Training Person-Specific Gaze Estimators from User Interactions with Multiple Devices

Xucong Zhang[1] **Michael Xuelin Huang**[1] **Yusuke Sugano**[2] **Andreas Bulling**[1]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
[2]Graduate School of Information Science and Technology, Osaka University, Japan
{xczhang,mhuang,bulling}@mpi-inf.mpg.de sugano@ist.osaka-u.ac.jp

ABSTRACT

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.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

Appearance-Based Gaze Estimation; Multi-Devices

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 [1, 2, 3, 4, 5]. Taken together, these advances promise to finally enable attentive user interface [6], eye-based user modelling [7, 8], and gaze interaction [9, 10, 11] 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.

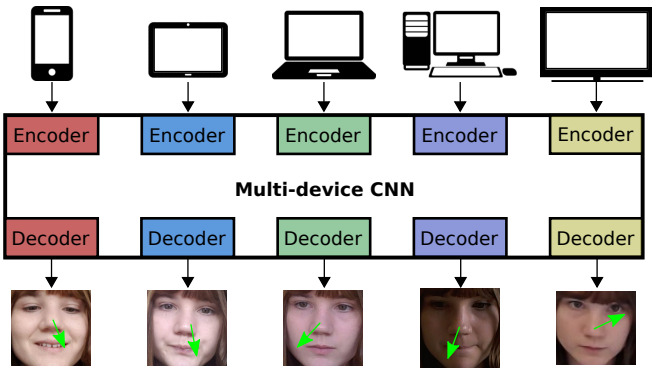

Figure 1. Our method for multi-device person-specific gaze estimation based on a convolutional neural network (CNN). It processes the person-specific images obtained from different devices with device-specific encoders and shared feature extraction layers, and gives out gaze estimates by different device-specific decoders.

In this work 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 *Multi-Device Person-Specific Gaze Estimation*. As illustrated in Figure 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 [12] or mouse clicks [13]. While

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ISBN 978-1-4503-5620-6/18/04…$15.00

DOI: https://doi.org/10.1145/3173574.3174198

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 [14]. 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 our work 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 paper. 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).

RELATED WORK

Our work is related to previous works on (1) learning-based gaze estimation and (2) multi-domain learning.

Learning-Based Gaze Estimation

Gaze estimation methods can be broadly differentiated into model-based and learning-based approaches. While model-based approaches typically fit a geometric eye model to an eye image to estimate gaze direction [15, 16, 17], learning-based methods directly map from the eye image pixels to a particular gaze direction [18, 19, 20]. Learning-based gaze estimation methods work with ordinary cameras under variable lighting conditions [3, 21]. Recently, a number of works have explored means to train one generic gaze estimator that can be directly applied to any device and user [1, 3, 4, 22]. To extend the coverage of the training data, latest efforts have focused on eye image synthesis [1, 22, 23] and eye image refinement [24] with some success. Zhang et al. recently demonstrated significant performance improvements of over 25% by using full-face images as input, instead of eye images [5]. 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 $7 \sim 10°$ [3, 24], and person-specific training data is necessary for good performance of about $3°$ [1].

User- or Device-Specific Adaptation

Traditional methods for learning-based gaze estimation assumed both user- and device-specific training data [18, 19, 20]. 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. Krafka et al. trained a gaze estimation CNN model on 1.5 million images collected from mobile phones and tablets, and achieved an accuracy of 2cm on the screen [4]. Huang et al. collected over 100,000 images during tablet use and applied a Random Forests regressor for on-screen gaze learning [25]. 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. proposed an alternative method that combined aggregation of gaze data from multiple users on a public display with an on-site training data collection [10]. In contrast, we focus on the *multi-device person-specific* training task that has not been explored in the gaze estimation literature so far.

