A Scenario-based Analysis of Front-facing Camera Eye Tracker for UX-UI Survey on Mobile Banking App

Conference Paper · January 2020

DOI: 10.1109/KST48564.2020.9059376

2 authors:

Wisuwat Sunhem
University of Glasgow
8 PUBLICATIONS **274** CITATIONS
SEE PROFILE

Kitsuchart Pasupa
King Mongkut's Institute of Technology Ladkrabang
146 PUBLICATIONS **1,253** CITATIONS
SEE PROFILE

A Scenario-based Analysis of Front-facing Camera Eye Tracker for UX-UI Survey on Mobile Banking App

Wisuwat Sunhem
Kasikorn Labs Co., Ltd.,
Kasikorn Business Technology Group,
Nonthaburi 11120, Thailand
Email: wisuwat.s@kbtg.tech

Kitsuchart Pasupa
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Bangkok 10520, Thailand
Email: kitsuchart@it.kmitl.ac.th

***Abstract*—Recently, User Experience and User Interface (UX-UI) have become important aspects in designing an effective mobile banking application. Traditionally, developers and designers have relied on explicit feedback derived from questionnaires to gain more insights into UX-UI. With the advancement of new technology, eye-tracking device has been introduced, and the approach has been used to provide a digital footprint indicating exact gazing positions of the users when using an application. So far, many studies have acknowledged the benefits of eye movement tracking and exploited such implicit feedback, alongside the result yielded from a survey. Successful uses of this eye-tracking device would further the development of mobile banking application. In this study, we aimed to build a device-free eye tracking software module that would work efficiently on mobile phones. To achieve this goal, we employed an existing Convolutional Neural Network model in our framework and evaluated the model when it was applied to the specific domain, i.e., UX-UI research design for mobile banking apps. We investigated a GazeCapture dataset, the first large-scale dataset for eye tracking, and conducted a data wrangling technique. The results show that fine-tuning the model with our wrangled data can improve the overall eye-tracking performance. Moreover, enabling user calibration can clearly enhance the predicting performance of the model.**

I. Introduction

With the advent of smartphones, our daily activities have been simplified, mainly because the capabilities of the hardware that can process data rapidly and accurately than conventional phones could in the past. Apart from the hardware being remarkably innovative, application software has been available for free download on all mobile platforms, i.e., Android and iOS. Worthwhile applications genuinely facilitate us to complete the productive daily tasks that we have to do. Daily activities include accessing business email, socialising, catching up on recent news, watching movies, and so on [1].

User-friendly design is one of the significant factors in building successful apps. A well-designed, competent appearance can attract the target group effectively. Before finalising the interface design, the designer usually conducts a survey for design evaluation. The survey data from participants can be qualitative or quantitative. This kind of explicit feedback is required to guarantee that the final product will reach the expected design quality [2].

Designing a mobile banking application is a challenging task for developers because of a number of particular development constraints; that is, to a great extent, a design is constrained by the standard regulations and highly restricted requirements of each particular bank. For example, security is considered the highest priority for mobile banking products [3]. On the other hand, impressive user experience also lies at heart of a design. However, it has been acknowledged that the profound authentication that serves the security aspect can be developed only at the expense of a friendly user-interface. Security restricts the flexibility of mobile banking application's design and, therefore, brings many constraints to the developers and designers.

To manage this demanding task and foster user experience (UX) and user interface (UI) on mobile banking application, apart from explicit feedback derived from questionnaires, eye-gazing information has been exploited to provide spontaneous footprints and implicit feedback [4]. According to Qu et al. (2017), users' personal perspectives revealed by their eye movement can be used alongside other evaluative information from surveys to give strong supportive evidence essential for the development of UX and UI [5].

Usually, to track eye movement requires a costly and inconvenient an additional plug-in physical device [6]; however, this study attempted to construct a device-free eye tracking software module for mobile phone. As we investigated potential studies in this task, Qiong Huang et al. [7] proposed a tree-based gaze estimator with fundamental hand-crafted features provided. Several variables were experimented to cope with possible varieties such as race, gender. However, the method was not able to reach the expected performance. Moreover, no pre-calibration procedure exists to enhance its performance beforehand. On the other hand, eye movement tracking based on the Convolutional Neural Network (CNN) architecture was proposed by Krafka and his colleagues [8]. In their work, the reliability of the module for a front-facing camera was tested in various environments and phone orientations, and that study revealed the universal capacities of this module. However, we hypothesised that, in this particular domain, i.e., assessing UX-UI quality of mobile banking, it is unnecessary to use all of

978-1-7281-4466-5/20/$31.00

the capabilities of the module.

