

โครงการเลขที่ วศ.คพ. P804-2/2565

เรื่อง

ระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้เทคนิคการเรียนรู้ของเครื่องแบบต่อเนื่อง

โดย

นายนฤสรณ์ กันจินะ รหัส 620612153

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่  
ปีการศึกษา 2565

**PROJECT No. CPE P804-2/2565**

**Facial authentication system for building access using adaptive machine  
learning technique**

**Naruson Kanchina 620612153**

**A Project Submitted in Partial Fulfillment of Requirements  
for the Degree of Bachelor of Engineering  
Department of Computer Engineering  
Faculty of Engineering  
Chiang Mai University  
2022**

หัวข้อโครงการ : ระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้เทคนิคการเรียนรู้ของเครื่องแบบต่อเนื่อง  
โดย : นายนฤศรัณ กันจินะ รหัส 620612153  
ภาควิชา : วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษา : ผศ.ดร. ภานุสรา แซ่บประเสริฐ  
ปริญญา : วิศวกรรมศาสตรบัณฑิต  
สาขา : วิศวกรรมคอมพิวเตอร์  
ปีการศึกษา : 2565

---

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์  
(รศ.ดร. สันติ พิทักษ์กิจนุกร)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ  
(ผศ.ดร. ภานุสรา แซ่บประเสริฐ)

..... กรรมการ  
(ผศ.ดร. กำพล วรดิษฐ์)

..... กรรมการ  
(ผศ.ดร. ยุทธพงษ์ สมจิต)

|                  |   |
|------------------|---|
| หัวข้อโครงการ    | : ระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้เทคนิคการเรียนรู้ของเครื่องแบบต่อเนื่อง |
|                  | : Facial authentication system for building access using adaptive machine learning technique                |
| โดย              | : นายนฤสรณ์ กันจินะ รหัส 620612153  |
| ภาควิชา          | : วิศวกรรมคอมพิวเตอร์   |
| อาจารย์ที่ปรึกษา | : ผศ.ดร. ภานุสกร แซ่บประเสริฐ   |
| ปริญญา           | : วิศวกรรมศาสตรบัณฑิต   |
| สาขา             | : วิศวกรรมคอมพิวเตอร์   |
| ปีการศึกษา       | : 2565  |

---

## บทคัดย่อ

เจียนบทคัดย่อของโครงการที่นี่

การเขียนรายงานเป็นส่วนหนึ่งของการทำโครงการวิศวกรรมคอมพิวเตอร์ เพื่อทบทวนทฤษฎีที่เกี่ยวข้อง อธิบายขั้นตอนวิธีแก้ปัญหาเชิงวิศวกรรม และวิเคราะห์และสรุปผลการทดลองอุปกรณ์ และระบบต่าง ๆ อย่างไร ก็ได้ การสร้างรูปเล่ารายงานให้ถูกต้องและมีความสวยงาม แม้ว่าจะมีต้นแบบสำหรับใช้ในโปรแกรม Microsoft Word แล้วก็ตาม แต่นักศึกษาส่วนใหญ่ยังคงค้นพบว่าการใช้งานมีความซับซ้อน และเกิดความผิดพลาดในการจัดรูปแบบ กำหนดเลขหัวข้อ และสร้างสารบัญอยู่ ภาควิชาวิศวกรรมคอมพิวเตอร์จึงได้จัดทำต้นแบบรูปเล่ารายงานโดยใช้ระบบจัดเตรียมเอกสาร  $\text{\LaTeX}$  เพื่อช่วยให้นักศึกษาเขียนรายงานได้อย่างสะดวกและรวดเร็วมากยิ่งขึ้น

Project Title : Facial authentication system for building access using adaptive machine learning technique

Name : Naruson Kanchina 620612153

Department : Computer Engineering

Project Advisor : Asst. Prof. Paskorn Champrasert, Ph.D.

Degree : Bachelor of Engineering

Program : Computer Engineering

Academic Year : 2022

---

## **ABSTRACT**

The abstract would be placed here. It usually does not exceed 350 words long (not counting the heading), and must not take up more than one (1) page (even if fewer than 350 words long).

Make sure your abstract sits inside the `abstract` environment.

## กิตติกรรมประกาศ

โครงการนี้จะไม่สำเร็จลุล่วงลงได้ ถ้าไม่ได้รับความกรุณาจาก ผศ.ดร.ภาสกร แข่นประเสริฐ ออาจารย์ที่ปรึกษา ที่ได้สละเวลาให้ความช่วยเหลือทั้งให้คำแนะนำ ให้ความรู้และแนวคิดต่าง ๆ รวมถึง ผศ.ดร.ยุรพงษ์ สมจิต และ ผศ.ดร.กำพล วรดิษฐ์ ที่ให้คำปรึกษาจนทำให้โครงการเล่มนี้สำเร็จ สมบูรณ์ไปได้

ขอบคุณห้องวิจัย OASYS ภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่เอื้อเฟื้อสถานที่ในการทำโครงการ สนับสนุนอุปกรณ์ต่าง ๆ และ ขอบคุณ นาย กมลพัฒน์ สุนทรพงษ์ ที่ คอยให้ความช่วยเหลือในการทำโครงการมาโดยตลอด

ขอบคุณทาง ITSC ที่ได้ให้ใช้เซิร์ฟเวอร์ และขอบคุณเพื่อน ๆ ที่ให้กำลังใจรวมถึงคำแนะนำที่ดีตลอด การทำโครงการที่ผ่านมา รวมทั้งขอบคุณอีกหลาย ๆ ท่านที่ไม่ได้เอียนามมา ณ ที่นี่ ที่ได้ให้ความช่วยเหลือตลอดมา หากหนังสือโครงการเล่มนี้มีข้อผิดพลาดประการใด กระผมขออภัยรับด้วยความยินดี

นายณัฐรัตน์ กันจินะ

12 ธันวาคม 2565

# สารบัญ

|  |           |
|--|-----------|
| บทคัดย่อ . . . . .   | ๑         |
| Abstract . . . . .   | ๒         |
| กิตติกรรมประกาศ . . . . .  | ๓         |
| สารบัญ . . . . .   | ๔         |
| สารบัญรูป . . . . .  | ๕         |
| สารบัญตาราง . . . . .  | ๖         |
| <b>1 บทนำ</b>  | <b>1</b>  |
| 1.1 ที่มาของโครงการ . . . . .  | 1         |
| 1.2 วัตถุประสงค์ของโครงการ . . . . .   | 1         |
| 1.3 ขอบเขตของโครงการ . . . . .   | 1         |
| 1.3.1 ขอบเขตด้านhardtแวร์ . . . . .  | 1         |
| 1.3.2 ขอบเขตด้านซอฟต์แวร์ . . . . .  | 2         |
| 1.4 ประโยชน์ที่ได้รับ . . . . .  | 2         |
| 1.5 เทคโนโลยีและเครื่องมือที่ใช้ . . . . .   | 2         |
| 1.5.1 เทคโนโลยีด้านhardtแวร์ . . . . .   | 2         |
| 1.5.2 เทคโนโลยีด้านซอฟต์แวร์ . . . . .   | 2         |
| 1.6 แผนการดำเนินงาน . . . . .  | 3         |
| 1.7 บทบาทและความรับผิดชอบ . . . . .  | 5         |
| 1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม . . . . .                 | 5         |
| <b>2 ทฤษฎีที่เกี่ยวข้อง</b>  | <b>6</b>  |
| 2.1 MediaPipe Holistic . . . . .   | 6         |
| 2.2 RESTful API . . . . .  | 6         |
| 2.3 Image Processing . . . . .   | 7         |
| 2.3.1 การปรับปรุงคุณภาพของภาพ (Image Enhancement and Restoration) . . . . .          | 7         |
| 2.3.2 การบีบอัดข้อมูลภาพ (Image compression) . . . . .                               | 7         |
| 2.4 การส่งข้อมูลโดยใช้โปรโตคอลอเขซทีพี (HyperText Transfer Protocol: HTTP) . . . . . | 8         |
| 2.5 Raspberry Pi . . . . .   | 8         |
| 2.6 Opensource Computer Vision (OpenCV) . . . . .                                    | 9         |
| 2.7 Deep Learning . . . . .  | 9         |
| 2.8 ส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface: GUI) . . . . .           | 10        |
| 2.9 Tkinter . . . . .  | 10        |
| 2.10 OpenFace (Open-source Face Recognition) . . . . .                               | 11        |
| 2.11 ความรู้ตามหลักสูตรชีวถุกนำมาใช้หรือบูรณาการในโครงการ . . . . .                  | 12        |
| 2.11.1 Logic and Digital Circuits และ Microprocessor and Interfacing . . . . .       | 12        |
| 2.11.2 Digital Image Processing . . . . .  | 12        |
| 2.11.3 Deep Learning . . . . .   | 12        |
| 2.11.4 CPE Lab . . . . .   | 12        |
| 2.12 ความรู้นอกหลักสูตรชีวถุกนำมาใช้หรือบูรณาการในโครงการ . . . . .                  | 12        |
| <b>3 โครงสร้างและขั้นตอนการทำงาน</b>   | <b>13</b> |
| 3.1 ภาพรวมโครงสร้างและการทำงานของระบบ . . . . .                                      | 13        |
| 3.1.1 โมดูลกล้อง (Camera Module) . . . . .   | 13        |
| 3.1.2 การส่งข้อมูลไปยังเซิร์ฟเวอร์ . . . . .   | 14        |