Implicit Calibration for Gaze Estimation

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 [26, 27, 28] or predicting fixations on images using a regression CNN [29]. Others proposed to leverage the correlation between gaze and user interactions. Jeff et al. explored multiple cursor activities for gaze prediction [30]. Sugano el al. used mouse-clicks to incrementally update the gaze estimator [31]. Papoutsaki et al. developed a browser-based eye tracker that learned from mouse-clicks and mouse movements [32]. Huang et al. further investigated the temporal and spatial alignments between keypresses, mouse-clicks and gaze signals for user-specific gaze learning [14]. 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.

Multi-Domain Learning

Multi-task learning has been researched in the machine learning literature for decades, such as for natural language processing [33], speech recognition [34], facial landmark detection [35], or facial expression recognition [36]. Kaiser et al. used a single model for multiple unrelated tasks by incorporating an encoder and a decoder for each task [37]. 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 recently proposed a multi-domain CNN for visual tracking composed of shared layers and multiple branches of domain-specific layers [38]. 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. The novelty of our work is to investigate the practical feasibility of a multi-domain approach in the context of gaze estimation.

MULTI-DEVICE PERSON-SPECIFIC GAZE ESTIMATION

The core idea explored in this work 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

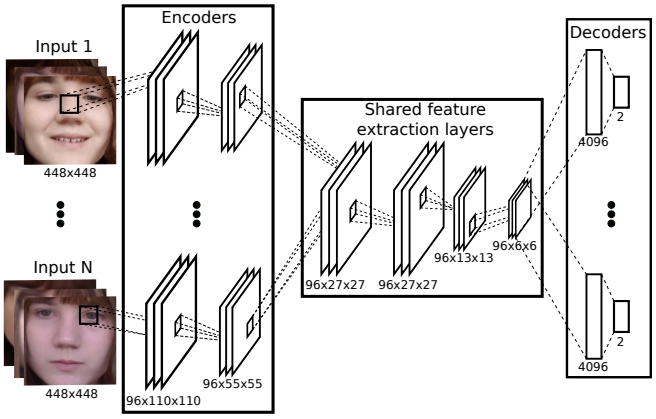

Figure 2. Architecture of the proposed multi-device gaze estimation method. Our method uses feature extraction layers shared across devices as well as device-specific encoders and decoders. It consists of two convolutional layers for each device-specific encoder, four convolutional layers for shared feature extraction layers, and two fully connected layers for each device-specific decoder. The numbers indicate the image resolution and the extracted feature dimension.

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.

Multi-Device CNN

Based on this idea, we propose a multi-device CNN as shown in Figure 2. Inspired by previous work [37], 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. [5].

We design our multi-device CNN based on the original AlexNet architecture [39], 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 features 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 [37], 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 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.

3D Gaze Estimation

The target of gaze estimation can either be an on-screen 2D gaze location [4, 25] or a 3D gaze direction in camera coordinates [1, 3, 5, 23]. In this work 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 [10]. 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 [40] and facial landmark detector [41] to process the input image, and then normalise the image data as suggested by Sugano et al. [1]. 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 [5], we set the size of the input face image to 448×448 pixels.

