the bottom two rows show maps depending on vertical gaze direction. As shown, different parts of the face become important depending on the gaze direction to be inferred. The eye region is most important if the gaze direction is straight ahead while the model puts higher importance on other regions if the gaze direction becomes more extreme.

Head Pose. While the head pose range in MPIIGaze is limited due to the recording setting, the EYEDIAP dataset contains a wide head pose range.

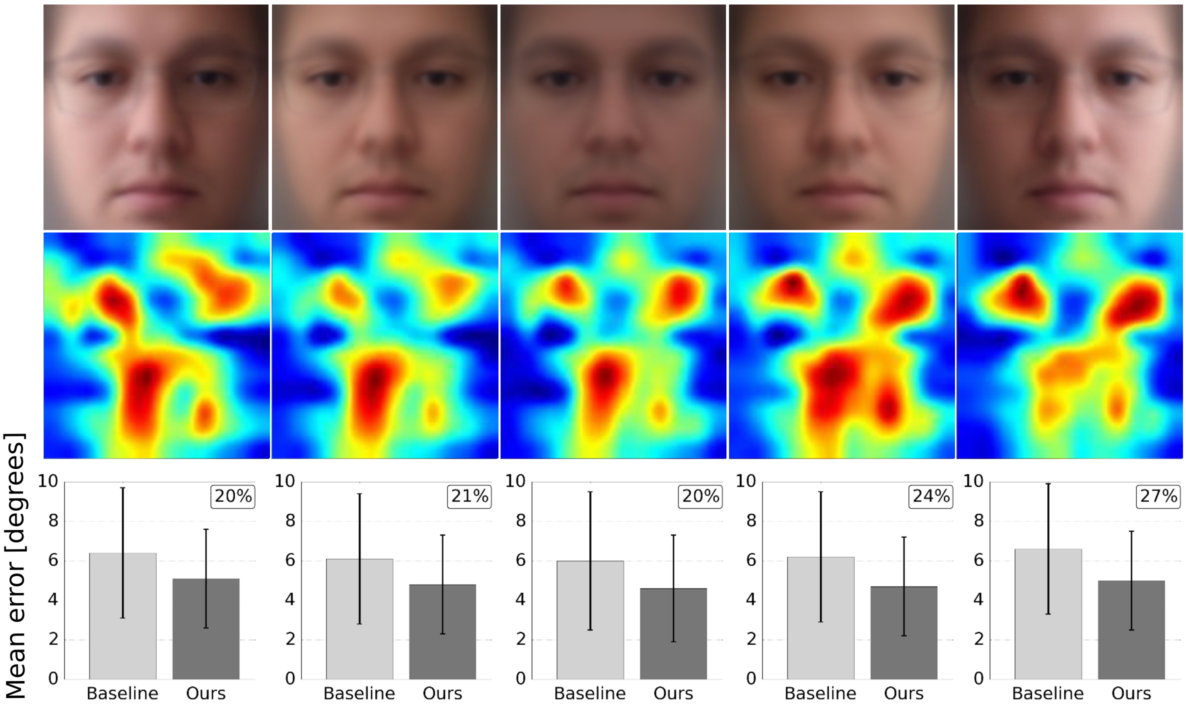


Figure 4.8: Region importance maps and corresponding mean face patches based on a clustering of face patches according to illumination conditions for the MPIIGaze dataset: From directional light on the right side of the face (left), over frontal light (center), to directional light on the left side of the face (right). Bar plots show the estimation error for the two eye model (baseline) and the proposed spatial weights CNN (ours), and the performance gain in percent in the top right corner. Error bars indicate standard deviations.

We therefore finally clustered images in EYEDIAP according to head pose in the same manner as before. The top two rows of Figure 4.10 show the corresponding region importance maps depending on horizontal head pose while the bottom two rows show maps depending on vertical head pose. In these cases, it can be clearly seen that the full-face input is particularly beneficial to improving estimation performance for extreme head poses. Non-eye facial regions also have in general higher importance compared to MPIIGaze, which indicates the benefit of using full-face input for low-resolution images.

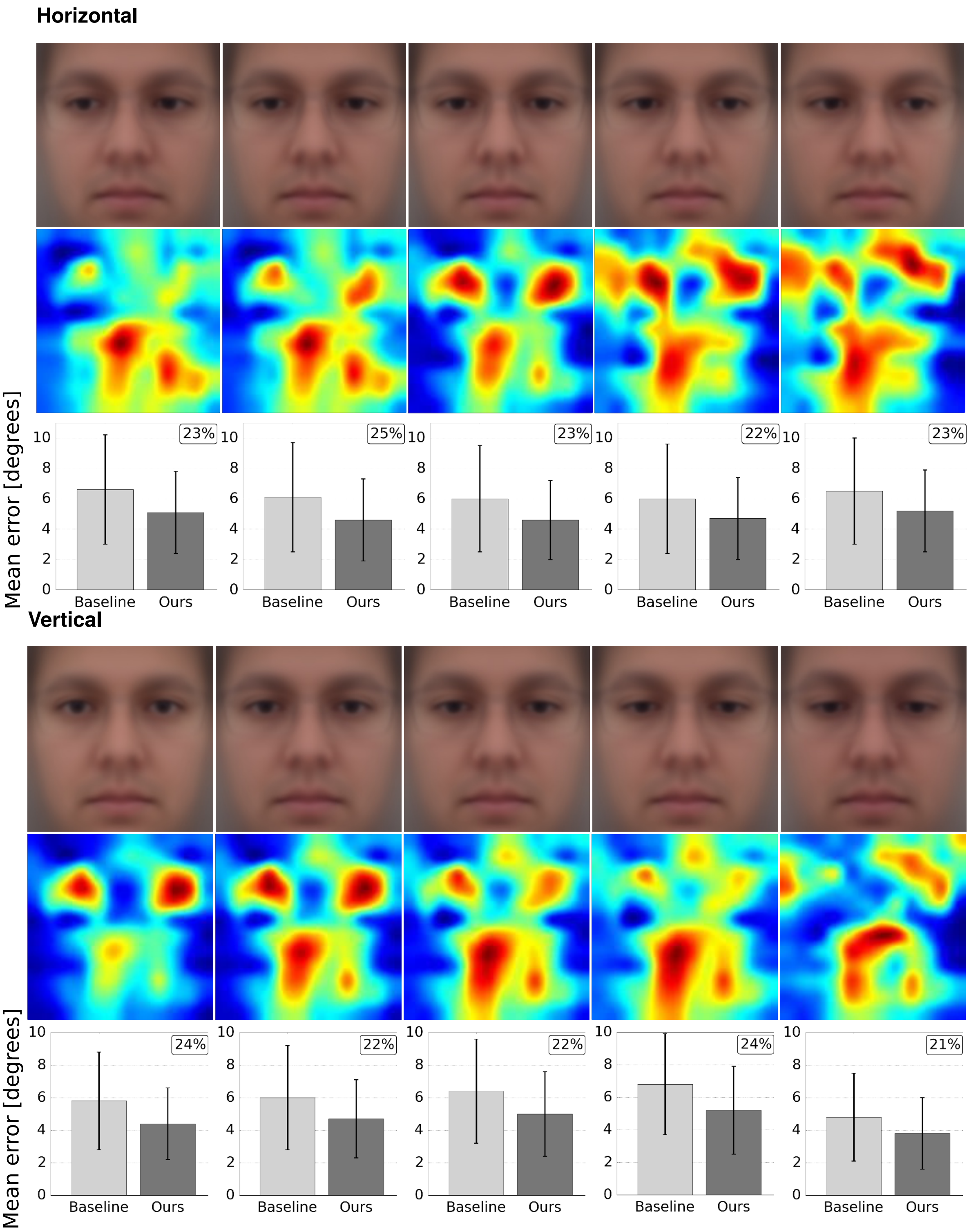


Figure 4.9: Region importance maps and corresponding mean face patches based on a clustering of images according to ground-truth horizontal (top) and vertical (bottom) gaze direction for the MPIIGaze dataset. Bar plots show the estimation error in the same manner as in Figure 4.8.

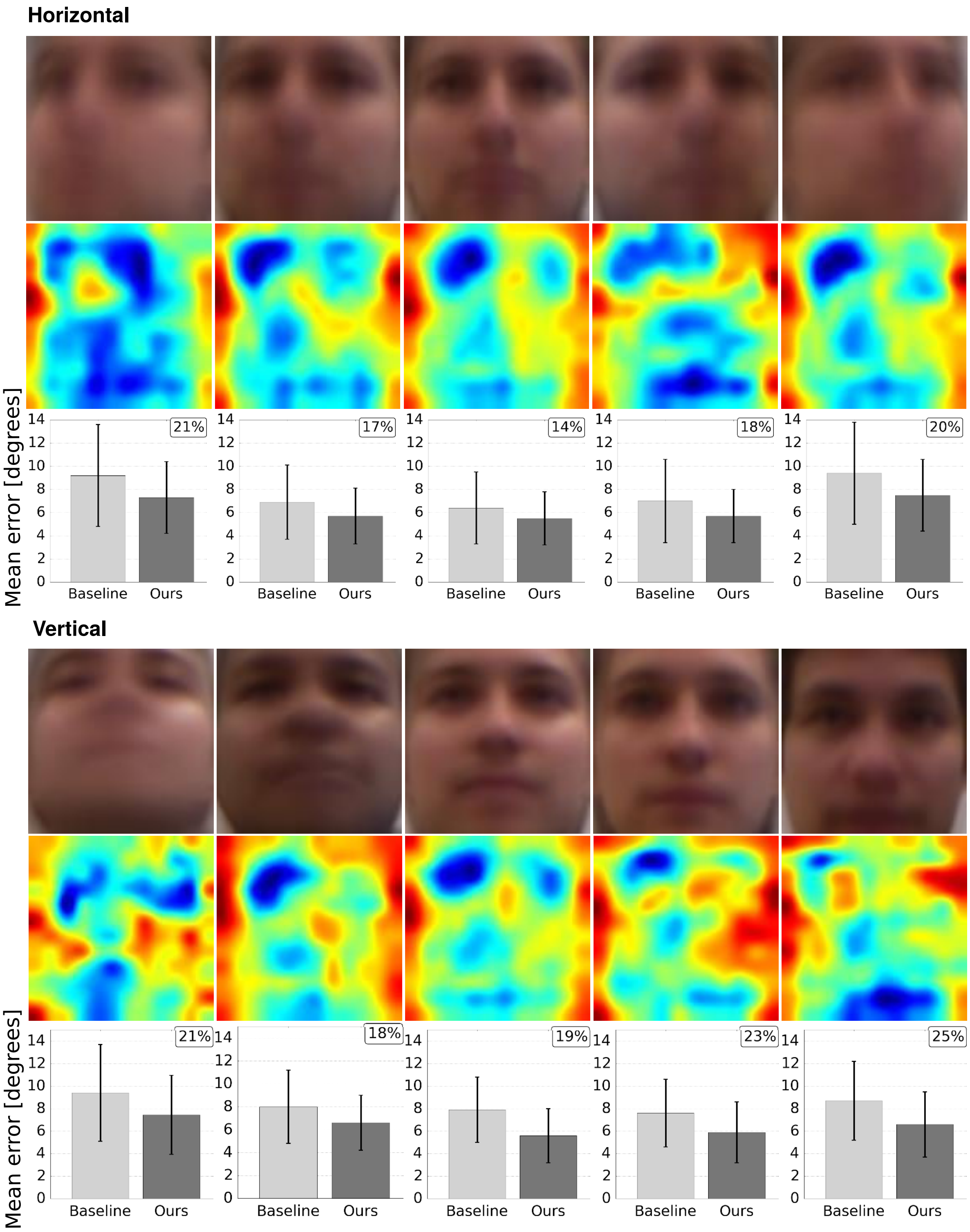


Figure 4.10: Region importance maps based on a clustering of images according to ground-truth horizontal (top) and vertical (bottom) head pose for the EYEDIAP dataset. Bar plots show the estimation error in the same manner as in Figure 4.8.

4.5 CONCLUSION

In this chapter we studied full-face appearance-based gaze estimation and proposed a spatial weights CNN method that leveraged information from the full face. We demonstrated that, compared to current eye-only and multi-region methods, our method is more robust to facial appearance variation caused by extreme head pose and gaze directions as well as illumination. Our method achieved an accuracy of 4.8° and 6.0° for person-independent 3D gaze estimation on the challenging in-the-wild MPIIGaze and EYEDIAP datasets, respectively – a significant improvement of 14.3% and 27.7% over the state of the art. We believe that full-face appearance-based gaze estimation leans itself closely to related computer vision tasks, such as face and facial feature detection, facial expression analysis, or head pose estimation. This chapter therefore points towards future learning-based methods that address multiple of these tasks jointly.

5 REVISITING DATA NORMALISATION FOR APPEARANCE-BASED GAZE ESTIMATION

Appearance-based gaze estimation is promising for unconstrained real-world settings, but the significant variability in head pose and user-camera distance poses significant challenges for training generic gaze estimators. Data normalisation was proposed to cancel out this geometric variability by mapping input images and gaze labels to a normalized space. Although used successfully in prior works, the role and importance of data normalisation remains unclear. To fill this gap, we study data normalisation for the first time using principled evaluations on both simulated and real data. We propose a modification to the current data normalisation formulation by removing the scaling factor and show that our new formulation performs significantly better (between 9.5% and 32.7%) in the different evaluation settings. Using images synthesized from a 3D face model, we demonstrate the benefit of data normalisation for the efficiency of the model training. Experiments on real-world images confirm the advantages of data normalisation in terms of gaze estimation performance.

5.1 INTRODUCTION

Driven by advances in deep learning and large-scale training image synthesis, appearance-based gaze estimation methods have recently received increased attention due to their significant potential for real-world applications (Zhang *et al.*, 2017c; Sugano *et al.*, 2016; Smith *et al.*, 2013; Zhang *et al.*, 2018a, 2017a). In contrast to their model- and feature-based counterparts (Hansen and Ji, 2010; Wang and Ji, 2017; Valenti *et al.*, 2012; Yamazoe *et al.*, 2008; Sesma *et al.*, 2012), appearance-based methods aim to directly map eye images to gaze directions, for example obtained using front-facing cameras already integrated into mobile devices (Krafka *et al.*, 2016). Early methods for appearance-based gaze estimation required a fixed head pose, e.g. enforced using a chin rest (Tan *et al.*, 2002; Williams *et al.*, 2006; Schneider *et al.*, 2014). While later works allowed for free head rotation (Funes Mora *et al.*, 2014; Deng and Zhu, 2017; He *et al.*, 2015; Yu *et al.*, 2016), the distance between user and camera was usually still assumed to be fixed and methods were mainly evaluated in controlled settings. Most recent works focused on the most challenging case, i.e. real-world environments without any constraints regarding head rotation and translation (Zhang *et al.*, 2018c, 2017d; Krafka *et al.*, 2016).

In principle, given a sufficient amount of training data, the variability caused by unconstrained head pose could be learned from the data. Previous works following this idea consequently focused on significantly increasing the number and diversity

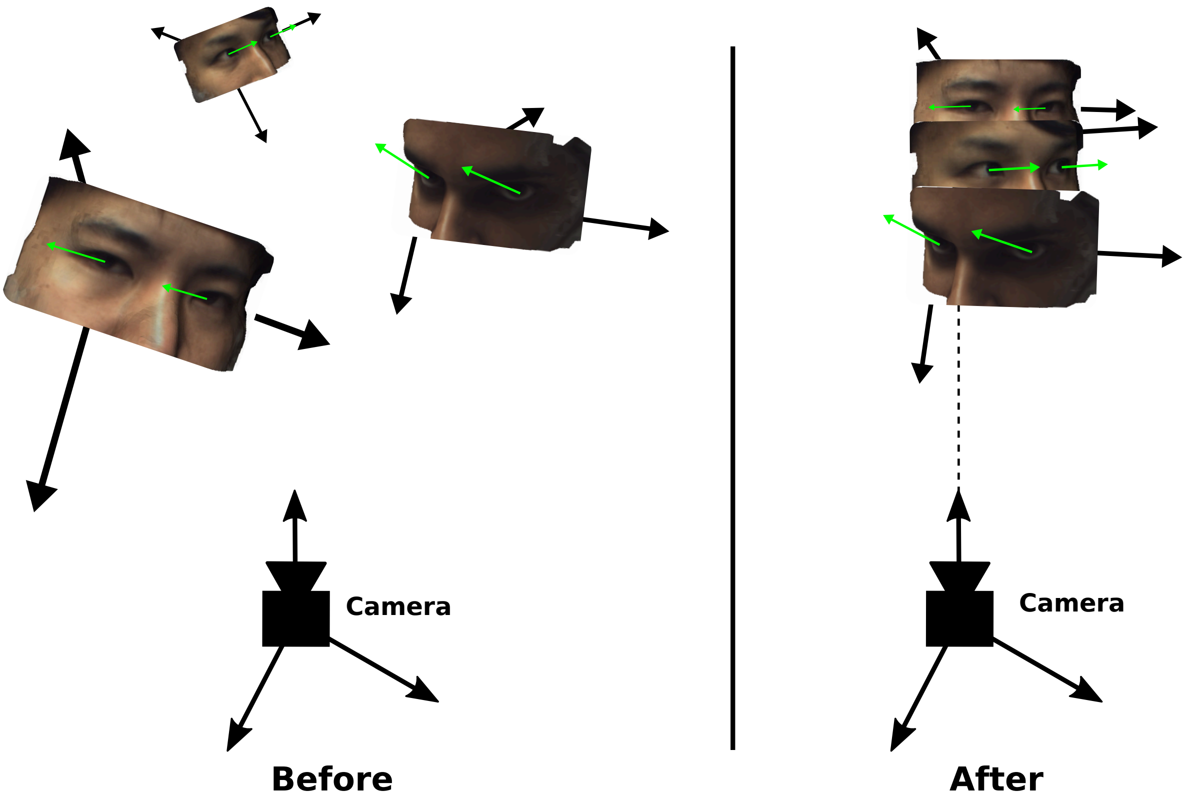


Figure 5.1: Data normalisation, as proposed for appearance-based gaze estimation, cancels out most variations caused by different head poses, by rotating and scaling the images.

of images to train the appearance-based gaze estimator (Zhang *et al.*, 2018c; Krafka *et al.*, 2016). While this approach resulted in significant performance improvements, manual collection and annotation of such large amounts of training data is time-consuming and costly. To reduce the burden of manual data collection, another recent line of work instead proposed to synthesize large numbers of eye images with arbitrary head poses using sophisticated 3D models of the eye region (Wood *et al.*, 2015, 2016b,a). However, for both of these approaches, covering all possible head poses is nearly impossible. In addition, this approach requires the gaze estimator to deal with a large amount of very similar and mostly redundant data and can result in prolonged training times and more difficult optimization of the loss function.

Data normalisation has been proposed to address the aforementioned challenge by reducing the training space and making the training more efficient. This is achieved by preprocessing the training data before it is used as input to the gaze estimator. As shown in Figure 5.1, the key idea is to normalize the data such that most of the variability caused by different head poses is canceled out. Originally proposed by Sugano et al. (Sugano *et al.*, 2014), this approach has subsequently been used very successfully in other works (Zhang *et al.*, 2015a, 2017d, 2018c; Shrivastava *et al.*, 2017). In a nutshell, data normalisation first rotates the camera to warp the eye images so that the x-axis of the camera coordinate system is perpendicular to the y-axis of the head coordinate system. Then, the image is scaled so that the (normalized) camera is located at a fixed distance away from the eye center. The final eye images have only 2 degrees of freedom in head pose for all the different data.

Although used successfully in prior works, the importance of rotation and translation/scaling of data normalisation remains unclear, and has not yet its impact on the gaze estimation performance been quantified. In this chapter we aim to fill this gap and, for the first time, explore the importance of data normalisation for appearance-based gaze estimation. The specific contributions of this chapter are two-fold. First, we explain the variability caused by different distances between camera and eye and discuss how data normalisation can cancel out some of this variability. Second, we demonstrate the importance of data normalisation for appearance-based gaze estimation with extensive experiments on both synthetic and real data. We first perform gaze estimation evaluations on synthesized eye images for different head poses to demonstrate the benefit of applying data normalisation. Afterwards, we evaluate within- and cross-dataset settings for gaze estimation and quantify the advantages of data normalisation with respect to performance. Third, we propose a modification to the original data normalisation formulation and demonstrate that this new formulation yields significant performance improvements for all evaluation settings studied.

5.2 DATA NORMALISATION

Data normalisation aims to align training and test data for learning-based gaze estimation by reducing the variability caused by head rotation and translation. In this section, we first demonstrate the problem setting and discuss why data normalisation is needed for canceling out such variability. We describe the detailed process of data normalisation presented in prior work (Sugano *et al.*, 2014; Zhang *et al.*, 2018c), and point out an issue when handling 2D images. We then introduce our modification on data normalisation with a stronger planarity assumption.

5.2.1 Problem Setting

As discussed earlier, most previous methods on appearance-based gaze estimation assume a frontal head pose, as shown on the left in Figure 5.2. However, in real-world settings we need to deal with head rotation, as shown on the right in Figure 5.2. The corresponding eyes are shown above the face image in Figure 5.2, and the goal of a pose-independent gaze estimator is to estimate 2D gaze positions or 3D gaze directions of eye images no matter how they appear in the original input images.

In addition, precisely speaking, scale/distance of the face also affects the eye appearance. Different distances between camera and eye obviously result in different sizes of eye in the captured images, and the eye appearance itself changes because the eye is not a planar object.

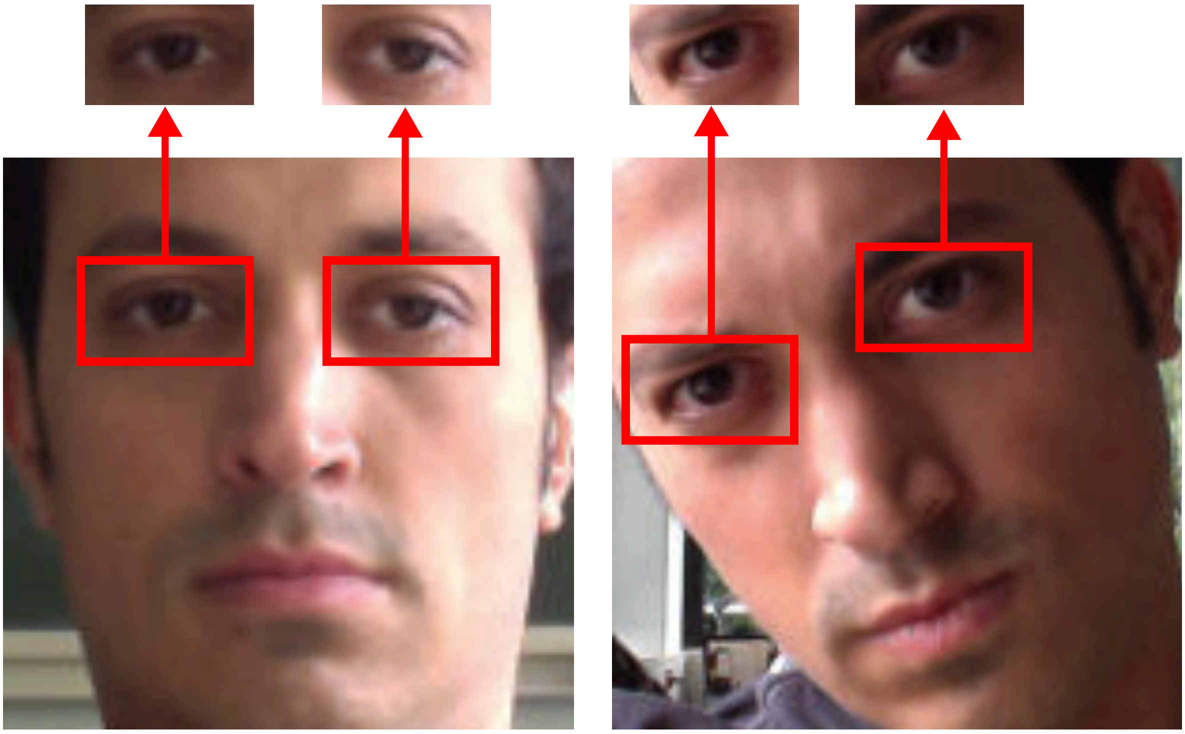


Figure 5.2: Visualization of head rotation factor. Left: face image and corresponding cropped eye images with nearly non-rotated head pose. Right: face image and corresponding cropped eye images with head pose rotation.

Figure 5.3 illustrates the effect of distance using a 3D eye region model from the UT Multiview dataset (Sugano *et al.*, 2014). We capture two eye images at two different distances between eye and camera (Figure 5.3a and Figure 5.3b). Closer distance (Figure 5.3a) naturally results in a larger image resolution, and usually image-based methods resize 2D input images (Figure 5.3d) so that they have the same resolution size. In this case, although Figure 5.3a and Figure 5.3b have the same 3D gaze direction, resized image Figure 5.3d and further distance image Figure 5.3b have slightly different image appearances. If we physically scale the 3D space by, e.g., changing the focal length (Figure 5.3c), the appearance difference between scaled (Figure 5.3c) and resized images (Figure 5.3d) is much smaller. This illustrates that image resizing is equivalent to 3D scaling rather than 3D shifting. It is important to precisely discuss the image resizing operation in data normalisation.

Pose-independent learning-based methods need to handle these factors causing appearance changes during training processes. However, practically speaking, it is almost impossible to train a gaze estimator with infinite variations of head poses and image resolutions. Therefore, image-based estimation methods require a *normalisation* technique to align training and test data and to constrain the input image to have a fixed range of variations. For example, image-based object recognition methods usually crop and resize the input image to a fixed image resolution while assuming that this operation does not affect the object label. The difficulty of data

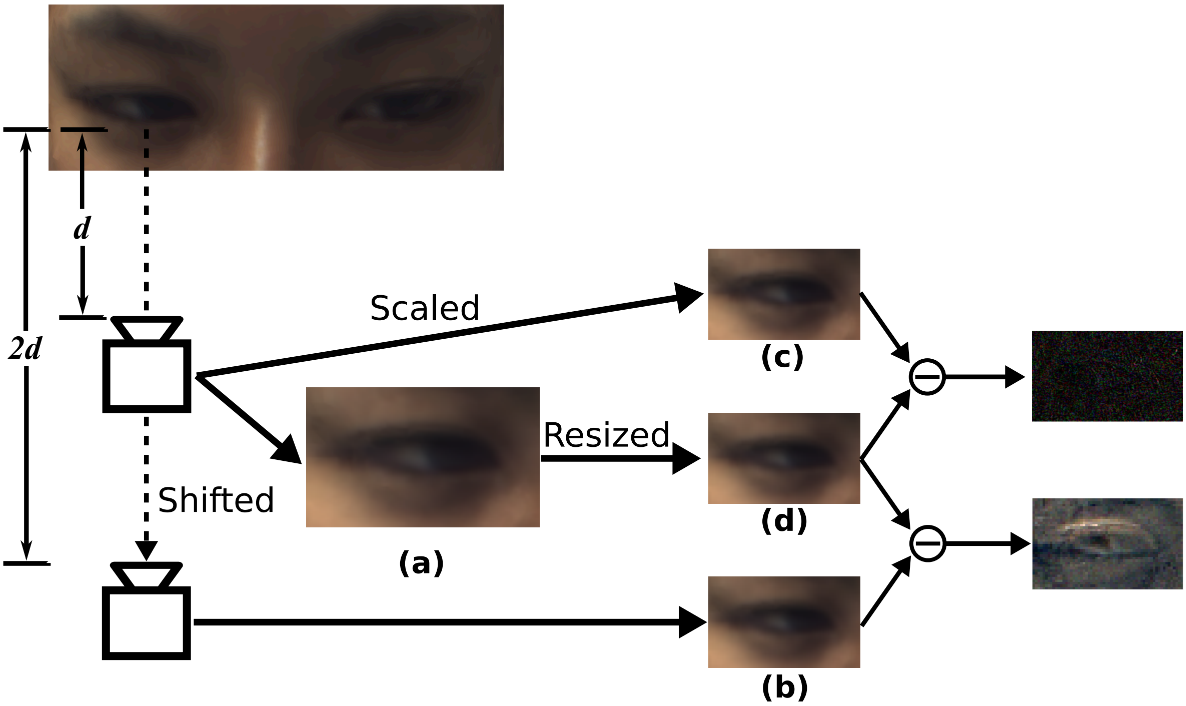


Figure 5.3: Visualization of distance factor. Eye image (b) is taken at distance d from the camera, and eye image (b) is shifted to distance $2d$ with half the size of (a). Eye images (c) and (d) are the eye images scaled and resized from (a). We calculate the image differences between (b) shifted and (d) resized, and (c) scaled and (d) resized, by subtracting each and normalizing the difference pixel values. Even though it is visually hard to tell, there is an appearance difference between the shifted and resized eye image.

normalisation in gaze estimation task is, however, the fact that eye image cropping, rotation, and resizing do affect their corresponding gaze labels. Gaze estimation is inevitably a geometric task, and it is important to properly formulate the normalisation operation.