|  |           |
|--|-----------|
| 3.1.3 การแสดงผลการระบุตัวตน . . . . .                                  | 14        |
| 3.1.4 เชิร์ฟเวอร์ (Server) . . . . .                                   | 15        |
| 3.1.5 การระบุตัวตน . . . . .   | 15        |
| 3.1.6 การจัดเก็บรูปภาพใบหน้า . . . . .                                 | 15        |
| 3.1.7 การเรียนรู้รูปภาพ . . . . .                                      | 16        |
| <b>4 การทดลองและผลลัพธ์</b>  | <b>17</b> |
| 4.1 ความแม่นยำของการระบุตัวตนด้วยใบหน้าในแต่ละวัน . . . . .            | 17        |
| 4.2 เวลาในการประมวลผลรูปภาพใบหน้า . . . . .                            | 17        |
| 4.3 ความพึงพอใจของการทดลอง . . . . .                                   | 18        |
| <b>5 บทสรุปและข้อเสนอแนะ</b>   | <b>19</b> |
| 5.1 สรุปผล . . . . .   | 19        |
| 5.2 ปัญหาที่พบและแนวทางการแก้ไข . . . . .                              | 19        |
| 5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ . . . . .                           | 19        |
| <b>บรรณานุกรม</b>  | <b>20</b> |
| <b>ก อุปกรณ์ต้นแบบและการแสดงผลบนหน้าจอ</b>                             | <b>22</b> |
| ก.1 รูปอุปกรณ์ที่ใช้ในการตรวจจับภาพใบหน้าและรับคำสั่งยืนยัน . . . . .  | 22        |
| ก.2 รูปภาพการแสดงผลบนหน้าจอ . . . . .                                  | 24        |
| <b>ข คู่มือการใช้งานระบบ</b>   | <b>25</b> |
| ข.1 คู่มือการติดตั้งโปรแกรมเพื่อตรวจจับใบหน้าบน Raspberry Pi . . . . . | 25        |
| ข.1.1 คู่มือการติดตั้ง OpenCV บน Raspberry Pi . . . . .                | 25        |
| ข.1.2 คู่มือการติดตั้ง TensorFlow lite บน Raspberry Pi . . . . .       | 25        |
| ข.1.3 คู่มือการติดตั้ง Mediapipe library บน Raspberry Pi . . . . .     | 25        |
| ข.1.4 คู่มือการติดตั้ง Tkinter library บน Raspberry Pi . . . . .       | 26        |
| ข.2 คู่มือการติดตั้งโปรแกรมเพื่อระบุตัวตนใบหน้าบน Server . . . . .     | 26        |
| ข.2.1 คู่มือการติดตั้ง Python บน Windows . . . . .                     | 26        |
| ข.2.2 คู่มือการติดตั้ง Flask framework . . . . .                       | 26        |
| ข.2.3 คู่มือการติดตั้ง OpenCV บน Windows . . . . .                     | 26        |
| ข.2.4 คู่มือการติดตั้ง Scikit learn บน Windows . . . . .               | 26        |
| ข.3 คู่มือการใช้งานการตรวจจับใบหน้า . . . . .                          | 27        |
| ข.4 คู่มือการใช้งานการระบุตัวตนด้วยใบหน้า . . . . .                    | 27        |
| <b>ประวัติผู้เขียน</b>   | <b>29</b> |

## สารบัญรูป

|   |    |
|---|----|
| 2.1 Poem . . . . .  | 7  |
| 2.2 Poem . . . . .  | 8  |
| 2.3 Poem . . . . .  | 8  |
| 2.4 Poem . . . . .  | 9  |
| 2.5 Poem . . . . .  | 10 |
| 2.6 Poem . . . . .  | 12 |
| 3.1 Poem . . . . .  | 13 |
| 3.2 Poem . . . . .  | 13 |
| 3.3 Poem . . . . .  | 15 |
| 3.4 Poem . . . . .  | 16 |
| 4.1 Poem . . . . .  | 17 |
| 4.2 Poem . . . . .  | 18 |
| 4.3 bar graph . . . . .   | 18 |
| ก.1 Module . . . . .  | 22 |
| ก.2 camera . . . . .  | 22 |
| ก.3 button . . . . .  | 23 |
| ก.4 inside module . . . . .   | 23 |
| ก.5 Raspberry Pi . . . . .  | 23 |
| ก.6 main page . . . . .   | 24 |
| ก.7 wrong predict . . . . .   | 24 |
| ข.1 แสดงหน้า GUI เมื่อโปรแกรมเริ่มทำงาน . . . . .                         | 27 |
| ข.2 แสดงหน้า Terminal เมื่อโปรแกรมระบุตัวตนด้วยใบหน้าเริ่มทำงาน . . . . . | 28 |

# ສາරບັດທາງ

## บทที่ 1

### บทนำ

#### 1.1 ที่มาของโครงการ

การยืนยันตัวตนในการเข้าสถานที่หลายรูปแบบ เช่น การแสดงลายนิ้วมือ การใช้บัตรประจำตัว การระบุเอกสารลักษณ์ด้วยคลื่นวิทยุ (Radio Frequency Identification: RFID) และอื่น ๆ อีกมาก many ซึ่งในปัจจุบันมีสถานการณ์โควิด-19 แพร่ระบาด ทำให้ผู้คนไม่สามารถพบปะกันได้ระหว่างพนักงานต้อนรับกับผู้ที่เข้าสถานที่ การยืนยันตัวตนในการเข้าสถานที่โดยใช้รูปถ่ายใบหน้าจะช่วยลดการแพร่ระบาดของเชื้อโรค และมีความสะดวกในการใช้งานไม่ต้องกับการยืนยันตัวตนแบบอื่น เมื่อใบหน้าของบุคคลมีการเปลี่ยนแปลงตลอดในทุกวัน เช่น มีหนวด ไม่มีหนวด ผอมสัน ผอมยาว ใส่แว่น ไม่ใส่แว่น เป็นต้น มีผลทำให้การระบุตัวตนด้วยการใช้ภาพใบหน้านั้นเกิดความผิดพลาด เป็นที่มาของการนำรูปภาพใบหน้าที่ได้รับเข้ามาใหม่ไปทำการเรียนรู้ใบหน้าให้ระบบสามารถจำภาพใบหน้าใหม่ที่มีการเปลี่ยนแปลง และความแม่นยำในการระบุตัวตนจะสูงขึ้นเมื่อรูปภาพใบหน้าที่เก็บไว้จำนวนมาก และหลายหลายรูปแบบ ทำให้สามารถแก้ปัญหาการเปลี่ยนแปลงใบหน้าได้

#### 1.2 วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาระบบตรวจใบหน้าใหม่ความแม่นยำที่สูง
- เพื่อให้ระบบมีความเหมาะสมสมของอุปกรณ์ที่ติดตั้ง
- เพื่อให้ระบบสามารถนำไปใช้งานได้จริง

#### 1.3 ขอบเขตของโครงการ

โดยระบบตรวจใบหน้าจะทำการติดตั้งหน้าทางเข้าห้องกลุ่มวิจัยทฤษฎีและการประยุกต์ใช้การหาค่าที่เหมาะสมที่สุดในระบบทางวิศวกรรม (OASYS Research Group Optimization Theory and Applications for Engineering SYStems Research Group: OASYS) และปรับให้มีความแม่นยำมากที่สุดให้ยังคงความพึงพอใจของผู้ใช้ห้องได้

##### 1.3.1 ขอบเขตด้านฮาร์ดแวร์

- ระบบจะสามารถค้นหาใบหน้าได้จะต้องมีพื้นที่ที่มีแสงสว่างเพียงพอ
- พื้นที่ที่ทำการติดตั้งต้องมีสัญญาณอินเทอร์เน็ตทั้งไร้สายหรือผ่านสายแลน
- พื้นที่ที่ทำการติดตั้งต้องไม่มีผู้คนพลุกพล่าน
- โปรแกรมการเรียนรู้ของเครื่องที่ไม่เกินกำลังด้านฮาร์ดแวร์ของเซิร์ฟเวอร์ที่ใช้เกรนไมเดล

### **1.3.2 ขอบเขตด้านซอฟต์แวร์**

1. สามารถจัดเก็บข้อมูลและรูปภาพใบหน้าได้
2. สามารถที่จะเรียนรู้รูปภาพใหม่ ที่เข้ามาจัดเก็บได้
3. ระบบใช้เวลาในการตรวจจับใบหน้า ส่งภาพไปยังเซิร์ฟเวอร์ ระบุตัวตน และส่งผลลัพธ์กลับมาแสดงจะให้เวลาไม่เกิน 40 วินาที

### **1.4 ประโยชน์ที่ได้รับ**

1. ผู้ที่เข้าสู่สถานที่ลดความเสี่ยงที่จะได้รับเชื้อโรค
2. ระบบสามารถที่จะระบุตัวตนในเวลาที่น้อย เพิ่มความสะดวกในการเข้าสู่สถานที่ได้
3. ระบบสามารถส่งต่อสัญญาณหรือข้อมูลไปยังส่วนอื่น ๆ ได้ เช่น บอกทางไปห้องทำงาน

### **1.5 เทคโนโลยีและเครื่องมือที่ใช้**

#### **1.5.1 เทคโนโลยีด้านฮาร์ดแวร์**

1. Raspberry Pi 4 Model B
2. Camera
3. Monitor
4. Keyboard
5. Server

#### **1.5.2 เทคโนโลยีด้านซอฟต์แวร์**

1. Python : ภาษาที่ใช้ในการค้นหาภาพใบหน้าบุคคล การส่งรูปภาพใบหน้า การทำเว็บเซิร์ฟเวอร์สำหรับรับรูปภาพ
2. OpenCV : ไลบรารี (Library) ใช้ในการค้นหาใบหน้าบุคคลแบบเรียลไทม์ และใช้ในการระบุตัวตน
3. TensorFlow : ไลบรารี (Library) สำหรับการเรียนรู้รูปภาพใบหน้าอອกมาเป็นโมเดลโดยสามารถใช้งานได้กับภาษา Python
4. Tkinter : ไลบรารี (Library) สำหรับการพัฒนา (Graphical User Interface: GUI) ที่ใช้ภาษา Python
5. Open Face : โมเดลที่ใช้ในการระบุต้น 얼굴บุคคล
6. MediaPipe : ไลบรารี (Library) ของ (Machine Learning: ML) หรือ (Deep Learning: DL) ที่พัฒนาโดย Google ใช้ในการตรวจจับใบหน้าบุคคล