To develop a more appropriate solution to mobile banking application, this study would conduct a number of controlled experiments to seek approaches that can effectively improve the prediction performance of the device-free eye tracking module on mobile phone. To do so, the existing pre-trained CNN model [8] would be drawn upon as a framework for this study. Although CNN is generally concerned for its computational extensive, and real-time functioning is almost impossible, in this case, a real-time condition is not a requirement in UX-UI research. We investigated GazeCapture dataset which is the first large-scale dataset for eye tracking [8]. All experiments were guided by two assumptions, vital to the construction of an eye tracking model, for this specialised scenario. Firstly, the majority of users access a mobile banking application through their smart phones. Secondly, it is more appropriate to display information in portrait mode than in landscape mode. This is because, unlike landscape mode which provides fascinatingly large view for video watchers and gamers, portrait mode offers feature-rich interface, namely a variety of components and controls in one page, and a lot of interactive touch points, e.g., for form completion, information search, etc [9]. Therefore, it is a preferable mode in mobile banking application. These assumptions were taken into consideration in the data wrangling procedure. Our wrangled data would be used further throughout the experiments in this work. Furthermore, Krafka et al.'s (2016) module utilised Apple's built-in libraries to detect both the user's face and eyes [8]. This is a limitation for employing the module on other platforms. Thus, we would adopt a well-known technique called Haar-cascade to perform the task. Our approach would not be restricted by any commercial libraries.

Following this introduction, the paper describes our methodology including domain adaptation in the deep learning model and the proposed data wrangling procedure in Section II. Section III described the evaluation of the model in a scenario where a new user starts to use the app. The scenario when CNN was tested on an unknown phone is explained in Section IV. Then Section V compares the performance between the model that has learned the previous user information and one that has not. Lastly, the conclusion is in Section VI.

II. Methodology

A. The Neural Network Architecture and Domain Adaptation

In this work, we investigated the iTracker that utilises CNN to track eye movements [8]. The model was trained with the original GazeCapture dataset that consists of 1,474 users and 2,445,504 video frames in total. Participants were asked to look at a set of generated points on a mobile screen in the data collection process. Each frame comprises only a point. Thus, it is labeled according to the location of the point shown to the participant. This point is assumed to be a 2D coordinate of eye-gazing. The gaze position is relative to the location of the camera at the origin $(0, 0)$.

We intend to replace the face and eye landmarks detection by Apple vision library by Haar-cascade detector to encourage the developing flexibility. As depicted in Figure 1, a video frame containing a participant's face is processed by the face-and-eye Haar-cascade classifiers. This detection algorithm results in four image components, namely face, left eye, right eye, and a binary matrix indicating face location in a video frame. Because of this, the CNN architecture composes of four input channels according to those components. All components, except the binary matrix, were fed into different blocks–sets of sequential convolutional layers–for feature extraction. The block of five convolutional layers for 'face' works independently while the blocks of 'left-eye' and 'right-eye' convolutional networks–contain four layers of convolution–use the same set of parameters for lower computational complexity. The outputs from the 'left-eye' and 'right-eye' networks were fed into a fully-connected layer. Spontaneously, the layers for 'face' and the face location matrix were assigned to separated fully-connected neural networks. After that, all the representations from the three independent networks were sent into the fully-connected neural network for predictive inference. The model predicts a 2D coordinate of eye-gazing for each frame in the model inference process.

As mentioned above, the architecture consists of connected differentiable computational units in an end-to-end fashion. This suits for future adaptation. Theoretically, although CNN is one of the most powerful machine learning techniques in this era, its ability to deal with a variety is limited under known-domain space. In other words, the model can yield outstanding performance as long as its input-output pattern is still in the training set distribution.

Many pieces of literature leverage a concept of 'learning to learn' in the area of machine learning studies–to utilise knowledge from previous tasks (source domain) for the benefit of solving a target problem. As investigated, they stated how this idea works in various definitions. A study of [10] summarises this concept based on distribution mismatch assumption. It divides the learning-to-learn concept into two groups that are domain adaptation and transfer learning. Domain adaptation is a problem that source domain data and target domain data are in different representative distributions while expecting a similar output pattern. At the same time, for transfer learning, the source and the target data are presumed to share some common distributions but having different patterns of predicted output.