For 3D data, such as UT Multiview (Sugano *et al.*, 2014), EYEDIAP (Funes Mora *et al.*, 2014) and UnityEye (Wood *et al.*, 2015), it is possible to render training and test samples so that they have the same camera at the same distance from the eye. However, for captured 2D images, such as MPIIGaze (Zhang *et al.*, 2018c) and GazeCapture (Krafka *et al.*, 2016), it is impossible to translate the eye. Nevertheless, we can still perform the approximation to crop the eye image properly.

5.2.2 Eye Image normalisation

We first summarize the detailed eye image normalisation procedure proposed in (Sugano *et al.*, 2014). The normalisation scheme aims at canceling variations in the eye image appearance as much as possible. The key idea is to standardize the translation

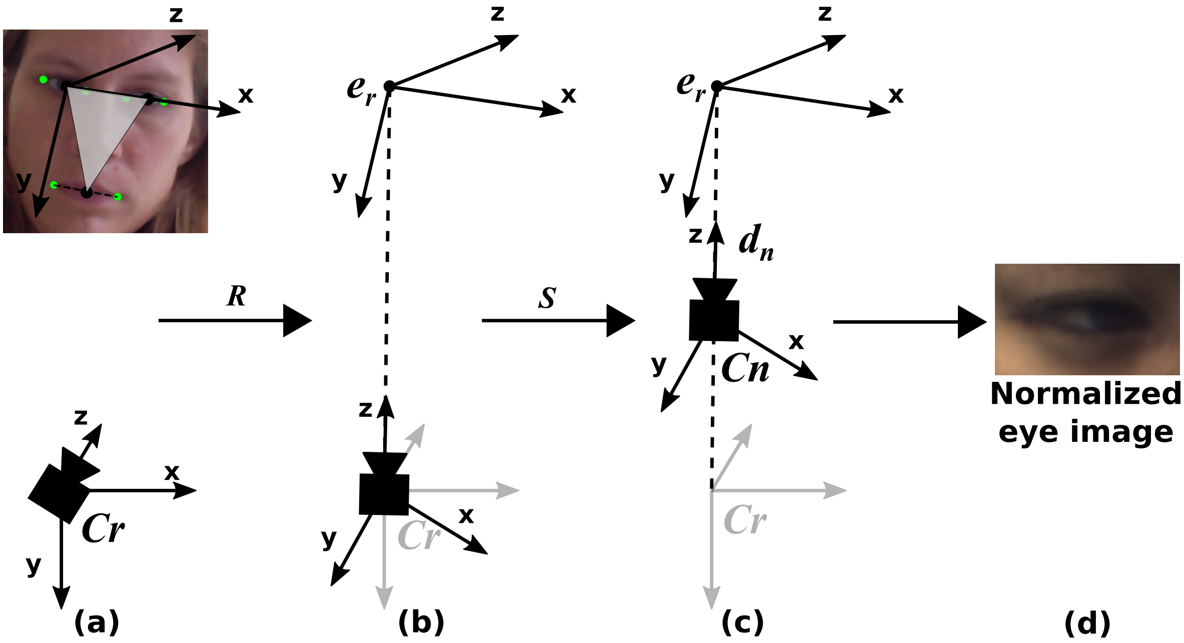


Figure 5.4: Basic concept of the eye image normalisation (Sugano *et al.*, 2014). (a) Starting from an arbitrary relationship between the head pose coordinate system centered at eye center $\boldsymbol{e}_r$ (top) and the camera coordinate system (bottom); (b) the camera coordinate system is rotated with a rotation matrix $\boldsymbol{R}$; (c) the world coordinate system is scaled with a scaling matrix $\boldsymbol{S}$; (d) the normalized eye images should be equivalent to the one captured with this *normalized* camera.

and rotation between camera and face coordinate system via camera rotation and scaling.

Figure 5.4 illustrates the basic concept of the eye image normalisation. As shown in Figure 5.4a, the process starts from an arbitrary pose of the target face. The pose is defined as a rotation and translation of the head coordinate system with respect to the camera coordinate system, and the right-handed head coordinate system is defined according to the triangle connecting three midpoints of the eyes and mouth. The x-axis is defined as the line connecting midpoints of the two eyes from right eye to left eye, and the y-axis is defined as perpendicular to the x-axis inside the triangle plane from the eye to the mouth. The z-axis is perpendicular to the triangle and pointing backwards from the face.

To simplify the notation of eye image normalisation, we use the midpoint of the right eye as the origin of the head coordinate system, and we denote the translation and rotation from the camera coordinate system to the head coordinate system as $\boldsymbol{e}_r$ and $\boldsymbol{R}_r$.

Given this initial condition, the normalisation process transforms the input image so that the normalized image meets three conditions. First, the *normalized* camera looks at the origin of the head coordinate system and the center of the eye is located at the center of the normalized image. Second, the x-axes of the head and camera coordinate systems are on the same plane, i.e., the x-axis of the head coordinate system appears as a horizontal line in the normalized image. Third, the normalized camera is located at a fixed distance d_n from the eye center and the eye always has the same size in the normalized image.

The rotation matrix $\boldsymbol{R}$ to achieve the first and second conditions can be obtained as follows. If we rotate the original camera to meet the first condition, the rotated z-axis of the camera coordinate system $\boldsymbol{z}_c$ has to be $\boldsymbol{e}_r$. To meet the second condition, the rotated y-axis has to be defined as $\boldsymbol{y}_c = \boldsymbol{z}_c \times \boldsymbol{x}_r$. $\boldsymbol{x}_r$ is the x-axis of the head coordinate system, and the y-axis of the rotated camera is defined to be perpendicular to both $\boldsymbol{z}_c$ and $\boldsymbol{x}_r$. Then, the remaining x-axis of the rotated camera is defined as $\boldsymbol{x}_c = \boldsymbol{y}_c \times \boldsymbol{z}_c$. Using these vectors, the rotation matrix can be defined as $\boldsymbol{R} = [\frac{\boldsymbol{x}_c}{\|\boldsymbol{x}_c\|}; \frac{\boldsymbol{y}_c}{\|\boldsymbol{y}_c\|}; \frac{\boldsymbol{z}_c}{\|\boldsymbol{z}_c\|}]$. In addition, the scaling matrix $\boldsymbol{S}$ to meet the third condition can be defined as $\text{diag}(1, 1, \frac{d_n}{\|\boldsymbol{e}_r\|})$. Therefore, the overall transformation matrix is defined as $\boldsymbol{M} = \boldsymbol{SR}$.

In the extreme case where the input is a 3D face mesh, the transformation matrix $\boldsymbol{M}$ can be directly applied to the input mesh and then it appears in the normalized space with a restricted head pose variation. Since the transformation is $\boldsymbol{M}$ defined as rotation and scaling, we can apply a perspective image warping with the transformation matrix $\boldsymbol{W} = \boldsymbol{C}_n \boldsymbol{M} \boldsymbol{C}_r^{-1}$ to achieve the same effect if the input is a 2D face image. $\boldsymbol{C}_r$ is the original camera projection matrix obtained from camera calibration, and $\boldsymbol{C}_n$ is the camera projection matrix defined for the normalized camera.

Sugano *et al.* (2014) introduced this idea to restrict the head pose variation when synthesizing training data for learning-based gaze estimation from 3D face meshes. Since we can assume test data always meets the above three conditions after normalisation, it is enough to render training images by placing virtual cameras on a viewing sphere around the eye center with radius d_n and rotating the camera to meet the first and second conditions. This data normalisation results in only 2 degrees of freedom, and significantly reduces the training space to be covered via learning-by-synthesis framework.

5.2.3 Modified Data normalisation

As discussed earlier, it is also important to properly handle the geometric transformation caused by the eye image normalisation and apply the same transformation to the gaze direction vector. If the input is training data and associated with a ground-truth gaze direction vector $\boldsymbol{g}_r$, it is necessary to compute the *normalized* gaze vector $\boldsymbol{g}_n$ which is consistent with the normalized eye image.

Assuming 3D data, Sugano *et al.* (2014) originally proposed to apply the same transformation matrix to the gaze vector as $\boldsymbol{g}_n = \boldsymbol{M}\boldsymbol{g}_r$. However, while in the 3D space the same rotation and translation should be applied to the original gaze vector $\boldsymbol{g}_r$, this assumption is not precise enough when dealing with 2D images. Since scaling does not affect the rotation matrix, the head rotation matrix after normalisation is computed only with rotation as $\boldsymbol{R}_n = \boldsymbol{R}\boldsymbol{R}_r$. For 2D images, image normalisation is achieved via perspective warping as $\boldsymbol{W} = \boldsymbol{C}_n\boldsymbol{M}\boldsymbol{C}_r^{-1}$. This operation implicitly assumes the eye region is a planar object, and if the eye is a planar object, scaling should not change the gaze direction vector.

Based on this discussion, in this chapter we propose a slightly modified version of the data normalisation process for 2D images. While the formulation of the image normalisation and the image transformation matrix $\boldsymbol{W}$ stays exactly the same, different with the original 2D data normalisation method, we propose to only rotate the original gaze vector to obtain the normalized gaze vector $\boldsymbol{g}_n = \boldsymbol{R}\boldsymbol{g}_r$. This formulation corresponds to an interpretation of the image transformation matrix $\boldsymbol{W}$ that the scaling $\boldsymbol{S}$ is applied to the camera projection matrix $\boldsymbol{C}_r$, instead of the 3D coordinate system. While this results in the exactly same image warping, it does not affect the physical space in terms of scaling and the gaze vector is only affected with the rotation matrix $\boldsymbol{R}$.

The transformation is also used to project back the estimated gaze vector to the original camera coordinate system. If the gaze vector estimated from the normalized eye image is $\hat{\boldsymbol{g}}_n$, the estimation result in the original camera coordinate system $\hat{\boldsymbol{g}}_r$ is obtained by rotating back $\hat{\boldsymbol{g}}_n$ as $\hat{\boldsymbol{g}}_r = \boldsymbol{R}^{-1}\hat{\boldsymbol{g}}_n$.

5.3 EXPERIMENTS

In this section, we validate the modified formulation of the data normalisation using both synthetic and real image datasets. In all experiments that follow, we used the AlexNet architecture (Krizhevsky *et al.*, 2012) as a basis for our appearance-based gaze estimation network and concatenated the normalized head angle vector and the first fully-connected layer, as done in (Zhang *et al.*, 2018c). The output of the network is a two-dimensional gaze angle vector $\boldsymbol{g}$ as polar angles converted from $\boldsymbol{g}_n$. As loss we used the Euclidean distance between estimated gaze angle vector $\hat{\boldsymbol{g}}$ and ground-truth gaze angle vector $\boldsymbol{g}$. During computing the final gaze estimation error, we first converted $\hat{\boldsymbol{g}}$ and $\boldsymbol{g}$ to $\hat{\boldsymbol{g}}_n$ and $\boldsymbol{g}_n$, and then projected them back to the original camera coordinate system to calculate the differences between direction vectors in degrees. We used the AlexNet pre-trained on the ImageNet dataset (Deng *et al.*, 2009) from the Caffe library (Jia *et al.*, 2014), and fine-tuned the whole network with gaze estimation training data depending on the particular experimental setting (see the respective section below for details). We used the Adam solver (Kingma and Ba, 2015) with the two momentum values set to $\beta_1 = 0.9$ and $\beta_2 = 0.95$, as well as the initial learning rate set to 0.00001. For data normalisation, we set the focal length for the normalized camera projection matrix and the distance d_n to be compatible with the UT Multiview dataset (Sugano *et al.*, 2014). The resolution of the normalized eye images was 60×36 pixels.

In this section, we refer to the original data normalisation method as *Original*, and the modified data normalisation method as *Modified*. We further analyze a naive baseline without any geometric transformation (*None*). For this baseline, we took the center of two eye corners as eye center, 1.5 times of the distance between two eye corners as eye width, and 0.6 times of eye width as eye height to crop the eye image. Last, we resized the eye image to the same 60 × 36 pixels.

5.3.1 Evaluation on Synthetic Images

While the main purpose of data normalisation is handling large variations in head pose, real-world datasets inevitably have limited head pose variations due to device constraints. To fully evaluate the effect of data normalisation on gaze estimation performance, we first use synthetic eye images with controlled head pose variations. We synthesized eye images using 3D face meshes of 50 participants provided by UT Multiview (Sugano *et al.*, 2014) to simulate 2D images that were captured with different head poses. We placed the 3D face mesh at random positions and rotations in the virtual camera coordinate system, and then rendered the image with the camera. The range of these randomizations was [-500 mm, 500 mm] for the x- and y-axes of the 3D face mesh position, [100 mm, 1500 mm] for the z-axis (distance between eye and camera), and [-30°, 30°] for head rotation around the roll, pitch and yaw axes, respectively. Note that we constrained the random position of the 3D face mesh so that the faces always fall inside the camera's field of view. The image resolution was set to 1280 × 720 pixels. Some examples of the rendered images are shown in the top row of Figure 5.5. Note that our own synthetic images were treated as 2D images in the following experiments, without access to the original 3D face mesh. The above process is introduced to simulate challenging input images with large head pose variations.

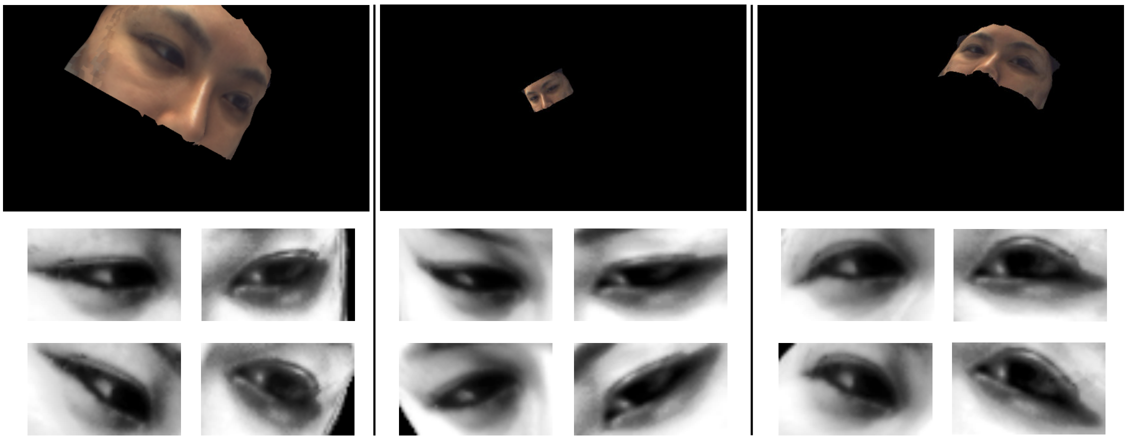


Figure 5.5: Example of our synthesized 2D images and corresponding eye images from UT Multiview. We first randomly rotated and translated the 3D face mesh in the camera coordinate system to render the 2D image (the top row), and performed the normalisation on the captured image to crop the eye image (the middle row), or directly crop the eye images (the bottom row) according to the eye corner landmarks as a naive baseline.

We then performed the data normalisation with *Original* or *Modified* methods on the rendered image to crop the eye images. The cropped eye images via 2D data normalisation are shown in the middle row of Figure 5.5, and we also show the cropped eye image from the *None* baseline as the bottom row of Figure 5.5. Note that the cropped eye images for *Original* and *Modified* are the same as the middle row of Figure 5.5, and the only difference is whether the gaze direction is scaled or not. UT Multiview contains 160 face meshes with different gaze directions for each 50 participant. Using this approach, we synthesized one 2D image for each face mesh, and flipped the cropped right eye images horizontally and trained them together with the left eye images. This finally resulted in $160 \times 2 = 320$ eye images for each of the 50 participants. Since this *None* baseline cannot take into account the eye position, we also prepared a position-restricted synthetic dataset to train and test a special version (*None (restricted)*) of the *None* baseline. During synthesis, we fixed the x- and y-axes of the 3D face mesh position and set them to zero, and the face center was always located in the image center. This way, only rotation and distance change in this dataset, and the *None (restricted)* baseline takes into account all information related to head pose variation.

5.3.1.1 *Test Data normalisation*

To evaluate the effectiveness of the data normalisation, we first evaluate the scenario where the training images are synthesized from 3D data under the normalized pose space, and 2D test images are cropped according to the normalisation schemes. We fine-tuned the AlexNet model on the synthetic data provided by the original UT Multiview dataset, and tested on our own synthetic samples that were processed with the *Original*, *Modified* or *None* baseline.

We converted the output gaze angle vector from the model $\boldsymbol{g}$ to a gaze direction vector $\boldsymbol{g}_n$, and then projected it back to the original camera coordinate system depending on the normalisation method: For *Original*, we computed the gaze direction vector in the original camera space with transformation matrix $\boldsymbol{M}$ as $\boldsymbol{g}_r = \boldsymbol{M}^{-1}\boldsymbol{g}_n$. For the *Modified* method, we computed the gaze direction vector in the original camera space with rotation matrix $\boldsymbol{R}$ as $\boldsymbol{g}_r = \boldsymbol{R}^{-1}\boldsymbol{g}_n$. For the *None* baseline, we directly took the output from the model as the final gaze direction $\boldsymbol{g}_r$ in the original camera space.

The results are shown in Figure 5.6 with the gaze estimation performances for *None*, *Original* and *Modified*. The bars show the mean gaze estimation error in degrees, and the error bars show the standard deviation across all participants. As can be seen from the figure, the *Modified* method outperforms the other two methods significantly. Since the only difference between *Original* and *Modified* is scaling the gaze direction or not, the better performance achieved by *Modified* over *Original* indicates the scaling on gaze direction actually hurts the performance. This is because the scaling factor is not suitable to apply on gaze direction here, since the eye region in the input image is a planar object. Such a scaling factor even makes the performance worse than the *None* baseline without any data normalisation. The *Modified* outperforms over the *None* baseline significantly with 32.0% (from 14.7 degrees to 10.0 degrees), clearly

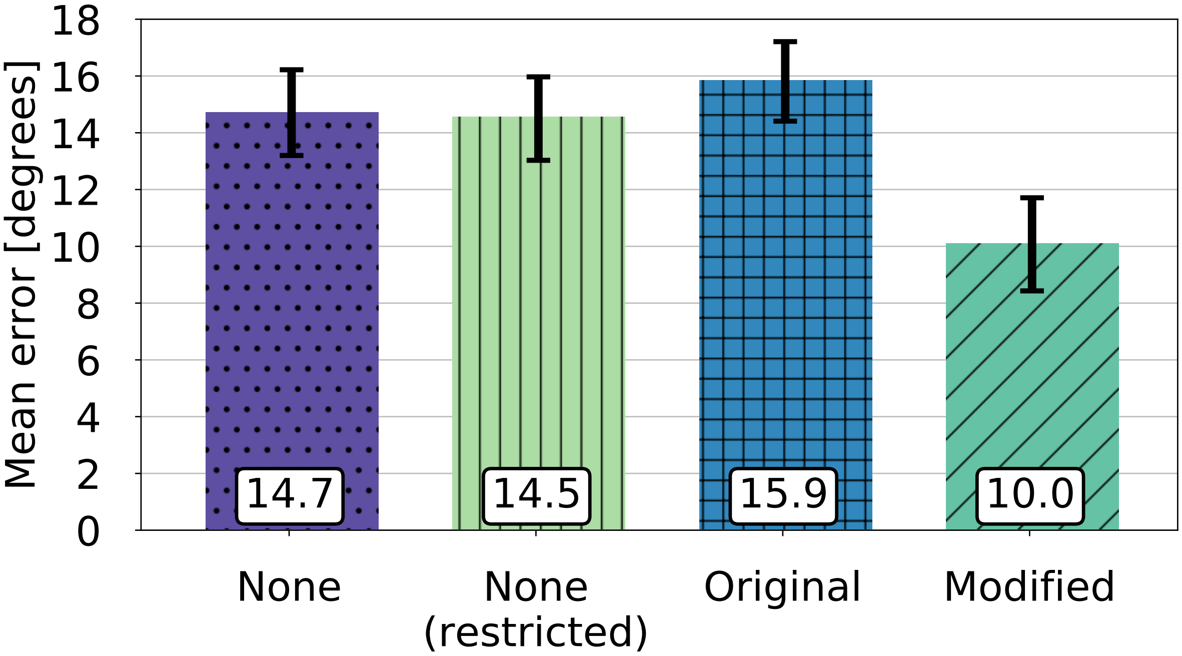


Figure 5.6: Gaze estimation error in degrees of visual angle for data normalisation methods *Original* and *Modified*, and *None* baselines with the gaze estimation network fine-tuned on UT Multiview, and tested on our synthetic samples. Bars show the mean error across participants and error bars indicate standard deviations.

showing the benefits of data normalisation for handling the variations caused by head poses. *None (restricted)* achieved slightly better but insignificant performance improvements ($p < 0.01$, paired Wilcoxon signed rank test) over the *None* baseline. This indicates that this naive baseline cannot achieve performance comparable to data normalisation even if face positions in the image are restricted.

Figure 5.7 further shows the gaze estimation error for different distances between camera and eye. To generate a smooth curve, we used least squares polynomial fitting. As can be seen from the figure, the gaze estimation error of the *Modified* method only slightly increases with increasing distance. A similar trend can also be observed for the *None* baseline. In contrast, the *Original* data normalisation method encodes distance information in the gaze direction. This results in an increased gaze estimation error, particularly for small distances. When projecting the gaze direction back to the original camera coordinate system, the gaze direction will be scaled with the inverse scaling matrix $\mathbf{S}$. In consequence, the gaze direction is narrowed when the sample has bigger distance than d_n, and the gaze direction is expanded when the sample has smaller distance than d_n. This causes the larger gaze estimation error on the smaller distances. Finally, given that the scaling matrix $\mathbf{S}$ for *Original* becomes the identity matrix when the distance between camera and eye is d_n, the gaze estimation error is the same for *Original* and *Modified* at that normalisation distance.

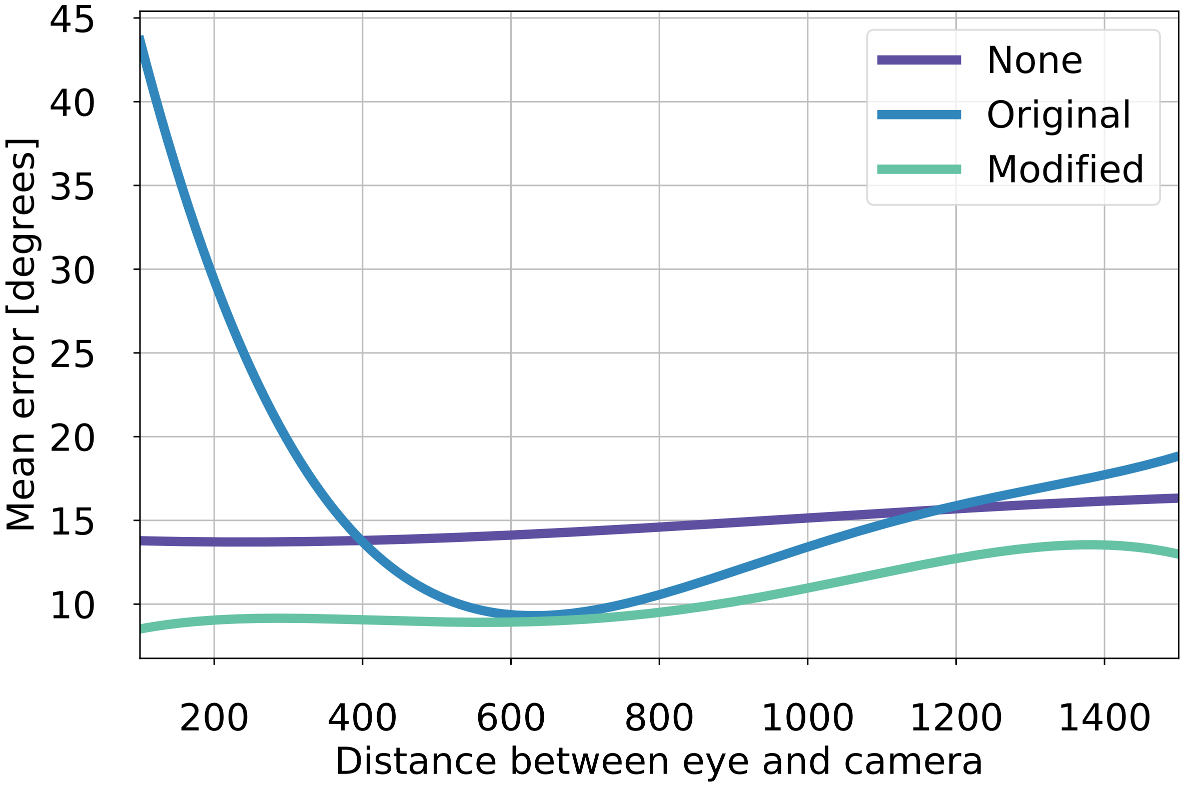


Figure 5.7: Gaze estimation error for the different normalisation methods and different distances between camera and eye. Curves plotted using least squares polynomial fitting.

5.3.1.2 *Training Data normalisation*

In this section, we further evaluate the model trained and tested on the data generated from 2D images. While the model was trained on the data generated directly with the 3D face mesh in the previous evaluation scenario and our synthetic data was used only as test data, in this section we split our synthetic images into training and test data. In this case, the training and test samples were both processed via the *Original*, *Modified* or *None* methods, respectively. We performed a 5-fold cross-person evaluation on the 16,000 synthesized samples from 50 participants.

The results of this evaluation are shown in Figure 5.8. The bars show the mean gaze estimation error in degrees, and the error bars show the standard deviation across all participants. As can be seen from the figure, in this setting, both data normalisation methods achieve better performances than the *None* baseline, suggesting that the data normalisation benefits the model training. The *None* baseline performed the worst because the noisy training data with head rotation makes the model training difficult. Restricting the face position does not improve performance, as indicated by the *None (restricted)* baseline. For *Original*, both training and test samples were rotated and also scaled in the same way, which corresponds to mapping the gaze direction into a scaled space. This does not result in large gaze

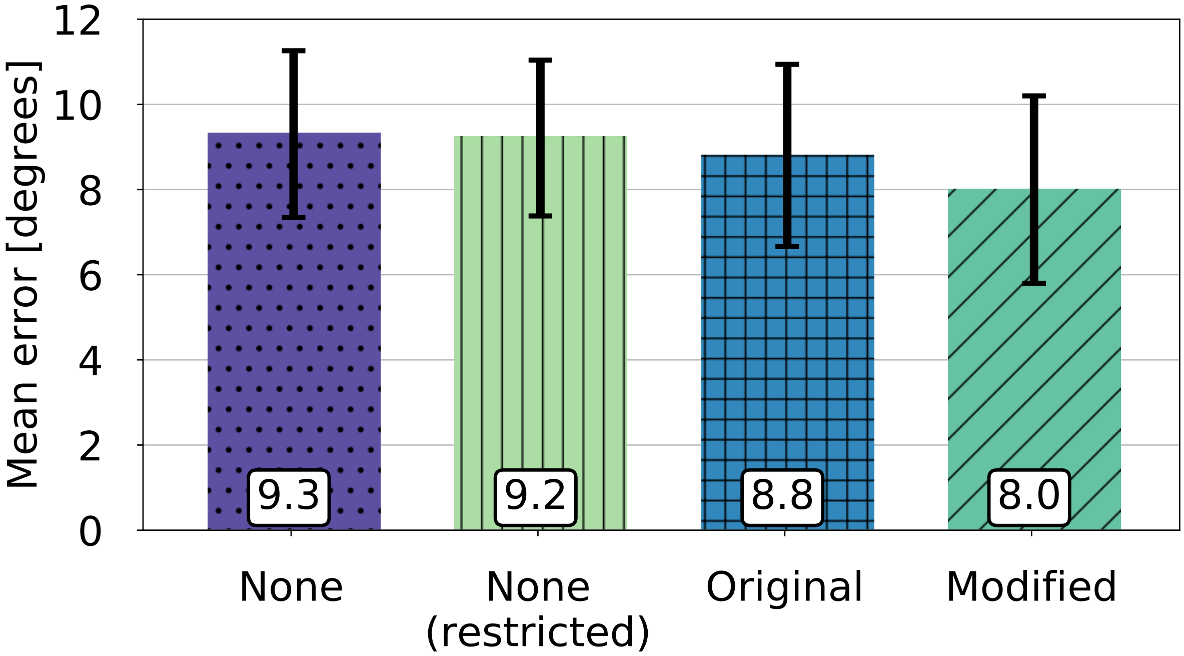


Figure 5.8: Gaze estimation error in degrees of visual angle for data normalisation method *Original* and *Modified*, and *None* baseline with the gaze estimation network fine-tuned and tested on our synthetic samples. Bars show the mean error across participants and error bars indicate standard deviations.

estimation error when projecting the gaze direction back to the original camera coordinate system. However, as we already saw, the *Modified* formulation handles the normalisation task more accurately and hence overall performance was still improved.

