Course Report for 'Human-Computer Interaction III"

Student Name: VAN SONHIENG

Student ID: 2020059118

Major: Human-Computer Interaction

International School, Jinan University

Email: songheingvan@gmail.com

THE FUTURE OF EYES INTERACTION ON SMART DEVICES

Abstract: At various degrees of complexity, different people process information. They transition between levels, concentrating at any given time on a certain level. The most typical tasks, like problem-solving, speaking, and designing, exhibit this pattern. This paper investigates the effects of design on creating interactive systems. It illustrates the concept by demonstrating the viability and desirableness of developing and testing a straightforward interaction of the smart devices and the eyes interaction system based on the concept of layers of Deep learning and CNN.

Keywords: *Design, Structure Design, Alternative Design, Computer Vision, Human-Computer Interaction, Deep learning, Artificial Intelligence.*

1. INTRODUCTION

Making the user's interaction with the system feel more natural is the goal of HCI design. It is an interdisciplinary area of study that focuses on computer technology design. The development of the contemporary user experience is a result of HCI research. Human objectives and requirements evolve quickly throughout time. Fifteen years ago, the main ways of interface with electronic devices were the mouse and the keyboard. Now, there are several methods for a system and a user to engage. One or more human senses, including as touch, sight, and sound, may be used to develop a simple user interface (UI) for an application.

2. DESCRIPTION

I. High Level design

Regarding the project's high-level design, there are two potential algorithms that may be used. One uses deepLapCut and GazeNet to participate in Deep Learning and Convolution Neural Network.

A system-level structural diagram

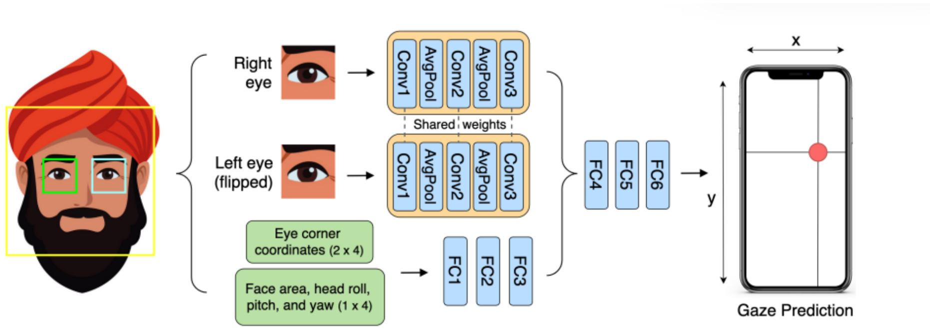

Fig.1 Architecture of the estimation CNN.

The core of our model is a Convolutional Neural Network (CNN) adapted from the state-of-the-art tracking model presented by Valliappan et al. We used TensorFlow to train this modified CNN model from and estimates the 2D estimate of the gaze in screen coordinate to train a device-calibrated support vector, we first create a gaze feature vector representing each frame of the iPhone's camera. This feature vector includes the output of the final three layers of our CNN, to which we append the following facial features: head yaw, pitch, and roll, the area of the face with respect to the frame, and the on-screen coordinates of the left and right eye corners.

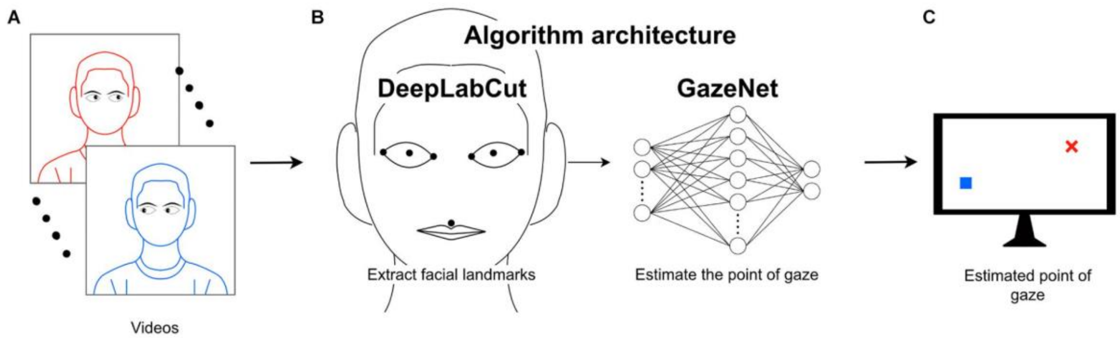

Fig.2 Architecture of the Deep learning with DeepLabcut and GazeNet

A. We detect the detect our eyes through the webcam of the computer or laptop.
B. Then we use the DeepLabCut [1] to detect on the human facial landmarks.
C. After that we estimate the point of gaze with GazeNet [2].
D. Finally, we automate the estimated point of gaze on the screen.

[1]DeepLabCut is a deep convolutional network that combines pretrained ResNets with deconvolutional layers, two essential components from techniques for object identification and semantic segmentation. The network is made up of a ResNets variation whose weights were developed using the popular, extensive ImageNet object recognition benchmark, on which it performs exceptionally well. We demonstrate the versatility of this framework by tracking various body parts in multiple species across a broad collection of behaviors. The package is open source, fast, robust, and can be used to compute 3D pose estimates.