Because the input processing technique and scenario usages are changed from the original task in our case, thus our work is classified as the domain adaptation problem. Here, we employ a neural network fine-tuning technique to tackle the domain changing problem.

B. Proposed Data Wrangling Procedure

As we described about the existing dataset. It is large and sufficient to train a model to predict where people look on the mobile screen. The model can handle various situations, such as head pose movement, different degree of illumination, wearing or not wearing glasses, using tablet, portrait or landscape modes. Moreover, many devices were used in data collection process including smartphones and tablets, see Fig. 2 for more details. Fig. 3a shows distributions of gazing points in this dataset. The gazing points are relative to the front-camera position as the origin $(0, 0)$ in centimetres. It is clearly seen that there are three regions, i.e., portrait mode with front-camera at the top, landscape mode with front-camera on the left and the right sides.

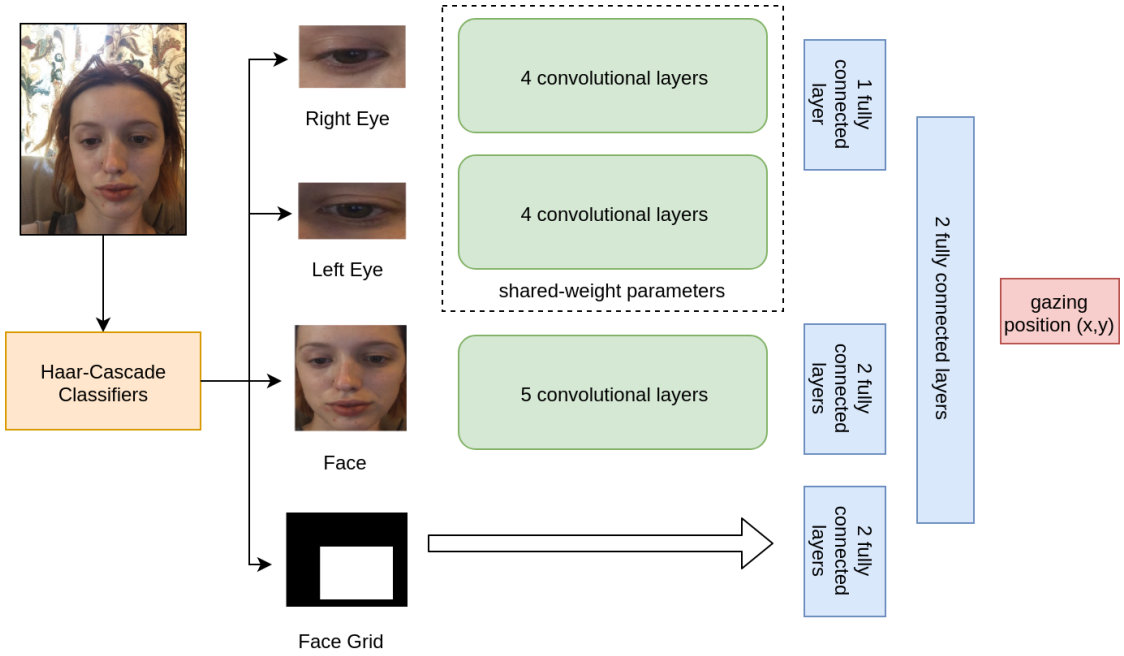


Fig. 1. The diagram illustrates how a video frame can be processed as an input to Haar-Casecade classifier and the convolutional Neural Network architecture to get the gazing position associated with the frame.

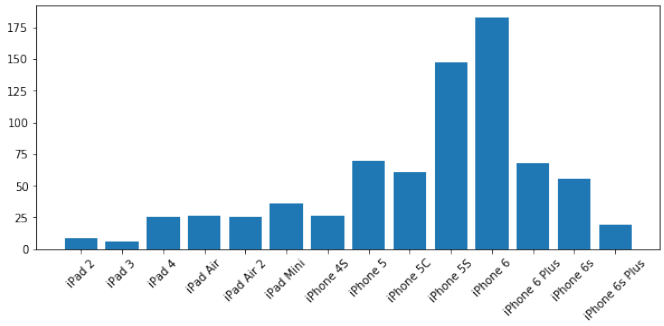


Fig. 2. A number of participants on each device used in data collection process.