5.3.2 Evaluation on Real Images

We then evaluated the impact of data normalisation using real images from the MPIIGaze dataset (Zhang *et al.*, 2018c). As discussed earlier, real images have stronger device constraints, and in terms of head pose, it has smaller variations than the previous case. The MPIIGaze dataset consists of a total of 213,659 images collected on the laptops of 15 participants over the course of several months using an experience sampling approach. Therefore, most of the head poses in the MPIIGaze dataset are restricted to the natural and typical ones in front of a laptop webcam. One important question is whether data normalisation contributes to the estimation performance even with a restricted head pose range.

5.3.2.1 *Test Data normalisation*

We first performed the simple cross-dataset evaluation, which we trained the model on the UT Multiview dataset and tested on the MPIIGaze dataset. We used the

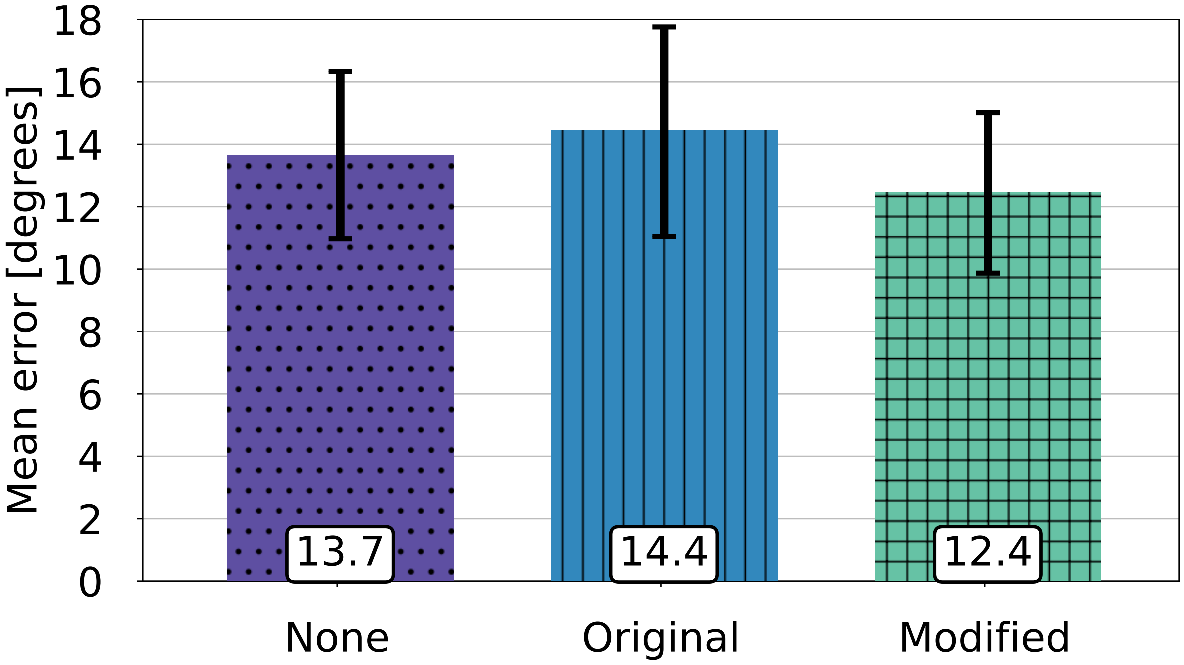


Figure 5.9: Gaze estimation error in degrees of visual angle for data normalisation method *Original* and *Modified*, and *None* baseline with the gaze estimation network fine-tuned on UT Multiview, and tested on MPIIGaze. Bars show the mean error across participants and error bars indicate standard deviations.

same normalized camera projection matrix, normalized distance ($d_n = 600mm$), and images size (60×36 pixels) as before.

The results are shown in Figure 5.9. The bars show the mean gaze estimation error in degrees, and the error bars show the standard deviation across all participants. As can be seen from the figure, the ranking in terms of performance is the same as in Figure 5.6. That is, the *Modified* method achieved the best performance and the *Original* method achieved the worst performance. The *None* baseline has the second-best performance. This analysis confirms that encoding distance information by scaling gaze direction in the *Original* method is not helpful since the eye region is planar in the input 2D image.

The relative improvement achieved by the *Modified* method over the *None* baseline becomes smaller compared to Figure 5.6 (9.5% vs 32.0%). This is because the head rotation in MPIIGaze data as shown in Zhang *et al.* (2018c) is much narrower compared to our synthesized samples from UT Multiview.

5.3.2.2 *Training Data normalisation*

Last, we repeated the training on 2D images evaluation on MPIIGaze using a leave-one-person-out approach. The training and test sets were both processed via the *Original*, *Modified* or *None* methods, respectively. The results are shown in Figure 5.10. The bars show the mean gaze estimation error in degrees, and the error bars show

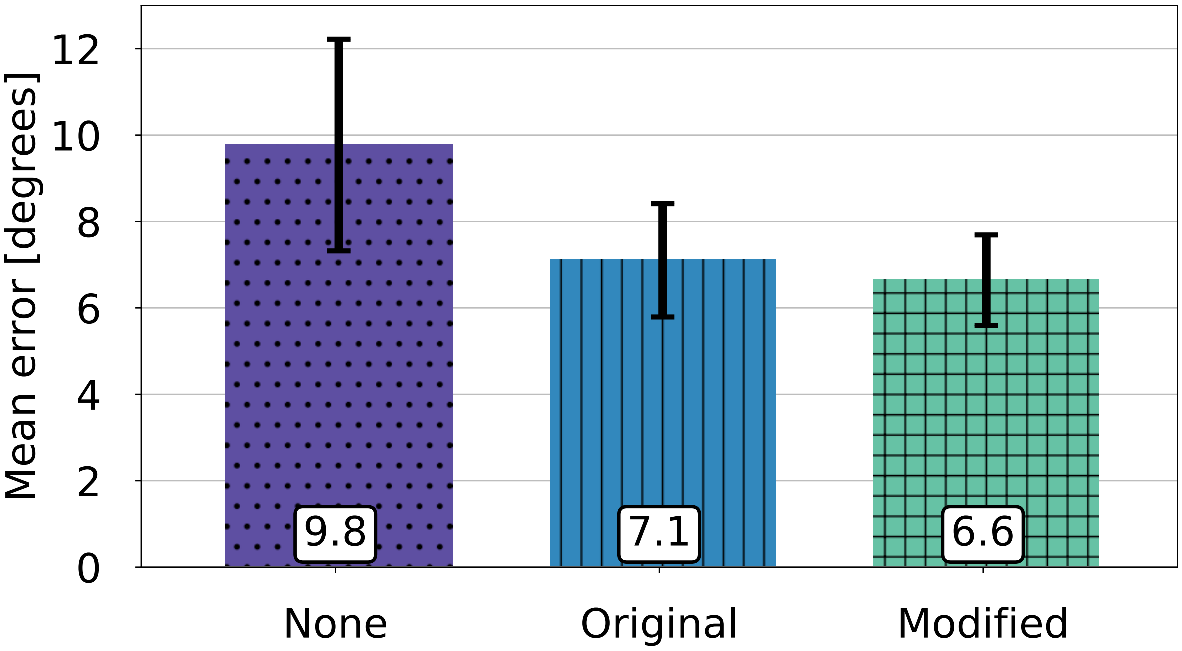


Figure 5.10: Gaze estimation error in degrees of visual angle for data normalisation methods *Original* and *Modified*, and *None* baseline with the gaze estimation network fine-tuned and tested on MPIIGaze. Bars show the mean error across participants and error bars indicate standard deviations.

the standard deviation across all participants. The figure shows that performance order for the different methods is similar to Figure 5.8. Both *Original* and *Modified* achieved better performances than the *None* baseline, while *Modified* again achieved the best performance. As such, this analysis confirms that the data normalisation can lead to better performance for both synthetic and real data, and that the *Modified* data normalisation method can achieve better performance than the *Original* data normalisation method.

The relative improvement achieved by the *Modified* method over the *None* baseline when evaluating on synthetic (see Figure 5.8) and real (see Figure 5.10) data increased from 16.3% to 32.7% despite the fact that the head rotation range is smaller for real data from MPIIgaze. This is most likely because for the real data, the model has to handle variations that never appeared in synthesized data, such as different illumination conditions. The variability caused by the head rotation becomes crucial during model learning for the *None* baseline since the model has to handle additional variations. This suggests that data normalisation is particularly beneficial for the case of training and testing on 2D images, which is the practically most relevant case for appearance-based gaze estimation.

5.4 CONCLUSION

In this chapter we modified the data normalisation method for appearance-based gaze estimation initially proposed in (Sugano *et al.*, 2014). We demonstrated the importance of eye image appearance variations caused by different head poses, and provided detailed explanations and discussions on how data normalisation can cancel most of these variation to make the model learning more efficient. We showed that data normalisation can result in significant performance improvements between 9.5% and 32.7% for different evaluation settings using both synthetic and real image data. These results underline the importance of data normalisation for appearance-based methods, particularly in unconstrained real-world settings. As such, we strongly recommend data normalisation as the default pre-processing step for appearance-based gaze estimation.

Part II

ATTENTIVE USER INTERFACES

Attentive user interfaces perform interaction according to current user attentional focus and capacity, and gaze-contingent systems are one type of applications (Bulling, 2016). With the development of our gaze estimation methods, we are first to be able to estimate gaze of users with a single webcam under challenging real-world environments. Given the current gaze estimation methods cannot provide practical accurate gaze estimates without dedicated personal calibration, it still requires effort to design an elegant interface that can work with raw results or smart way to collect domain-specific data. In this part, we specific propose a novel eye contact detection method for arbitrary objects without any calibration, and a personal gaze estimator can use data from multiple personal devices.

In Chapter 6, we contribute a detector for both human-object and human-human eye contacts. Our method collects and labels the samples and improve accuracy since it is deployed by assuming the target object is visually salient and closest to the camera. Since it only needs the relative position of camera and target object, there is no need for prior knowledge about the camera, object and scene. Consequently, our method can work with arbitrary cameras and objects without the need for tedious and time-consuming manual data annotation. In Chapter 7, we propose the first solution to train a personal gaze estimator using data from multiple devices. To address the key challenge as ambiguity caused by device-specific properties, we use shared feature layers to learn the generic feature for gaze estimation task, and encoders/decoders to adapt the device-specific features. Detailed evaluations on a new dataset of interactions with five common devices demonstrate the significant potential of multi-device training.

6 EVERYDAY EYE CONTACT DETECTION USING UNSUPERVISED GAZE TARGET DISCOVERY

Eye contact is an important non-verbal cue in social signal processing and promising as a measure of overt attention in human-object interactions and attentive user interfaces. However, robust detection of eye contact across different users, gaze targets, camera positions, and illumination conditions is notoriously challenging. We present a novel method for eye contact detection that combines a state-of-the-art appearance-based gaze estimator with a novel approach for unsupervised gaze target discovery, i.e. without the need for tedious and time-consuming manual data annotation. We evaluate our method in two real-world scenarios: detecting eye contact at the workplace, including on the main work display, from cameras mounted to target objects, as well as during everyday social interactions with the wearer of a head-mounted egocentric camera. We empirically evaluate the performance of our method in both scenarios and demonstrate its effectiveness for detecting eye contact independent of target object type and size, camera position, and user and recording environment.

6.1 INTRODUCTION

Eye contact plays an important role in the social, behavioural, and computational sciences. Eye contact on objects in the environment contains valuable information for understanding everyday attention allocation (Land and Hayhoe, 2001), while eye contact between humans is fundamental for social communication (Kleinke, 1986). As a consequence, eye contact detection emerged as an important building block for attentive user interfaces (Shell *et al.*, 2003a), assistive systems (Ye *et al.*, 2012), lifelogging (Dickie *et al.*, 2004a), or human-robot interaction (Imai *et al.*, 2003).

A large body of work has explored the use of eye tracking for eye contact detection (Shell *et al.*, 2004; Zhang *et al.*, 2015b). However, existing commercial eye tracking systems require dedicated hardware, such as infrared illumination, personal calibration, or high-quality images to achieve good performance. While state-of-the-art appearance-based methods have improved in terms of robustness (Zhang *et al.*, 2017d), gaze estimation accuracy is still not sufficient to detect eye contact on small objects. Also, current methods still require adaptation to the specific environment as well as camera-screen setup (Sugano *et al.*, 2016). Generic, yet still accurate, long-range gaze estimation remains challenging.

While gaze estimation can be seen as a regression task to infer arbitrary gaze directions, eye contact detection is a binary classification task to output whether the user is looking at a target or not. Consequently, several previous works tried

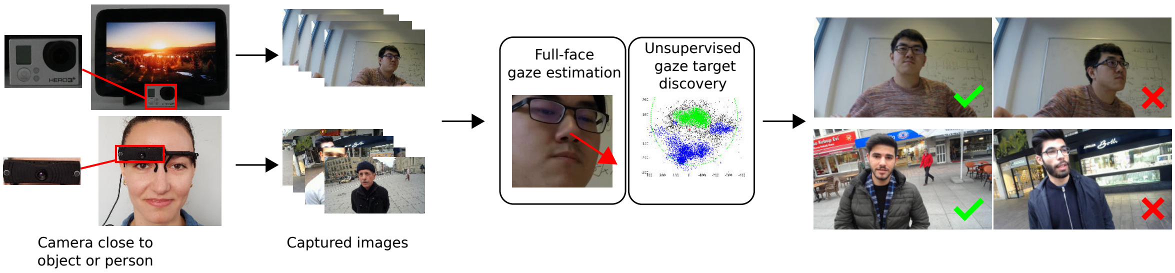

Camera close to object or person
Captured images
Full-face gaze estimation
Unsupervised gaze target discovery

Figure 6.1: We present a method for everyday eye contact detection. Our method takes images recorded from an off-the-shelf RGB camera close to a target object or person as input. It combines a state-of-the-art appearance-based gaze estimator with a novel approach for unsupervised gaze target discovery, i.e. without the need for tedious and time-consuming manual data annotation.

to transform the task by designing dedicated eye contact detectors, for example using embedded sensors consisting of a camera and infrared LEDs (Shell *et al.*, 2004, 2003b; Vertegaal *et al.*, 2002) or by using machine learning (Edmunds *et al.*, 2017; Smith *et al.*, 2013; Ye *et al.*, 2015). While the shift from regression to classification can potentially make the eye contact detection task easier, from a practical perspective two fundamental challenges remain.

First, the classification boundary between eye contact and non-eye-contact always depends on the target object. For learning-based eye contact detection, the algorithm first needs to identify the size and location of the target object with respect to the camera, and requires dedicated training data for training the target-specific eye contact detector. Without such prior knowledge, training a generic eye contact detector, i.e. a detector that works even for very small target sizes and locations close to the camera, is as difficult as training a generic gaze estimator. Second, the difficulty of handling different environments still prevents robust and accurate detection. Bridging the gap between training and test data is one of the most difficult issues even with state-of-the-art machine learning algorithms, and preparing appropriate training data for target users and environments is almost impossible in practical scenarios.

In this chapter we approach appearance-based eye contact detection from a novel perspective. We exploit the fact that visual attention tends to be biased towards the centre of objects and faces and that the fixation distribution consequently has a centre-surround structure around gaze targets (Nuthmann and Henderson, 2010). Our key idea is to use an *unsupervised* data mining approach for collecting on-site training data. Instead of training a generic eye contact detector beforehand, our method automatically acquires training data during deployment and adaptively learns an eye contact detector specific to the target user, object, and environment. The appearance-based gaze estimation model (Zhang *et al.*, 2015a) is first used to infer an inaccurate spatial gaze distribution, and we show that eye contact images can be identified by clustering analysis even with such low-precision gaze data. The clustering result is used to create positive and negative training labels and to train a dedicated eye contact detector. Our method transforms arbitrary cameras into eye contact sensors that perform accurately when the target is visually salient and the closest to the camera.

Our contributions are threefold. First, we present a novel camera-based method for eye contact detection, which automatically adapts to the arbitrary eye contact target object. Second, we also present a new *in-the-wild* dataset for eye contact detection, under two different and complementary settings: stationary object-mounted and mobile head-mounted cameras. Third, using the dataset, we quantify the performance of our method and discuss the fundamental limitation of existing approaches on eye contact detection.

6.2 UNSUPERVISED EYE CONTACT DETECTION

Our method for unsupervised eye contact detection only requires a single off-the-shelf RGB camera placed close to the target object. As illustrated in Figure 3.1, during training, our method first detects the face and facial landmarks in the images obtained from the camera and then applies a state-of-the-art full-face appearance-based gaze estimation method. Estimated gaze directions are clustered and the sample cluster corresponding to the target object is identified. The clustering result is then used to label samples with positive and negative eye contact labels, and the labelled samples are used to train a two-class SVM for eye contact detection from high-dimensional features extracted from the gaze estimation CNN. During testing, the input CNN features are fed into the learned two-class SVM to predict eye contact on the desired target object.

6.2.1 Gaze Estimation and Feature Extraction

In this chapter, we use the full-face method proposed in (Zhang *et al.*, 2017d) for the initial gaze estimation. We train the CNN model using two publicly available gaze datasets, MPIIGaze (Zhang *et al.*, 2015a) and EYEDIAP (Funes Mora *et al.*, 2014), to maximise variability in illumination conditions, as well as head pose and gaze direction ranges. We use the same face detection (King, 2009), facial landmark detection (Baltrušaitis *et al.*, 2016) and data normalisation methods as in (Zhang *et al.*, 2017d). Data normalisation is employed to handle different hardware setups using a perspective warp from an input face image to a normalised space with fixed camera parameters and reference point location. The face image is fed into the CNN model to predict a gaze direction vector $\boldsymbol{g}$. Assuming dummy camera parameters, the gaze direction vector $\boldsymbol{g}$ is projected to the camera image plane and converted to on-plane gaze locations $\boldsymbol{p}$. While the gaze estimation results are used for sample clustering, we also extract a 4096-dimensional face feature vector $\boldsymbol{f}$ from the first fully-connected layer of the CNN. To leverage the full descriptive power of the CNN model, this feature vector is used as input to the eye contact detector.

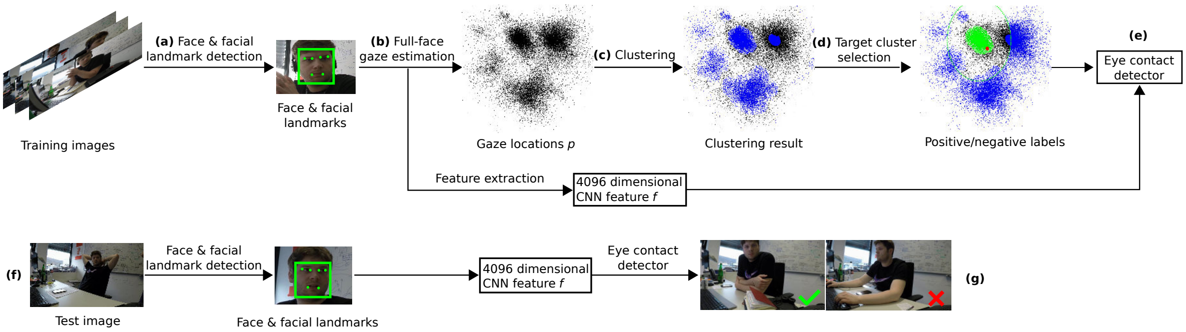


Figure 6.2: Overview of our method. Taking images from the camera as input, our method first detects the face and facial landmarks (a) and estimates gaze directions $\boldsymbol{p}$ and extracts CNN features $\boldsymbol{f}$ using a full-face appearance-based gaze estimation method (b). During training, the gaze estimates are clustered (c) and samples in the cluster closest to the camera get a positive label while all others get a negative label (d). These labelled samples are used to train a two-class SVM for eye contact detection (e). During testing (f), the learned features $\boldsymbol{f}$ are fed into the two-class SVM to predict eye contact on the desired target object or face (g).

6.2.2 Sample Clustering and Target Selection

The estimated gaze direction $\boldsymbol{g}$ is not accurate enough even using a state-of-the-art method, and it cannot be mapped directly to the physical space without accurate camera parameters. However, it at least indicates the relative gaze direction from the camera position, and hence gaze direction clusters corresponding to physical objects can be observed. Consequently, in the next step, gaze directions are clustered into different clusters that are assumed to correspond to different objects. The cluster closest to the camera position is finally selected as belonging to the target object. To filter out unreliable samples from the clustering process, we reject samples whose facial landmark alignment score is below a threshold θ. Since these unreliable samples often correspond to non-frontal faces, we directly use them as negative samples during training. We then use the OPTICS algorithm (Ankerst *et al.*, 1999) to cluster the samples. Since the OPTICS algorithm is a density-based hierarchical clustering algorithm, it tends to create a child cluster at the centre of a parent cluster with the same centroid. In our method we discard such a recursive hierarchy, and adopt the largest cluster spatially separated from other clusters.

Given that our method assumes that the camera is close to the target object, samples in the nearest cluster to the camera position (the origin of the camera image plane) are used as positive training samples. Other clusters are assumed to correspond to other objects, and samples from these clusters are used as negative samples. In addition, given that there tend to be many samples labelled as noise by the OPTICS algorithm, we set a safe margin d around the positive cluster, and we also use samples outside the safe margin as negative samples.

6.2.3 Eye Contact Detection

Labelled samples obtained from the previous step are used to train the eye contact classifier. Since the number of positive and negative samples can be highly unbalanced, we use a weighted SVM classifier (Xanthopoulos and Razzaghi, 2014). As mentioned before, we use a high-dimensional feature vector f extracted from the gaze estimation CNN to leverage richer information instead of only gaze locations. We first apply PCA to the training data and reduce the dimensionality so that the PCA subspace retains the 95% variance. After the training phase, input images are fed into the same preprocessing pipeline with the face and facial landmark detection, and feature f is extracted from the same gaze estimation CNN. It is then projected to the PCA subspace, and the SVM classifier is applied to output eye contact labels.

6.3 EXPERIMENTS

To evaluate our method for eye contact detection, we collected two real-world datasets with complementary characteristics in terms of target object type and size, stationary and mobile setup, as well as single-user and multi-user assumptions. We evaluated our method and different baselines on both datasets and analysed performance across different objects, camera positions, and duration of training data collection.

6.3.1 Data Collection

Data collection was performed for two challenging real-world scenarios: In the office scenario (see Figure 6.3, left) cameras were *object-mounted* and we aimed to detect eye contact of a single user with these target objects during everyday work at their workplace. We used the participant's main work display as one of the targets, and put the camera in three different but imprecisely defined locations: above, below, and next to the display. In addition, we placed a tablet or clock as target objects on the participant's desk and put a camera close to them. The tablet was configured to show different videos and images in a loop, simulating a digital picture frame. This was to make the dataset more variable with respect to target object saliency/distractiveness, as well as target size and position with respect to the user and camera. We recorded 14 participants in total (five females) and each of them recorded four videos: one for the clock, one for the tablet, and two for the display with two different camera positions. The recording duration for each participant ranged between three and seven hours.

In the interaction scenario (see Figure 6.3, right) a user was wearing a *head-mounted* camera while being engaged in everyday social interactions. This scenario was complementary to the office scenario in that the user's head/face was the target and we aimed to detect eye contact of different interlocutors. We recruited three *recorders* (all male) and recorded them while they interviewed multiple people on

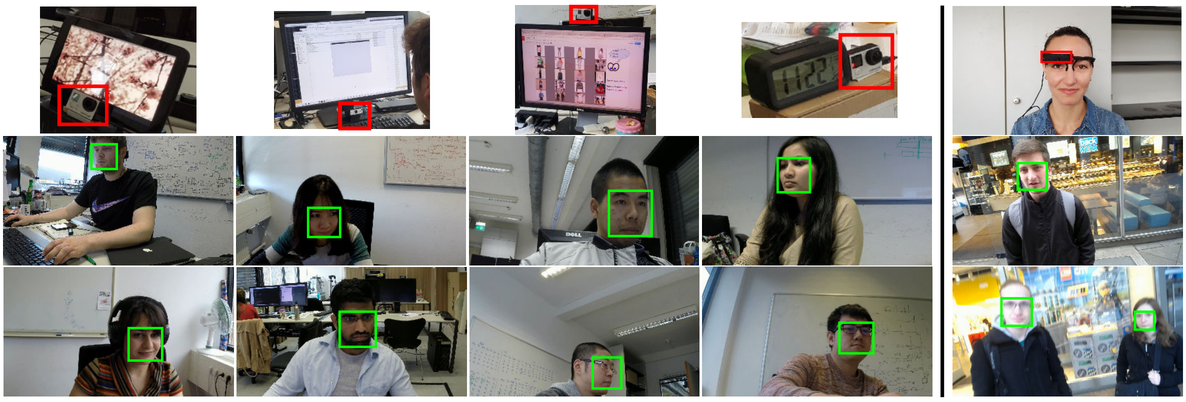


Figure 6.3: Sample recording settings and images for eye contact detection using object-mounted (left) and head-mounted (right) cameras. The first row shows the targets with cameras marked in red; the second and third rows show sample images captured by the camera, as well as detected face bounding boxes. The images show the considerable variability in terms of illumination, face appearance, and head pose as well as motion blur (in case of the head-mounted camera).

the street. In total, this resulted in five hours of video covering 28 social interactions. As can be seen from Figure 6.3, the head-mounted camera used in our experiments is rather bulky, and we cannot exclude the possibility that it attracted the attention of the second person, instead of the face. However, the camera was positioned close to the centre of the face, so we expected the resulting error to be very low and visual inspection of a random subset of the images confirmed this expectation.

For both scenarios, we used the first 75% of the data for training our method (the training set) and the remaining 25% for testing (the test set). We uniformly sampled 5,000 images from the test set and asked two annotators to manually annotate them with binary ground-truth eye contact labels, i.e. if the person was looking at the target (object or person) or not. Each of the two annotators annotated disjunct halves of the test sets. We also asked another third annotator to check these annotations and flag incorrect ones, and flagged images were annotated again by the same corresponding annotator. Annotators were asked to judge eye contact from the detected face, with detailed knowledge about the physical setup of each recording, including the target object and camera locations.

6.3.2 Implementation Details

We set the facial landmark detection threshold θ to -0.7 (-1.0 is the best detection and 1.0 is the worst), which rejected 45.7% of the detected faces during training. The minimum number of samples per cluster in the OPTICS algorithm was set to $N/50$, where N is the total number of samples used for clustering. The safe margin d was 10σ, where σ is the standard deviation of the sample distances from the centre of the cluster. On a PC with an Intel(R) Xeon(R) 3.30GHz CPU and an Nvidia GeForce

GTX TITAN GPU our method achieved 14 fps.

6.3.3 Baseline Methods

We compared the performance of our method with the following five baselines, covering both prior works as well as variants of our proposed method.

GazeLocking. The *GazeLocking* method proposed in Smith *et al.* (2013) performs eye contact detection by training a SVM classifier in a fully supervised manner using an eye image dataset with ground-truth labels. It assumes aligned faces recorded of people using a chin rest. For a fair comparison, we adapted the *GazeLocking* method to use the same CNN-based classification architecture as our proposed method to train the eye contact detector from the Columbia dataset. When evaluated on the test set of Columbia, it was confirmed that the adapted method achieved a similar performance (MCC = 0.83) as reported in Smith *et al.* (2013).