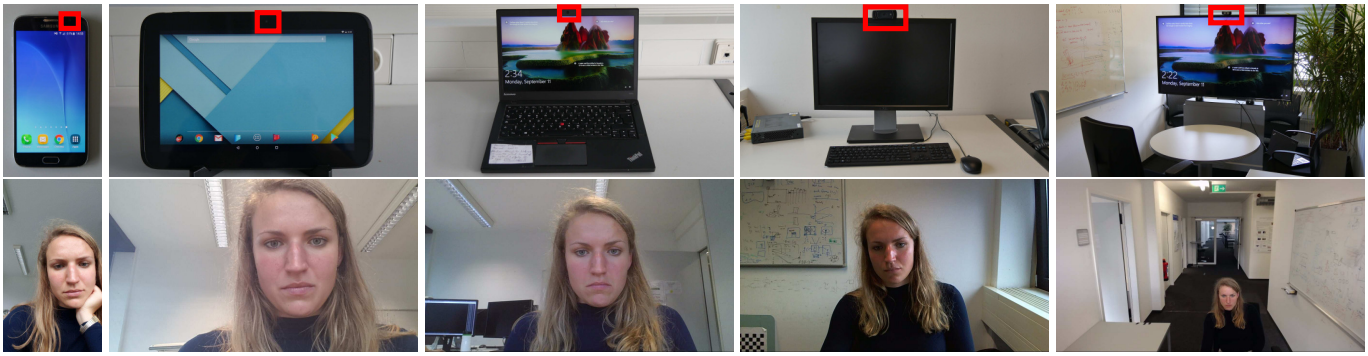

Figure 3. Top row: Personal devices used in the data collection. These devices have different sizes and camera-screen relationships (cameras are marked by red squares). Bottom row: Sample images from each device. As can be seen, the image resolution, noise level, illumination condition, face-to-screen distance, and head pose vary significantly across devices and environments, thus posing significant challenges for multi-device training.

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 [12] 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.

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 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 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 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 [42], 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 [43], we picked text editing as the recording activity for the laptop and desktop computer. Participants were asked to compose a document

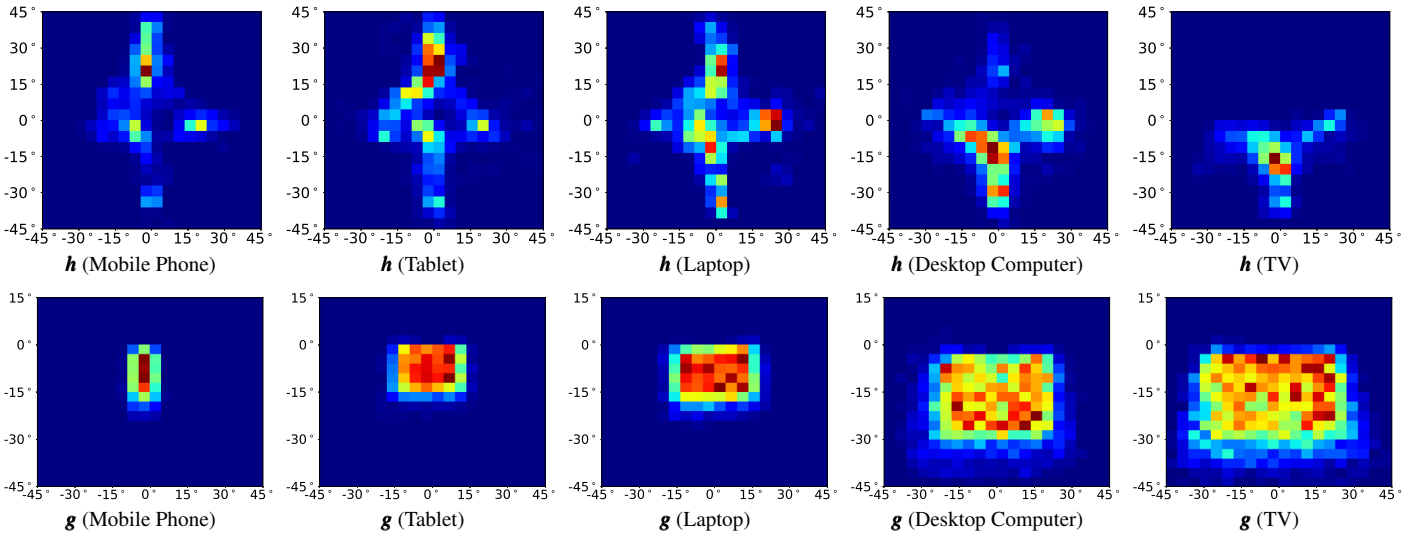

Figure 4. Distributions of head angle (h) and gaze angle (g) in degrees for mobile phone, tablet, laptop, desktop computer, and smart TV, created with explicit calibration data. The overall range of head poses and gaze directions differed across devices. Participants mostly looked down while using mobile 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.

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

Table 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.

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.

Dataset Characteristics

Figure 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 appearance information remains consistent to a large degree.

[1] https://kodi.tv/

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 [3]. Figure 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.