We hypothesise that the more specific the model to the task is, the high accuracy the model is. When it comes to utilise the pre-trained model–is trained by the original dataset– to predict gaze points for our task, the prediction performance might not be as good as the model that is trained by the more specific dataset. Therefore, we propose a data variation reduction technique via a data wrangling procedure to enhance the performance of the model to use for our task. Unnecessary data points will be filtered out according to the following steps:

i) Mobile banking apps are commonly available in the smartphone. Thus, all videos recorded by tablets are removed. Then, only 'iPhone 5', 'iPhone 5C', 'iPhone 5S', 'iPhone 6', 'iPhone 6 Plus', 'iPhone 6s', 'iPhone 6s Plus' are considered in this task.
ii) All videos recorded in the landscape mode will be filtered out because the mobile banking app is always in the portrait mode.
iii) All participants with glasses will be removed.
iv) We randomly select one hundred points of each user. It is noted that all users who have less than 100 points will be neglected.

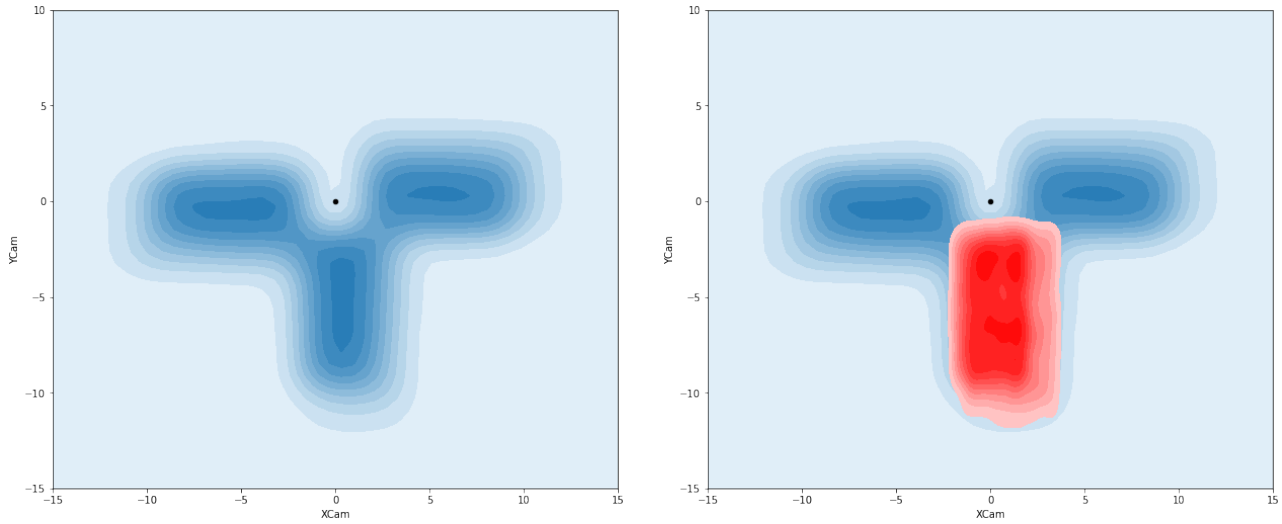
According to these steps, we finally obtain a compact dataset that includes 439 users with 16,243 frames. Its distribution is shown in Fig. 3b. It is clear that the new solution space is less variation than the previous one. As the model requires four components from a frame, we utilise eye and face Haar-cascade detectors to obtain them. We further discard all frames that Haar-cascade fails to detect the components. As a result, the data are skimmed down to 426 users with 11,195 frames.

III. An Evaluation of the Wrangled Dataset

We evaluate the model under a new user scenario. Assuming that a new user uses the model without calibration–no information of this user in the training set. This is done by randomly assign 20 % (86 users) of the data as a test set. The remaining 340 users were further divided into training set and validation set with 80:20 ratio. We compare three approaches as follows.

i) Baseline prediction with centroid: the predicted gaze point for all test samples is a centroid of the labels in training set.
ii) Pre-trained iTracker model: the pre-trained model is publicly available to download[1]. The model is evaluated on the test set of our compact dataset.
iii) Fine-tuned model: the pre-trained model is fine-tuned with the training set of our compact dataset prior to the evaluation on test set. In training phase, batch size is set to 20 and the maximum number of epochs is 10. All instances of test set are predicted by the best model at the optimal epoch. The optimal epoch is selected based on the minimum mean square error in validation set.

