Poster: Combining Software-Based Eye Tracking and a Wide-Angle Lens for Sneaking Detection

Dayananda Herurkar
University of Kaiserslautern &
German Research Center for
Artificial Intelligence (DFKI)
Kaiserslautern, Germany
Dayananda.Herurkar@dfki.de

Andreas Dengel
University of Kaiserslautern &
German Research Center for
Artificial Intelligence (DFKI)
Kaiserslautern, Germany
Andreas.Dengel@dfki.de

Shoya Ishimaru
University of Kaiserslautern &
German Research Center for
Artificial Intelligence (DFKI)
Kaiserslautern, Germany
Shoya.Ishimaru@dfki.de

UbiComp/ISWC'18 Adjunct, October 8–12, 2018, Singapore, Singapore
ACM 978-1-4503-5966-5/18/10.
https://doi.org/10.1145/3267305.3267675

Abstract
This paper proposes *Sneaking Detector*, a system which recognizes sneaking on a laptop screen by other people and alerts the owner through several interventions. We utilize a pre-trained deep learning network to estimate eye gaze of sneakers captured by a front-facing camera. Since most of the cameras equipped on laptop computers cannot cover a wide enough range, a commercial wide-angle lens attachment and an image processing are applied in our system. On the dataset involving nine participants following four experiments, it has been realized that our system can estimate the horizontal eye gaze and recognizes whether a sneaker is looking at a screen or not with 78% accuracy.

Author Keywords
Sneaking; eye tracking; deep learning; image processing

ACM Classification Keywords
H.5.0 [Information interfaces and presentation (e.g., HCI)]: General

Introduction
Sneaking – a process of watching something secretly – can happen in several situations in daily life including in working environments and public places. Another person behind an owner can see what has been typed or displayed on mobile devices or computers without the owner's knowledge.

Figure 1: Application flow chart which includes 3 main steps: image undistortion, gaze estimation, and sneaking detection

(a) Without a wide-angle lens

(b) With a wide-angle lens

(c) Undistorted

Figure 2: Examples of images taken by our proposed system

One of the most promising approaches to recognize sneaking is to utilize eye tracking. Eye tracking technologies are getting more and more pervasive. There are already lots of eye tracking devices in the market today which tracks human eye gaze on a screen. But these devices are designed for recognizing only one user's eye gaze. (i.e., eye tracking for multiple people is not supported, and the range for detecting eyes is limited.) Additionally, these devices cannot be attached to laptop computers all the time as it would not be comfortable to carry mobiles with attached devices.

Recent inventions and improvements in the field of computer vision help to avoid these drawbacks [2, 3, 4]. Also, these work demonstrate that gaze estimation can be possible with only software and without any additional hardware. Such results in the field of software-based eye tracking is an encouragement in our work on sneaking detection.

The remaining problem is the range of eye tracking. In this paper, we solve this problem by attaching a wide-range lens to a camera (see Figure 1). Contributions of this work are 2-fold. (1) We propose *Sneaking Detector*, a system which recognizes sneaking on a laptop and informs the incident to the user. (2) We demonstrate the potentials and limitations of gaze estimation on undistorted images.

Approach

Figure 1 shows an overview of the proposed system. The following section describes each procedure.

Image Undistortion

A video frame taken through a wide-angle lens can detect a person diagonally behind a user (see Figure 2). However, due to an inbuilt feature of a wide-angle lens, we are not able to use the distorted image for gaze estimation. We remove the distortion by inbuilt functions in OpenCV. As a first step, we calculate camera matrix K and distortion coefficient D for a given laptop camera by taking pictures of a checkerboard in different angles. This step is required once for one combination of a laptop computer and a wide-angle lens. For a given input video frame we calculate new camera matrix using K and D, then computed the undistortion and rectification maps for image transformation. Finally using remap function created a new frame without the fisheye effect, which applies generic geometric transformation for the frame. These steps have been repeated for every frame to remove a fisheye effect from the video.

Gaze estimation

After evaluating several open eye tracking approaches [3, 1], we selected work by Krafka et. al. [2] to be integrated into our system. A pre-trained network is available on their

Figure 3: An experimental set up (Exp. 4). The participant is sitting on the sneak's position and looking at targets on the paper.

project's web site[1]. As they detect a face in an image by Apple's face detection function[2], we follow their procedure. Additionally, we convert the pre-trained model into an appropriate format of CoreML[3] to estimate gaze in real time.

Sneaking detection
In preliminary trials, we found that estimated gaze coordinates have error towards the position of the camera even if we do not attach a wide-range lens (e.g., eye gaze on the left or right side of a camera are shifted to the center, and all eye gaze on the screen is shifted to the top). This might be because the original network was trained based on images taken by smartphones and mobile tablets. We apply Linear Support Vector Regression to scale the gaze estimation output from a sneaker. If the sneaker's gaze is on the screen, The system gives an alert to the user.