Frequency of Interaction

Table 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

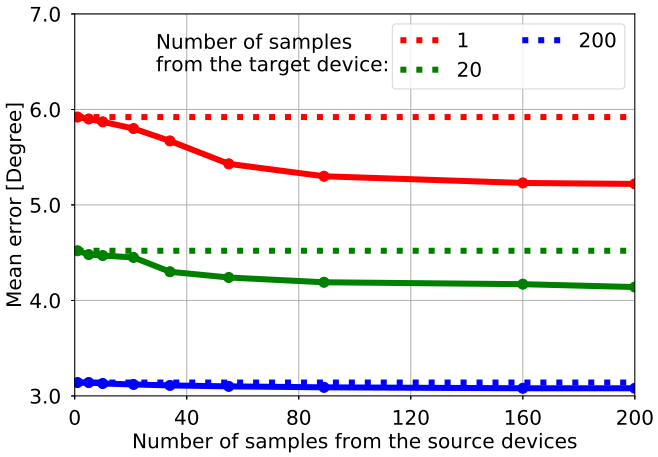

Figure 5. Gaze estimation error for the explicit calibration setting, comparing the proposed multi-device CNN (solid lines), a baseline single-device CNN trained on 1, 20 or 200 samples from the target device (dashed lines), as well as a single-device CNN trained on data from all source devices (dotted lines). The results were averaged over all five devices.

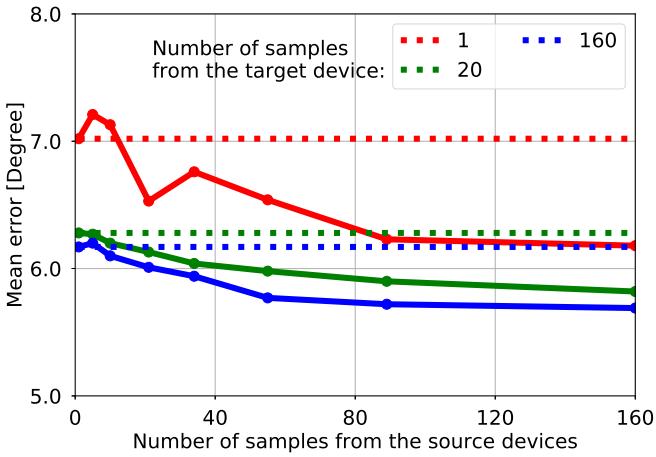

Figure 6. Gaze estimation error for the implicit calibration setting, comparing the proposed multi-device CNN (solid lines), a baseline single-device CNN trained on 1, 20 or 160 samples from the target device (dashed lines), as well as a single-device CNN trained on data from all source devices (dotted lines). The results were averaged over all five devices.

(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 [44]. 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.

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 [45] library to implement our model based on a modified AlexNet [39] pre-trained on ImageNet [46]. We fine-tuned the multi-device and single-device CNNs on MPIIGaze [3] and EYEDIAP [2]. We used the Adam solver [47] 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, we selected the first 200 for training and the remaining 100 samples for testing.

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.

Performance for Explicit Calibration

We first investigate the explicit calibration setting that yields high-quality training data and thus represents the ideal situation. Figure 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