[2]GazeNet is an innovative framework for developing event detectors that do not require custom signal characteristics or signal thresholding. It uses an end-to-end deep learning technique to classify raw eye-tracking data into fixations, saccades, and post-saccadic oscillations. As a result, our strategy calls into question an established implicit assumption that hand-crafted features are required in the design of event detection algorithms.

A system-level behavior diagram

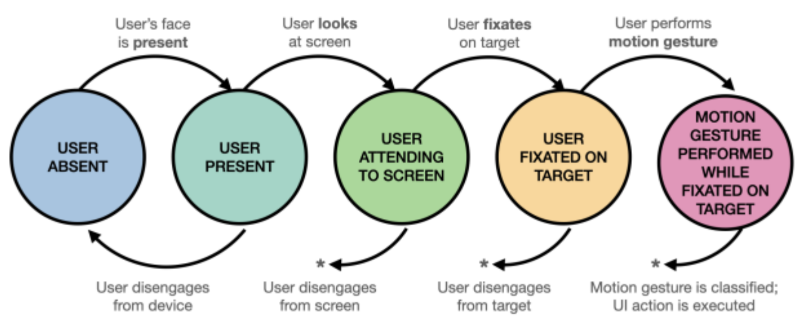

Fig 3: Behavior Diagram

The figure above represents the behavior diagram, which represents the entire interaction of the product's operation. We categorized it into five major stages, such as;

1. USER ABSENT

This stage, which often detects the user's presence, is in charge of the user's absence. It will advance to the next step if the user's face is visible. Otherwise, it will disengage from the devices or keep looping at the same stage.

2. USER PRESENT

This stage's duty is to check whenever the user looks at the screen. If the user is physically present in front of the screen but does not look at it. This stage is still not activated, so wait until the user looks at the screen. Otherwise, it will disengage from the screen.

3. USER ATTENDING TO SCREEN.

The program's task at this point is to identify what the user intends to do with their eyes, such as which app they wish to launch. Does the user seem to be fixated on the target? If not, the software must stop interacting with the screen.

4. USER FIXATED ON TARGET.

After that, the program can detect which app is the target of the user now. The program will wait for the user's command to perform a motion gesture. Which of these actions do these users want to take? Otherwise, the user will disengage from the target if the program cannot understand the user's command.

5. MOTION GESTURE PERFORMED WHILE FIXATED ON TARGET.

The last stage is in charge of the program's logic, therefore getting to it requires successfully completing the previous four stages without making any mistakes. This indicates that the motion gesture has been classified and the UI action is prepared to be carried out in accord with the command order.

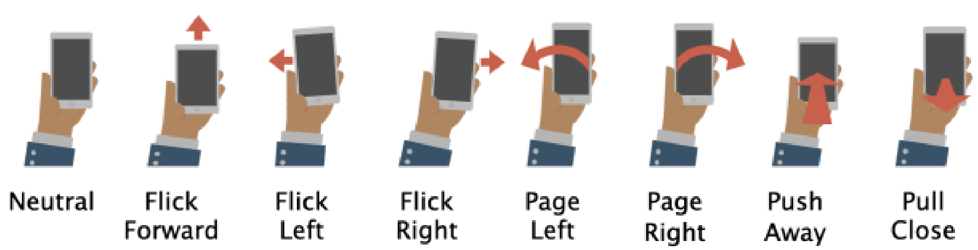

Fig 4. Motion Gesture

II. Static interface design

Home Screen: In this demonstration, a user moves their phone nearer (A) to access the calendar icon after glancing at it on their home screen (C). By pushing a button, the calendar is reduced (D). The user then concentrates on the weather symbol and pulls the phone nearer (E) in order to launch the app (F).

For better understanding Let's divide the picture into two parts: the above one which is the picture of users and the one below which is the interface of the iPhone. I name the first one "interaction interface" and the second "iPhone interface."

Picture A: When the user is staring at the phone, as shown in the interaction interface on the iPhone interface, we can see that the red circle is wrapped around the app that the user is staring at. At the moment, the user is staring at the calendar. On picture B, the user starts pulling the phone closer, and then suddenly the calendar starts opening. But when the user is pushing away, the phone suddenly closing the app which has shown in Picture D.

Once again, the user starts to repeat the process again, but for this time, the user is staring at another app, which we can see in the iPhone interface in Picture E. the user start pulling the phone closer again and then the app which we have been staring at will launch the app immediately.

III. Alternative design

For this project, there is not much user interface required because the important part of this project is the logical way in which this program is smart enough to interact with smart devices. Therefore, there is no such thing as an "alternative design." But the only alternative user interface that could possibly replace it is the color when it wraps around the detected app or the color theme for the whole interface. The color circled around the app that is being detected is red, as shown in the figure below. As a result, there are some options for the user to change the color to their preferred color for the alternative design concepts.