7. Flask Framework : เป็นโครงสร้างของ Restful API ที่ใช้ในการทำเว็บเซิร์ฟเวอร์ที่ปรับรูปภาพโดยเป็นภาษาไพธอน (Python) ทำให้สามารถเรียกใช้งาน OpenCV หรือ TensorFlow เมื่อรับรูปภาพสำเร็จและส่งผลลัพธ์
8. Rest API : ใช้ในการสร้างเว็บเซิร์ฟเวอร์สำหรับรับรูปภาพบนเซิร์ฟเวอร์
9. Application Programming Interface : ใช้ในการส่งรูปภาพผ่านอินเทอร์เฟซ (HyperText Transfer Protocol: HTTP)
10. Virtual Studio Code : ใช้ในการพัฒนาการค้นหาใบหน้าแบบเรียลไทม์และทำเว็บเซิร์ฟเวอร์สำหรับรับรูปภาพ

## 1.6 แผนการดำเนินงาน

| ขั้น ตอน การ ดำเนินงาน  | บ.ค. 2564 | ม.ค. 2565 | ก.พ. 2565 | มี.ค. 2565 | เม.ย. 2565 | พ.ค. 2565 | มิ.ย. 2565 | ก.ค. 2565 | ส.ค. 2565 | ก.ย. 2565 | ต.ค. 2565 | พ.ย. 2565 | ธ.ค. 2565 | ม.ค. 2566 | ก.พ. 2566 | มี.ค. 2566 |
|---|-----------|-----------|-----------|------------|------------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| ศึกษา และ การ ตรวจ จับ ใบหน้า และ การ ทำงาน บน raspbian os และการปีบอัดไฟล์ รูปภาพ                |           |           |           |            |            |           |            |           |           |           |           |           |           |           |           |            |
| ศึกษา และทดลอง การ ส่ง รูปภาพ ผ่าน RESTful API และ Python Flask framework และเทคนิค การปรับรูปภาพ |           |           |           |            |            |           |            |           |           |           |           |           |           |           |           |            |
| ศึกษา และทดลอง การ ทำงาน ของ DNN และ การ เรียนรู้ภาพใบหน้า บุคคล หรือ Train model                 |           |           |           |            |            |           |            |           |           |           |           |           |           |           |           |            |

| ชื่น ตอน การดำเนินงาน   | ธ.ค. 2564 | ม.ค. 2565 | ก.พ. 2565 | มี.ย. 2565 | พ.ค. 2565 | ก.ย. 2565 | ต.ค. 2565 | พ.ย. 2565 | ธ.ค. 2566 | ม.ค. 2566 | ก.พ. 2566 | มี.ย. 2566 |
|---|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| เก็บ ข้อมูล รูปภาพ<br>ใบหน้า ผู้ใช้งาน<br>ห้องวิจัย OASYS<br>และ ออกรูปแบบ ให้<br>ระบบ สามารถ<br>ตรวจจับใบหน้าได้<br>ดีขึ้น และส่งภาพ<br>ใบหน้าเร็วขึ้น                     |           |           |           |            |           |           |           |           |           |           |           |            |
| ติดตั้ง และทดสอบ<br>ระบบ และ ออกรูปแบบ<br>และ พัฒนา<br>โมเดล การ เรียนรู้<br>ภาพ ใบหน้า บุคคล<br>บนเซิร์ฟเวอร์ และ <sup>การส่งผลลัพธ์</sup>                                 |           |           |           |            |           |           |           |           |           |           |           |            |
| ออกแบบ และ <sup>พัฒนา GUI</sup><br>ตอบ รับ ผลลัพธ์<br>ส่ง ผลลัพธ์ ไป ยัง <sup>เซิร์ฟเวอร์</sup> และ <sup>เซิร์ฟเวอร์</sup> จัดการ<br>กับผลลัพธ์ ที่ได้รับ <sup>กลับมา</sup> |           |           |           |            |           |           |           |           |           |           |           |            |
| ทดสอบ ทั้ง ระบบ<br>ปรับปรุง ระบบ<br>และ ปรับ แต่ง<br>ระบบ ให้มีประสิทธิภาพขึ้น  |           |           |           |            |           |           |           |           |           |           |           |            |
| เขียน รายงาน สรุป<br>ผลการทำงาน   |           |           |           |            |           |           |           |           |           |           |           |            |

## **1.7 บทบาทและความรับผิดชอบ**

รับผิดชอบทุกส่วนของโครงการนี้ โดยที่ต้องใช้ความรู้ด้าน Computer vision, Web service, Storage, Rest API, Machine Learning และพัฒนาการเรียนรู้รูปภาพใบหน้า

## **1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม**

สามารถช่วยลดการแพร่ระบาดของเชื้อโควิด-19 ของพนักงานในสถานที่ มีการเก็บรูปภาพบุคคลที่เข้าสถานที่ โดยเมื่อมีเหตุการณ์ก่อร้ายที่บันทึกมาใช้เป็นหลักฐานได้โดยรูปภาพใบหน้านั้นจะไม่อนุญาตให้ผู้อื่นนำไปใช้ได้ จะสามารถใช้ได้ก็ต่อเมื่อมีการขออนุญาตเรียบร้อยซึ่งจะไม่ขัดกับกฎหมาย รูปภาพที่ส่งไปให้เชิร์ฟเวอร์นั้นมี การเข้ารหัสเพื่อป้องกันการโจรมมได้ เมื่อยืนยันตัวตนสำเร็จก็สามารถนำข้อมูลหรือสัญญาณไปยังระบบอื่น ๆ ได้ เช่นระบบบันทึกการเข้างาน ระบบบอกทางไปยังห้องทำงาน เป็นต้น ทำให้เป็นอีกช่องทางในการยืนยัน ตัวตนเพื่อเข้าสู่สถานที่

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

การทำโครงการ เริ่มต้นด้วยการศึกษาค้นคว้า ทฤษฎีที่เกี่ยวข้อง หรือ งานวิจัย/โครงการ ที่เคยมีผู้นำเสนอไว้ แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับการอธิบายถึงสิ่งที่เกี่ยวข้องกับโครงการ เพื่อให้ผู้อ่านเข้าใจเนื้อหาในบท ถัดไปได้ดียิ่งขึ้น เนื้อหาในบทนี้จะแบ่งออกเป็นดังนี้

#### 2.1 MediaPipe Holistic

อัลกอริทึมในการตรวจจับการเคลื่อนไหวของท่าทาง ใบหน้า และมือได้แบบเรียลไทม์ และสามารถที่จะรองรับอุปกรณ์ ทำให้เป็นวิธีการตรวจจับใบหน้าที่มีประสิทธิภาพ จุดเด่นหลักของ MediaPipe คือความรวดเร็วของการประมวลผลรูปภาพแบบเรียลไทม์ ซึ่งการใช้งานส่วนใหญ่นิยมใช้กับ OpenCV ที่ใช้ภาษาไพธอน (Python) [7] โดยแอปพลิเคชันที่ MediaPipe สามารถทำได้มีดังนี้

1. การตรวจจับใบหน้า
2. การตรวจจับท่วงท่า
3. การตรวจจับสิ่งของ
4. การตรวจจับเส้นผมบนหัว
5. การตรวจจับท่าทางของมือ

ซึ่งในโครงการนี้เลือกที่จะเอาการตรวจจับใบหน้ามาใช้งาน

#### 2.2 RESTful API

เป็นแนวทางในการสร้างเว็บเซอร์วิส (Web Service) แบบเรียบง่าย โดยเรียกใช้ผ่านทางเมท็อด GET / POST / PUT / DELETE โดย RESTful จะอยู่บนพื้นฐานของเกณฑ์วิธีนิยมส่งข้อความหลายมิติ (Hypertext Transfer Protocol: HTTP) โดยผู้รับบริการ (Client) จะส่ง คำขอ (Request) ไปยังรหัสสีบดัน ข้อมูลซึ่งระบุแหล่งที่อยู่ของทรัพยากรที่ต้องการ (Uniform Resource Locator: URI) ที่กำหนด และรับ Response กลับมาเป็น Payload ในรูปแบบของ (HyperText Markup Language: HTML), (Document Markup Language: XML), (JavaScript Object Notation: JSON) หรือรูปแบบ (format) อื่น ๆ [6] ซึ่งในโครงการนี้จะใช้ Payload แบบ JSON โดย RESTful API จะประกอบไปด้วย

- Client - ผู้ที่เข้ามาเป็น Request resource
- Server - ผู้ที่ให้บริการ Resource



รูปที่ 2.1: แสดงผังการทำงานของ RESTful

## 2.3 Image Processing

เป็นกระบวนการจัดการและวิเคราะห์รูปภาพให้เป็นข้อมูลในแบบดิจิทัล โดยใช้คอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงคุณภาพและปริมาณ (ขนาด รูปร่าง) [3] โดยกระบวนการจัดการและวิเคราะห์รูปภาพที่ใช้ในโครงงานนี้ มีดังนี้