[1]https://github.com/CSAILVision/GazeCapture



(a) (b)

Fig. 3. The two-dimensions contour plot illustrates the gaze points distribution of samples in GazeCapture dataset. Black dot at the origin is a camera position. (a) shows the original gaze points distribution, and the red area in (b) shows gaze points distribution after data wrangling process.

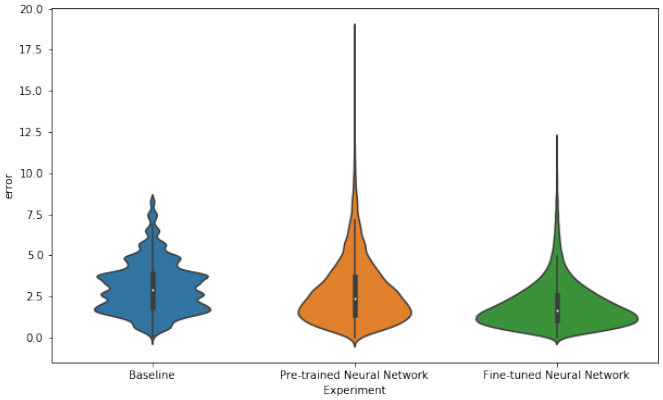


Fig. 4. The violin plots demonstrate gaze position prediction performance in term of error including the baseline, the pre-trained model, and the fine-tuned model.

The experiment was executed 10 times with different random split. The average error of the baseline prediction with centroid is 3.08 ± 1.52 centimetres and its median error is 2.90 centimetres. As expected, the pre-trained model gave 2.82 ± 1.96 centimetres on average error and 2.36 centimetres on median error. The best model is the fine-tuned model that demonstrates error with 2.00 ± 1.41 centimetres on average and 1.67 centimetres on median. It is clearly seen that distance between the mean and the median of the baseline method is relatively small compared to the others. That is to say, the error distributions of pre-trained and fine-tuned models are right-skewed. We further show the violin plots as shown in Fig. 4. Being right-skewed, in most situations, illustrates the high possibility of getting less prediction error than the error represented by mean. Therefore, both models can perform better than the baseline.

IV. When the Predicting Model is Employed in an Unseen Device

In the previous section, we assume that the model is used for predicting a new user but does not consider the device model factor. All the device models are found in both training and test sets. Here, we aim to evaluate the predicting model when it comes to the new arrival of a product model–especially with different screen size ratio. There is a high chance that we need to capture gaze points from the users on various model devices. This is because the survey might require the user to evaluate on their own device. Therefore, we need to evaluate how well the model can predict when there is an unseen device model.

The experimental framework is illustrated in Fig. 5. All device models in the dataset are divided into three groups based on the screen size [11]. Here, we perform leave-one-group-out cross-validation. The two device model groups for predicting model training purpose are split into training and validation sets for model optimization purpose (optimal epoch) in fine-tuning process. The remaining group is used for model testing in order to testify how robust the model will be on arrival of the new device.

The experimental results are reported in Table I. As previously mentioned, samples are assigned to each group based on the mobile-screen sizes–small (A), medium (B), and large (C). When the medium-screen-size mobiles (Group B) has been left out, the prediction performance of the model yield the best. This might be because of the prediction space in the trained model. When Group B has been left out, the model has been trained by small and large screen size samples. Hence, the prediction model could estimate prediction area between large and small device model efficiently. On the other hand, when groups A or C is left out, the model might misconceive some prediction regions generated by excessive large or small screen size devices. In overall picture, the average error across all three groups is 2.104 centimetres. It can be seen that the overall performance slightly drops–compare with a new user