Face Clustering. Some recent work (Duffner and Garcia, 2016; Recasens *et al.*, 2015) used face images to infer coarse gaze directions. A key advantage of our method is that it relies on a state-of-the-art appearance-based gaze estimator to obtain the initial features for the unsupervised gaze target discovery. To evaluate the benefits of this approach, for this baseline we directly used the face features $\boldsymbol{f}$ extracted from the CNN model as input to the clustering.

Gaze Classification. Similarly, our method uses face features $\boldsymbol{f}$ for training the eye contact detector. To assess the contribution of the face feature representation, for this baseline we instead used gaze locations $\boldsymbol{p}$ for both sample clustering and eye contact detector training.

Gaze Projection. Raw gaze direction has recently been used to estimate visual attention on public displays (Huang *et al.*, 2016a; Sugano *et al.*, 2016). For this baseline, we manually measured the physical size of the target object and its position related to the camera, and projected the object as bounding box on the camera image plane. The input image was classified as eye contact if the estimated gaze location was inside the bounding box. Therefore, this method assumes accurate knowledge of the target object location.

Head Orientation Projection. Finally, head orientation has also been used for visual attention estimation (Stiefelhagen *et al.*, 2002; Voit and Stiefelhagen, 2008; Yoo *et al.*, 2010), especially when the target face image is low-resolution and accurate gaze estimation cannot be expected. Hence, for this baseline, we obtained 3D head orientations from input faces by fitting 2D facial landmark detections to a 3D face model as in Zhang *et al.* (2015a), and calculated the intersection of the head orientation vector and camera image plane. The input frame was classified as eye contact if the intersection is inside the object bounding box as we described the in

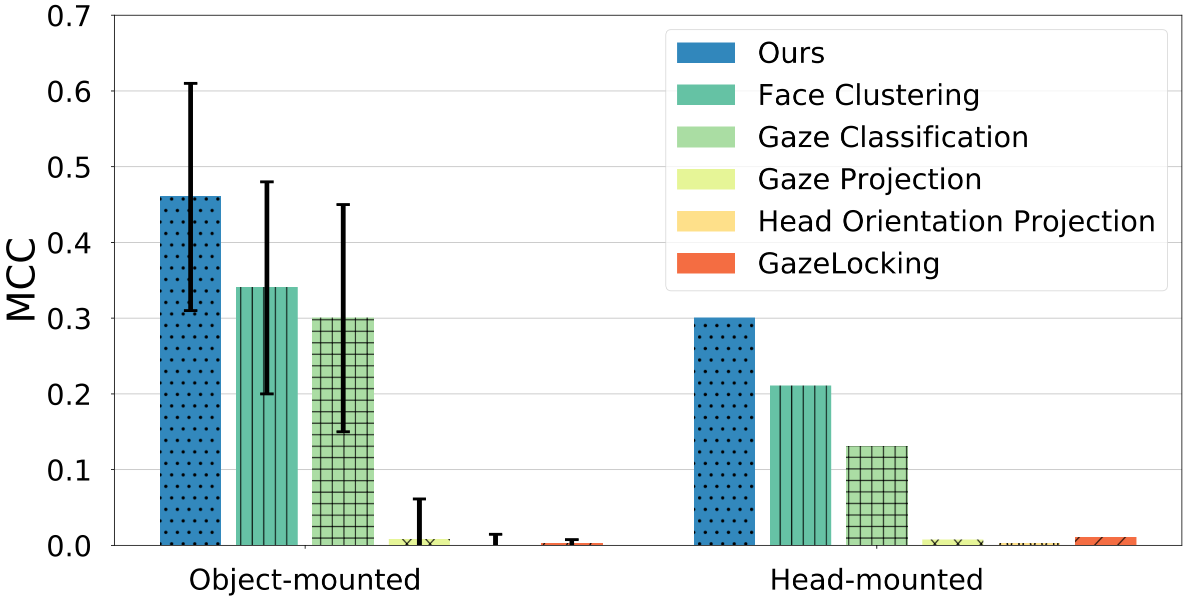


Figure 6.4: Performance of the different methods for the *object-mounted* (left) and *head-mounted* setting (right) across participants. The bars are the MCC value and error bars indicate standard deviations across participants.

Gaze Projection method.

In our experimental setup, we achieved 16 frames per second (FPS) for *GazeLocking*, 18 FPS for *Face Clustering*, 14 FPS for *Gaze Classification*, 13 FPS for *Gaze Projection*, and 22 FPS for *Head Orientation Projection*. The *Head Orientation Projection* was the fastest, given that it is the only method that did not use any CNN models.

6.4 PERFORMANCE EVALUATION

Eye contact detection results of all methods in both settings are shown in Figure 6.4. Given that positive and negative samples are highly unbalanced, we use the MCC (Matthews Correlation Coefficient) metric to evaluate eye contact detection performance as in Smith *et al.* (2013). An MCC of 1.0 represents perfect classification, an MCC of -1.0 represents completely incorrect classification, and an MCC of 0.0 represents random guessing. The bars represent the MCC and the error bars indicate standard deviation across participants. From left to right, we show the proposed method, *Face Clustering*, *Gaze Classification*, *Gaze Projection*, *Head Orientation Projection* and *GazeLocking*. For the *object-mounted* setting, we report the average performance across all 14 participants. The proposed method achieves the best performance with a significant margin (35% in the *object-mounted* setting and 43% in the *head-mounted* setting) from the second best *Face Clustering* method (t-test, $p < 0.01$).

The *Face Clustering* is a strong baseline, but it can also cluster very limited samples that have similar face appearance. However, due to different head poses, the face appearance could be different even if the person looks at the same object.

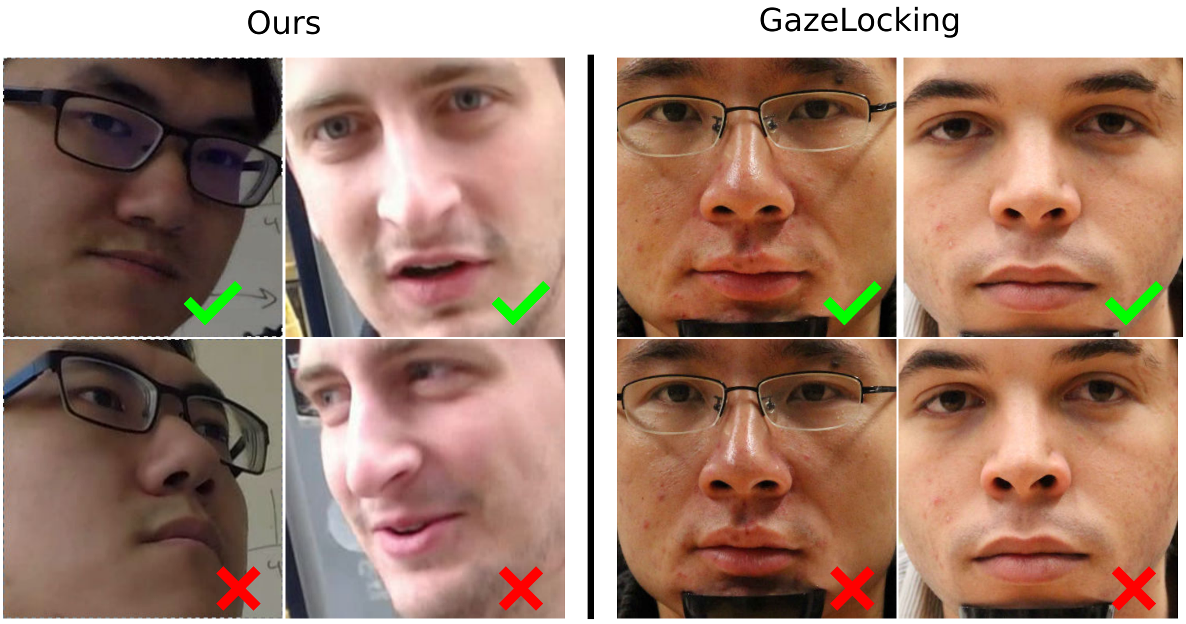


Figure 6.5: Sample images from our (left) and the Columbia Gaze dataset (right) illustrating the considerable differences in the naturalness of illumination, head pose, and gaze range. The first row shows positive and the second row shows negative samples from each dataset.

In contrast to our method and *Face Clustering*, *Gaze Classification* uses the gaze location $\boldsymbol{p}$ instead of the face feature $\boldsymbol{f}$ to train the eye contact detector, which achieved worse results than ours or *Face Clustering*. This indicates that the face feature $\boldsymbol{f}$ is better than the gaze locations $\boldsymbol{p}$ for the eye contact training, which has better representation of the faces to capture the appearance variations.

Gaze Projection is directly based on the low accuracy gaze estimation results, and *Head Orientation Projection* is estimated from the detected facial landmarks, which are not reliable for non-frontal faces. These projection-based methods also require prior knowledge about the physical scene structure, and also suffer from errors in camera calibration and object location measurement.

The *GazeLocking* method determines whether a person is looking at the camera, which is not sufficient for eye contact detection on arbitrary objects. Figure 6.5 shows sample images from our and the Columbia Gaze dataset (Smith *et al.*, 2013), further illustrating the considerable differences in the naturalness of illumination, head pose, and gaze range. Training the *GazeLocking* method with the labelled data in our dataset instead of the Columbia Gaze dataset could result in a better performance. However, the difficulty of collecting such fully annotated on-site training data is the key issue we addressed in the proposed method, and hence we opted for the evaluation using their own dataset.

The performance of the different methods for the *head-mounted* setting is lower than for the *object-mounted* setting given the more challenging outdoor environment. Motion blur is pervasive for the *head-mounted* camera, which affects both the facial

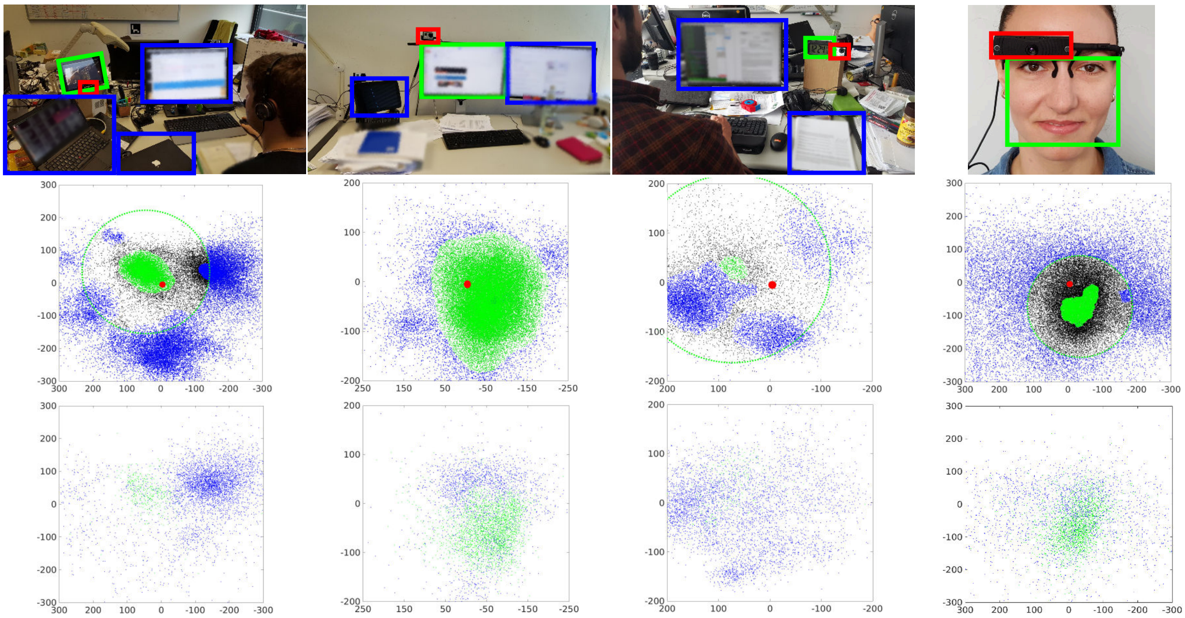


Figure 6.6: Examples of gaze locations distribution for the *object-mounted* (tablet, display, and clock) and *head-mounted* settings. The first row shows the recording setting with marked target objects (green), camera (red), and other distractive objects (blue). The displays were pixelated for privacy reasons. The second row shows the gaze locations clustering results with the target cluster in green and negative cluster in blue. The red dot is the camera position, and the green dotted line indicates the safe margin *d*. The third row shows the ground-truth gaze locations from a subset of 5,000 manually annotated images with positive (green) and negative (blue) samples.

landmark detection and the appearance-based gaze estimation. The gaze estimation is also applied to multiple unknown users, which is similar to the most difficult cross-dataset evaluation as discussed in Zhang *et al.* (2015a).

Examples of gaze location distribution for different object configurations and their corresponding clustering results are shown in Figure 6.6. In the first row of Figure 6.6 are the recording settings for the different objects, and we mark the target object (green rectangle), camera (red rectangle) positions and other distractive objects (blue rectangle). The second row of Figure 6.6 shows the sample clustering results where the target cluster is marked as green dots and all other negative samples are marked with blue dots. The noise samples are marked as black and the big red dot is the camera position (coordinate (0,0)). The dotted green line indicates the range of safe margin *d* where the samples outside the margin were also been selected as negative.

From the second row of Figure 6.6, we can see that our sample clustering methods can achieve good clustering results. The safe margin *d* also works quite well to find additional negative samples, especially for the *object-mounted* setting where only one cluster is created.

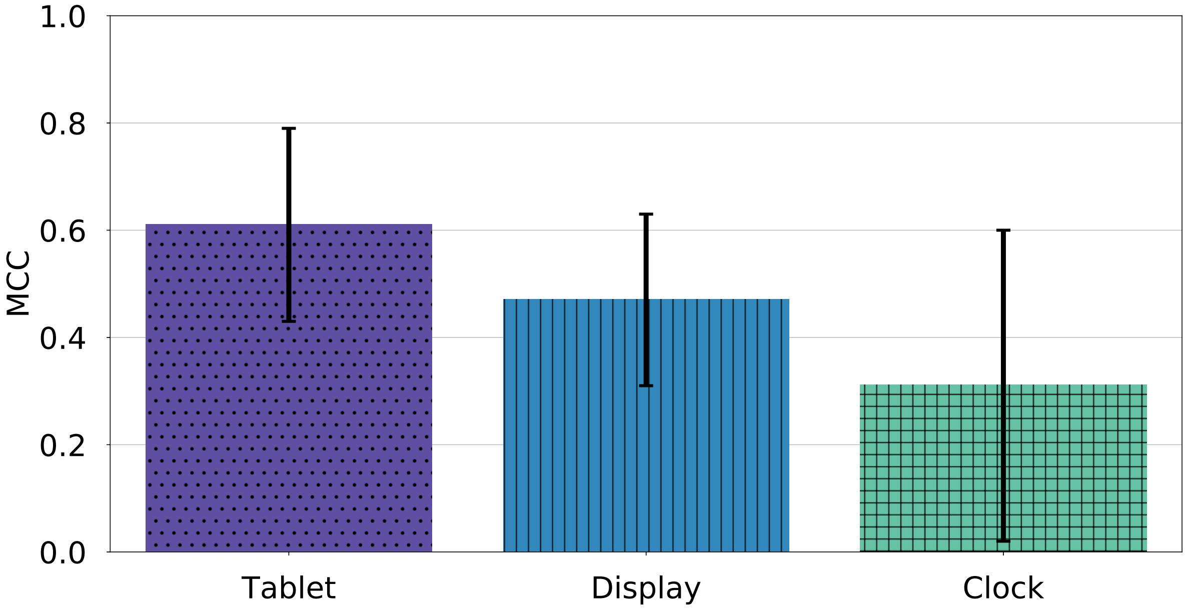


Figure 6.7: Performance of the proposed method for eye contact detection with different objects: tablet, display, and clock across participants. The bars are the MCC value and error bars indicate standard deviations across participants.

6.4.1 Object Categories and Camera Positions

The *object-mounted* setting uses three different objects (tablet, display and clock) with different sizes and attractiveness. Figure 6.7 shows the performance of the proposed method for each of the three objects. Each bar corresponds to the mean MCC value and error bars indicate standard deviations across all participants, and the performance for *Display* is also averaged across different camera positions.

In Figure 6.8, we show the confusion matrix of the proposed method for the three objects in the *object-mounted* setting. We normalise each element by dividing the sum of the each row so that the top left cell is the sensitivity (true positive rate), and the bottom right cell is the specificity (true negative rate). There are also biases in the ground-truth label distribution among test data, and percentage of positive test samples were 18.6%, 58% and 5.4% for the tablet, display and clock objects respectively.

The clock becomes the worst case among the three objects, because it attracts less attention from the participants, as illustrated in the third column of Figure 6.6, and hence has the lowest amount of positive training data. Figure Figure 6.8c also shows that the clock has low sensitivity but high specificity, which indicates that the model mostly predicted the samples to be negative. While, on the other hand, the display and tablet are expected to attract a similar level of user attention, our method achieved the best performance for the tablet. Although the display is attracting enough user attention in terms of amount of training data, gaze distribution is not concentrated at the centre, as shown in the second column of Figure 6.6. This is

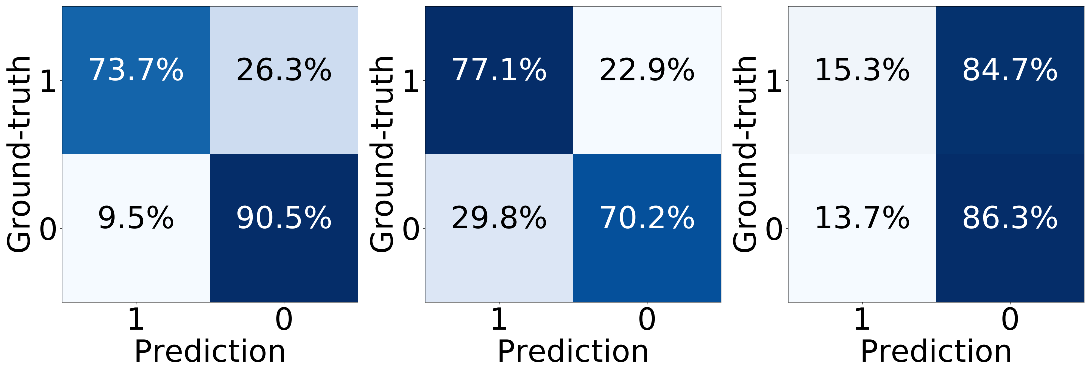


Figure 6.8: The confusion matrix of the proposed method for eye contact detection with different objects: tablet, display, and clock. The label 1 means positive eye contact and 0 means negative eye contact. We normalise each element by dividing the sum of the row.

expected to be because of its larger physical size and the fact that displayed contents can create different *target* areas even inside the display. Hence the cluster structure tends to be more complex, and the positive sample selection becomes more difficult.

In addition, there are three positions we set for the recording, which results in 10 videos for the above display, 9 videos for the below display, and 7 videos for the next to the display. We compare the MCC for these three different positions in Figure 6.9. The results show that our method works equally well for different camera positions. The above-display position has the best performance since usually there is no other salient object to affect the sample clustering, and it gives a good view of the participant's face. The below-display position also has a good view of the participant's face, but there could be some other object close to the camera that attracted the participant's attention. In our evaluation, for example, we find that there are two cases where the sample clustering picks the cluster belonging to the keyboard as the target cluster, so that the MCC becomes to near 0 values. When the camera is placed next to the display, the camera's view is not as good, thereby effectively reducing gaze estimation accuracy and resulting in noisy sample clustering.

6.4.2 Duration of Training Data Collection

Since our method requires a certain amount of data for sample clustering, here we test the performance across different times for the training data collection. We evaluated the three objects under the *object-mounted* setting, and picked the samples collected from the period of time according to the time sequence. We kept the test set the same as for the previous evaluation. In Figure 6.10, we plot the performance

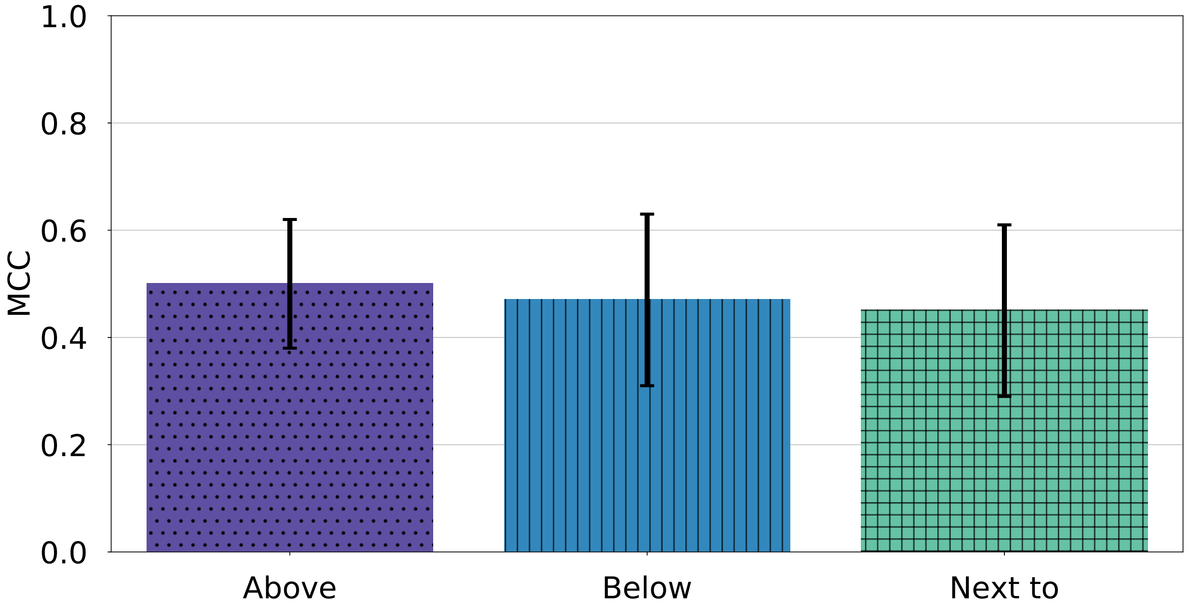


Figure 6.9: Performance of the proposed method for eye contact detection with the display for different camera positions across participants. We evaluated three positions: above, below, and next to the display. The bars are the MCC value and error bars indicate standard deviations across participants.

across the amount of time for training data collection. It can be seen that the eye detection performance in general increases with longer data collection, while the performance converges after around 3.0 hours. However, it can be also seen that the performance of the clock case has not yet fully converged, and this indicates that longer training duration for small objects can partly address the above-mentioned issue of the smaller number of positive samples. In our *object-mounted* recording, the average number of samples per hour is around 13,000.

6.4.3 Cross-Person Evaluation

We finally evaluate the tablet sessions of our *object-mounted* dataset across all users. To this end, we used the training data from all participants, tested on each respective test set, and averaged the individual performance numbers. This simulates an application scenario in which multiple users share a space, such as an office, and there is a single target object for which eye contact detection from all users should be analysed. As shown in Figure 6.11, our method achieved the best performance for this setting with MCC 0.43, outperforming the second best *Face Clustering* method by 34%. Note that the proposed method achieved an MCC of 0.61 in the person-specific evaluation (Figure 6.7), i.e. there is still lots of room for further performance improvement. Compared to the *object-mounted* setting, which is also cross-user and in which our method achieved an MCC of 0.30, the *object-mounted* setting is easier due to the higher quality images.

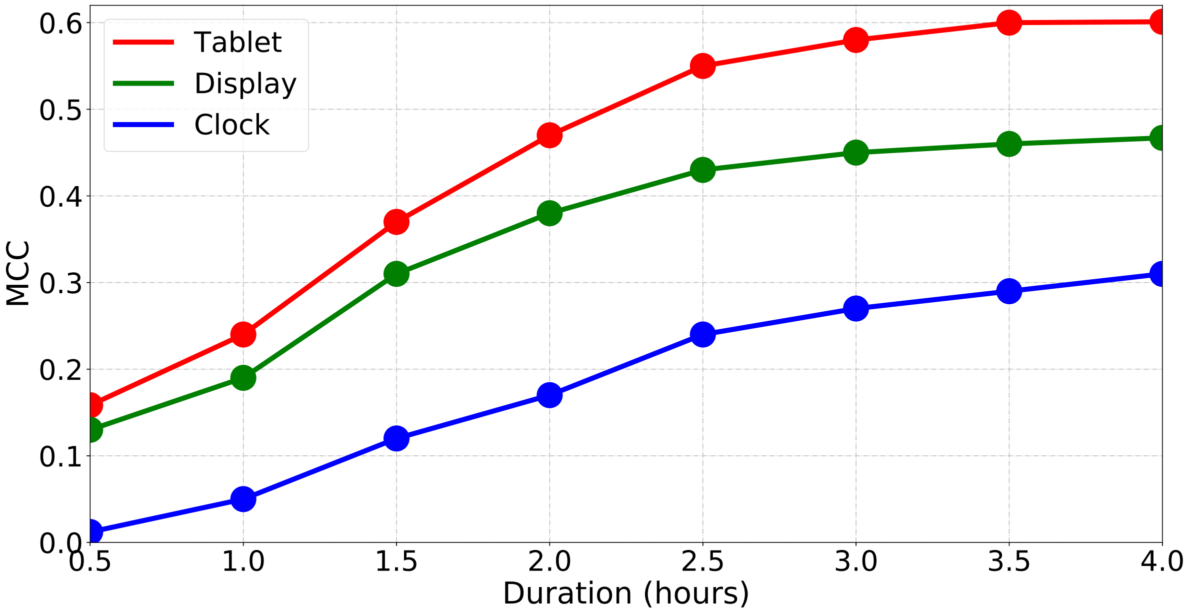


Figure 6.10: Performance of our method depending on the duration (number of hours) of training data collection. We show the performance for the three objects in the *object-mounted* setting.

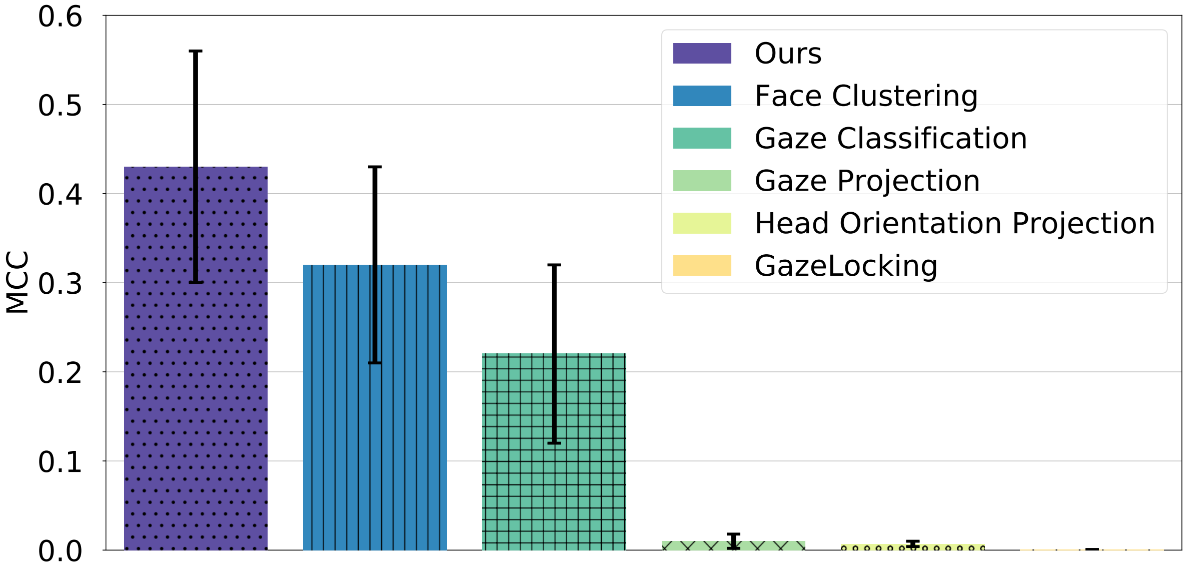


Figure 6.11: Performance of the different methods for eye contact detection with the tablet using cross-person evaluation. The bars are the MCC value and error bars indicate standard deviations across participants.