### 2.3.1 การปรับปรุงคุณภาพของภาพ (Image Enhancement and Restoration)

การปรับปรุงคุณภาพของภาพเป็นการปรับปรุงหรือซ่อมแซมให้ข้อมูลภาพที่มีอยู่นั้นมี คุณภาพดีขึ้น เช่น ภาพที่ได้มาอาจมีความคมชัด (Contrast) น้อยหรือเบลอ ไม่คมชัด เราสามารถปรับภาพให้คมชัดได้ด้วยเทคนิค เช่น การปรับค่าความคมชัด (Contrast Enhancement) หรือการปรับเน้นเส้นขอบภาพ (Edge Enhancement) หรือในกรณีที่ภาพที่มี อยู่มีความไม่สมบูรณ์ เช่น มีสัญญาณรบกวน (Noise) เราสามารถใช้เทคนิค การกรองสัญญาณภาพ (Image Filtering) เพื่อกำจัดสัญญาณรบกวนได้

### 2.3.2 การบีบอัดข้อมูลภาพ (Image compression)

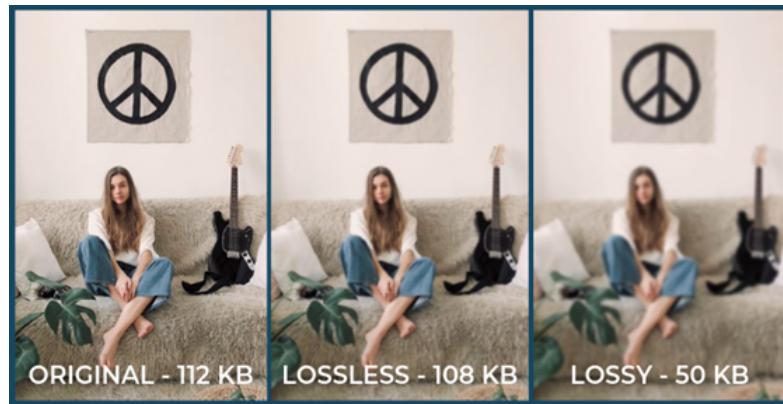
#### 1. การบีบอัดแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless compression)

ค่าความสว่างของแต่ละจุดภาพจะยังคงอยู่เหมือนเดิมทุกประการ หรือไม่มีการเปลี่ยนแปลงค่าของแต่ละจุดภาพ ซึ่งการบีบอัดวิธีนี้จะอาศัยเทคนิคการจัดเก็บข้อมูลเชิงเลขในการลดขนาดของข้อมูล

#### 2. การบีบอัดแบบสูญเสียรายละเอียดข้อมูล (Lossy compression)

วิธีการนี้จะมีการเปลี่ยนแปลงค่าความสว่างของจุดภาพนั้นหมายความว่า วิธีการนี้ไม่เหมาะสมสำหรับข้อมูลภาพที่ต้องมีการจำแนกข้อมูล (Classification)

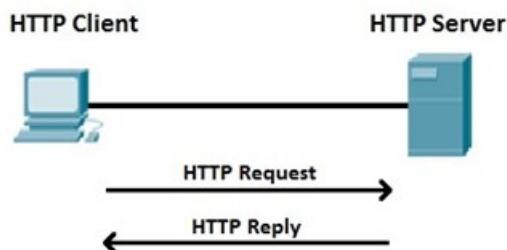
โดยในโครงงานนี้จะใช้การใช้การบีบอัดรูปภาพแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless compression)



รูปที่ 2.2: แสดงความแตกต่างของการบีบอัดข้อมูล

## 2.4 การส่งข้อมูลโดยใช้โปรโตคอลเอชทีพี (HyperText Transfer Protocol: HTTP)

เป็นโปรโตคอลที่ใช้งานในด้านเว็บไซต์และในระบบอินเทอร์เน็ต สามารถสื่อสารกับข้ามแพลตฟอร์มมักรู้จักนิยมใช้งาน HTTP เนื่องจากเป็นโปรโตคอลมาตรฐานที่มีมาให้ใช้งานในทุกภาษา และทุกอุปกรณ์ที่เขื่อมต่ออินเทอร์เน็ตได้ พื้นฐานของ HTTP มาจากโปรโตคอล (Transmission Control Protocol: TCP) ที่มีการใช้เพื่อรับ-ส่งข้อมูลในรูปแบบตามมาตรฐาน และใช้พอร์ต 80 เป็นค่าเริ่มต้น โดยผู้รับบริการ (HTTP Client) จะส่งข้อมูลผ่านคำสั่งการร้องขอแบบ POST เป็นคำสั่งที่ให้ส่งข้อมูลโดยแฟ้มข้อมูลไปกับเลขที่อยู่ไอพี (IP address) และใช้ร้องขอข้อมูลจากผู้ให้บริการ (HTTP Server) [8] ดังรูป



รูปที่ 2.3: โปรโตคอล HTTP

## 2.5 Raspberry Pi

เครื่องคอมพิวเตอร์ขนาดเล็ก มีคุณสมบัติดีน คือ ติดต่อ และความคุ้มอุปกรณ์วิเล็กทรอนิกส์ได้โดยใน Raspberry Pi ได้รวมเอาซีพียู (CPU) หน่วยความจำ (Memory) และพอร์ต (Port) ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัว殼เดียว กัน และสามารถเชื่อมต่ออินเทอร์เน็ตผ่านพอร์ตแลนหรือผ่านเครือข่ายไร้สาย [10] เช่น WiFi ในโครงงานนี้ได้เลือกใช้ Raspberry Pi มาเป็นอุปกรณ์ในการรับรูปภาพและค้นห้าใบหน้าในรูปภาพแบบเรียลไทม์ ส่งรูปภาพไปยังเซิร์ฟเวอร์ รอรับ

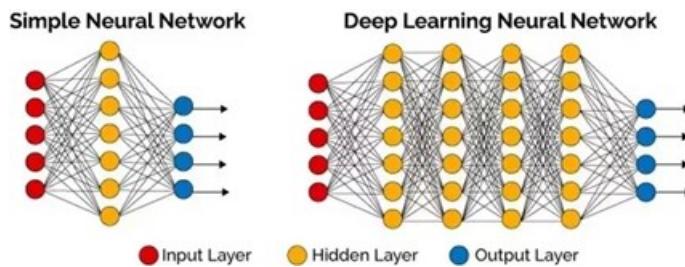
ผลลัพธ์กลับมาแสดงผล ซึ่งใช้พลังงานต่ำ กินกระแสไม่เกิน 2A ในสภาวะการทำงานปกติ และสามารถเชื่อมต่ออินเทอร์เน็ตผ่านระบบไวไฟ (Wi-Fi) เพื่อในการรับส่งข้อมูลจากเซิร์ฟเวอร์

## 2.6 Opensource Computer Vision (OpenCV)

ไลบรารีโอเพ่นซอร์สที่นิยมสำหรับการประมวลผลภาพขั้นพื้นฐาน เช่น การเบลอภาพ การผสมภาพ การเพิ่มคุณภาพของภาพ เพิ่มคุณภาพของวิดีโอ การรู้จำวัตถุต่าง ๆ ในภาพ หรือ การตรวจจับใบหน้าหรือวัตถุต่าง ๆ ในภาพและวิดีโอด้วย ปัจจุบัน (ปี 2022) OpenCV ได้พัฒนามาจนถึงรุ่นที่ 4 (Version 4) โดยในโครงงานนี้ได้เลือก OpenCV มาใช้ในการปรับแต่งรูป การตรวจจับใบหน้าแบบเรียลไทม์ และการระบุตัวตน [4]

## 2.7 Deep Learning

ศาสตร์แขนงหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning: ML) ที่เลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) โดยนำระบบโครงข่ายประสาท (Neural Network) มาซ้อนกันหลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูล ดังกล่าวจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจัดหมวดหมู่ข้อมูล (Classify the Data) ดังนั้นความสามารถของมันในอนาคตอาจจะเหนือมนุษย์ เนื่องจากสามารถเพิ่มพลังประมวลผลได้ไม่จำกัด ซึ่ง (Deep Learning: DL) คือโครงข่ายประสาทเทียม (Artificial Neural Network: ANN) ที่มีชั้นภายใน (Hidden layer) หลายชั้น เพื่อความสามารถในการคิดที่มากกว่าปกติ และสะท้อนสมองคนได้ดีขึ้น [5]



รูปที่ 2.4: แสดงโครงสร้าง Deep Learning

จะเห็นว่ามี 3 ส่วนคือ ชั้nrับข้อมูล (Input layer) ชั้นภายใน (Hidden layer) และชั้นแสดงผล (Output layer)

1. ชั้nrับข้อมูล (Input layer) เป็นจุดเริ่มต้นของการทำงานสำหรับ ANN จะทำหน้าที่ส่งข้อมูลไปยังแต่ละจุดต่อ (Node) ของชั้น (Layer)
2. ชั้นภายใน (Hidden layer) เป็นส่วนที่ทำหน้าที่ส่งต่อข้อมูลไปยังชั้นแสดงผล (Output Layer) โดยแต่ละครั้งที่ข้อมูลการฝึกอบรม(Training Data) ผ่านชั้น (Layer) นี้ไป แต่ละจุดต่อ (Node) จะค่อย ๆ ปรับน้ำหนัก (Weight) ให้เข้ากับข้อมูล (Data) มากขึ้นหรือถ้าอธิบายแบบเป็นทางการ Hidden Layer จะพยายามกักเก็บความซับซ้อนของชั้น (Layer) อื่น ๆ โดยการหาความสัมพันธ์ระหว่างคุณสมบัติ (Feature) ของข้อมูล (Data)

3. ชั้นแสดงผล (Output layer) เป็นส่วนที่จะแสดงผล (Output) ซึ่งจำนวนจุดต่อ (Node) ในชั้นแสดงผล (Output layer) จะขึ้นอยู่กับจำนวนประเภท (Class) ในข้อมูล (Data) อย่างเช่นจะสร้าง ANN เพื่อจำแนกหมวดหมู่ เช่น ก็ต้องมีจุดต่อแสดงผล (Output Node) 2 จุด และเมื่อใช้กับปัญหาการถดถอย (Regression Problems) ก็ต้องมี 1 จุดต่อ (Node) เท่านั้น เพราะหมายความว่า (Predict) แค่ตัวเลข