[1] http://gazecapture.csail.mit.edu/
[2] https://developer.apple.com/documentation/coreimage/cidetector
[3] https://developer.apple.com/documentation/coreml

Experiment	Task	Position	Lens
1	9-point	User	No
2	9-point	User	Yes
3	9-point	Sneaker	Yes
4	5-point	Sneaker	Yes

Table 1: Conditions of each experiment

Evaluation

To evaluate our proposed system, we asked nine college students to participate in the following four experiments.

Experimental Design
Figure 3 shows an overview of the setup. We prepared 13-inch MacBook Pro, a software to guide eye gaze, a fish-eye lens, and two chairs. One chair in front of the computer is for a position of a user, and another is for a position of a sneaker. As shown in Table 1, participants followed two tasks on two positions, with and without a fish-eye lens.

In the 9-point task, 3x3 targets were displayed with equal spaces on the screen (25%, 50%, and 75% of the width and height of the screen). Each target was highlighted with a sound every 1.5 second (*one* to *nine*, corresponding to the position), and participants followed the highlights. In the 5-point task, they looked at targets on A4 papers attached to the screen. They followed the order with voice announcements of the position mark. The order of the highlights was Top, Right, Bottom, Left, and Center. Each task includes three intervals. Therefore we collected 96 samples from one participant. Models of Linear Support Vector Regression were trained with a leave-one-participant-out approach. Data from eight participants are used as training samples to scale data of the remaining one participant.

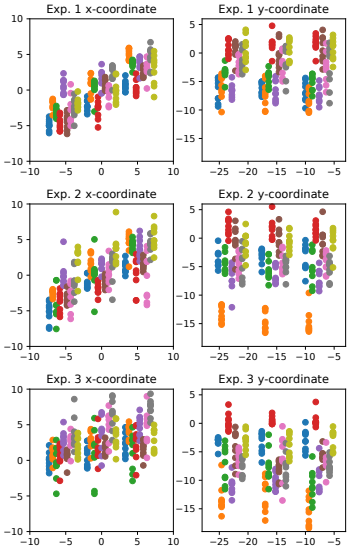

Figure 4: Gaze estimation results (x-axis: ground truth, y-axis: predictions). Data from each participant are displayed as different color, and linearly shifted for the visualization purpose.

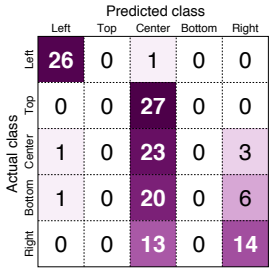

Figure 5: A confusion matrix of our sneaking detection involving nine participants (Exp. 4).

We conducted the first experiment to calculate the baseline of the gaze estimation. In the second experiment, we evaluate an effect of a wide-angle lens on the gaze estimation of the user. Even if there is not any distortion in the center of the image, estimating gaze on a small face should be a challenging task. The third and fourth experiment is for evaluating performances of gaze estimation of the sneaker.

Results and Discussion

Figure 4 represents the results of gaze estimations of the first, second, and third experiment. Mean values of absolute errors are 2.24cm (SD: 1.67), 2.56cm (SD: 2.04), and 4.09cm (SD: 2.69), respectively. Unfortunately, the system was not able to estimate y-coordinate of eye gaze regardless of the condition of a wide-angle lens. We tested two implementations (online estimation on CoreML and offline estimation on Caffe), but estimation results were similar.

The accuracy of the five-class classification in the fourth experiment is 47% (see Figure 5 for the confusion matrix). As we investigated in the first experiment, gaze estimation of y-coordinate seems not be reliable. If we simplify the problem to three-class classification (Left, Center or Right), the accuracy increases to 78%.

Conclusion and Future Work

We have proposed an approach of combining software-based gaze estimation and a wide-range lens in order to detect sneaking. Evaluations realized that gaze could be enough estimated on images taken through a wide-range lens and undistortion by image processing.

We classified a person sitting diagonally behind the user is looking at the left side, the right side, or the screen with 78% accuracy. However, even if we did not attach the lens, the system was not able to estimate the vertical eye gaze.

Future work includes investigation of other gaze estimation method to be integrated. The intervention (how to notify the sneaking to a user) is also in future work. Since displaying an alert window is disturbing for a user, other approaches.

Acknowledgements

This work was supported by JSPS KAKENHI (17K12728).

REFERENCES

1. Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*. IEEE, 59–66.
2. Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2176–2184.
3. Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. 2016. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*.
4. Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4511–4520.