6.5 DISCUSSION

Our method provides a light weight yet robust and generic approach for learning-based eye contact detection. The experimental results show that while pre-trained eye contact detectors do not perform well in real-world environments, our method constantly achieves good performance even for challenging cases, such as the small clock on a cluttered desk or the face moving around outdoor environments.

6.5.1 Application Scenarios

The main advantage of our approach over state-of-the-art methods is that it has very few requirements with respect to the camera and target objects. Potential users simply have to attach an arbitrary camera to the target object and the system automatically collects evidence for eye contact detection, and starts running as an eye contact detector after the initial training phase. This approach thus allows for continuous training data collection during deployment, which also allows the method to handle dynamic environments. As such, our method opens up a variety of exciting new applications.

The first promising application area is attentive smart home or office environments in which eye contact detection can be a signal of user intention to start an interaction, e.g. with household appliances. The group of users is also typically limited in such settings, and our method thus has a good chance to train a robust eye contact detector even for multiple users. Another application area is eye contact detection on mobile devices, such as smartphones and smartwatches. As shown in our experiments, our method also allows such mobile cases, and since these devices typically assume a single user, we can expect better classification accuracy than for the most challenging head-mounted setup. Our method therefore has significant potential to enable new types of mobile glance-based interactions. Sensing driver attention in cars is another application area in which there is a single user under dynamic changes in lighting conditions. In such a scenario, our method could learn and detect the driver's eye contact from, for instance, a camera-equipped car navigation system.

Although mobile and multi-user scenarios are the most challenging setting, eye contact detection from wearable cameras has a great potential for, e.g., extracting important moments from lifelogs. Our method is also not limited to the head-mounted case, and provides flexibility for designing new wearable eye contact sensors. Similarly, eye contact from robots has many potential application scenarios, and our method has the advantage that it can be embedded into almost any kind of configuration including humanoid robots, vehicles, or drones. Finally, eye contact detection can also serve as an important input cue for public displays, and our method also allows such multi-user cases. It could allow public displays to dynamically change their content according to the amount of eye contact from audiences, and eye contact statistics provide valuable information to analyse the display usage. Unlike the approach proposed in Sugano *et al.* (2016), our method can also be applied to static displays, billboards, posters etc. for analytical purposes.

6.5.2 Technical Limitations

The key requirement of our method is that the eye contact target is the salient object nearest to the camera. This holds true in most of the above-mentioned application scenarios, however, there are cases that our method cannot handle properly. For example, if the camera is placed exactly between two equally salient objects, it is difficult to robustly identify both target objects. This also happens in our experiments when the camera is installed between the display and keyboard, and sometimes the keyboard is chosen as the target object. Essentially, it is an ill-posed problem to choose the target object cluster from multiple candidates without any information. Hence, this requires a hardware design consideration, or additional human supervision.

The size of the target object also affects the performance of our method. If the target object is not salient enough, like the small clock in our experiments, estimated gaze locations do not show a clear cluster structure at the target location and the performance degrades. On the other hand, if the target object is too large, such as public displays or the main work display in our experiment, multiple attention clusters can occur even within the same target object. These issues may be addressed by introducing a long-term training phase or by developing new methods that are able to distinguish or merge multiple clusters for large objects.

The performance of our method is directly linked to the accuracy of the underlying appearance-based gaze estimation method. It will therefore be important to improve the baseline performance of these methods. However, even with perfect accuracy, our approach still has advantages because 1) it can exploit the scene structure to find the decision boundary between the target object and other objects, and 2) it can also focus on target users and environments, which is expected to be consistently better than a generic gaze estimator assuming arbitrary users and environments. While currently we extract the face features from the same gaze estimation CNN, there is also room for improvement by investigating feature extraction networks optimised for the eye contact detection task. Future work could also investigate methods to exploit temporal aspects of human gaze. Users naturally fixate on the target object for a certain amount of time, and such temporal information could help the clustering process.

6.6 CONCLUSION

In this chapter we studied the challenging task of detecting eye contact with objects and people in real-world office and social interaction settings. We proposed a method for eye contact detection that combines a state-of-the-art appearance-based gaze estimator with a novel approach for unsupervised gaze target discovery. Evaluations on a novel dataset demonstrated that our method is robust across different users, gaze target types and sizes, camera positions, and illumination conditions. The method can perform real-time eye contact detection with a target object for single or multiple users, and achieved an MCC of 0.46 and 0.30 for both settings – a significant improvement of 35% and 43% over the second-best baseline method and with the state-of-the-art method only at chance level. Our findings are significant and pave the way for a new class of attentive systems that sense and respond to eye contact.

7 TRAINING PERSON-SPECIFIC GAZE ESTIMATORS FROM USER INTERACTIONS WITH MULTIPLE DEVICES

Learning-based gaze estimation has significant potential to enable attentive user interfaces and gaze-based interaction on the billions of camera-equipped handheld devices and ambient displays. While training accurate person- and device-independent gaze estimators remains challenging, person-specific training is feasible but requires tedious data collection for each target device. To address these limitations, we present the first method to train person-specific gaze estimators across multiple devices. At the core of our method is a single convolutional neural network with shared feature extraction layers and device-specific branches that we train from face images and corresponding on-screen gaze locations. Detailed evaluations on a new dataset of interactions with five common devices (mobile phone, tablet, laptop, desktop computer, smart TV) and three common applications (mobile game, text editing, media center) demonstrate the significant potential of cross-device training. We further explore training with gaze locations derived from natural interactions, such as mouse or touch input.

7.1 INTRODUCTION

Cameras are being integrated in an ever-increasing number of personal devices, such as mobile phones and laptops. At the same time, methods for learning-based gaze estimation, i.e. methods that directly map eye images to on-screen gaze locations/3D gaze directions, have considerably matured (Sugano *et al.*, 2014; Funes Mora *et al.*, 2014; Zhang *et al.*, 2015a; Krafka *et al.*, 2016; Zhang *et al.*, 2017d). Taken together, these advances promise to finally enable attentive user interface (Bulling, 2016), eye-based user modelling (Seifert *et al.*, 2017; Huang *et al.*, 2016b), and gaze interaction (Kristensson and Vertanen, 2012; Sugano *et al.*, 2016; Zhang *et al.*, 2017b) on devices that we all use in everyday life. Despite this potential, current learning-based methods still require dedicated person- and device-specific training data to achieve a practically useful accuracy of 2°∼4°. This requires a so-called explicit calibration in which users have to iteratively fixate on predefined locations on the device screen. This calibration data is then used to train a person-specific gaze estimator. However, this approach is both tedious and time-consuming given that the calibration has to be performed on each device separately. This has hindered the adoption of gaze input in a wider range of HCI applications.

In this chapter we are the first to propose a solution to this problem, namely to learn a gaze estimator for a particular user across multiple devices – so-called

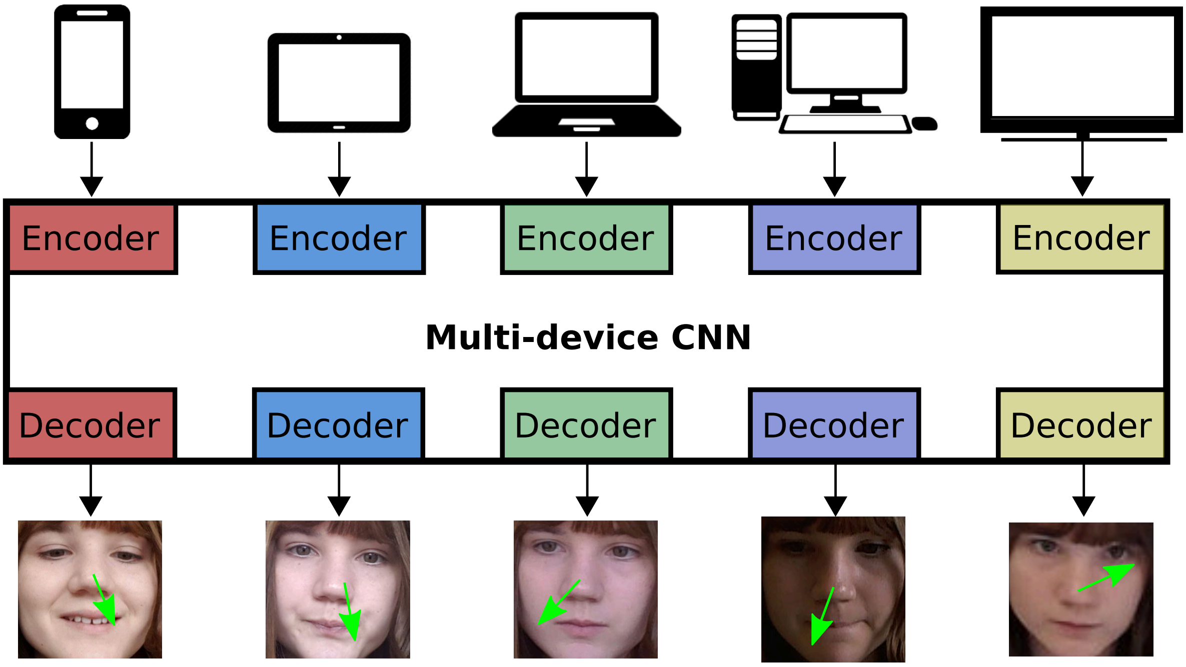


Figure 7.1: Our method for multi-device person-specific gaze estimation based on a convolutional neural network (CNN). It processes the person-specific images obtained from different devices with device-specific encoders and shared feature extraction layers, and gives out gaze estimates by different device-specific decoders.

Multi-Device Person-Specific Gaze Estimation. As illustrated in Figure 7.1, the key idea is to train a single gaze estimator, in our case based on a convolutional neural network (CNN), with *shared feature extraction layers* and *device-specific encoder/decoder branches*. While the shared feature extraction layers encode device-independent image information indicative for different gaze directions, the encoders and decoders adapt these shared features to device-specific camera and screen properties, such as image quality and screen resolution. Key advantages of this approach are that it is scalable, i.e. it can use data from an arbitrary number and type of devices a user might own, and that it leverages whatever amount of data may be available from these devices. To the best of our knowledge, this is the first work to explore person-specific gaze estimation using multi-device learning. In addition, we demonstrate how our approach can be combined with implicit calibration into a highly practical solution for person-specific gaze estimation. In contrast to explicit calibration, implicit calibration exploits the correlation between gaze and interaction events naturally occurring on the device, such as touches (Weill-Tessier and Gellersen, 2017) or mouse clicks (Sugano *et al.*, 2008). While implicit calibration can yield large amounts of data without imposing any additional user effort, ground-truth gaze location labels are less reliable than the data from conventional explicit calibration. In addition, implicit calibration fundamentally suffers from the low input frequency, and thus low amount of data, on some devices, such as TVs (Huang *et al.*, 2016a). Multi-device person-specific gaze estimation can alleviate this issue by leveraging data from other personal devices, and by sharing the learned person-specific feature across all devices.

The contributions of this chapter are three-fold. First, we propose the first method to train person-specific gaze estimators across multiple devices. Second, we conduct detailed evaluations demonstrating the effectiveness and significant potential of multi-device person-specific gaze estimation. To facilitate these evaluations, we further collected a new 22-participant dataset of images and user interactions with five device types (mobile phone, tablet, laptop, desktop computer, smart TV). We will release this dataset to the community free of charge upon acceptance of this chapter. Third, we propose a practical approach that combines multi-device person-specific gaze estimation with implicit calibration and evaluate it on data collected while users interacted with three common applications (mobile game, text editing, media center).

7.2 MULTI-DEVICE PERSON-SPECIFIC GAZE ESTIMATION

The core idea explored in this chapter is to train a single person-specific gaze estimator across multiple devices. We assume a set of training data, i.e., face images and ground-truth on-screen gaze locations, to be available from multiple personal devices. Facial appearance of a particular user can vary across devices due to different camera properties, and the physical relationship between the camera and screen coordinate system also depends on the hardware configuration. Furthermore, typical head pose with respect to the camera also greatly depends on the hardware design and its use case. This causes highly device-specific head pose/gaze distributions and input image qualities, and results in a large performance gap between generic and device-specific estimators. However, the fundamental features for person-specific gaze estimation should be independent of the devices, and a gaze estimation function should thus be able to use a shared facial appearance feature for multi-device gaze estimation.

7.2.1 Multi-Device CNN

Based on this idea, we propose a multi-device CNN as shown in Figure 7.2. Inspired by previous work (Kaiser *et al.*, 2017), the proposed CNN architecture handles the data variation across different devices by device-specific encoder/decoder and exploits the shared knowledge of the personal appearance by the shared layers. Each encoder and decoder accommodates the attributes of one specific device, while the shared feature extraction layers learn the shared gaze representation across devices. Inputs to the model are full-face images as suggested by Zhang *et al.* (2017d).

We design our multi-device CNN based on the original AlexNet architecture (Krizhevsky *et al.*, 2012), which has five convolutional layers and three fully connected layers. We use the same number of layers and number of nodes in each layer as AlexNet. The first two layers are used as encoders to distil the common visual fea-

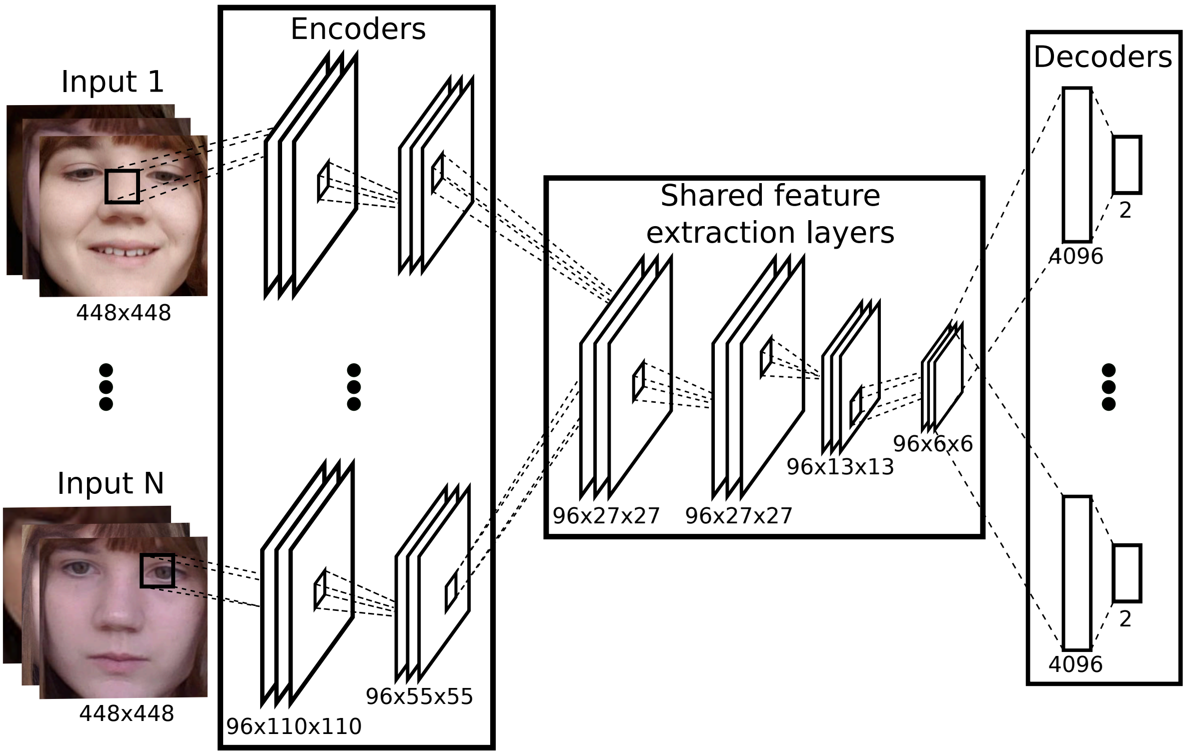


Figure 7.2: Architecture of the proposed multi-device gaze estimation method. Our method uses feature extraction layers shared across devices as well as device-specific encoders and decoders. It consists of two convolutional layers for each device-specific encoder, four convolutional layers for shared feature extraction layers, and two fully connected layers for each device-specific decoder. The numbers indicate the image resolution and the extracted feature dimension.

tures from different cameras. More specifically, given *N* devices, our model contains *N* device-specific encoders, each of which consists of two convolutional layers. These layers learn the local features from the input images of the corresponding device and encode the image differences caused by camera parameters, head poses and face-to-camera distances. It is important to note that although our CNN architecture is shallow compared to common networks for multi-device learning tasks (Kaiser *et al.*, 2017), our method is not restricted to the AlexNet model and can be easily extended to deeper architectures.

The key property of our model is to learn a gaze representation that is generic across devices but specific to a particular user. This is achieved by shared feature extraction layers after the device-specific encoders. We replaced the sixth fully connected layer of the original AlexNet with a convolutional layer, resulting in a total of four convolutional layers for shared feature extraction. After the shared feature extraction layers, the decoders consisting of two fully-connected layers are used for mapping the shared feature representation to device-specific on-screen gaze spaces. We systematically evaluated different numbers of layers for the encoders and decoders, and found these numbers to result in the best performance. In the

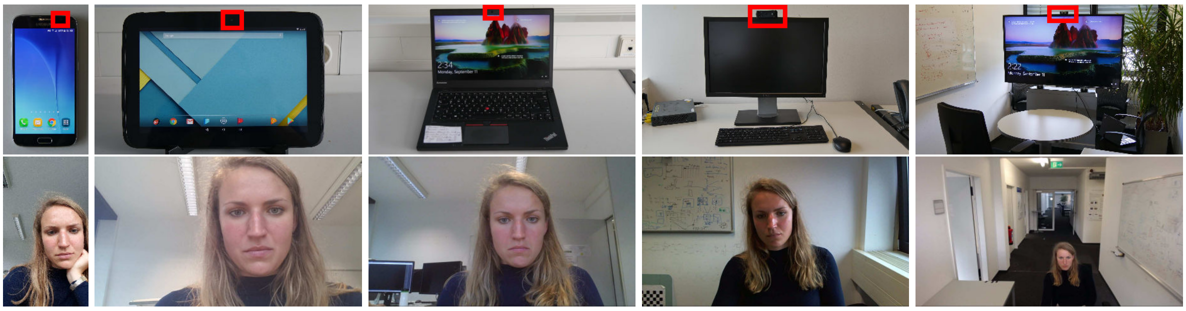


Figure 7.3: Top row: Personal devices used in the data collection. These devices have different sizes and camera-screen relationships (cameras are marked by red squares). Bottom row: Sample images from each device. As can be seen, the image resolution, noise level, illumination condition, face-to-screen distance, and head pose vary significantly across devices and environments, thus posing significant challenges for multi-device training.

training phase, the shared feature extraction layers are updated according to all of the user-specific training data from different devices, while device-specific encoders and decoders are updated only with their corresponding device-specific training data. In the test phase, the shared feature extraction layers process the local features produced by a specific encoder and pass the shared gaze representation to the corresponding decoder.

7.2.2 3D Gaze Estimation

The target of gaze estimation can either be an on-screen 2D gaze location (Krafka *et al.*, 2016; Huang *et al.*, 2017) or a 3D gaze direction in camera coordinates (Sugano *et al.*, 2014; Zhang *et al.*, 2015a, 2017d; Wood *et al.*, 2016b). In this chapter we use a 3D gaze estimation task formulation for multi-device personal training. Although the direct 2D regression from face images to on-screen coordinates is straightforward, it requires a dedicated mapping function for each device to compensate for hardware configurations, such as the camera-screen relationship. In contrast, the 3D formulation explicitly incorporates geometric knowledge, such as camera intrinsics and hardware configurations. However, the 3D formulation only addresses a subset of the technical challenges involved in multi-device training, specifically not device-specific gaze and head pose distribution biases. There is still a large performance gap between multi-device and device-specific training, with device-specific training typically improving gaze estimation performance significantly (Sugano *et al.*, 2016). As such, our multi-device person-specific training approach complements the 3D formulation: While the shared visual features can be learned more efficiently thanks to the 3D task formulation, the performance gap between generic and device-specific training is considerably reduced by the proposed encoder/decoder architecture.

The 3D gaze direction is usually defined as a unit vector originating from a 3D reference point (e.g the centre of the eyes) and pointing along the optical axis. In practice, to estimate 3D gaze and reduce the device biases, we first apply the face detector (King, 2009) and facial landmark detector (Baltrušaitis *et al.*, 2014) to process the input image, and then normalise the image data as suggested by Sugano *et al.* (2014). Specifically, we transform the face image through a perspective warping to compensate for the scaling and rotation of the camera. This process results in a normalised image space with fixed camera parameters and reference point location. After this normalisation, we can get the cropped face image and gaze direction in the camera coordinate system, which can also be projected back to the specific screen coordinate system. Following (Zhang *et al.*, 2017d), we set the size of the input face image to 448 × 448 pixels.

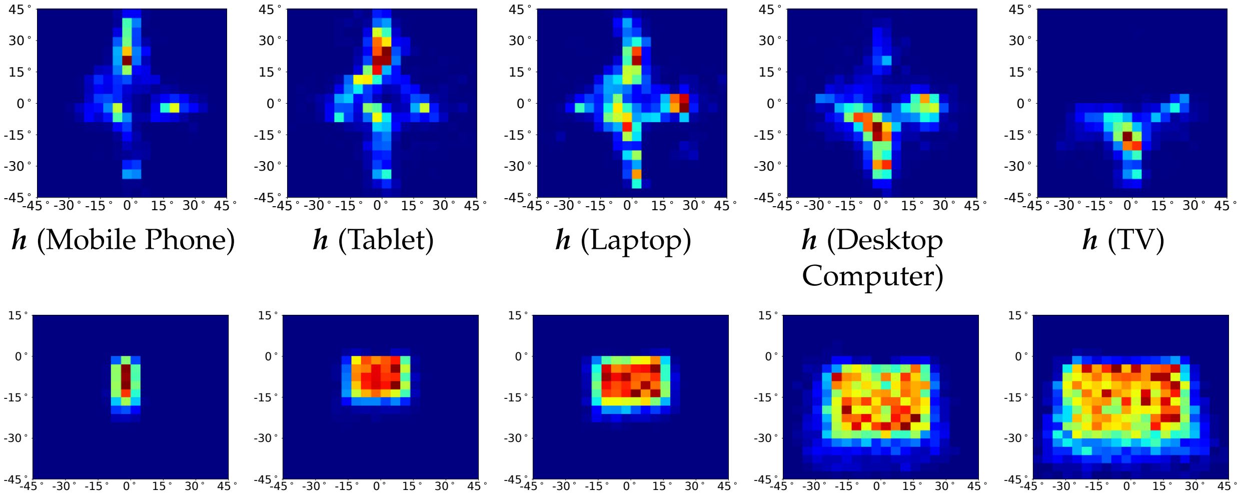
7.3 DATA COLLECTION

The data collection was designed with two main objectives in mind: 1) To obtain face images and corresponding ground truth on-screen gaze annotations in a principled way, i.e. one that could be used for quantitative evaluation of our method, and 2) to obtain face images during natural interactions. We therefore opted to collect data 1) using an explicit calibration routine that involved users visually fixating on predefined locations on the screen and confirming each location with a mouse click or touch to obtain highly accurate ground truth annotations, and 2) by logging face images as well as interaction data, such as mouse, keyboard and touch input, in the background that are known to correlate with gaze (Weill-Tessier and Gellersen, 2017) during different activities.

Activities were selected in such a way as to match common device usage and the dominant input modality available on a device in the real world. For example, while the activity of choice on the laptop was text editing using mouse and keyboard input, the predominant activity on mobile devices is digital games operated using touch input.

7.3.1 Participants and Procedure

We recruited 22 participants through university mailing lists and notice boards (10 female, aged between 19 and 44 years). Data of two male participants had to be excluded due to a too large number of face detection failures. Our participants were from eight different countries with 14 from Asia and the other six from Europe. Ten of them wore glasses during the recording. To evaluate our multi-device person-specific training method, each participant interacted with five devices, including a 5.1-inch mobile phone, a 10-inch tablet, a 14-inch laptop, a 24-inch desktop computer, and a 60-inch smart TV. These devices were chosen because of their popularity and pervasiveness; billions of interactions are performed with such devices every day worldwide. The top row of Figure 7.3 shows the five devices in our data collection with their camera locations highlighted in red. To capture participants' faces, we used the built-in cameras of the mobile phone, tablet, and laptop. We mounted a



h (Mobile Phone) *h* (Tablet) *h* (Laptop) *h* (Desktop Computer) *h* (TV)

g (Mobile Phone) *g* (Tablet) *g* (Laptop) *g* (Desktop Computer) *g* (TV)

Figure 7.4: Distributions of head angle (*h*) and gaze angle (*g*) in degrees for mobile phone, tablet, laptop, desktop computer, and smart TV, created with **explicit calibration data**. The overall range of head poses and gaze directions differed across devices. Participants mostly looked down while using mobile phone. In contrast, they often looked up while using the desktop computer and smart TV. In general, the range of gaze directions increases as the size of the device screen increases.

Logitech C910 on the monitor of the desktop computer, and a Logitech C930e on the smart TV. The camera resolutions for each device were: 1440 × 2560 pixels for the mobile phone, 2560 × 1600 pixels for the tablet, 1920 × 1080 pixels for the laptop, 1920 × 1200 pixels for the desktop computer and 1920 × 1080 pixels for the smart TV. The camera was always placed at the top of the screen. On each device we adopted two calibration methods for data collection.

Explicit calibration requires special user effort but provides the most reliable training data. For explicit calibration, participants were instructed to fixate on a shrinking circle on the screen and perform a touch/click when the circle had shrunk to a dot, at which point our recording software captured one image from the camera. We did not log the corresponding data point if participants failed to perform the touch/click action within half a second. The explicit calibration took around 10 minutes. For each participant, we collected 300 samples through explicit calibration at the beginning and end of the interaction with each device.

Implicit calibration was performed by monitoring users in the background while they interacted naturally with these devices. As implicit calibration does not rely on explicit user input, it is more practical in real use but also much more challenging. Thus, evaluation on the implicit calibration is also of interest and may provide in-depth insights for our method. In the sessions of implicit calibration, we recorded the face video from the frontal camera, the time stamps of each frame, and the locations of interaction events, such as clicks, touches, and key-presses. Each event

	Mobile Phone	Tablet	Laptop	Desktop Computer	Smart TV
Mean	810	358	802	636	165
STD	242	112	179	234	41

Table 7.1: Mean and standard deviation (STD) of the number of samples collected in the implicit calibration sessions over 20 participants and five devices. The number of samples differs across devices and activities.

position was considered as gaze ground truth and trained with the corresponding face image with the same time stamp. On each device, participants performed a specific activity, which lasted for 10 minutes and yielded on average 554 samples. The activities included gaming, text editing, and interacting with a media center.

Mobile Phone and Tablet. Since nowadays people spend a lot of time on mobile game playing (Seok and DaCosta, 2015), we asked participants to play five games on the mobile phone and five games on tablet during data collection. These games required participants to touch specific on-screen targets to increase their game score and win.

Laptop and Desktop Computer. As text editing is prevalent in computer use (Pearce and Rice, 2013), we picked text editing as the recording activity for the laptop and desktop computer. Participants were asked to compose a document with texts and figures about a familiar topic. All the texts were be typed manually, and the figures could be found on and downloaded from the Internet. Participants were also encouraged to format the document, such as adding bulleted lists, changing fonts and font types, or structuring the document into sections and subsections, etc.

Smart TV. We simulated a typical video retrieval activity using media center software[1]. Participants were instructed to search for interesting videos using the on-screen keyboard, quickly skim the video by clicking the progress bar, and add a bookmark to any video they found interesting. We asked them to perform at least three searches and at least one bookmark for each search.

7.3.2 Dataset Characteristics

Figure 7.3 shows sample images of one participant looking at the center of the screen for five different devices: mobile phone, tablet, laptop, desktop computer, and smart TV (from left to right). As can been seen from these images, the resolution, noise level, illumination condition, face-to-screen distance, and head pose vary significantly across devices and environments. However, the inherent personal

[1] `https://kodi.tv/`

appearance information remains consistent to a large degree.