DL มีชั้นภายใน (Hidden layer) หลายชั้นทำให้มันสามารถคำนวณอะไรที่ซับซ้อนได้ และสามารถใช้เทคนิคต่าง ๆ ได้มากขึ้น และคิดอย่างเป็นขั้น เป็นตอนได้ดังรูป



รูปที่ 2.5: แสดง Deep Learning ที่มีหลาย Hidden layer

## 2.8 ส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface: GUI)

การติดต่อกับผู้ใช้โดยใช้ภาษาสัญลักษณ์ เป็นการออกแบบส่วนของโปรแกรมคอมพิวเตอร์ให้มีการโต้ตอบกับผู้ใช้ โดยการใช้ สัญลักษณ์ (Icon) รูปภาพ และสัญลักษณ์อื่น เพื่อแทนลักษณะต่าง ๆ ของโปรแกรม แทนที่ผู้ใช้จะพิมพ์คำสั่งต่าง ๆ ในการทำงาน ช่วยทำให้ผู้ใช้งานสามารถทำงานได้ง่าย และรวดเร็วขึ้น ไม่ต้องจดจำคำสั่งต่าง ๆ ของโปรแกรม เป็นวิธีการให้ความสะดวกแก่ผู้ใช้คอมพิวเตอร์ ให้ติดต่อสื่อสารกับระบบโดยผ่านทางภาพ เช่น ใช้มาสก์คลิปสัญลักษณ์ (Icon) แทนการพิมพ์คำสั่งดังแต่ก่อน โดยเฉพาะในบางโปรแกรมที่มีคำสั่งจำนวนมาก ซึ่งทำให้เมื่อต้องพิมพ์คำสั่งต่าง ๆ ทางแป้นพิมพ์ ช่วยทำให้เกิดความรวดเร็วในการทำงาน และไม่ต้องเสียเวลาในการเรียนรู้และจำคำสั่งที่ต้องการมากนัก เพียงดูจากไอคอนที่ปรากฏในโปรแกรมก็สามารถใช้งานได้ทันที [9]

## 2.9 Tkinter

Tkinter หรือ Tk เป็นโมดูลที่พัฒนามาจาก Tk GUI Toolkit ซึ่งทำงานอยู่บนระบบปฏิบัติการยูนิฟอร์ม ก่อน Python (Python) ได้เลือกโมดูลนี้ในการพัฒนากราฟิกบน Python (Python) เป็นหลัก ประกอบไปด้วย

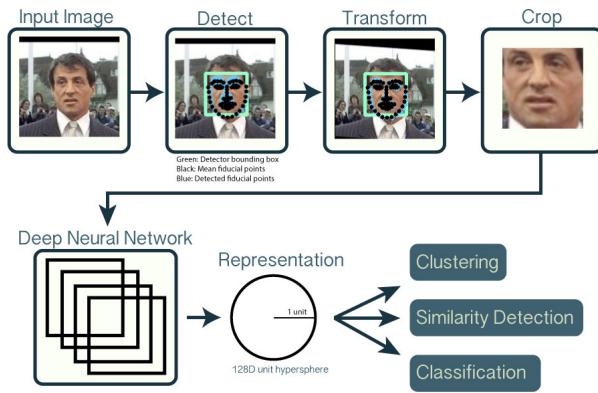
3 ส่วนที่สำคัญคือ วิดเจ็ต (Widgets) การจัดการรูปทรงเรขาคณิตให้กับวิดเจ็ต (Geometry management) และ การจัดการกับเหตุการณ์ต่าง ๆ (Event Handling) [2] ซึ่งมีรายละเอียดดังนี้

1. วิดเจ็ต (Widgets) คือ สิ่งต่าง ๆ หรือเรียกว่า օբเจกต์ (Object) ที่ปรากฏอยู่บนจอภาพ เช่น ปุ่ม (Button) ตัวหนังสือ (Label) เฟรม (Frame) กล่องเลือก (Checkbox) วิวตันไม้ (Tree views) แท็บเลื่อน (Scrollbars) และกล่องข้อความ (Text areas) เป็นต้น
2. การจัดการรูปทรงเรขาคณิตให้กับวิดเจ็ต (Geometry management) คือ การวางวิดเจ็ต (Widgets) ลงบนเฟรม (Frame) นั้นจะต้องกำหนดตำแหน่งในการวาง โดยอาศัยศาสตร์ทางด้านเรขาคณิตเข้าช่วย เพื่อให้วิดเจ็ต (Widgets) ที่วางอยู่ในตำแหน่งที่เหมาะสม ซึ่ง Python มี 3 เมธอดในการจัดการเกี่ยวกับเรขาคณิตของวิดเจ็ต (Widgets) ประกอบไปด้วยเมธอด pack(), grid() และ place()
3. การจัดการกับเหตุการณ์ต่าง ๆ (Event Handling) คือ เหตุการณ์ต่าง ๆ ที่ผู้ใช้งานกระทำกับวิดเจ็ต (Widgets) บน GUI เช่น การกดปุ่ม การกดปุ่มใด ๆ บนแป้นพิมพ์ การคลิกเมาส์ การปรับขนาดของหน้าต่างวินโดว์ เป็นต้น เหตุการณ์ต่าง ๆ เหล่านี้จะถูกจัดการโดย Tk ซึ่งเรียกว่าวนรอบเหตุการณ์ (Event loop) โดยจะทำงานร่วมกับระบบปฏิบัติการโดยตรง เช่น เมื่อคลิกเมาส์ไปปุ่มจะส่งผลให้ปุ่มดังกล่าวจะเปลี่ยนสี และเมื่อคลิกเมาส์ออกจากปุ่มจะทำให้สีของปุ่มกลับไปเป็นสีเดิม เป็นต้น

## 2.10 OpenFace (Open-source Face Recognition)

โมดูลที่ใช้ในการระบุตัวตนด้วยรูปภาพใบหน้าของมนุษย์ที่ทำงานร่วมกับ DNN และเป็นโมดูลแบบโอเพนซอร์ส [1] โดยมีหลักการทำงาน ดังนี้

1. ตรวจจับใบหน้าด้วยโมเดลที่ผ่านการฝึกอบรมล่วงหน้า
2. แปลงใบหน้าสำหรับโครงข่ายประสาทเทียม ทำงานร่วมกับ OpenCV เพื่อทำให้ดวงตา และริมฝีปากล่างปรากฏในตำแหน่งเดียวกัน และสัมพันในแต่ละภาพ
3. ใช้โครงข่ายประสาทเทียมเชิงลึก (DNN) เพื่อแสดงหรือฝังใบหน้าบนหน่วยไซเบอร์สเฟียร์ 128 มิติ การฝังเป็นการแสดงความแตกต่างสำหรับใบหน้าของครick กีต้าม ซึ่งแตกต่างจากใบหน้าอื่น ๆ การฝังนี้ทำให้เห็นคุณสมบัติที่ต่างกันมากขึ้นระหว่างใบหน้าสองใบหน้า ซึ่งทำให้ทราบว่าใบหน้านั้นไม่น่าจะใช่คนคนเดียวกันหรือเป็นคนคนเดียวกัน คุณสมบัตินี้ทำให้การจัดกลุ่ม การตรวจจับความคล้ายคลึงกัน และการจัดหมวดหมู่ทำได้ง่ายกว่าเทคนิคการจัดจำใบหน้าอื่น ๆ โดยที่ระยะห่างแบบยุคคลิธรรมะว่างคุณลักษณะต่าง ๆ ไม่มีความหมาย
4. ใช้เทคนิคการจัดกลุ่ม หรือการจัดหมวดหมู่ที่สนใจกับคุณสมบัติต่าง ๆ เพื่อทำการจัดจำของใบหน้าบุคคล



รูปที่ 2.6: แสดงการทำงานของ OpenFace

## 2.11 ความรู้ตามหลักสูตรชี้งถูกนำมาใช้หรือบูรณาการในโครงการ

### 2.11.1 Logic and Digital Circuits และ Microprocessor and Interfacing

ใช้ความรู้จากสิ่งวิชานี้ในการออกแบบระบบการทำงานของอุปกรณ์ของโครงสร้างชั้นงานในแต่ละส่วน และการเขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์

### 2.11.2 Digital Image Processing

ใช้ความรู้จากสิ่งวิชานี้ในการออกแบบการบีบอัดรูปภาพ การจัดเก็บรูปภาพ การทำให้รูปภาพมีคุณภาพที่ดีขึ้น เพื่อให้ความแม่นยำของการทำงานやすูงขึ้น

### 2.11.3 Deep Learning

เรียนรู้โครงสร้าง และการทำงานของ Neuron Network เพื่อหาโมเดลการรีบูตัวตนที่เหมาะสมกับงาน