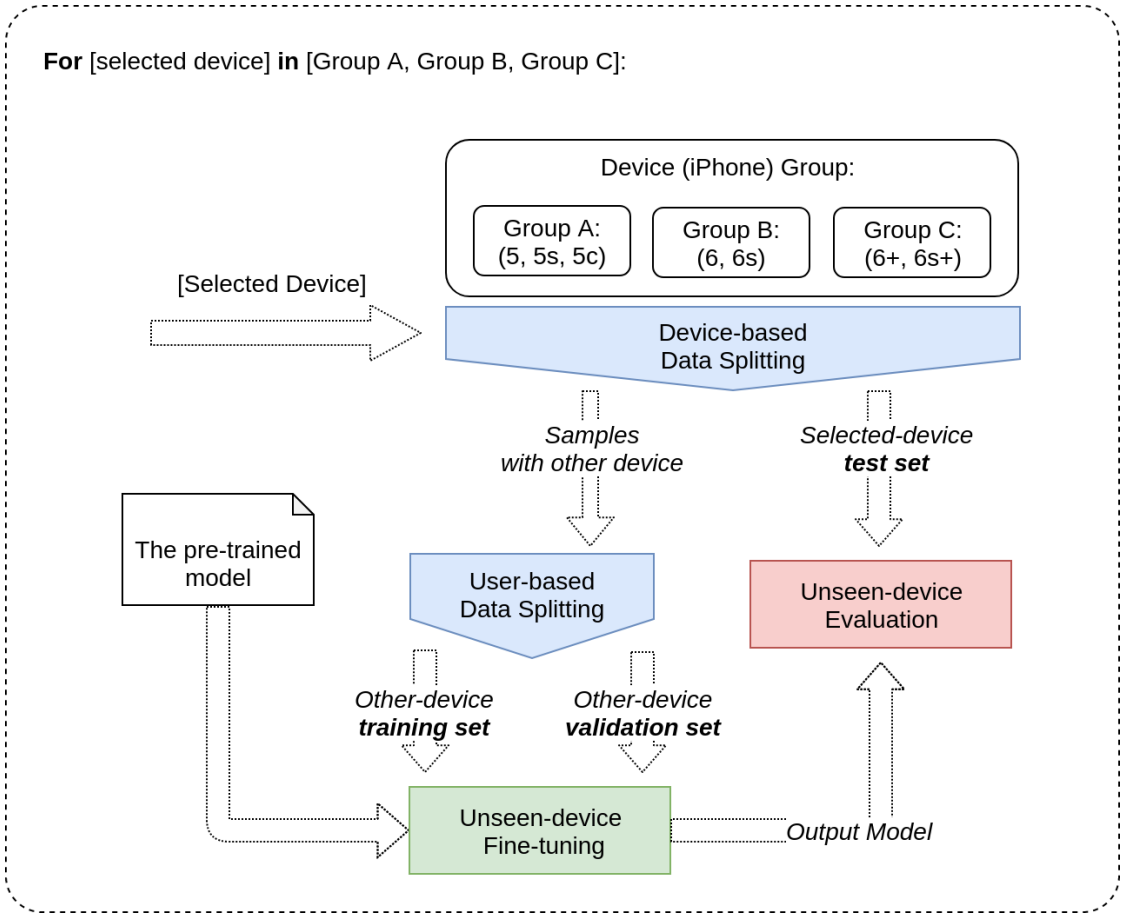


Fig. 5. The work-flow diagram shows the unseen device evaluation processes.

TABLE I. NEW-DEVICE EVALUATION RESULTS ON 10 RANDOM DATA SPLITS. EACH COLUMN IDENTIFIES EACH DEVICE GROUP HAVING BEEN VALIDATED. EACH ELEMENT REPRESENTS AN ERROR.

Seed	A	B	C
1	2.211	1.941	2.159
2	2.198	1.936	2.191
3	2.240	1.938	2.225
4	2.143	1.930	2.158
5	2.194	1.947	2.251
6	2.146	1.943	2.143
7	2.176	1.957	2.235
8	2.205	1.912	2.169
9	2.167	1.975	2.180
10	2.174	1.926	2.160
Mean	2.185	1.941	2.187

scenario in the previous section. Therefore, when there is an availability of new device model, it is not necessary to fine-tune the model–especially if the screen size of new device is under the covered range of screen size in training samples.

V. IS USER CALIBRATION NECESSARY?

We assume that the model has already learned some information from the user in this section. Thus, we will include sample images that are belong to the same user in the training set. This means that the model is calibrated by user information. Unlike the scenario explained in Section III, there is no user information included in the model.