7.3.2.1 *Distribution of Head and Gaze Angles*

We measured the 3D head pose by fitting a generic 3D face model to the detected facial landmarks, and transformed the on-screen gaze location to the 3D direction vector in the camera coordinate system as in Zhang *et al.* (2015a). Figure 7.4 shows the distributions of head and gaze angle in degrees on the five devices in the explicit calibration setting. The figure shows clear differences between devices due to the different hardware setups and the way participants interacted with them. For explicit calibration, a large proportion of the data from the mobile phone and tablet appears with positive angles of head pitch (looking up/down), meaning that participants were looking down on the screen. In contrast, most of the data recorded on the desktop computer and smart TV shows negative angles of head pitch, while the data from the laptop is quite evenly distributed. This suggests that data from different devices will be likely to complement each other and that training a gaze estimator using the combined data of different devices, especially from those with distinct use patterns of head poses, should be advantageous.

Although the sizes of the tablet (10") and the laptop (14"), as well as of the desktop computer (24") and the smart TV (60") are rather different, the ranges of gaze angles are similar between tablet and laptop as well as between desktop and smart TV due to the distance from the users. However, the differences are still prominent among three device groups: mobile phone, tablet/laptop, and desktop/TV. These differences in head pose and gaze direction distributions illustrate the difficulty of training a generic gaze estimator across multiple devices, even with the 3D gaze estimation formulation.

7.3.2.2 *Frequency of Interaction*

Tablet 7.1 summarises the amount of data that we collected using implicit calibration from all 20 participants. The two most efficient implicit calibrations are game playing on the mobile phone (overall 1.4 samples/sec) and text editing on the laptop (overall 1.3 samples/sec). There are also differences in sampling rate for the same task performed on different devices. That is, game playing on the mobile phone yielded more data than on the tablet, as did text editing on the laptop compared to the desktop computer. The former may be because the tablet has a larger screen, resulting in longer travelling times between touches and a possibly higher chance of muscle fatigue. The latter could be due to the differences of typing on different keyboards under varied typing skills (Szeto and Lee, 2002). In addition, as expected, implicit calibration is not particularly efficient on the smart TV (overall 0.3 samples/sec). In summary, these differences in data acquisition efficiency support our idea of multi-device training. Our method can especially contribute to the gaze estimation on devices with limited and skewed person-specific data. It is important to note that the activity data that we collected cannot represent the universal quality and amount of data from the corresponding device.

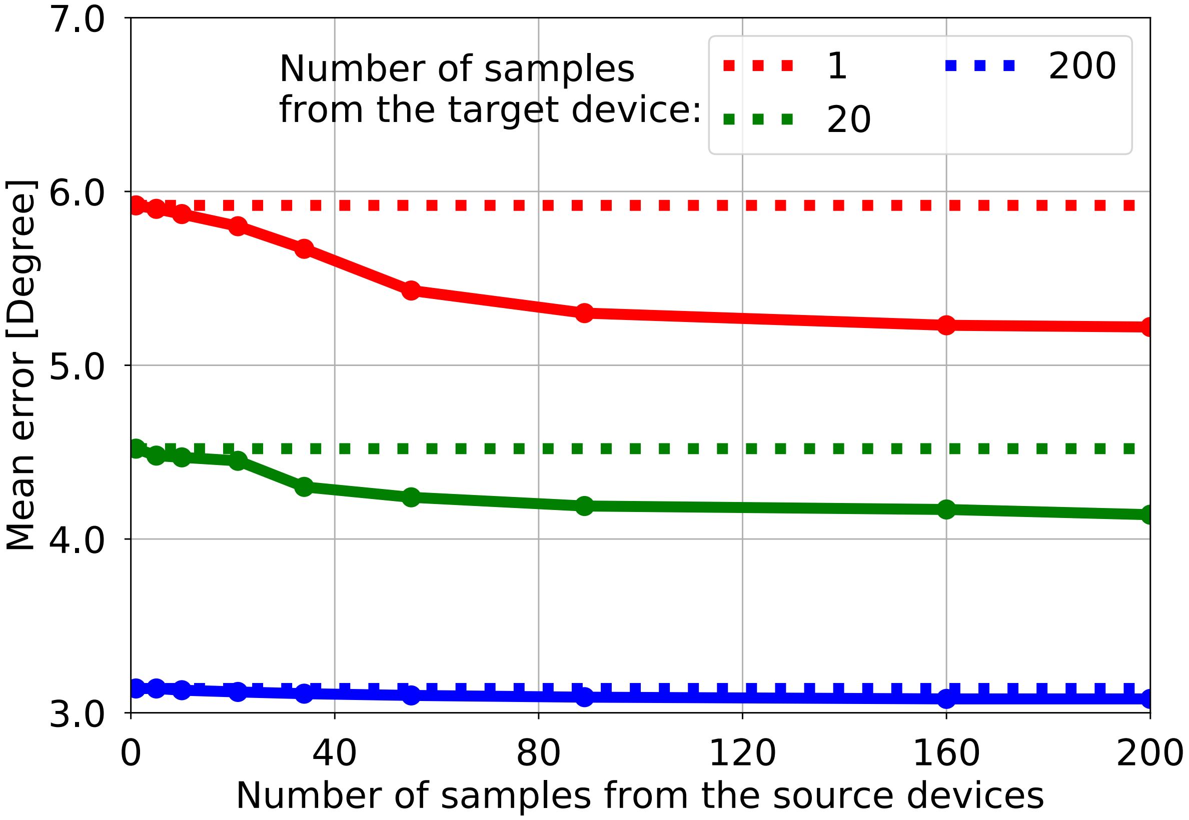


Figure 7.5: Gaze estimation error for the explicit calibration setting, comparing the proposed multi-device CNN (solid lines), a baseline single-device CNN trained on 1, 20 or 200 samples from the target device (dashed lines), as well as a single-device CNN trained on data from all source devices (dotted lines). The results were averaged over all five devices.

7.4 EXPERIMENTS

We conducted several experiments to evaluate our method for multi-device person-specific gaze estimation. We first compare our multi-device CNN with a single-device CNN, and discuss the results for each device in more detail. We then evaluate another scenario where an increasing number of samples from the target device was used for training. We conducted all of these experiments for both the explicit and implicit calibration settings. Finally, we analyse the contribution of the different devices for multi-device learning when using explicit calibration data.

We used the Caffe (Jia *et al.*, 2014) library to implement our model based on a modified AlexNet (Krizhevsky *et al.*, 2012) pre-trained on ImageNet (Deng *et al.*, 2009). We fine-tuned the multi-device and single-device CNNs on MPIIGaze (Zhang *et al.*, 2015a) and EYEDIAP (Funes Mora *et al.*, 2014). We used the Adam solver (Kingma and Ba, 2015) with a learning rate of 0.00001 and stopped training after 60 epochs. From the 300 samples collected for each participant during the explicit calibration,

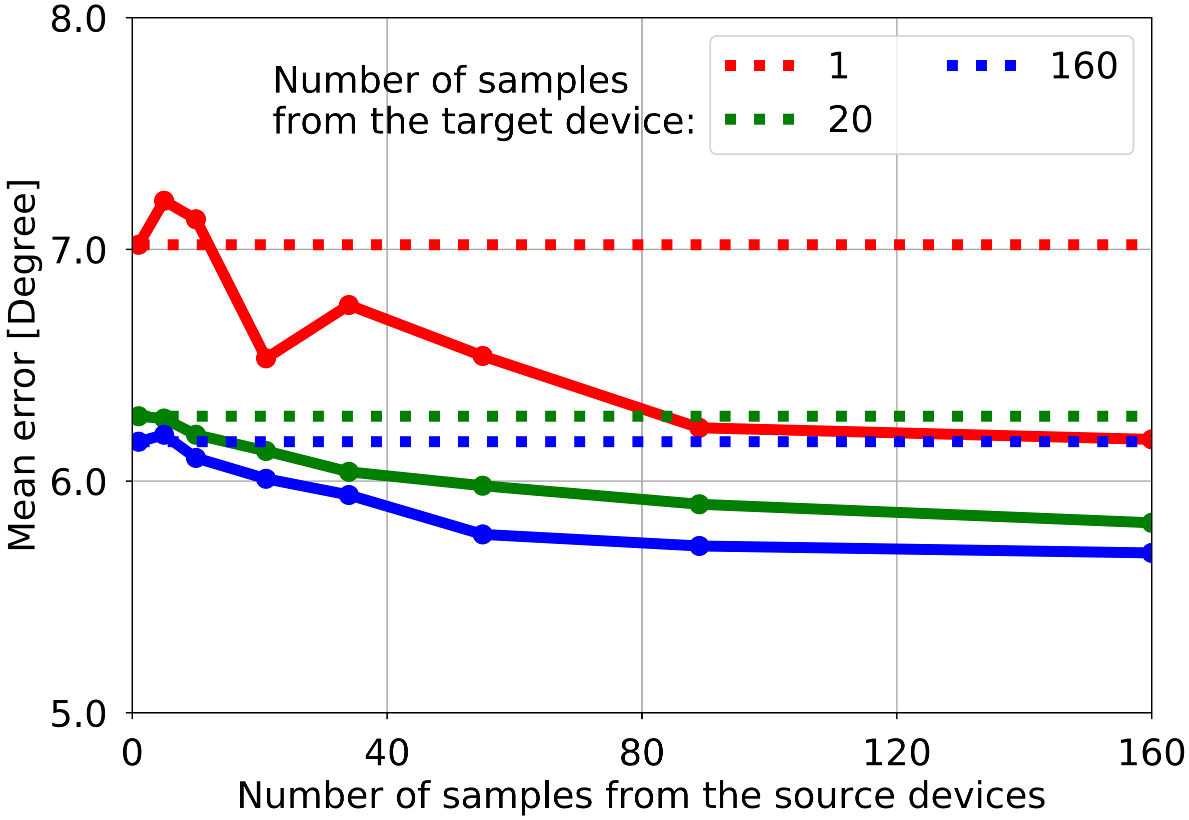


Figure 7.6: Gaze estimation error for the implicit calibration setting, comparing the proposed multi-device CNN (solid lines), a baseline single-device CNN trained on 1, 20 or 160 samples from the target device (dashed lines), as well as a single-device CNN trained on data from all source devices (dotted lines). The results were averaged over all five devices.

we selected the first 200 for training and the remaining 100 samples for testing.

7.4.1 Multi-Device vs. Single-Device Performance

To compare the multi-device CNNs and single-device CNNs, we performed a leave-one-device-out cross-validation, where each time we took one device as the *target device* for evaluation and the other four devices as *source devices*. Last, the results were averaged across all five target devices. The proposed multi-device CNN takes the data from both target and source devices as input, while the single-device CNN only uses samples from the target device, as in previous works. In addition, we additionally trained the same single-device CNN with data from all devices to evaluate the effectiveness of our proposed network architecture.

We evaluate the gaze estimation performance for different amounts of training data from the target device. Specifically, we are interested in the following cases: performance 1) with one sample from the target device, which is close to the case of 1-point calibration; 2) with 20 samples, which takes a feasible time (around half a minute) to collect in the explicit calibration; and 3) with the maximum number of samples (200 for the explicit calibration, and a variable number for the implicit calibration), which gives us the upper bound performance.

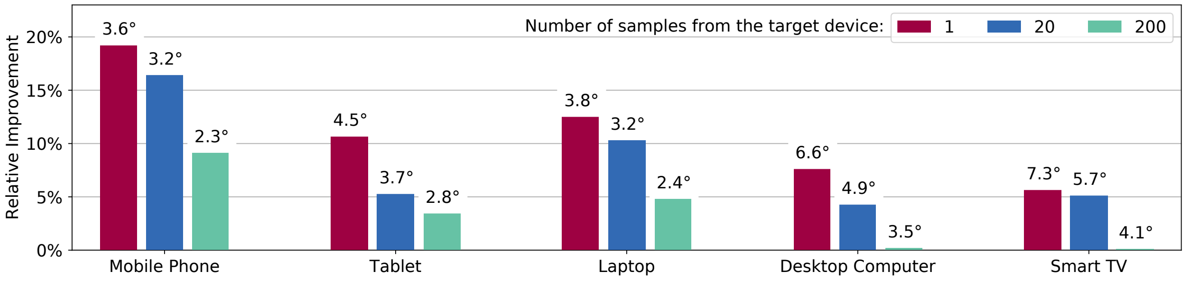


Figure 7.7: Relative improvement in gaze estimation error of our multi-device CNN over the single-device baseline in the explicit calibration setting when training on 1, 20 and 200 target samples. The numbers at the top of each bar are the mean error in degrees achieved by the multi-device CNN.

Performance for Explicit Calibration. We first investigate the explicit calibration setting that yields high-quality training data and thus represents the ideal situation. Figure 7.5 shows the performance of our multi-device CNN compared to the single-device baseline. The single-device baseline was trained on 1, 20 or 200 *target samples* (samples from the target device), while the multi-device CNN was trained with the corresponding amount of target samples together with the data from the source devices. The figure also shows the single-device architecture trained with the same multi-device training data as the proposed multi-device CNN. The results were averaged across multiple devices, including mobile phone, tablet, laptop, desktop, and smart TV. The red, green, and blue lines indicate the cases with one, 20, and 200 target samples. The dashed lines denote the mean error in degrees of the single-device CNN, the dotted lines show the results from the single-device architecture trained with data from all devices, and the solid lines are the results of the multi-device CNN trained on a growing amount of *source samples* (up to 200) from the source devices. As can be seen from the figure, the multi-device CNN outperforms the single-device CNN. In particular, there is a significant 11.8% improvement (paired t-test: $p < 0.01$) in the 1-sample case (red lines), corresponding to a mean error of 5.22° when trained with 200 source samples. The single-device architecture trained with data from all devices performs considerably worse. This is expected given that this represents the challenging cross-device gaze estimation task, one of the holy grails in learning-based gaze estimation (Zhang *et al.*, 2015a). Our multi-device CNN significantly improves over this performance using device-specific encoders and decoders to better leverage cross-device data.

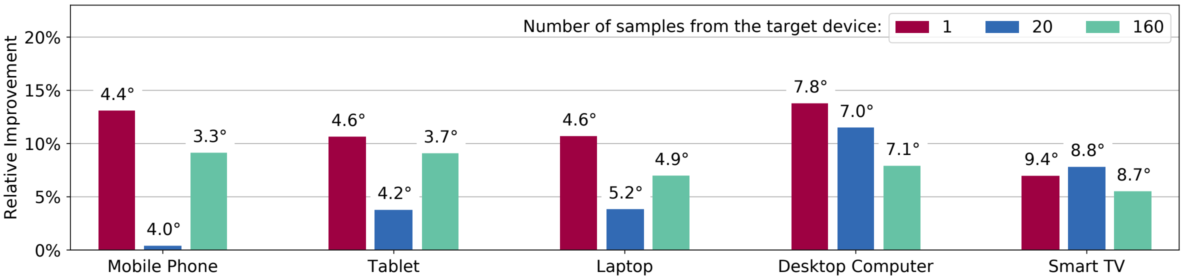


Figure 7.8: Relative improvement in gaze estimation error of our multi-device CNN over the single-device baseline in the implicit calibration setting when training on 1, 20 and 160 target samples. The numbers at the top of each bar are the mean error in degrees achieved by the multi-device CNN.

Performance for Implicit Calibration. Figure 7.6 shows the corresponding results for the implicit calibration setting when using one, 20, and up to 160 samples from target devices for training. The test sets were the same as for the explicit calibration setting. We picked 160 samples given that it is the average number of samples collected on the smart TV, and thus the minimum number among the five devices (see Table 7.1). Prior work has shown that the performance for implicit calibration can be affected by the temporal and spatial misalignment between interaction events, e.g. key presses or mouse clicks, and gaze locations, leading the performance to deteriorate. However, encouragingly, with only a few exceptions in the case of the 1-sample calibration (red lines), training with multi-device data generally produced a significant 12% improvement (paired t-test: $p < 0.01$) over the single-device CNN, corresponding to a mean error of 6.18°, when it was trained with 160 source samples. The single-device architecture trained with data from all devices again achieved the worst performance due to the difficulty of cross-device gaze estimation training.

Most importantly, for the practically most useful 1-sample case, our multi-device CNN reaches the best performance of the single-device CNN with 160 target samples (blue dashed line). This is exciting as it, for instance, means that we can use a 1-point calibration for a new personal device to achieve the same performance as when training on over a hundred device-specific implicit calibration samples. This can significantly enhance the usability of gaze-based applications. In addition, similar to the explicit calibration setting discussed before, training with multi-device data can further improve the device-specific performance. Unlike the explicit calibration setting, though, our multi-device CNN can achieve a much lower mean error (5.69°) in the 160-sample case (blue lines) than the single-device CNN (6.17°), when it has been trained with 160 source samples. This demonstrates that multi-device person-specific training is clearly preferable in terms of performance.

7.4.2 Performance on Different Target Devices

We then evaluate the performance of the multi-device and single-device CNN baseline on the different target devices.

Performance for Explicit Calibration. Figure 7.7 shows the relative improvement of our multi-device CNN over the single-device baseline in the explicit calibration setting averaged over 20 participants. The numbers at the top of each bar are the mean error in degrees achieved by the multi-device CNN. Following the previous discussion, the single-device CNN was trained on 1, 20 or 200 target samples, while the multi-device CNN was trained on 200 additional samples from each source device, i.e. 800 source samples in total. The numbers at the top of each bar are for each devices, and their average is shown at the far right of Figure 7.5. The angular gaze estimation error with 200 samples corresponds to the distance of 1.4 cm on the mobile phone screen, 2.2 cm on the tablet, 2.5 cm on the laptop, 3.5 cm on the desktop computer, and 8.6 cm on the smart TV.

In all cases, the multi-device CNN achieves a clear improvement over single-device CNN. Although the improvements are negligible for the desktop computer and smart TV in the 200-sample case, the improvements for the mobile phone, tablet, and laptop are clear. Most encouragingly, the improvements for the 1-sample case (red bars) on different devices are considerable, over 5% across all devices and reaching almost 20% for the mobile phone. The 20-sample case (blue bars) also gives promising results with an improvement of almost 5% across all devices. It is also interesting to see that the relative improvements increase as the size of the target device display decreases, most obviously for the mobile phone. This is most likely because more samples from other devices share similar gaze directions with the mobile phone, thus contributing to the multi-device training (see Figure 7.4, the second row).

Performance for Implicit Calibration. As before, we compare the multi-device CNN against the single-device CNN in the implicit calibration setting on the same test set as for the explicit calibration. We intended to compare performance with increasing training samples from the target device. As before, our multi-device CNN was trained on the target samples along with 160 source samples from other source devices, i.e. 640 source samples in total. Source samples were ordered randomly. The results are shown in Figure 7.8. The bars show the relative improvement of the multi-device over the single-device CNN. The numbers at the top of each bar are for each devices, and their average is shown at the far right of Figure 7.6. The angular error with 200 samples corresponds to the distance of 1.8 cm on the mobile phone screen, 2.6 cm on the tablet, 4.7 cm on the laptop, 7.4 cm on the desktop computer, and 18.1 cm on the smart TV.

Encouragingly, for all cases, our multi-device CNN can still outperform single-device CNN. For the 1-sample case (red bars), the achieved improvements over the single-device CNN are more than 10% for four devices (mobile phone, tablet,

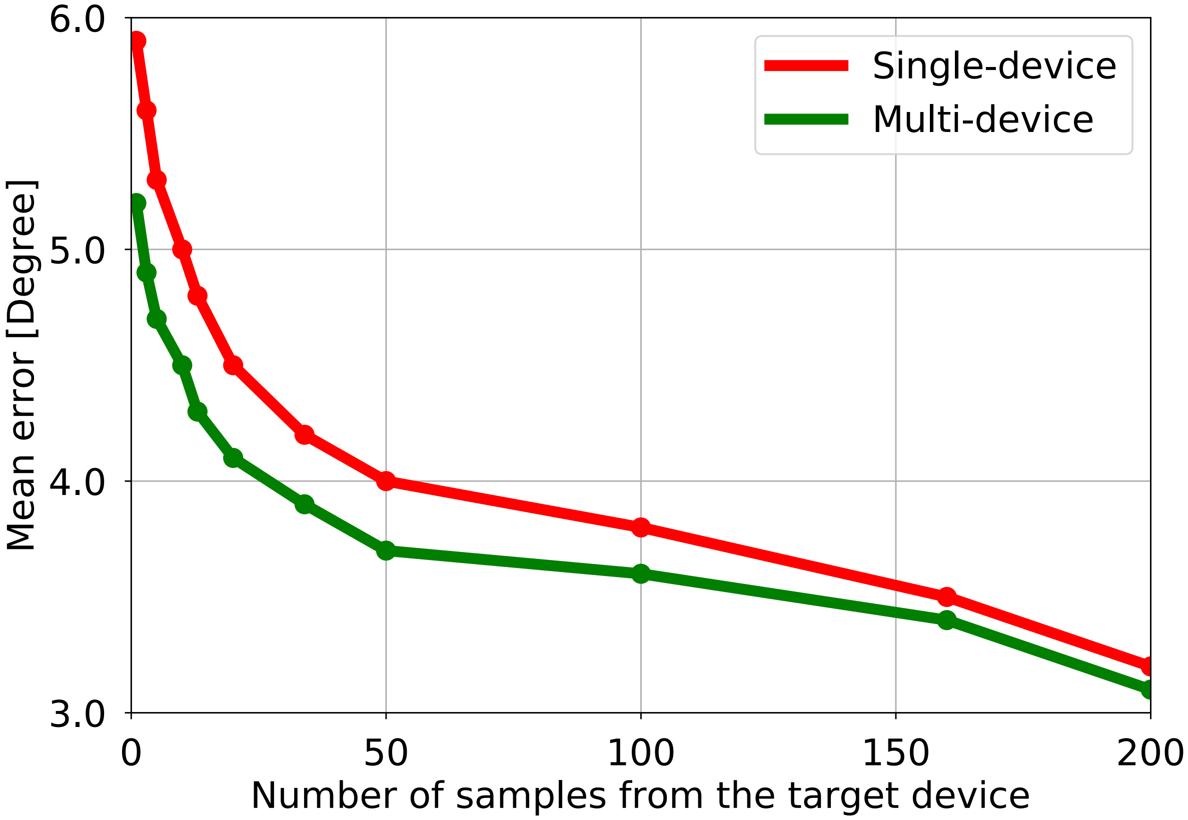


Figure 7.9: Mean error when adding a new device to the multi-device CNN compared to the single-device CNN in the explicit calibration setting. The single-device CNN was trained on increasing target samples from one to 200, while the multi-device CNN was trained additionally on 200 source samples from each source devices. The green line indicates the averaged performance of the multi-device CNN over five target devices; the red line shows that of the single-device CNN.

laptop, and desktop). For the 160-sample case (green bars), our models achieved an improvement of more than 5% for all devices. However, the improvements with 20 target samples (blue bars) are not consistent with the other cases, probably due to the noise in the implicit calibration data.

7.4.3 Adding a New Device to the Multi-Device CNN

To shed light on the performance of our method in practical use, we investigate the scenario of a user adding a new personal device to the multi-device CNN already trained using a certain amount of data from existing devices. To this end, we treated this new device as the target device and the other four devices as source devices. We repeated this procedure for each device and averaged the resulting error numbers.

In the case of explicit calibration, the single-device CNN was trained on an increasing number of target samples from one to 200, while the multi-device CNN

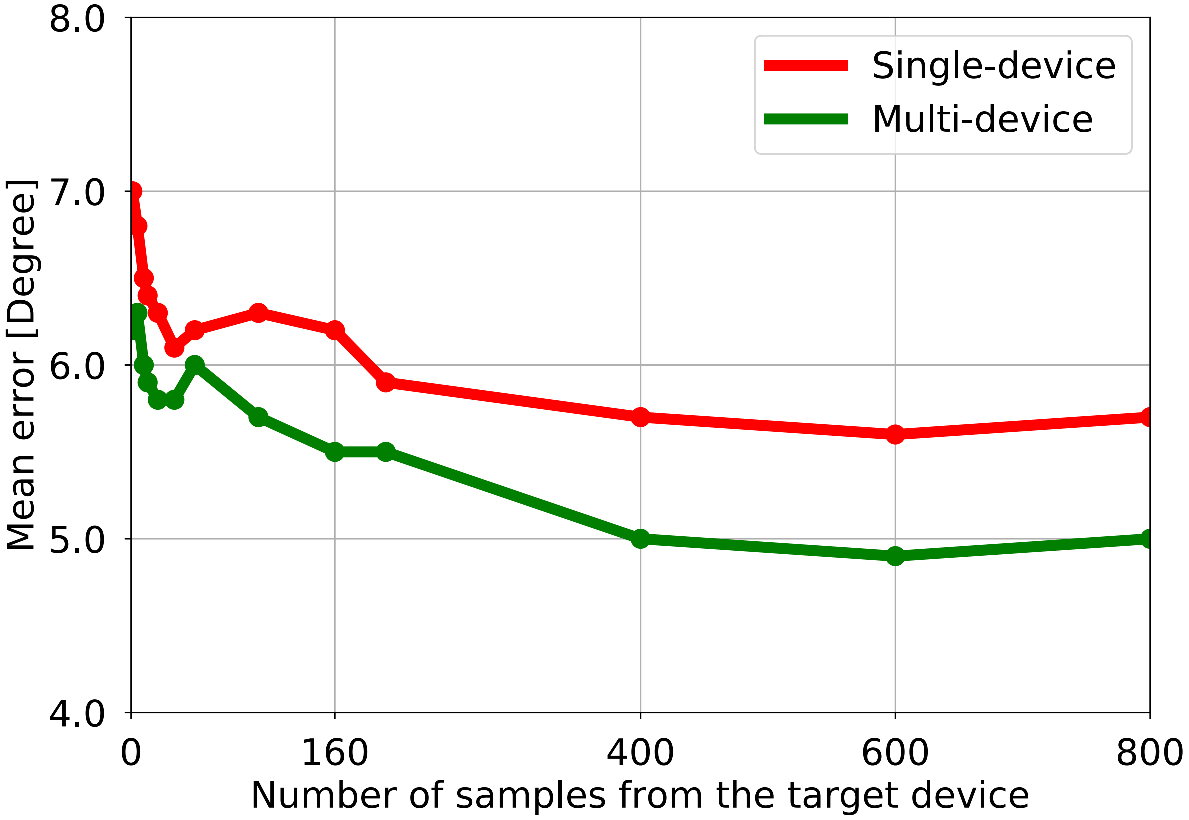


Figure 7.10: Evaluation of adding a new device to the multi-device CNN compared to the single-device CNN in the implicit calibration setting. The single-device CNN was trained on increasing target samples, which depended on the actual collected data on each device. The multi-device CNN was trained additionally on source samples from the source devices. The green line indicates the averaged performance of the multi-device CNN over five target devices; the red line shows that of the single-device CNN.

was trained additionally on 200 samples from each source device. Figure 7.9 shows the resulting performance. The x-axis indicates the number of target samples and the y-axis is the mean error in degrees averaged across the five devices and 20 participants. As can be seen from the figure, the proposed multi-device CNN generally outperformed the single-device counterparts, and achieved higher improvements with less data from the target device.