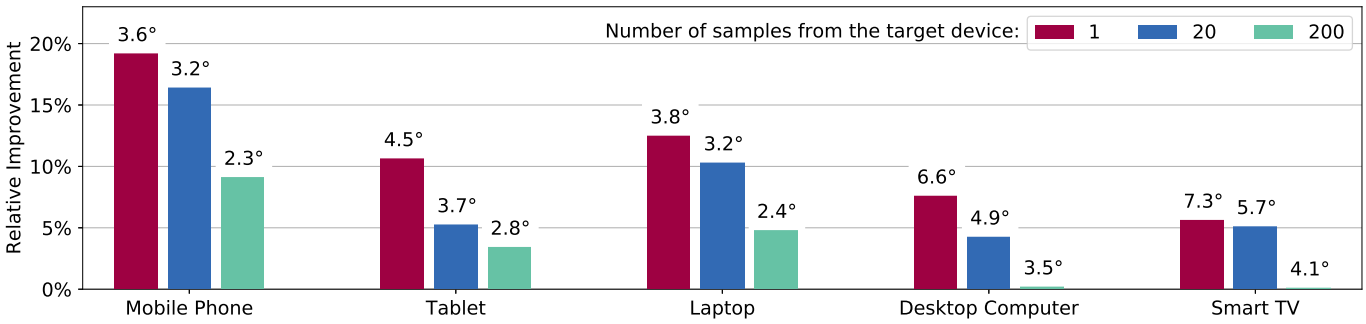

Figure 7. Relative improvement in gaze estimation error of our multi-device CNN over the single-device baseline in the explicit calibration setting when training on 1, 20 and 200 target samples. The numbers at the top of each bar are the mean error in degrees achieved by the multi-device CNN.

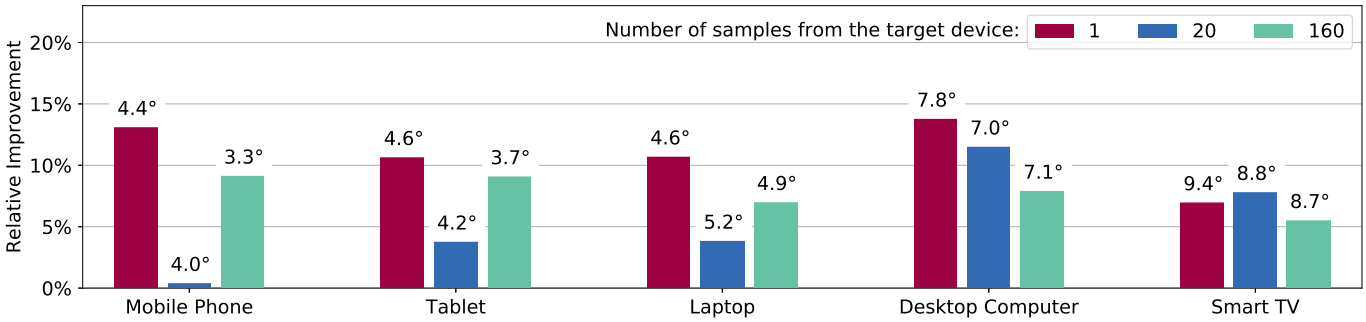

Figure 8. Relative improvement in gaze estimation error of our multi-device CNN over the single-device baseline in the implicit calibration setting when training on 1, 20 and 160 target samples. The numbers at the top of each bar are the mean error in degrees achieved by the multi-device CNN.

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 [3]. Our multi-device CNN significantly improves over this performance using device-specific encoders and decoders to better leverage cross-device data.

Performance for Implicit Calibration

Figure 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 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.

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 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 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 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 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 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, 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.

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.

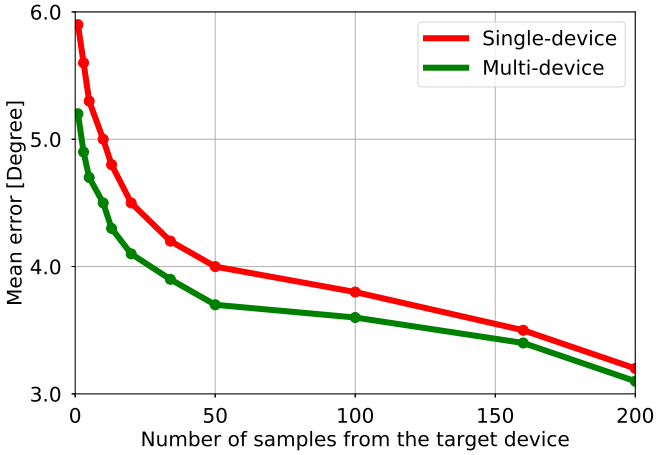

Figure 9. Mean error when adding a new device to the multi-device CNN compared to the single-device CNN in the explicit calibration setting. The single-device CNN was trained on increasing target samples from one to 200, while the multi-device CNN was trained additionally on 200 source samples from each 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.