### 2.11.4 CPE Lab

ใช้ในการทำ Web service และออกแบบ GUI

## 2.12 ความรู้นอกหลักสูตรชี้งถูกนำมาใช้หรือบูรณาการในโครงการ

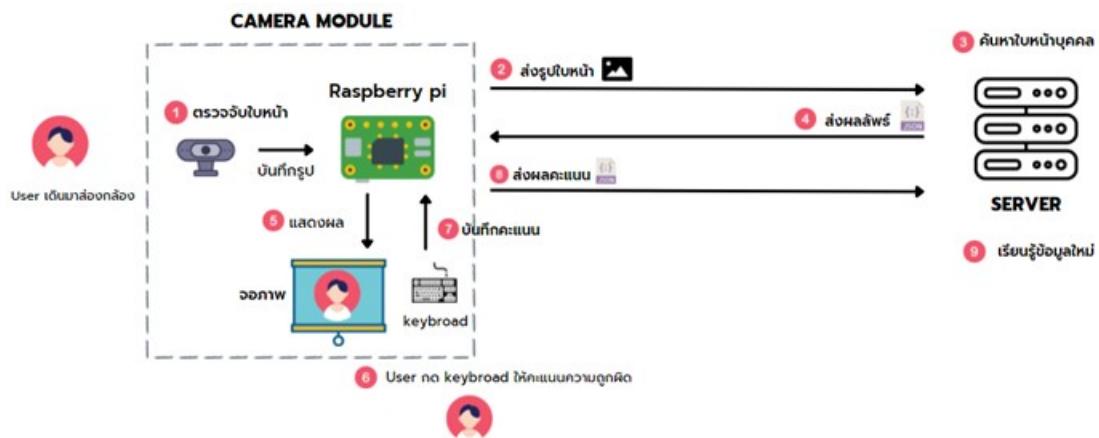
- การใช้งาน Raspbian OS ของ Raspberry Pi
- การเชื่อมต่อและใช้ GPIO ของ Raspberry Pi เพื่อติดตั้งชุดระบบายความร้อน

### บทที่ 3

## โครงสร้างและขั้นตอนการทำงาน

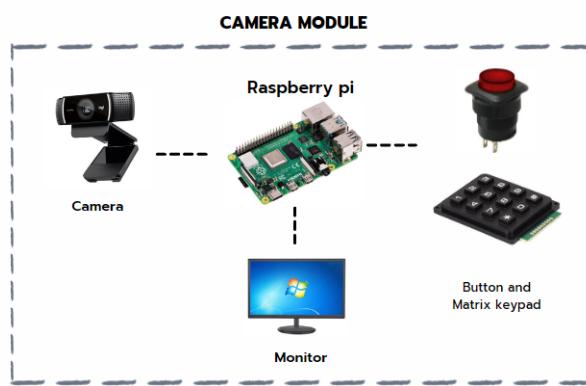
ในบทนี้จะกล่าวถึงโครงสร้างของระบบในภาพรวม และขั้นตอนการทำงานของระบบ โดยขั้นตอนการทำงานจะแบ่งเป็น 2 ส่วนใหญ่ ๆ คือส่วนของโมดูลกล้องและเซิร์ฟเวอร์

### 3.1 ภาพรวมโครงสร้างและการทำงานของระบบ



รูปที่ 3.1: Flow Diagram ภาพรวมของระบบ

#### 3.1.1 โมดูลกล้อง (Camera Module)



รูปที่ 3.2: ส่วนประกอบต่าง ๆ ที่ใช้ในโมดูลกล้อง (Camera Module)

อุปกรณ์ที่ใช้ในโมดูลกล้องสำหรับการตรวจจับใบหน้า มีดังนี้

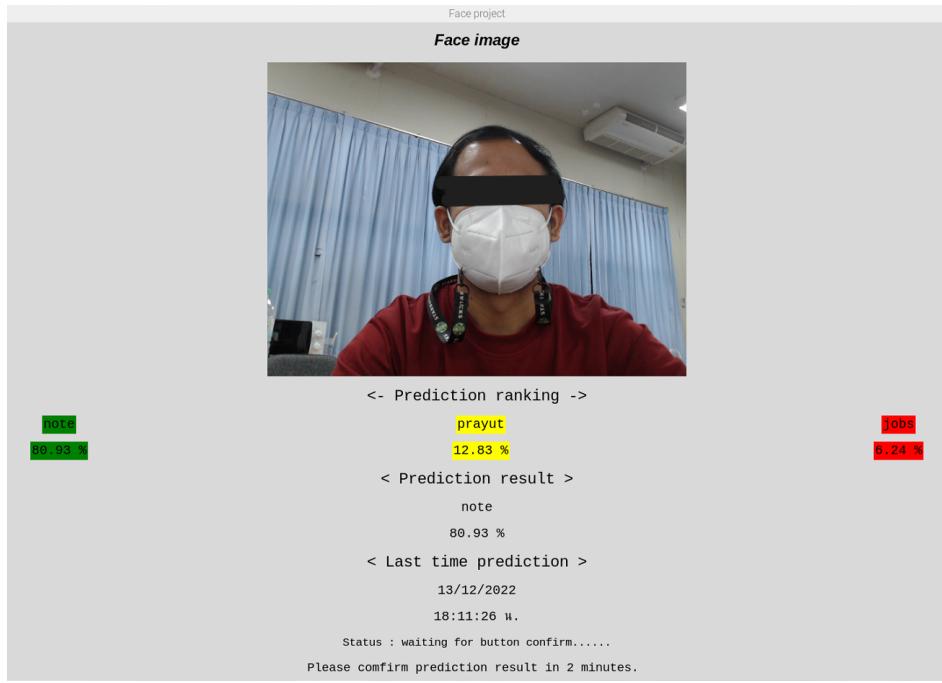
1. Raspberry Pi 4 Model B : แพลตฟอร์มที่ใช้ในการค้นหาใบหน้าบุคคล ซึ่งคุณสมบัติที่จำเป็นได้แก่ มีขนาดเล็ก สามารถส่งข้อมูลผ่านเครื่องข่ายไร้สายไวไฟ (WI-FI) หรือผ่านเครือข่ายที่ใช้สาย (LAN) สามารถอ่านข้อมูลภาพจากกล้องถ่ายภาพ และส่งรูปภาพไปยังเซิร์ฟเวอร์และรอรับผลลัพธ์ และส่งผลลัพธ์จากปุ่มกดกลับไปยังเซิร์ฟเวอร์
2. Camera : กล้องเว็บแคมที่มีใช้มาการส่งภาพใบหน้าไปยัง Raspberry Pi
3. Monitor : หน้าจอแสดงผลที่ใช้ในการแสดงผลลัพธ์ของการระบุต้นตอน
4. Button และ Matrix keypad : ใช้ในการรับการให้คำแนะนำการแสดงผลลัพธ์และแก้ไขความถูกผิดของการแสดงผลลัพธ์

### 3.1.2 การส่งข้อมูลไปยังเซิร์ฟเวอร์

การส่งรูปภาพจากโมดูลกล้องไปยังเซิร์ฟเวอร์นั้นในโมดูลกล้องใช้คำสั่ง ภาษาไพธอน (Python) ในการใช้สั่ง คำสั่งของระบบ (System) คือเคิร์ล (Client for URLs: cURL) ในการส่งรูปภาพผ่าน HTTP ไปยัง เซิร์ฟเวอร์ที่เป็น (RESTful Web Services: RWS) ผ่านเลขที่อยู่ไอพี (IP Address) ของเซิร์ฟเวอร์ โดย RWS นั้นใช้ Flask Framework ในการสร้างเนื้องจาก Flask Framework นั้นใช้ภาษาไพธอน (Python) ในการเขียนทำให้มีความสะดวกในการเรียก TensorFlow และ OpenCV มาใช้งาน

### 3.1.3 การแสดงผลการระบุตัวตน

การแสดงผลที่หน้าจอที่เชื่อมต่อกับ Raspberry Pi โดยรับข้อมูลมาจากเซิร์ฟเวอร์ที่ส่งข้อมูลบุคคลที่มีความ ใกล้เคียงจำนวน 5 คน แต่จะต้องมีความใกล้เคียงกับรายชื่อในฐานข้อมูลมากกว่า 80 เปอร์เซ็นต์จึงจะส่งผล- ลัพธ์ได้ แต่ถ้าไม่มีความใกล้เคียงมากกว่า 80 เปอร์เซ็นต์ก็จะแสดงผลว่าไม่รู้จัก โดยเซิร์ฟเวอร์ส่งข้อมูลแบบ JSON กลับมาให้ Raspberry Pi แบบการสนอง (Response) เมื่อรับข้อมูลจากเซิร์ฟเวอร์จะทำการนำไป แสดงผลที่หน้าจอ และเมื่อให้คำแนะนำความถูกผิดแล้วนั้นจะส่งผลคะแนนกลับไปยังเซิร์ฟเวอร์เพื่อทำการย้าย ไปยังที่จัดเก็บตามรายชื่อ



รูปที่ 3.3: แสดงผลลัพธ์การระบุตัวตน

### 3.1.4 เชิร์ฟเวอร์ (Server)

ทำหน้าที่ในการเป็นเว็บเซอร์วิส (Web service) ในการรับรูปภาพเพื่อนำรูปภาพมาระบุตัวตนโดยใช้ OpenCV เพื่อบอกว่ารูปนี้มีความใกล้เคียงกับบุคคลโดยโมเดลที่ใช้ในการระบุตัวตนนั้นมาจาก TensorFlow ในกรณีนำรูปภาพจากที่จัดเก็บ (Storage) มาทำการเรียนรู้จนได้โมเดลไปใช้งานและเมื่อ OpenCV บอกผลลัพธ์ได้แล้ว จึงทำการส่งขอร้อง (Request) กลับไปยังโมดูลกล้องแล้วทำการรอรับคะแนนเพื่อที่จะนำรูปภาพย้ายไปยังตำแหน่งที่จัดเก็บของบุคคลนั้น ๆ เมื่อจบวันในทุก ๆ วันเชิร์ฟเวอร์จะทำการสั่ง TensorFlow เรียนรู้รูปภาพใหม่และนำโมเดลใหม่ไปใช้งาน โดยตัวเชิร์ฟเวอร์จะมีความต้องการด้าน咽าร์ดแวร์คือต้องมีความจุมากกว่า 1 เทราไบต์ (Terabyte) หน่วยความจำขนาด 16 กิกะไบต์ (Gigabyte) ในการประมวลผล