The corresponding results for the implicit calibration setting are shown in Figure 7.10. In this setting, the number of target samples depended on the actual interactions performed with each device during data collection (see Table 7.1). The y-axis shows the mean error in degrees averaged across the five devices and the 20 participants. As the figure shows, the performance for both multi-device and single-device CNN fluctuates as the number of target samples increases, most likely because the implicit calibration results in more noise in the training data. However, our multi-device CNN still consistently outperforms the single-device baseline given

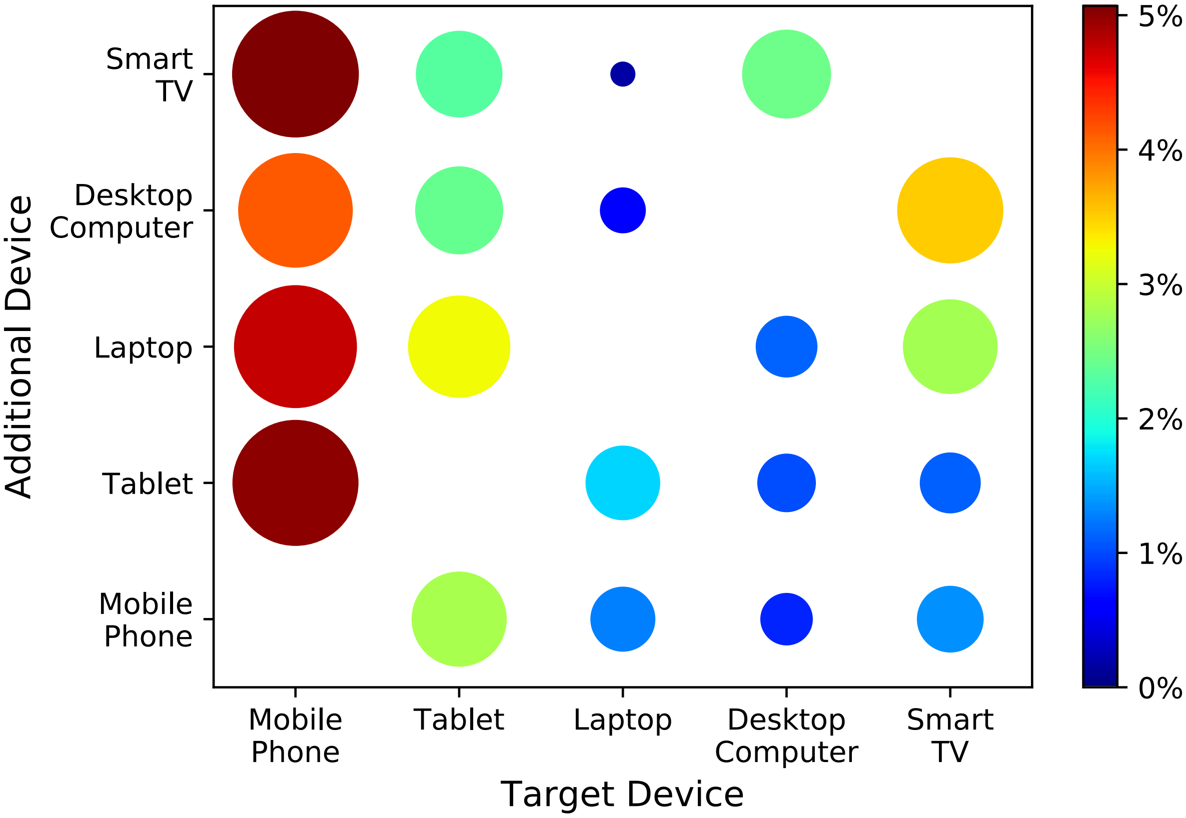


Figure 7.11: Relative performance improvements of a two-device CNN over the single-device CNN trained only on the target device samples in the explicit calibration setting. The x-axis shows the target device and the y-axis shows the source device. We used 20 target samples and 200 source samples from another device for multi-device CNN training. The bubble size and colour are proportional to the relative improvement.

sufficient target samples, indicating that the multi-device CNN is more robust to such noisy data.

7.4.4 Which Device Contributes Most to the Performance?

We finally conducted a fine-grained analysis of the contribution of the different source devices for multi-device learning on the target device. We took 20 explicit calibration samples from the target device for single-device CNN training, and trained our multi-device CNN with additional 200 explicit calibration samples from one source device. Figure 7.11 shows the relative improvement from this two-device CNN over the single-device CNN. We see that the relative improvement on the devices depends on their range of gaze directions. That is, the relative improvement is higher if the gaze direction ranges are similar (see Figure 7.4). For example, the desktop computer and smart TV have a higher impact on each other compared to the other devices, and all the other four devices lead to high relative improvements on the mobile phone since their ranges of gaze direction cover that of the mobile phone.

7.5 DISCUSSION

In this chapter we proposed a novel method for multi-device person-specific gaze estimation – to the best of our knowledge the first of its kind. Our extensive experiments on a novel 20-participant dataset of interactions with five different common device types demonstrated significant performance improvements and practical advantages over state-of-the-art single-device gaze estimation. We first demonstrated these improvements for an explicit calibration setting that resembles a standard 9-point calibration procedure widely used in eye tracking. We additionally demonstrated how to combine our method with an implicit calibration scheme in which we train with gaze locations derived from natural interactions, such as mouse or touch input. Our results also demonstrated significant performance improvements in this setting.

Tedious personal calibration is one of the most important obstacles and a main reason why learning-based gaze estimation has not yet made its way into many interactive systems deployed in public. As personal devices become ever more ubiquitous, the requirement to perform personal calibration on every single device is even more time-consuming and tedious. Our proposed multi-device gaze estimation method turns the ubiquity of personal devices and the large number of interactions that users perform with these devices on a daily basis into an advantage. It does so by leveraging both the shared and complementary image information across devices to significantly improve over most common single-device CNNs. Even more importantly, as we show experimentally, our proposed multi-device CNN can not only reach the same performance as a single-device CNN, but does so with much less training data. A single-device method could achieve a better performance, but only with an extensive data collection on each device at the cost of limited practicality and drastically reduced user experience. Our approach provides an alternative solution to this problem by leveraging training data from devices on which implicit data collection is more efficient. This is of particular importance for those devices on which implicit calibration data occurs infrequently, such as smart TV.

In summary, our method has significant potential to pave the way for a whole new range of gaze-based applications in the wild. Although we have experimented on five devices in our study, the proposed method is by nature scalable to different numbers of devices. With ongoing advances in smart homes and sensor-rich mobile devices, cameras are integrated into a variety of objects, such as devices or even just walls. Users may not intentionally interact with these objects. However, given only one training sample, our method can produce an acceptable gaze estimator for each camera. Therefore, every object that users face or interact with could understand their visual attention (Bulling, 2016) or even cognitive states (Huang *et al.*, 2016b).

Our experiments also revealed a fundamental challenge of learning from implicit and thus unreliable calibration data. Although we have not implemented any data alignment technique to handle this unreliability so far, our method could leverage the useful data from different devices to facilitate gaze learning. This shows that our method offers a certain robustness against noisy implicit calibration data. We expect the use of alignment techniques (Huang *et al.*, 2016a; Sugano *et al.*, 2015) to further improve the performance and practicality of our approach. Besides, our experimental results (Figure 7.11) also highlight the different contributions of different device/activity data. We believe that a future study can use an intelligent learning strategy to jointly optimise the source selection of training data as well as the data reliability.

7.6 CONCLUSION

In this chapter we proposed the first method for multi-device person-specific gaze estimation. Our method leverages device-specific encoders/decoders to adapt to device differences and uses shared feature extraction layers to encode the relation between personal facial appearance and gaze directions in a single representation shared across multiple devices. Our experiments demonstrated that our multi-device CNN outperforms single-device baselines for five different target devices. Furthermore, it could still improve the single-device CNN if it was trained with a sufficient amount of device-specific data. We also found that our method was more robust to noisy data than the single-device CNN. With the growing availability of camera-equipped devices, our method provides a practical and highly promising solution to personal gaze learning, thus opening up numerous opportunities for gaze-based applications in HCI and affective/cognitive computing.

8 CONCLUSIONS AND FUTURE PROSPECTS

Traditional gaze estimation methods, including model-based and feature-based methods (Hansen and Ji, 2010), were well studied in the past decades and facilitate gaze-based interactive systems. In the ideal case, these gaze estimation systems can reach 0.5 to one degree gaze estimation error. However, due to the requirement of eye feature detection, traditional gaze estimation methods usually require dedicated hardware such as high-resolution cameras and additional infrated light sources (Hansen and Pece, 2005; Ishikawa *et al.*, 2004). Although there are recent works that try to enable these methods to be used with a single camera, accurate eye feature detection is always the key bottleneck for their implementation in practice.

In contrast, appearance-based gaze estimation methods can work with low-quality images captured by webcam since they directly learn the mapping from input images to gaze targets (Tan *et al.*, 2002; Lu *et al.*, 2014b). However, due to the large dimensionality of input eye images, appearance-based gaze estimation methods, in general, suffer from low estimation accuracy, such as around 4 degrees of estimation error (Sugano *et al.*, 2014). Besides, appearance-based gaze estimation require more person-specific training data than traditional approaches to cover the significant variability in eye appearance caused by head poses, gaze directions and illumination conditions. Therefore, recent works proposed person-independent gaze estimation as training the model with a large group of participants that generalise to arbitrary persons without requiring additional input (Schneider *et al.*, 2014; Funes Mora and Odobez, 2013; Krafka *et al.*, 2016; Deng and Zhu, 2017). Despite significant advances in person-independent gaze estimation on the calibration-free direction, all previous works assumed training and test data to come from the same dataset. This essentially already assumes a prior domain-specific knowledge which cannot be acquired in real-world settings.

In this thesis, we focused on bringing gaze estimation and corresponding attentive user interfaces into real-world settings with a single webcam and without personal calibration. As the starting point of our research on gaze estimation methods, we collected a gaze estimation dataset, MPIIGaze, the first of its kind, under real-world environments from long-term recordings. With this dataset, we were able to test performances of state-of-the-art gaze estimation methods in real-world settings. We conducted the cross-dataset evaluations as training and testing on different datasets, along with cross-person evaluations and other detailed analysis on key challenges. These evaluations became an important step toward the unconstrained gaze estimation task as gaze estimation from a monocular RGB camera without assumptions regarding the user, environment or device. To extend the input space of appearance-based gaze estimation methods to a large input space, we studied gaze estimation with a single full-face patch. We proposed a novel spatial weights mechanism to efficiently encode different regions of the face, which achieved significant improvements over baselines. For the second part of this thesis, we validated our developed gaze estimation methods with applications of attentive user interfaces. We first developed an eye contact detection method that can be adapted to any arbitrary objects by assuming the target object is visually salient and closest to the camera. Our method uses an unsupervised data mining approach for collecting on-site training data, instead of tedious and time-consuming manual data annotation. We then proposed the first personal gaze estimation with multiple devices, such as cellphones, tablets, and laptop. We exploit the fact that a user usually has multiple personal devices, and develop a multi-device CNN to utilize data from multiple devices. Our method handles device-specific properties with encoders and decoders, and learns generic gaze estimation features with shared feature extraction layers.

In this chapter we further discuss the contributions of this thesis (Section 8.1) and review open problems as well as potential future prospects (Section 8.1).

8.1 DISCUSSION OF CONTRIBUTIONS

The overall goal of this thesis is to enable gaze-based attentive user interfaces in real-world settings. We tackled two specific research topics: *gaze estimation in real-world settings* and *attentive user interfaces*. In the following, we will discuss the main findings and insights of this thesis with respect to the individual chapters.

8.1.1 Appearance-based gaze estimation in real-world settings

The first part of this thesis is want to demonstrate the task of unconstrained gaze estimation and establish the research foundation for it.

Most of the previous works were evaluated in controlled laboratory settings (Sugano *et al.*, 2014; Funes Mora *et al.*, 2014) due to difficulties in handling the significant variability caused by challenging environments. In Chapter 3, we proposed a gaze estimation dataset, MPIIGaze, the first of its kind, collected from real-world settings. MPIIGaze includes a total of 213,659 images from 15 participants across ages, genders, and iris colours. Participants collected the data over different time periods ranging from 9 days to 3 months without constraints on location or time. Consequently, this dataset covers significant variation in illumination as well as natural head poses. With our MPIIGaze dataset, we were able to evaluate the state-of-the-art methods in real-world settings. We performed cross-dataset evaluation where the model was trained on large synthetic data, UT-Multiview, and tested on our MPIIGaze and also EYEDIAP datasets. This challenging evaluation has important practical meaning for how the state-of-the-art methods could perform for real-world settings with the model pre-trained on a different domain. It turned out to be a challenging task, although our proposed GazeNet achieved leading performance. We then studied key challenges including target gaze range, illumination conditions, and facial appearance variation. Our exclusive evaluations show that image resolution and the use of both eyes affect gaze estimation performance, while head pose and pupil centre information are less informative.

In Chapter 4, we proposed full-face gaze estimation to handle the high variability in input image appearance caused by head poses, gaze directions and illumination conditions. Most previous appearance-based gaze estimation works only used single eye image as input to the regressor and only a few considered alternative approaches, such as using two images, one of each eye (Huang *et al.*, 2017), or a single image covering both eyes (He *et al.*, 2015; Rikert and Jones, 1998). Krafka *et al.* (2016) recently presented a multi-region 2D gaze estimation method that took individual eye images, the face image, and a face grid as input. Their results suggested that adding the face image can be beneficial for appearance-based gaze estimation. Our full face-patch gaze estimation methods provide a detailed analysis of the potential of the full-face approach for 2D and 3D appearance-based gaze estimation tasks. Without telling the model the eye positions and head pose, our model achieved significant improvement with such an end-to-end approach over existing eye-only (Zhang *et al.*, 2015a) and multi-region (Krafka *et al.*, 2016) methods. We proposed a spatial weights mechanism to learn spatial weights on the activation maps of the convolutional layers, reflecting that the information is contained in different facial regions, which improves the model training efficiency.

In Chapter 5, we studied the data normalisation for the first time in a principled way and propose a modification that yields significant performance improvements. Data normalisation has been proposed to address the variability caused by unconstrained head pose by reducing the training space and making the training more efficient (Sugano *et al.*, 2014). This is achieved by preprocessing the training data before it is used as input to the gaze estimator. Although used successfully in prior works, the importance of rotation and translation/scaling of data normalisation remains unclear and its impact on the gaze estimation performance has not ye been quantified. We first explained the variability caused by different distances between camera and eye and discuss how data normalisation can cancel out some of this variability. Then, we demonstrated the importance of data normalisation for appearance-based gaze estimation with extensive experiments on both synthetic and real data.

8.1.2 Attentive user interfaces

In the last two chapters, we presented two applications of attentive user interfaces in real-world settings as validations of our developed gaze estimation methods.

In Chapter 6, we studied eye contact detection with a single webcam attached to the target object. Here, eye contact includes both human-object eye contact and human-human eye contact. Due to the difficulties of challenging real-world environments, existing gaze estimation methods cannot provide accurate eye contact detection. Compared to previous binary eye contact detection methods (Shell *et al.*, 2003b; Smith *et al.*, 2013), our method does not require dedicated hardware, prior geometry information, or human annotations. We exploited the fact that humans tend to have centre bias when looking at an object, which naturally results separated in gaze target clusters. By assuming the target object is visually salient and nearest to the camera, our method automatically labels the samples into positive and negative groups according to their relative positions to the camera. Detailed evaluations on two real-world scenarios, namely detecting eye contact at the workplace as well as during everyday social situations demonstrates that our method is effective for detecting eye contact independent of target object type and size, camera position, and user and recording environment.

In Chapter 7, we explored personal gaze estimation with multiple devices. Traditional methods for learning-based gaze estimation assume both user- and device-specific training data (Tan *et al.*, 2002; Williams *et al.*, 2006; Lu *et al.*, 2014a). While they could achieve better performance, it is usually quite impractical to assume large amounts of training data from each target user and device. Sugano *et al.* (2016) proposed an alternative method that combined aggregation of gaze data from multiple users on a public display with an on-site training data collection. We focused on another mode, a multi-device person-specific training task that has not been explored in the gaze estimation literature so far. The main difficulty for previous works has been that training a generic model for all devices causes too much unnecessary ambiguity due to device-specific properties, such as captured image quality and screen resolution. We addressed this issue by learning such device-specific features with device-specific encoders and decoders, and generic gaze estimation features with shared feature extraction layers. Exclusive evaluations on a new dataset of interactions with five common devices (mobile phone, tablet, laptop, desktop computer, smart TV) demonstrate the significant potential of multi-device training. We further explored training with gaze locations derived from natural interactions with three common applications (mobile game, text editing, media centre).

8.2 FUTURE PROSPECTS

In this section, we discuss several challenges that remain for enabling gaze-based interaction in real-world settings, and possible solutions to address them.

Unconstrained gaze estimation dataset. Although recent gaze estimation datasets (Sugano *et al.*, 2014; Wood *et al.*, 2016a; Krafka *et al.*, 2016), including our own (Zhang *et al.*, 2018c), cover multiple variations, a fully unconstrained dataset is still missing. This ideal dataset should include arbitrary head poses, gaze directions, illumination conditions and personal appearances. However, collecting such a dataset could be very expensive, especially for covering all of the aspects at the same time. For head pose capture, rendered 3D face meshes from multiple cameras could save effort since arbitrary head poses could be synthesised with computer graphic technologies (Sugano *et al.*, 2014; Deng and Zhu, 2017). Note that these methods require synchronised cameras and reliable rendering methods. Illumination conditions could be simulated in the indoor setting with additional lighting sources (Jones *et al.*, 2006), while how different such simulations would be from real illuminations is unclear. Gaze direction is another issue, as currently existing datasets do not consider the large range of depth for 3D gaze targets (Mansouryar *et al.*, 2016). Image synthesis with graphic models is another potential solution for generating large-scale unconstrained datasets with reliable ground truth (Wood *et al.*, 2016a). It has been proved that synthesised eye images can be modified to be realistic with adversarial training (Shrivastava *et al.*, 2017). For the graphic model, there is another challenge on modelling both eye and face, which should be consistent with each other. Considering all of these issues, building the unconstrained gaze estimation dataset is still a challenging task.

Investigation of full-face gaze estimation. It still remains a mystery how other face regions besides the eyes functionally improve gaze estimation performance. Intuitively, eye regions are sufficient for estimating eye gaze, and the rest of face regions should not provide more information than the head pose. More strictly speaking, only the iris contour shape can accurately estimate eye gaze by fitting to a pre-defined 3D eyeball model. However, our experiments in Chapter 4 clearly showed great benefit by using the full-face patch for gaze estimation rather than only eye regions. Very interestingly, we showed that the models trained with the face regions except for eyes could perform better than head pose information in Figure 4.7 in Chapter 4, which suggests these regions contribute more than just head pose information. There is the possibility that the model can estimate the head pose implicitly by taking the face-patch as input, which could outperform explicit head pose estimation by facial landmark fitting using the EPnP algorithm (Lepetit *et al.*, 2009). Besides this, illumination conditions can also be more easily estimated from face-patch rather than eye images. Another assumption is that the model can take other regions as reference features to estimate the eye feature locations. To further utilise the hidden information inside the face-patch, attention mechanisms (Xu *et al.*, 2015a) and multi-task learning (Zhang *et al.*, 2014b) could be potential solutions within this area of research.

Generic gaze estimation. Learning a generic gaze estimator to handle all the variability is a challenging task. Although we could leverage large-scale data with a powerful model, gaze estimation as a special task requires both sensitive feature extraction and tolerance of variability. Since pixel-level differences in the input eye/face images can cause significant deviation of gaze directions, the model has to be able to sense such subtle difference. Meanwhile, the input image appearances can vary due to head poses, illumination conditions and personal appearance, even though they share the same gaze direction. Therefore the model has to be able to ignore these ambiguities, which could be very noisy. These two counter-factors make the generic gaze estimation task very challenging, which calls for elegant model design. This generic gaze estimation across person appearances, head poses, gaze directions, environments and devices is the holy grail in learning-based gaze estimation.

Combining with other interactive methods. There is great potential to combine eye gaze and other interactive methods, such as hand gesture (Rautaray and Agrawal, 2015), human pose (Yao and Fei-Fei, 2010) and brain activity (Ferrez and Millán, 2008). Eye gaze, head pose and hand pose clearly have their own benefits in terms of accuracy and pervasiveness (Hansen *et al.*, 2004; Zhang *et al.*, 2015c), which can be utilized for interactive systems. Eye gaze could be taken as a target object selection method as it naturally reflects user attention, and another interactive method, such as hand gesture, could perform the actual action. Previously, the lack of accurate gaze estimation methods forced some interactive works to use head orientation as gaze information (Yoo *et al.*, 2010; Reale *et al.*, 2011). With the development of gaze estimation methods with a single webcam, we could now integrate eye gaze with other interactive methods, which share a single webcam as the input sensor. Brain activity is another potential interactive method can cooperate with eye gaze. There are many similarities between the tow modality, as hardware, personal calibration, noisy samples are challenges for both brain activity classification and gaze estimation. There could be a way to improve their performance due to the strong connection to user current attention. In summary, there are many possibilities for eye gaze combining with other interactive methods.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

F. Alnajar, T. Gevers, R. Valenti, and S. Ghebreab (2013). Calibration-Free Gaze Estimation Using Human Gaze Patterns, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) 2013*. Cited on page 15.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander (1999). OPTICS: ordering points to identify the clustering structure, in *ACM Sigmod record 1999*. Cited on page 90.

T. Baltrušaitis, P. Robinson, and L.-P. Morency (2016). Openface: an open source facial behavior analysis toolkit, in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on 2016*. Cited on page 89.

T. Baltrušaitis, P. Robinson, and L.-P. Morency (2014). Continuous Conditional Neural Fields for Structured Regression, in *Proc. Eur. Conf. Comput. Vis. (ECCV) 2014*. Cited on pages 32, 33, 34, 37, 58, and 110.

S. Baluja and D. Pomerleau (1994). Non-intrusive gaze tracking using artificial neural networks, in *Advances in Neural Inf. Process. Syst. 1994*. Cited on pages 14, 17, and 53.

J. J. Bengoechea, J. J. Cerrolaza, A. Villanueva, and R. Cabeza (2014). Evaluation of accurate eye corner detection methods for gaze estimation, *Journal of Eye Movement Research*, vol. 7(3). Cited on page 14.

R. Biedert, A. Dengel, G. Buscher, and A. Vartan (2012). Reading and estimating gaze on smart phones, in *Proceedings of the symposium on eye tracking research and applications 2012*. Cited on page 1.

G. Bradski (2000). The OpenCV Library, *Dr. Dobb's J. Softw. Tools*, vol. 25, pp. 120–123. Cited on page 30.

F. Brudy, D. Ledo, S. Greenberg, and A. Butz (2014). Is Anyone Looking? Mitigating Shoulder Surfing on Public Displays through Awareness and Protection, in *Proceedings of The International Symposium on Pervasive Displays 2014*. Cited on page 19.

A. Bulling (2016). Pervasive Attentive User Interfaces., *IEEE Computer*, vol. 49(1), pp. 94–98. Cited on pages 1, 19, 85, 105, and 122.

D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun (2014). Joint cascade face detection and alignment, in *Proc. Eur. Conf. Comput. Vis. 2014*. Cited on page 37.

J. Chen and Q. Ji (2008). 3D Gaze estimation with A Single Camera without IR Illumination, in *Proc. IEEE Int. Conf. Pattern Recognit. 2008*. Cited on page 14.

J. Chen and Q. Ji (2011). Probabilistic Gaze Estimation without Active Personal Calibration, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2011*. Cited on pages 15 and 21.

J. Chen, X. Liu, P. Tu, and A. Aragones (2013). Learning person-specific models for facial expression and action unit recognition, *Pattern Recognition Letters*, vol. 34(15), pp. 1964–1970. Cited on page 21.

J. Choi, B. Ahn, J. Parl, and I. S. Kweon (2013). Appearance-based Gaze Estimation Using Kinect, in *Proc. IEEE Conf. Ubiquitous Robots and Ambient Intell. 2013*. Cited on page 15.

R. Collobert and J. Weston (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning, in *Proceedings of the 25th international conference on Machine learning 2008*. Cited on page 21.

S. Cristina and K. P. Camilleri (2016). Model-based Head Pose-free Gaze Estimation for Assistive Communication, *Comput. Vis. Image Understanding*, vol. 149, pp. 157–170. Cited on pages 1 and 14.

P. Debevec (2006). Image-based lighting, in *ACM SIGGRAPH 2006 Courses 2006*. Cited on page 7.

H. Deng and W. Zhu (2017). Monocular Free-head 3D Gaze Tracking with Deep Learning and Geometry Constraints, in *Computer Vision (ICCV), 2017 IEEE International Conference on 2017*. Cited on pages 16, 69, 125, and 128.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database, in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on 2009*. Cited on pages 76 and 114.

L. Deng, G. Hinton, and B. Kingsbury (2013). New types of deep neural network learning for speech recognition and related applications: An overview, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on 2013*. Cited on page 21.

C. Dickie, R. Vertegaal, D. Fono, C. Sohn, D. Chen, D. Cheng, J. S. Shell, and O. Aoudeh (2004a). Augmenting and sharing memory with eyeBlog, in *Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences 2004*. Cited on page 87.

C. Dickie, R. Vertegaal, J. S. Shell, C. Sohn, D. Cheng, and O. Aoudeh (2004b). Eye contact sensing glasses for attention-sensitive wearable video blogging, in *CHI'04 extended abstracts on Human factors in computing systems 2004*. Cited on page 20.

S. D'Mello, A. Olney, C. Williams, and P. Hays (2012). Gaze tutor: A gaze-reactive intelligent tutoring system, *International Journal of human-computer studies*, vol. 70(5), pp. 377–398. Cited on page 53.

S. Duffner and C. Garcia (2016). Visual focus of attention estimation with unsupervised incremental learning, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26(12), pp. 2264–2272. Cited on page 93.

S. R. Edmunds, A. Rozga, Y. Li, E. A. Karp, L. V. Ibanez, J. M. Rehg, and W. L. Stone (2017). Brief Report: Using a Point-of-View Camera to Measure Eye Gaze in Young Children with Autism Spectrum Disorder During Naturalistic Social Interactions: A Pilot Study, *Journal of Autism and Developmental Disorders*, pp. 1–7. Cited on page 88.

A. Esteves, E. Velloso, A. Bulling, and H. Gellersen (2015). Orbits: Gaze interaction for smart watches using smooth pursuit eye movements, in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology 2015*. Cited on page 1.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin (2008). LIBLINEAR: A Library for Large Linear Classification, *J. Mach. Learning Res.*, vol. 9, pp. 1871–1874. Cited on page 39.

A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. Morris (2017). Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems 2017*. Cited on pages 1 and 2.

P. W. Ferrez and J. d. R. Millán (2008). Error-related EEG potentials generated during simulated brain–computer interaction, *IEEE transactions on biomedical engineering*, vol. 55(3), pp. 923–929. Cited on page 130.

K. A. Funes Mora, F. Monay, and J.-M. Odobez (2014). EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras, in *Proc. ACM Symp. Eye Tracking Res. Appl. (ETRA) 2014*. Cited on pages 16, 26, 27, 28, 31, 39, 51, 59, 69, 73, 89, 105, 114, 126, and 139.

K. A. Funes Mora and J.-M. Odobez (2012). Gaze Estimation From Multimodal Kinect Data, in *IEEE Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW) 2012*. Cited on page 15.

K. A. Funes Mora and J.-M. Odobez (2013). Person Independent 3D Gaze Estimation From Remote RGB-D Cameras, in *Proc. IEEE Int. Conf. on Image Process. 2013*. Cited on pages 15, 17, 25, 38, and 125.

K. A. Funes Mora and J.-M. Odobez (2014). Geometric Generative Gaze Estimation (G3E) for Remote RGB-D Cameras, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2014*. Cited on pages 2, 4, and 14.

T. Gao, D. Harari, J. Tenenbaum, and S. Ullman (2014). When Computer Vision Gazes at Cognition, Technical report, Cambridge, MA, USA: Center for Brains, Minds and Machines. Cited on page 15.

R. Girshick (2015). Fast R-CNN, in *International Conference on Computer Vision (ICCV) 2015*. Cited on page 18.

E. D. Guestrin and M. Eizenman (2006). General theory of remote gaze estimation using the pupil center and corneal reflections, *IEEE Transactions on biomedical engineering*, vol. 53(6), pp. 1124–1133. Cited on page 14.

Q. Guo and E. Agichtein (2010). Towards Predicting Web Searcher Gaze Position from Mouse Movements, in *Ext. Abstracts CHI 2010*. Cited on page 1.

D. W. Hansen and Q. Ji (2010). In the Eye of the Beholder: A Survey of Models for Eyes and Gaze, *IEEE Trans. Pattern Anal. and Mach. Intell. (PAMI)*, vol. 32(3), pp. 478–500. Cited on pages 1, 13, 53, 69, and 125.

D. W. Hansen and A. E. Pece (2005). Eye tracking in the wild, *Computer Vision and Image Understanding*, vol. 98(1), pp. 155–181. Cited on pages 13 and 125.

J. P. Hansen, K. Tørning, A. S. Johansen, K. Itoh, and H. Aoki (2004). Gaze typing compared with input by head and hand, in *Proceedings of the 2004 symposium on Eye tracking research & applications 2004*. Cited on page 130.

K. He, X. Zhang, S. Ren, and J. Sun (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition, in *European Conference on Computer Vision (ECCV) 2014*. Cited on page 18.