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 was trained additionally on 200 samples from each source device. Figure 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 10. In this setting, the number of target samples depended on the actual interactions performed with each device during data collection (see Table 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 sufficient target samples, indicating that the multi-device CNN is more robust to such noisy data.

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 11 shows the relative improvement from this two-device CNN over the single-device CNN. We see that the relative improvement on

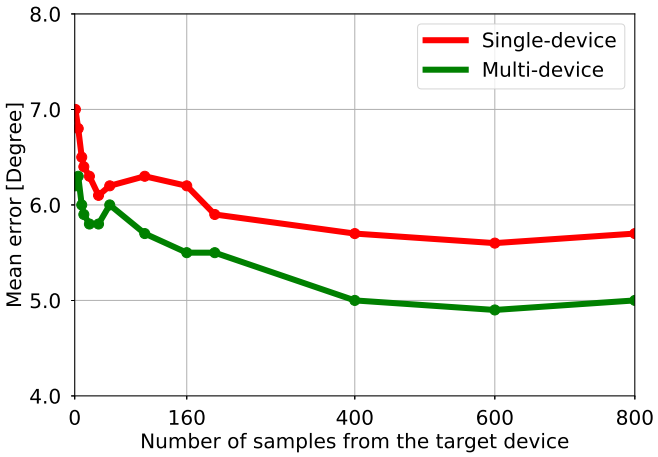

Figure 10. Evaluation of adding a new device to the multi-device CNN compared to the single-device CNN in the implicit calibration setting. The single-device CNN was trained on increasing target samples, which depended on the actual collected data on each device. The multi-device CNN was trained additionally 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.

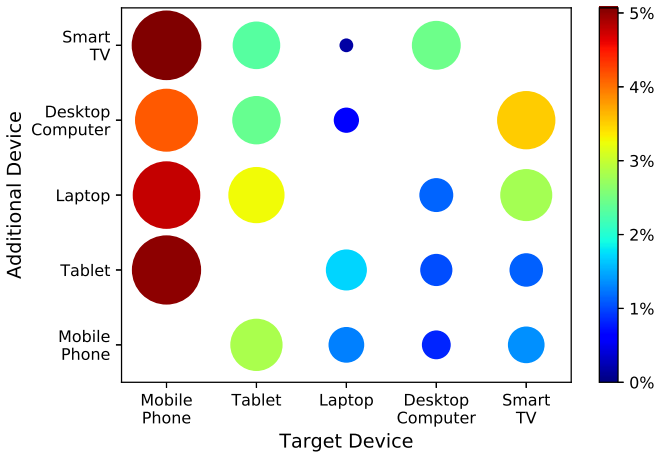

Figure 11. Relative performance improvements of a two-device CNN over the single-device CNN trained only on the target device samples in the explicit calibration setting. The x-axis shows the target device and the y-axis shows the source device. We used 20 target samples and 200 source samples from another device for multi-device CNN training. The bubble size and colour are proportional to the relative improvement.

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 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.

DISCUSSION

In this work 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 [6] or even cognitive states [8].

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 [14, 31] to further improve the performance and practicality of our approach. Besides, our experimental results (Figure 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.

CONCLUSION

In this work 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.

ACKNOWLEDGMENTS

This work was supported, in part, by the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University, Germany, as well as a JST CREST research grant (JPMJCR14E1), Japan.