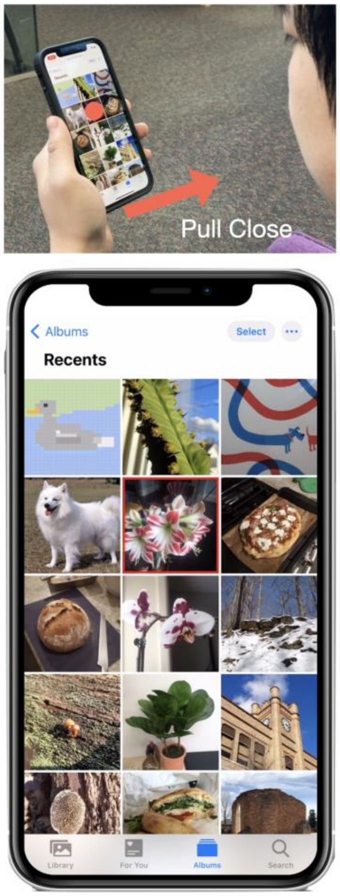

Fig.A

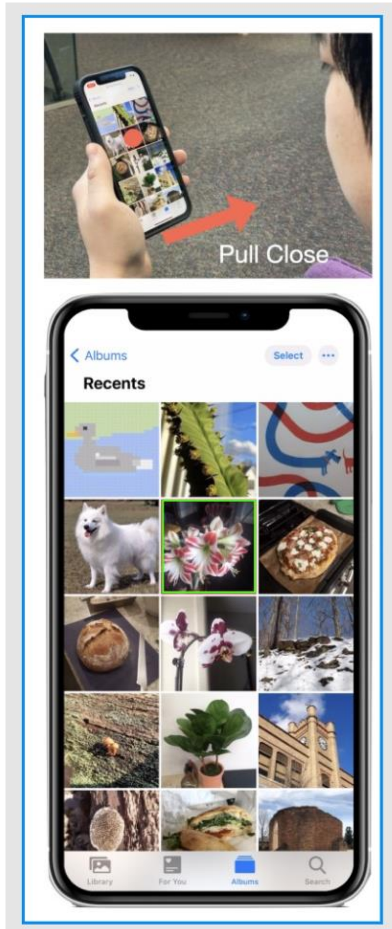

Fig.B

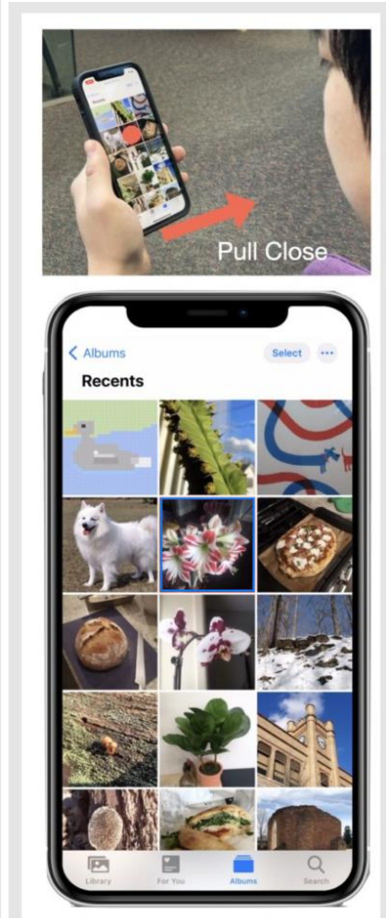

Fig.C

I have designed the alternative design for the color that is shown in the figure above. As shown in Figure A, which has the color red as the theme; Figure B, which has the color green as the theme; and Figure 3, which has the color blue as the theme. So, therefore, users can pick any color they want according to their convenience.

3. CONCLUSION

In conclusion, design has played a very important role in the field of human-computer interaction. without a well-designed HCI field. The user will be inconvenienced when using it.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my professor, Ph.D. LONG Jin Yi , who guided me throughout this research or the report paper and also thank for his awesome and perfect lecture performance. I would also like to thank my friends and family who supported me and offered deep insight into the study.

REFERENCES

[1]Computer vision - Wikipedia. Computer Vision - Wikipedia 2018. https://en.wikipedia.org/wiki/Computer_vision

[2]Convolutional neural network - Wikipedia. Convolutional Neural Network - Wikipedia 2019. https://en.wikipedia.org/wiki/Convolutional_neural_network

[3]yihuacheng. GitHub – yihuacheng Itracker: Gaze estimatin code. The Pytorch Implementation of “Eye Tracking for Everyone”. GitHub 2021. https://github.com/yihuacheng/Itracker.

[4]Eye Tracking for Everyone. Eye Tracking for Everyone n.d. https://gazecapture.csail.mit.edu/

[5]EyeMU Interactions: Gaze + IMU Gestures on Mobile Devices. YouTube 2021. https://www.youtube.com/watch?v=-HwcmWRAsaA

[6]Flores J. Training a TensorFlow model to recognize emotions. Medium 2018. https://medium.com/@jsflo.dev/training-a-tensorflow-model-to-recognize-emotions-a20c3bcd6468

[7]Disability Inclusion Overview. World Bank n.d. https://www.worldbank.org/en/topic/disability