### 3.1.5 การระบุตัวตน

การระบุตัวตนจะใช้ภาพถ่ายใบหน้าที่ได้รับมาจากการจัดเก็บ โดยใช้ OpenCV ในการระบุตัวตนโดยรับตัวโมเดลที่ใช้ในการนำรูปภาพใบหน้าว่ามีความใกล้เคียงมากน้อยเพียงใด แล้วจึงทำการคัดกรองรูปภาพที่มีความใกล้เคียงกับฐานข้อมูลมากกว่า 80 เปอร์เซ็นต์จะส่งข้อมูลไปให้เชิร์ฟเวอร์ทำการสั่ง แต่เมื่อไม่มีรูปภาพที่มีความใกล้เคียงกับฐานข้อมูลมากกว่า 80 เปอร์เซ็นต์ ก็จะส่งไปบอกเชิร์ฟเวอร์ว่า ”ไม่รู้จักบุคคลนี้”

### 3.1.6 การจัดเก็บรูปภาพใบหน้า

จัดเก็บรูปภาพใบหน้าลงในที่เก็บข้อมูลในเครื่อง (Local storage) บนเชิร์ฟเวอร์โดยแบ่งเป็นแฟ้มข้อมูล (Folder) ในแต่ละแฟ้มก็จะเป็นรายชื่อของบุคคลที่ลงทะเบียนหรือเป็นผู้ที่ใช้งานห้อง โดยเมื่อได้รับคะแนนจากการระบุตัวตนก็ไปเช็คกับผลลัพธ์จากการระบุแล้วจึงจะย้ายรูปภาพไปยังแฟ้มของรายชื่อนั้น ๆ

```
DATASET
|---john
|   1.jpg
|   10.jpg
|   2.jpg
|   3.jpg
|   4.jpg
|   5.jpg
|   6.jpg
|   7.jpg
|   8.jpg
|   9.jpg
|
|---kaitlyn
|   1.jpg
|   2.jpg
|   3.jpg
|   4.jpg
|   5.jpg
|   6.jpg
|   7.jpg
|   8.jpg
|
|---michel
|   20220401_135235.jpg
|   20220401_135237.jpg
|   20220401_135240.jpg
|   20220401_135253.jpg
|   20220401_135257.jpg
|   20220401_135301.jpg
|   20220401_135306.jpg
|   20220401_135319.jpg
```

รูปที่ 3.4: แผนภาพแสดงการจัดเก็บรูปภาพใบหน้า

### 3.1.7 การเรียนรู้รูปภาพ

ใช้ OpenFace ในการเรียนรู้รูปภาพ โดย OpenFace นั้นใช้ภาษาไพธอน (Python) ในการเขียนโปรแกรม เพื่อเรียนรู้รูปภาพใบหน้า เริ่มจากการนำรูปภาพของบุคคลที่บันทึกไว้ในที่จัดเก็บ (Storage) และรายชื่อของ บุคคลที่มีรูปภาพใบหน้าในที่จัดเก็บ (Storage) แล้วทำการเรียนรู้ด้วย OpenFace จะได้โน้มเดลการเรียนรู้เพื่อ นำไปใช้ในการระบุตัวตนของบุคคล โดยขั้นตอนการเรียนรู้จะใช้เวลาขึ้นอยู่กับจำนวนของรูปภาพที่จัดเก็บไว้ และใช้ทรัพยากรในการเรียนรู้รูปภาพใบหน้าที่สูงจึงนำ OpenFace ไปทำการเรียนรู้ที่เซิร์ฟเวอร์

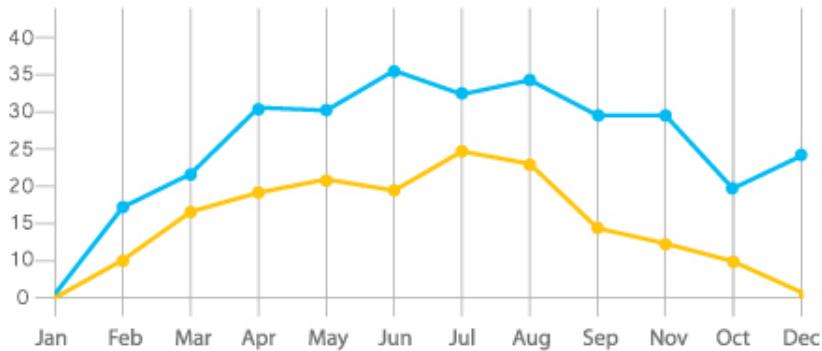
## บทที่ 4

### การทดลองและผลลัพธ์

ในบทนี้จะทดสอบระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้า โดยมีผู้ทดลอง 5 คน ทำการทดลองเป็นเวลา 5 วันเพื่อดูผลการทดลองในแต่ละวัน ได้แก่ ความแม่นยำของการระบุตัวตนด้วยใบหน้า เวลาในการตรวจจับใบหน้าตลอดถึงการแสดงผล และความพึงพอใจของผู้ทดลองต่อระบบ

#### 4.1 ความแม่นยำของการระบุตัวตนด้วยใบหน้าในแต่ละวัน

การทดสอบนี้จะเป็นการทดสอบเพื่อวัดผลความแม่นยำในการระบุตัวตนนั้นมีความแม่นยำเพิ่มขึ้นหรือลดลงตามวันเวลาที่ตรวจจับภาพใบหน้า โดยจะบันทึกความแม่นยำหลังจากการนำยผลรูปภาพใบหน้าทุก ๆ ครั้งที่มีการตรวจจับภาพใบหน้าและนำไปแสดงผล

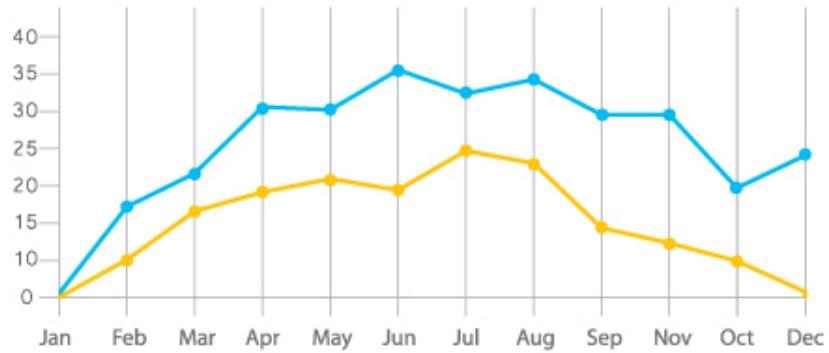


รูปที่ 4.1: กราฟแสดงของความแม่นยำของการระบุตัวตนในแต่ละวัน

จากราฟจะเห็นได้ว่าความแม่นยำเมื่อเวลาในการจัดเก็บข้อมูลนั้นมากขึ้นก็จะทำให้ความแม่นยำสูงขึ้นแต่จะมีจุดที่ความแม่นยำสูงที่สุดและลดลงเรื่อยๆเนื่องจาก ข้อมูลรูปภาพใบหน้ามีจำนวนที่เกินไปทำให้ความแม่นยำนั้นจะไม่สูงไปกว่าจัดนี้อีกแล้ว

#### 4.2 เวลาในการประมวลผลรูปภาพใบหน้า

การทดสอบต่อไปนี้เป็นการทดสอบเพื่อวัดผลว่าเวลาในการประมวลผลรูปภาพใบหน้าต่อจำนวนบุตัวตนที่ออกแบบนั้น มีเวลาที่มากหรือน้อยเพียงใดจากการบันทึกผลที่ Raspberry Pi โดยจะเปรียบเทียบที่วัดได้ในครั้งที่มีการตรวจจับภาพใบหน้า เปรียบเทียบอุปกรณ์เป็นกราฟเส้นดังนี้

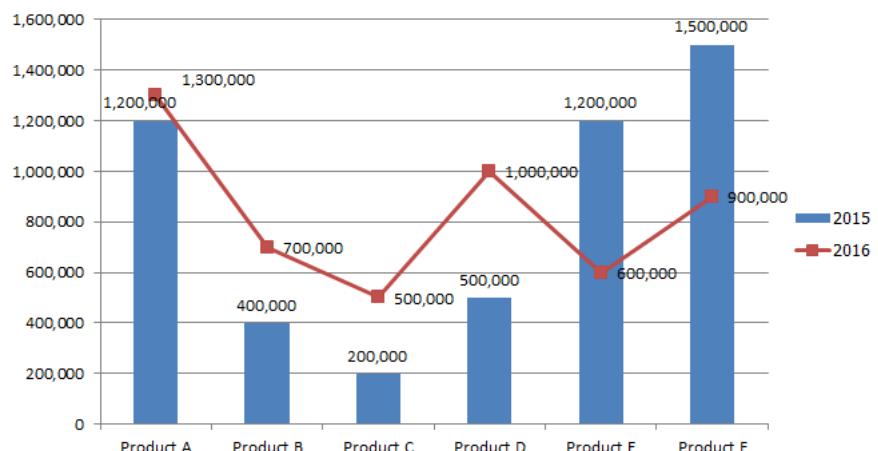


รูปที่ 4.2: กราฟแสดงเวลาในการประมวลผลของระบบในแต่ละเดือน

จากการจะเห็นได้ว่าเวลาในการประมวลรูปภาพใบหน้า ส่งรูปภาพไปยังเซิร์ฟเวอร์ การระบุตัวตน และการส่งผลลัพธ์กลับมาแสดงผลในหน้าจอันนั้นมีเวลาเฉลี่ย .... มิลลิวินาที

### 4.3 ความพึงพอใจของการทดลอง

จากการทดลองที่ห้องวิจัย OASYS ได้ทำการบันทึกความพึงพอใจของผู้ทดลองต่อระบบระบุตัวตนด้วยรูปภาพใบหน้าบุคคลที่ โดยจะได้ผลสรุปดังนี้