Q. He, X. Hong, X. Chai, J. Holappa, G. Zhao, X. Chen, and M. Pietikäinen (2015). OMEG: Oulu Multi-Pose Eye Gaze Dataset, in *Image Anal. 2015*, pp. 418–427. Cited on pages 16, 18, 28, 51, 69, and 127.

C. Hennessey, B. Noureddin, and P. Lawrence (2006). A Single Camera Eye-gaze Tracking System with Free Head Motion, in *Proc. ACM Symp. Eye Tracking Res. Appl. 2006*. Cited on pages 1 and 14.

E. Horvitz, C. Kadie, T. Paek, and D. Hovel (2003). Models of attention in computing and communication: from principles to applications, *Communications of the ACM*, vol. 46(3), pp. 52–59. Cited on page 19.

J. Huang, R. White, and G. Buscher (2012). User See, User Point: Gaze and Cursor Alignment in Web Search, in *Proc. CHI 2012*. Cited on pages 1 and 21.

M. X. Huang, T. C. Kwok, G. Ngai, S. C. Chan, and H. V. Leong (2016a). Building a personalized, auto-calibrating eye tracker from user interactions, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems 2016*. Cited on pages 21, 93, 106, and 123.

M. X. Huang, T. C. Kwok, G. Ngai, H. V. Leong, and S. C. Chan (2014). Building a Self-Learning Eye Gaze Model from User Interaction Data, in *Proc. Int. Conf. on Multimedia 2014*. Cited on page 15.

M. X. Huang, J. Li, G. Ngai, and H. V. Leong (2016b). StressClick: Sensing Stress from Gaze-Click Patterns, in *Proceedings of the 2016 ACM on Multimedia Conference 2016*. Cited on pages 19, 105, and 122.

Q. Huang, A. Veeraraghavan, and A. Sabharwal (2017). TabletGaze: Dataset and Analysis for Unconstrained Appearance-Based Gaze Estimation in Mobile Tablets, *Mach. Vis. Appl.*, vol. 28(5), pp. 445–461. Cited on pages 16, 18, 28, 33, 51, 55, 109, and 127.

T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey (1989). Human-computer interaction using eye-gaze input, *IEEE Transactions on systems, man, and cybernetics*, vol. 19(6), pp. 1527–1534. Cited on page 1.

M. Imai, T. Ono, and H. Ishiguro (2003). Physical relation and expression: Joint attention for human-robot interaction, *IEEE Transactions on Industrial Electronics*, vol. 50(4), pp. 636–643. Cited on page 87.

T. Ishikawa, S. Baker, I. Matthews, and T. Kanade (2004). Passive Driver Gaze Tracking with Active Appearance Models, in *Proc. 11th World Congr. Intell. Transportation Syst. 2004*. Cited on pages 13 and 125.

M. Jaderberg, K. Simonyan, A. Zisserman, *et al.* (2015). Spatial transformer networks, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on page 18.

L. A. Jeni and J. F. Cohn (2016). Person-independent 3D Gaze Estimation using Face Frontalization, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW) 2016*. Cited on page 15.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell (2014). Caffe: Convolutional Architecture for Fast Feature Embedding, in *Proc. Int. Conf. Multimedia 2014*. Cited on pages 38, 59, 76, and 114.

L. Jianfeng and L. Shigang (2014). Eye-Model-Based Gaze Estimation by RGB-D Camera, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops 2014*. Cited on page 14.

A. Jones, A. Gardner, M. Bolas, I. Mcdowall, and P. Debevec (2006). Simulating spatially varying lighting on a live performance. Cited on page 129.

L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit (2017). One Model To Learn Them All, *arXiv preprint arXiv:1706.05137*. Cited on pages 21, 107, and 108.

M. Kassner, W. Patera, and A. Bulling (2014). Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction, in *Proc. Int. Conf. Pervasive Ubiquitous Comput.: Adjunct Publication 2014*. Cited on page 29.

D. E. King (2009). Dlib-ml: A Machine Learning Toolkit, *J. Mach. Learning Res.*, vol. 10, pp. 1755–1758. Cited on pages 33, 34, 89, and 110.

D. Kingma and J. Ba (2015). Adam: A Method for Stochastic Optimization, *The Int. Conf. on Learning Representations*. Cited on pages 38, 76, and 114.

C. L. Kleinke (1986). Gaze and eye contact: a research review., *Psychological bulletin*, vol. 100(1), p. 78. Cited on pages 7 and 87.

K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba (2016). Eye Tracking for Everyone, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2016*. Cited on pages 5, 6, 15, 17, 18, 19, 28, 33, 51, 53, 54, 55, 59, 60, 69, 70, 73, 105, 109, 125, 127, and 128.

P. O. Kristensson and K. Vertanen (2012). The potential of dwell-free eye-typing for fast assistive gaze communication, in *Proceedings of the Symposium on Eye Tracking Research and Applications 2012*. Cited on pages 1, 19, and 105.

A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). Imagenet Classification with Eeep Convolutional Neural Networks, in *Proc. NIPS 2012*. Cited on pages 18, 58, 76, 107, and 114.

M. F. Land and M. Hayhoe (2001). In what ways do eye movements contribute to everyday activities?, *Vision research*, vol. 41(25), pp. 3559–3565. Cited on page 87.

R. Larson and M. Csikszentmihalyi (1983). The Experience Sampling Method, *New Directions for Methodology of Social & Behavioral Science*, vol. 15, pp. 41–56. Cited on page 29.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based Learning Applied to Document Recognition, *Proc. of the IEEE*, vol. 86(11), pp. 2278–2324. Cited on pages 38 and 59.

V. Lepetit, F. Moreno-Noguer, and P. Fua (2009). EPnP: An Accurate O(n) Solution to the PnP Problem, *Int. J Comput. Vis.*, vol. 81(2), pp. 155–166. Cited on pages 35 and 129.

Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille (2014). The Secrets of Salient Object Segmentation, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2014*. Cited on page 26.

K. Liang, Y. Chahir, M. Molina, C. Tijus, and F. Jouen (2013). Appearance-based Gaze Tracking with Spectral Clustering and Semi-supervised Gaussian Process Regression, in *Proc. Conf. Eye Tracking South Africa 2013*. Cited on page 14.

F. Lu, T. Okabe, Y. Sugano, and Y. Sato (2014a). Learning Gaze Biases with Head Motion for Head Pose-free Gaze Estimation, *Image and Vis. Comput.*, vol. 32(3), pp. 169 – 179. Cited on pages 2, 5, 15, 17, 21, 55, and 128.

F. Lu, Y. Sugano, T. Okabe, and Y. Sato (2012). Head Pose-free Appearance-based Gaze Sensing via Eye Image Synthesis, in *Proc. IEEE Int. Conf. Pattern Recognit. 2012*. Cited on pages 2 and 15.

F. Lu, Y. Sugano, T. Okabe, and Y. Sato (2014b). Adaptive Linear Regression for Appearance-Based Gaze Estimation, *IEEE Trans. Pattern Anal. and Mach. Intell. (PAMI)*, vol. 36(10), pp. 2033–2046. Cited on pages 4, 15, 37, 38, 53, 55, and 125.

F. Lu, Y. Sugano, T. Okabe, and Y. Sato (2015). Gaze Estimation From Eye Appearance: A Head Pose-Free Method via Eye Image Synthesis, *IEEE Transactions on Image Processing*, vol. 24(11), pp. 3680–3693. Cited on pages 5, 17, and 55.

P. P. Maglio, R. Barrett, C. S. Campbell, and T. Selker (2000a). SUITOR: An attentive information system, in *Proceedings of the 5th international conference on Intelligent user interfaces 2000*. Cited on page 19.

P. P. Maglio, T. Matlock, C. S. Campbell, S. Zhai, and B. A. Smith (2000b). Gaze and speech in attentive user interfaces, in *Advances in Multimodal Interfaces-ICMI 2000 2000*, pp. 1–7, Springer. Cited on page 19.

P. Majaranta and A. Bulling (2014). Eye Tracking and Eye-based Human-computer Interaction, in *Advances in physiological computing 2014*, pp. 39–65, Springer. Cited on pages 25 and 29.

P. Majaranta and K.-J. Räihä (2002). Twenty years of eye typing: systems and design issues, in *Proceedings of the 2002 symposium on Eye tracking research & applications 2002*. Cited on page 1.

M. Mansouryar, J. Steil, Y. Sugano, and A. Bulling (2016). 3d gaze estimation from 2d pupil positions on monocular head-mounted eye trackers, in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications 2016*. Cited on page 129.

C. D. McMurrough, V. Metsis, J. Rich, and F. Makedon (2012). An Eye Tracking Dataset for Point of Gaze Detection, in *Proc. ACM Symp. Eye Tracking Res. Appl. (ETRA) 2012*. Cited on pages 16, 28, and 51.

J. Merchant, R. Morrissette, and J. L. Porterfield (1974). Remote measurement of eye direction allowing subject motion over one cubic foot of space, *IEEE transactions on biomedical engineering*, (4), pp. 309–317. Cited on page 14.

K. A. F. Mora and J.-M. Odobez (2016). Gaze estimation in the 3D space using RGB-D sensors-yowards head-pose and user invariance., *Int. J. Comput. Vis.*, vol. 118(2), pp. 194–216. Cited on pages 17, 19, 42, and 55.

C. H. Morimoto, A. Amir, and M. Flickner (2002). Detecting Eye Position and Gaze from A Single Camera and 2 Light Sources, in *Proc. IEEE Int. Conf. Pattern Recognit. 2002*. Cited on page 14.

P. Müller, M. X. Huang, X. Zhang, and A. Bulling (2018). Robust Eye Contact Detection in Natural Multi-Person Interactions Using Gaze and Speaking Behaviour, in *Proc. International Symposium on Eye Tracking Research and Applications (ETRA) 2018*. Cited on page 18.

B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita (2009). Footing in human-robot conversations: how robots might shape participant roles using gaze cues, in *International Conference on Human-Robot Interaction (HRI) 2009*. Cited on page 53.

H. Nam and B. Han (2016). Learning multi-domain convolutional neural networks for visual tracking, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016*. Cited on page 21.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng (2011). Multimodal deep learning, in *Proc. Int. Conf. Mach. Learning 2011*. Cited on page 37.

A. Nuthmann and J. M. Henderson (2010). Object-based attentional selection in scene viewing, *Journal of vision*, vol. 10(8), pp. 20–20. Cited on page 88.

A. Oh, H. Fox, M. Van Kleek, A. Adler, K. Gajos, L.-P. Morency, and T. Darrell (2002). Evaluating look-to-talk: a gaze-aware interface in a collaborative environment, in *CHI'02 Extended Abstracts on Human Factors in Computing Systems 2002*. Cited on page 19.

A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays (2016). Webgazer: Scalable webcam eye tracking using user interactions, in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016 2016*. Cited on page 21.

K. E. Pearce and R. E. Rice (2013). Digital divides from access to activities: Comparing mobile and personal computer Internet users, *Journal of Communication*, vol. 63(4), pp. 721–744. Cited on page 112.

V. Ponz, A. Villanueva, and R. Cabeza (2012). Dataset for the Evaluation of Eye Detector for Gaze Estimation, in *Proc. ACM Conf. Ubiquitous Comput. 2012*. Cited on page 51.

S. S. Rautaray and A. Agrawal (2015). Vision based hand gesture recognition for human computer interaction: a survey, *Artificial Intelligence Review*, vol. 43(1), pp. 1–54. Cited on page 130.

M. J. Reale, S. Canavan, L. Yin, K. Hu, and T. Hung (2011). A multi-gesture interaction system using a 3-D iris disk model for gaze estimation and an active appearance model for 3-D hand pointing, *IEEE Transactions on Multimedia*, vol. 13(3), pp. 474–486. Cited on page 130.

A. Recasens, A. Khosla, C. Vondrick, and A. Torralba (2015). Where are they looking?, in *Advances in Neural Information Processing Systems 2015*. Cited on pages 20 and 93.

A. Recasens, C. Vondrick, A. Khosla, and A. Torralba (2016). Following Gaze Across Views, *arXiv preprint arXiv:1612.03094*. Cited on page 20.

T. D. Rikert and M. J. Jones (1998). Gaze estimation using morphable models, in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on 1998*. Cited on pages 18 and 127.

R. Rodrigues, J. a. Barreto, and U. Nunes (2010). Camera Pose Estimation Using Images of Planar Mirror Reflections, in *Proc. Eur. Conf. Comput. Vis. 2010*. Cited on page 30.

H. Sattar, S. Müller, M. Fritz, and A. Bulling (2015). Prediction of Search Targets From Fixations in Open-World Settings, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2015*. Cited on page 25.

T. Schneider, B. Schauerte, and R. Stiefelhagen (2014). Manifold Alignment for Person Independent Appearance-Based Gaze Estimation, in *Proc. IEEE Int. Conf. Pattern Recognit. 2014*. Cited on pages 15, 25, 38, 39, 69, and 125.

J. Schrammel, E. Mattheiss, S. Döbelt, L. Paletta, A. Almer, and M. Tscheligi (2011). Attentional behavior of users on the move towards pervasive advertising media, in *Pervasive Advertising 2011*, pp. 287–307, Springer. Cited on page 1.

C. Seifert, A. Mitschick, J. Schlötterer, and R. Dachselt (2017). Focus Paragraph Detection for Online Zero-Effort Queries: Lessons learned from Eye-Tracking Data, in *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval 2017*. Cited on pages 19 and 105.

T. Selker, A. Lockerd, and J. Martinez (2001). Eye-R, a glasses-mounted eye motion detection interface, in *CHI'01 extended abstracts on Human factors in computing systems 2001*. Cited on page 20.

S. Seok and B. DaCosta (2015). Predicting video game behavior: An investigation of the relationship between personality and mobile game play, *Games and Culture*, vol. 10(5), pp. 481–501. Cited on page 112.

L. Sesma, A. Villanueva, and R. Cabeza (2012). Evaluation of pupil center-eye corner vector for gaze estimation using a web cam, in *Proceedings of the symposium on eye tracking research and applications 2012*. Cited on pages 14 and 69.

W. Sewell and O. Komogortsev (2010). Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network, in *Ext. Abstr. ACM CHI Conf. on Human Factors in Comput. Syst. 2010*. Cited on pages 2, 4, and 14.

J. S. Shell, T. Selker, and R. Vertegaal (2003a). Interacting with groups of computers, *Communications of the ACM*, vol. 46(3), pp. 40–46. Cited on page 87.

J. S. Shell, R. Vertegaal, D. Cheng, A. W. Skaburskis, C. Sohn, A. J. Stewart, O. Aoudeh, and C. Dickie (2004). ECSGlasses and EyePliances: using attention to open sociable windows of interaction, in *Proceedings of the 2004 symposium on Eye tracking research & applications 2004*. Cited on pages 8, 20, 87, and 88.

J. S. Shell, R. Vertegaal, and A. W. Skaburskis (2003b). EyePliances: attention-seeking devices that respond to visual attention, in *CHI'03 extended abstracts on Human factors in computing systems 2003*. Cited on pages 20, 88, and 127.

S.-W. Shih and J. Liu (2004). A novel approach to 3-D gaze tracking using stereo cameras, *IEEE Trans. Syst. Man Cybern., Part B: Cybernetics*, vol. 34(1), pp. 234–245. Cited on page 14.

A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb (2017). Learning from simulated and unsupervised images through adversarial training, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 7, 16, 18, 20, 70, and 129.

K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, in *Int. Conf. Learning Representations 2015*. Cited on pages 18, 37, 38, and 132.

B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar (2013). Gaze locking: passive eye contact detection for human-object interaction, in *Proc. ACM Symp. User Interface Softw. Technol. 2013*. Cited on pages 8, 15, 16, 20, 27, 28, 51, 69, 88, 93, 94, 95, 128, and 139.

J. D. Smith, R. Vertegaal, and C. Sohn (2005). ViewPointer: lightweight calibration-free eye tracking for ubiquitous handsfree deixis, in *Proceedings of the 18th annual ACM symposium on User interface software and technology 2005*. Cited on page 20.

K. Smith, S. O. Ba, J.-M. Odobez, and D. Gatica-Perez (2008). Tracking the visual focus of attention for a varying number of wandering people, *IEEE transactions on pattern analysis and machine intelligence*, vol. 30(7), pp. 1212–1229. Cited on page 19.

R. Stiefelhagen, J. Yang, and A. Waibel (2002). Modeling focus of attention for meeting indexing based on multiple cues, *IEEE Transactions on Neural Networks*, vol. 13(4), pp. 928–938. Cited on page 93.

Y. Sugano and A. Bulling (2015). Self-Calibrating Head-Mounted Eye Trackers Using Egocentric Visual Saliency, in *Proc. ACM Symposium on User Interface Software and Technology (UIST) 2015*. Cited on page 21.

Y. Sugano, Y. Matsushita, and Y. Sato (2013). Appearance-Based Gaze Estimation Using Visual Saliency, *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 35(2), pp. 329–341. Cited on pages 15 and 21.

Y. Sugano, Y. Matsushita, and Y. Sato (2014). Learning-by-synthesis for appearance-based 3D gaze estimation, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2014*. Cited on pages 2, 3, 4, 6, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28, 31, 33, 34, 35, 36, 37, 38, 41, 49, 51, 56, 70, 71, 72, 73, 74, 75, 76, 77, 84, 105, 109, 110, 125, 126, 127, 128, 135, and 139.

Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike (2008). An incremental learning method for unconstrained gaze estimation, in *Proc. Eur. Conf. Comput. Vis. 2008*. Cited on pages 15 and 106.

Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike (2015). Appearance-based gaze estimation with online calibration from mouse operations, *Trans. Human-Mach. Syst.*, vol. 45(6), pp. 750–760. Cited on pages 17, 21, 37, 55, and 123.

Y. Sugano, X. Zhang, and A. Bulling (2016). AggreGaze: Collective Estimation of Audience Attention on Public Displays, in *Proc. ACM Symp. User Interface Softw. Technol. (UIST) 2016*. Cited on pages 9, 18, 19, 21, 25, 51, 69, 87, 93, 102, 105, 109, and 128.

L. Sun, M. Song, Z. Liu, and M.-T. Sun (2014). Real-time gaze estimation with online calibration, *IEEE MultiMedia*, vol. 21(4), pp. 28–37. Cited on page 14.

G. P. Szeto and R. Lee (2002). An ergonomic evaluation comparing desktop, notebook, and subnotebook computers, *Archives of physical medicine and rehabilitation*, vol. 83(4), pp. 527–532. Cited on page 113.

K.-H. Tan, D. J. Kriegman, and N. Ahuja (2002). Appearance-based eye gaze estimation, in *Proc. IEEE Workshop Appl. Comput. Vis. 2002*. Cited on pages 2, 4, 14, 17, 21, 53, 55, 69, 125, and 128.

J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler (2015). Efficient object localization using convolutional networks, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 18, 57, and 58.

M. Tonsen, X. Zhang, Y. Sugano, and A. Bulling (2016). Labelled Pupils in the Wild: A Dataset for Studying Pupil Detection in Unconstrained Environments, in *Proc. ACM Symp. Eye Tracking Res. Appl. (ETRA) 2016*. Cited on page 32.

A. Torralba and A. A. Efros (2011). Unbiased look at dataset bias, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2011*. Cited on page 26.

R. Valenti, N. Sebe, and T. Gevers (2012). Combining head pose and eye location information for gaze estimation, *IEEE Trans. Image Process.*, vol. 21(2), pp. 802–815. Cited on pages 14, 17, 55, and 69.

R. Vertegaal, C. Dickie, C. Sohn, and M. Flickner (2002). Designing attentive cell phone using wearable eyecontact sensors, in *CHI'02 extended abstracts on Human factors in computing systems 2002*. Cited on pages 20 and 88.

R. Vertegaal and J. S. Shell (2008). Attentive user interfaces: the surveillance and sousveillance of gaze-aware objects, *Social Science Information*, vol. 47(3), pp. 275–298. Cited on page 19.

A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza (2013). Hybrid method based on topography for robust detection of iris center and eye corners, *Trans. Multimedia Comput., Commun. Appl.*, vol. 9(4), p. 25. Cited on pages 16 and 28.

A. Vinciarelli, M. Pantic, H. Bourlard, and A. Pentland (2008). Social signal processing: state-of-the-art and future perspectives of an emerging domain, in *Proceedings of the 16th ACM international conference on Multimedia 2008*. Cited on page 53.

M. Voit and R. Stiefelhagen (2008). Deducing the visual focus of attention from head pose estimation in dynamic multi-view meeting scenarios, in *Proceedings of the 10th international conference on Multimodal interfaces 2008*. Cited on page 93.

K. Wang and Q. Ji (2017). Real Time Eye Gaze Tracking with 3D Deformable Eye-Face Model, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. Cited on pages 14 and 69.

K. Wang, S. Wang, and Q. Ji (2016). Deep eye fixation map learning for calibration-free eye gaze tracking, in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications 2016*. Cited on page 21.

K. Wang, R. Zhao, and Q. Ji (2018). A Hierarchical Generative Model for Eye Image Synthesis and Eye Gaze Estimation, in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on 2018*. Cited on page 16.

U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann (2007). A comprehensive head pose and gaze database, in *Proc. 3rd IET Int. Conf. Intell. Environ. 2007*. Cited on pages 16, 28, and 51.

P. Weill-Tessier and H. Gellersen (2017). Touch Input and Gaze Correlation on Tablets, in *International Conference on Intelligent Decision Technologies 2017*. Cited on pages 106 and 110.

F. Wilcoxon (1945). Individual Comparisons by Ranking Methods, *Biometrics Bulletin*, vol. 1(6), pp. 80–83. Cited on page 40.

O. Williams, A. Blake, and R. Cipolla (2006). Sparse and Semi-supervised Visual Mapping with the Sˆ 3GP, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2006*. Cited on pages 14, 17, 21, 53, 69, and 128.

E. Wood, T. Baltrusaitis, L.-P. Morency, P. Robinson, and A. Bulling (2016a). A 3D Morphable Eye Region Model for Gaze Estimation, in *Proc. Eur. Conf. Comput. Vis. (ECCV) 2016*. Cited on pages 7, 16, 70, 128, and 129.

E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling (2016b). Learning An Appearance-based Gaze Estimator from One Million Synthesised Images, in *Proc. ACM Symp. Eye Tracking Res. Appl. (ETRA) 2016*. Cited on pages 16, 25, 70, and 109.

E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling (2015). Rendering of Eyes for Eye-Shape Registration and Gaze Estimation, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) 2015*. Cited on pages 7, 9, 16, 17, 18, 19, 26, 39, 41, 42, 44, 55, 56, 70, and 73.

E. Wood and A. Bulling (2014). EyeTab: Model-based gaze estimation on unmodified tablet computers, in *Proc. ACM Symp. Eye Tracking Res. Appl. 2014*. Cited on pages 14, 25, 39, 41, 49, and 51.

P. Xanthopoulos and T. Razzaghi (2014). A weighted support vector machine method for control chart pattern recognition, *Computers & Industrial Engineering*, vol. 70, pp. 134–149. Cited on page 91.

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio (2015a). Show, attend and tell: Neural image caption generation with visual attention, in *International Conference on Machine Learning 2015*. Cited on page 129.

P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao (2015b). Turkergaze: Crowdsourcing saliency with webcam based eye tracking, Technical report, Princeton, N.J, USA: Department of Comput. Sci. Cited on page 17.

H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe (2008). Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions, in *Proc. ACM Symp. Eye Tracking Res. Appl. 2008*. Cited on pages 14 and 69.

B. Yao and L. Fei-Fei (2010). Modeling mutual context of object and human pose in human-object interaction activities, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on 2010*. Cited on page 130.

Z. Ye, Y. Li, A. Fathi, Y. Han, A. Rozga, G. D. Abowd, and J. M. Rehg (2012). Detecting eye contact using wearable eye-tracking glasses, in *Proceedings of the 2012 ACM conference on ubiquitous computing 2012*. Cited on page 87.

Z. Ye, Y. Li, Y. Liu, C. Bridges, A. Rozga, and J. M. Rehg (2015). Detecting bids for eye contact using a wearable camera, in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on 2015*. Cited on pages 20 and 88.

B. Yoo, J.-J. Han, C. Choi, K. Yi, S. Suh, D. Park, and C. Kim (2010). 3D user interface combining gaze and hand gestures for large-scale display, in *CHI'10 Extended Abstracts on Human Factors in Computing Systems 2010*. Cited on pages 93 and 130.

D. H. Yoo and M. J. Chung (2005). A novel non-intrusive eye gaze estimation using cross-ratio under large head motion, *Comput. Vis. and Image Understanding*, vol. 98(1), pp. 25–51. Cited on page 14.

P. Yu, J. Zhou, and Y. Wu (2016). Learning Reconstruction-Based Remote Gaze Estimation, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2016*. Cited on pages 15 and 69.

M. D. Zeiler and R. Fergus (2014). Visualizing and understanding convolutional networks, in *European Conference on Computer Vision (ECCV) 2014*. Cited on page 63.

X. Zhang, M. X. Huang, Y. Sugano, and A. Bulling (2018a). Training Person-Specific Gaze Estimators from Interactions with Multiple Devices, in *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI) 2018*. Cited on pages 11, 19, and 69.

X. Zhang, H. Kulkarni, and M. R. Morris (2017a). Smartphone-Based Gaze Gesture Communication for People with Motor Disabilities, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems 2017*. Cited on pages 1 and 69.

X. Zhang, Y. Sugano, and A. Bulling (2017b). Everyday Eye Contact Detection Using Unsupervised Gaze Target Discovery, in *Proc. ACM Symp. User Interface Softw. Technol. (UIST) 2017*. Cited on pages 11, 19, and 105.

X. Zhang, Y. Sugano, and A. Bulling (2017c). Everyday Eye Contact Detection Using Unsupervised Gaze Target Discovery, in *Proc. ACM Symp. User Interface Softw. Technol. (UIST) 2017*. Cited on pages 18, 51, and 69.

X. Zhang, Y. Sugano, and A. Bulling (2018b). Revisiting Data Normalization for Appearance-Based Gaze Estimation, in *Proc. International Symposium on Eye Tracking Research and Applications (ETRA) 2018*. Cited on pages 10 and 17.

X. Zhang, Y. Sugano, M. Fritz, and A. Bulling (2015a). Appearance-based Gaze Estimation in the Wild, in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2015*. Cited on pages 6, 9, 10, 16, 17, 18, 19, 20, 32, 38, 39, 40, 42, 51, 53, 54, 55, 58, 59, 63, 70, 88, 89, 93, 96, 105, 109, 113, 114, 116, and 127.

X. Zhang, Y. Sugano, M. Fritz, and A. Bulling (2017d). It is Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation, in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2017*. Cited on pages 10, 16, 18, 69, 70, 87, 89, 105, 107, 109, and 110.

X. Zhang, Y. Sugano, M. Fritz, and A. Bulling (2018c). MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. Cited on pages 10, 69, 70, 71, 73, 76, 81, 82, and 128.

Y. Zhang, A. Bulling, and H. Gellersen (2013). SideWays: A Gaze Interface for Spontaneous Interaction with Situated Displays, in *Proc. ACM CHI Conf. Human Factors in Comput. Syst. 2013*. Cited on page 25.

Y. Zhang, M. K. Chong, J. Müller, A. Bulling, and H. Gellersen (2015b). Eye tracking for public displays in the wild, *Personal and Ubiquitous Computing*, vol. 19(5-6), pp. 967–981. Cited on page 87.

Y. Zhang, J. Müller, M. K. Chong, A. Bulling, and H. Gellersen (2014a). GazeHorizon: enabling passers-by to interact with public displays by gaze, in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing 2014*. Cited on page 1.

Y. Zhang, S. Stellmach, A. Sellen, and A. Blake (2015c). The costs and benefits of combining gaze and hand gestures for remote interaction, in *Human-Computer Interaction 2015*. Cited on page 130.

Z. Zhang, P. Luo, C. C. Loy, and X. Tang (2014b). Facial Landmark Detection by Deep Multi-task Learning, in *Proc. Eur. Conf. Comput. Vis. 2014*. Cited on pages 21 and 129.

Z. Zhu and Q. Ji (2005). Eye gaze tracking under natural head movements, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2005*. Cited on page 14.

Z. Zhu, Q. Ji, and K. P. Bennett (2006). Nonlinear eye gaze mapping function estimation via support vector regression, in *Proc. IEEE Int. Conf. Pattern Recognit. 2006*. Cited on page 14.