REFERENCES

1. Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, 2014.
2. Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 255–258. ACM, 2014.
3. Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2015.
4. Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2176–2184, 2016.
5. Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 2299–2308. IEEE, 2017.
6. Andreas Bulling. Pervasive attentive user interfaces. *IEEE Computer*, 49(1):94–98, 2016.
7. Christin Seifert, Annett Mitschick, Jörg Schlötterer, and Raimund Dachselt. 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*, pages 301–304. ACM, 2017.
8. Michael Xuelin Huang, Jiajia Li, Grace Ngai, and Hong Va Leong. Stressclick: Sensing stress from gaze-click patterns. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1395–1404. ACM, 2016.
9. Per Ola Kristensson and Keith Vertanen. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 241–244. ACM, 2012.
10. Yusuke Sugano, Xucong Zhang, and Andreas Bulling. Aggregaze: Collective estimation of audience attention on public displays. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 821–831. ACM, 2016.
11. Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Everyday eye contact detection using unsupervised gaze target discovery. In *30th Annual Symposium on User Interface Software and Technology*. ACM, 2017.
12. Pierre Weill-Tessier and Hans Gellersen. Touch input and gaze correlation on tablets. In *International Conference on Intelligent Decision Technologies*, pages 287–296. Springer, 2017.
13. Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. An incremental learning method for unconstrained gaze estimation. *Computer Vision–ECCV 2008*, pages 656–667, 2008.
14. Michael Xuelin Huang, Tiffany CK Kwok, Grace Ngai, Stephen CF Chan, and Hong Va Leong. Building a personalized, auto-calibrating eye tracker from user interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5169–5179. ACM, 2016.
15. Jixu Chen and Qiang Ji. 3d gaze estimation with a single camera without ir illumination. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
16. Roberto Valenti, Nicu Sebe, and Theo Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 21(2):802–815, 2012.
17. Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 207–210. ACM, 2014.
18. Kar-Han Tan, David J Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 191–195. IEEE, 2002.

19. Oliver Williams, Andrew Blake, and Roberto Cipolla. Sparse and semi-supervised visual mapping with the sˆ 3gp. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 230–237. IEEE, 2006.

20. Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32(3):169–179, 2014.

21. Kenneth A Funes-Mora and Jean-Marc Odobez. Gaze estimation in the 3d space using rgb-d sensors. *International Journal of Computer Vision*, 118(2):194–216, 2016.

22. Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.

23. Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016.

24. Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. 2017.

25. Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: unconstrained appearance-based gaze estimation in mobile tablets. *arXiv preprint arXiv:1508.01244*, 2015.

26. Jixu Chen and Qiang Ji. Probabilistic gaze estimation without active personal calibration. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 609–616. IEEE, 2011.

27. Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Appearance-based gaze estimation using visual saliency. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):329–341, 2013.

28. Yusuke Sugano and Andreas Bulling. Self-calibrating head-mounted eye trackers using egocentric visual saliency. In *Proc. ACM Symposium on User Interface Software and Technology (UIST)*, pages 363–372, 2015.

29. Kang Wang, Shen Wang, and Qiang Ji. Deep eye fixation map learning for calibration-free eye gaze tracking. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 47–55. ACM, 2016.

30. Jeff Huang, Ryen White, and Georg Buscher. User see, user point: gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1341–1350. ACM, 2012.

31. Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. Appearance-based gaze estimation with online calibration from mouse operations. *IEEE Transactions on Human-Machine Systems*, 45(6):750–760, 2015.

32. Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. Webgazer: Scalable webcam ‘ tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*, 2016.

33. Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

34. Li Deng, Geoffrey Hinton, and Brian Kingsbury. 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*, pages 8599–8603. IEEE, 2013.

35. Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer, 2014.

36. Jixu Chen, Xiaoming Liu, Peter Tu, and Amy Aragones. Learning person-specific models for facial expression and action unit recognition. *Pattern Recognition Letters*, 34(15):1964–1970, 2013.

37. Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.

38. Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

39. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

40. Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

41. Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 354–361, 2013.

42. Soonhwa Seok and Boaventura DaCosta. Predicting video game behavior: An investigation of the relationship between personality and mobile game play. *Games and Culture*, 10(5):481–501, 2015.

43. Katy E Pearce and Ronald E Rice. Digital divides from access to activities: Comparing mobile and personal computer internet users. *Journal of Communication*, 63(4):721–744, 2013.

44. Grace P Szeto and Raymond Lee. An ergonomic evaluation comparing desktop, notebook, and subnotebook computers. *Archives of physical medicine and rehabilitation*, 83(4):527–532, 2002.

45. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

46. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

47. Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2015.