We employ the same data split and work-flow similar to the scenario presented in Section III in order to directly compare the results. It is noted that, we included gaze information of the considered user in the training set in the user calibration case. The experimental results is shown in Fig. 6. Referring to the performance of the non-calibration framework reported in Section III, the error is 2.00 ± 1.41 centimetres on average with a median of 1.67 centimetres. It is clear that the model with a calibration can achieve a better performance because an average error is significantly small at 1.62 ± 1.13 centimetres and median error at 1.39. It can be implied that calibration process is necessary to achieve accurate eye-gazing prediction.

VI. CONCLUSION

In this study, we proposed an idea on how to adapt the iTracker–the front-facing camera eye tracker–for mobile banking UX-UI research design survey. Our investigation reveals some findings to support awareness of the model usability in each scenario. The experiments report that (i) the fine-tuning the model with our wrangled data can improve the overall performance; (ii) it is unnecessary to fine-tune the eye tracking model for the new device, especially, when the screen size of new device is under the covered range of screen size in training samples; and (iii) user calibration should be conducted because

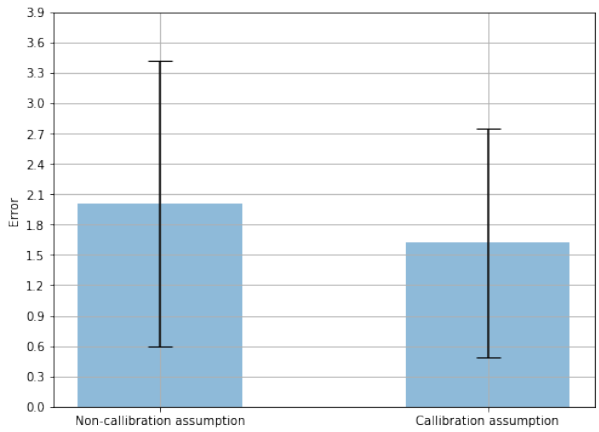


Fig. 6. A comparison between the model with and without user calibration. The chart shows average prediction errors with the error bars.

it can improve the predicting performance of the model. Thus, if we want precise prediction, user calibration is required.

ACKNOWLEDGMENT

This work was supported by Kasikorn Business-Technology Group under grant agreement number KLABS-G-2019-2.

REFERENCES

[1] ExactTarget. (2014) Daily activities on smartphones and tablets. (Accessed: June 01, 2019). [Online]. Available: https://www.marketingcharts.com/industries/retail-and-e-commerce-41027/attachment/exacttarget-daily-activities-smartphones-tablets-feb2014

[2] V. Roto, H. Rantavuo, and K. Väänänen-Vainio-Mattila, "Evaluating user experience of early product concepts," in *Proceedings of the 4th International Conference on Designing Pleasurable Products and Interfaces (DPPI), Compiegne, France*, 2009, pp. 199–208.

[3] T. Jorgensen. (2014) Mobile banking apps – the good, the bad, and the hybrid. (Accessed: June 01, 2019). [Online]. Available: https://banknxt.com/38531/mobile-banking-apps-good-bad-hybrid

[4] A. Bojko, "Eye tracking in user experience testing: How to make the most of it," in *Proceedings of the 14th Annual Conference of the Usability Professionals Association (UPA). Montre'al, Canada*, 2005, pp. 1–9.

[5] Q.-X. Qu, L. Zhang, W.-Y. Chao, and V. Duffy, "User experience design based on eye-tracking technology: a case study on smartphone apps," in *Advances in Applied Digital Human Modeling and Simulation*. Springer, 2017, pp. 303–315.

[6] V. Janthanasub and P. Meesad, "Evaluation of a low-cost eye tracking system for computer input," *King Mongkut's University of Technology North Bangkok International Journal of Applied Science and Technology*, vol. 8, no. 3, pp. 185–196, 2015.

[7] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets," *Machine Vision and Applications*, vol. 28, no. 5-6, pp. 445–461, 2017.

[8] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, United States*, 2016, pp. 2176–2184.

[9] A. Itzkovitch. (2012) Designing for device orientation: From portrait to landscape. (Accessed: June 01, 2019). [Online]. Available: https://www.smashingmagazine.com/2012/08/designing-device-orientation-portrait-landscape

[10] N. Patricia and B. Caputo, "Learning to learn, from transfer learning to domain adaptation: A unifying perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1442–1449.

[11] A. Fruhinsholz. (2019) Apple iPhone product line comparison. (Accessed: June 01, 2019). [Online]. Available: http://socialcompare.com/en/comparison/apple-iphone-product-line-comparison