รูปที่ 4.3: กราฟแสดงความพึงพอใจของผู้ทดลอง

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 สรุปผล

นศ. ควรสรุปถึงข้อจำกัดของระบบในด้านต่างๆ ที่ระบบมีในเนื้อหาส่วนนี้ด้วย

#### 5.2 ปัญหาที่พบและแนวทางการแก้ไข

ในการทำโครงการนี้ พบร่วมกับปัญหาหลักๆ ดังนี้

#### 5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะเพื่อพัฒนาโครงการนี้ต่อไป มีดังนี้

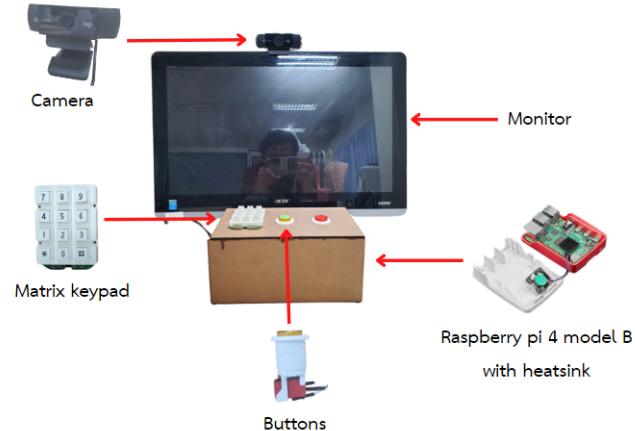
## บรรณานุกรม

- [1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.
- [2] Digitalschool.club. พื้นฐานการใช้โมดูล tkinter สำหรับ graphical user interface. <http://www.digitalschool.club/elearningcom/elearning/python/lesson9/index.php>, เมษายน 2560. [ออนไลน์; สืบค้นเมื่อวันที่ 28 พฤษภาคม 2565].
- [3] Navapat Jesadapatrakul. Image Processing. <https://medium.com/tni-university/image-processing-981c65c26289>, ตุลาคม 2562. [ออนไลน์; สืบค้นเมื่อวันที่ 24 ธันวาคม 2564 ].
- [4] Nuttakan Chuntra. OpenCV คืออะไร? <https://medium.com/@nut.ch40/opencv-คืออะไร-8771e2a4c414>, ธันวาคม 2561. [ออนไลน์; สืบค้นเมื่อวันที่ 5 ธันวาคม 2564].
- [5] phusitsom | zerohika. พื้นฐาน deep learning (ทฤษฎี): Intro. <https://phusitsom.medium.com/พื้นฐาน-deep-learning-ทฤษฎี-intro-7479d961cfef>, เมษายน 2563. [ออนไลน์; สืบค้นเมื่อวันที่ 10 กุมภาพันธ์ 2565].
- [6] Sakul Montha. Rest กับ restful api ต่างกันนะรู้ยัง. <https://iamgique.medium.com/restful-api-กับ-rest-api-ต่างกันนะรู้ยัง-2c70c42990e3>, มกราคม 2562. [ออนไลน์; สืบค้นเมื่อวันที่ 18 มกราคม 2565].
- [7] Sertis. Mediapipe holistic อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า มือ และท่าทาง ได้ในเวลาเดียวกัน. <https://sertiscorp.medium.com/mediapipe-holistic-อุปกรณ์ที่สามารถจับการเคลื่อนไหวของใบหน้า-มือ-และท่าทางได้ในเวลาเดียวกัน-e1185469e111>, มกราคม 2564. [ออนไลน์; สืบค้นเมื่อวันที่ 1 ธันวาคม 2565].
- [8] Tanabodin Kamol. Basic http. <https://medium.com/icreativesystems/basic-http-3a2b05e5aa19>, พฤษภาคม 2562. [ออนไลน์; สืบค้นเมื่อวันที่ 10 ธันวาคม 2564].
- [9] Wikipedia contributors. Graphical user interface — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Graphical\\_user\\_interface&oldid=1122648120](https://en.wikipedia.org/w/index.php?title=Graphical_user_interface&oldid=1122648120), ธันวาคม 2565. [ออนไลน์; สืบค้นเมื่อวันที่ 20 พฤษภาคม 2565].
- [10] Wikipedia contributors. Raspberry pi — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Raspberry\\_Pi&oldid=1126331603](https://en.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=1126331603), ธันวาคม 2565. [ออนไลน์; สืบค้นเมื่อวันที่ 12 ธันวาคม 2565].

ภาคผนวก

ภาคผนวก ก  
อุปกรณ์ต้นแบบและการแสดงผลบนหน้าจอ

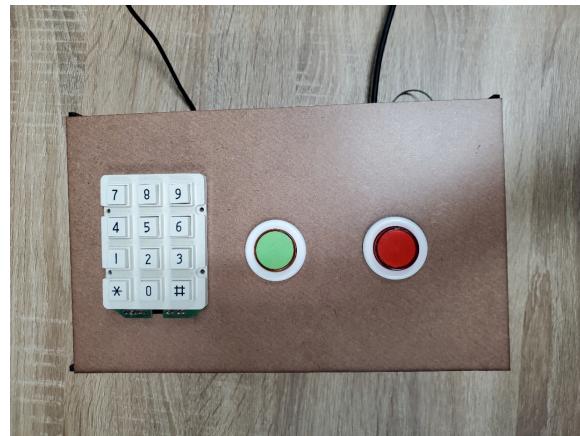
ก.1 รูปอุปกรณ์ที่ใช้ในการตรวจจับภาพใบหน้าและรับคำสั่งยืนยัน



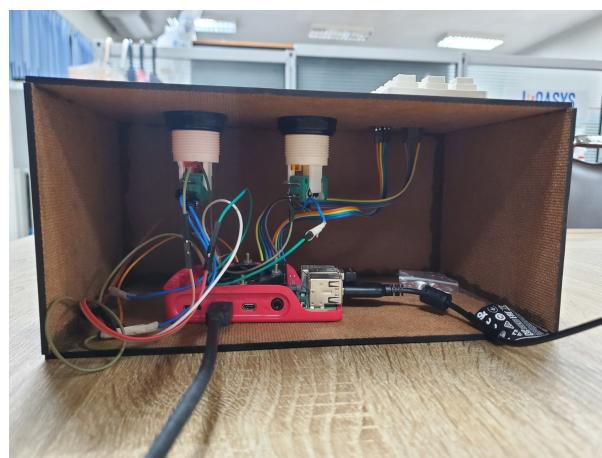
รูปที่ ก.1: แสดงอุปกรณ์ตรวจจับใบหน้า แสดงผลและรับผล



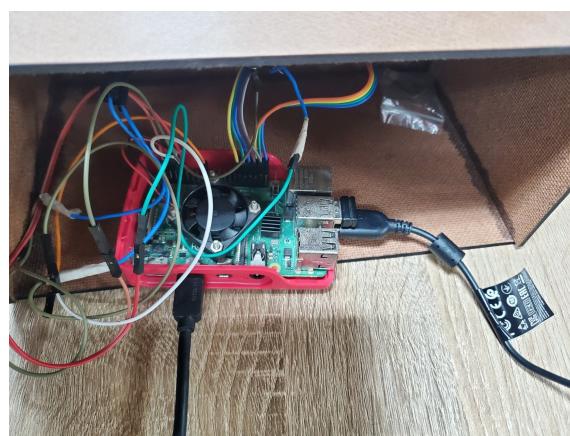
รูปที่ ก.2: กล้องที่ใช้ในการตรวจจับใบหน้าบุคคล



รูปที่ ก.3: ปุ่มกดให้คะแนนความถูกต้อง

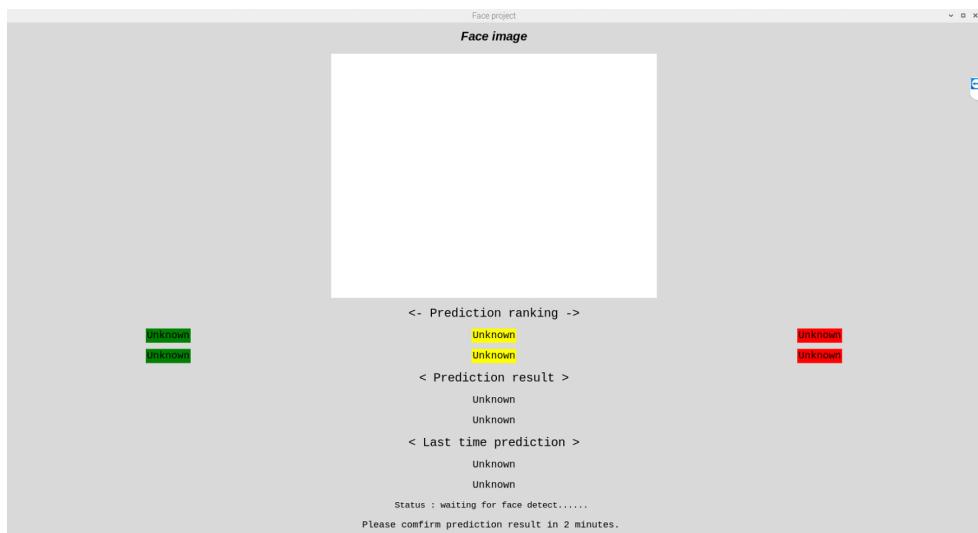


รูปที่ ก.4: แสดงภายในมอduลการตรวจจับใบหน้าและแสดงผล



รูปที่ ก.5: แสดงการเชื่อมต่อ Raspberry Pi

## ก.2 รูปภาพการแสดงผลบนหน้าจอ



รูปที่ ก.6: แสดงหน้า GUI เมื่อโปรแกรมเริ่มทำงาน



รูปที่ ก.7: แสดงหน้า GUI เมื่อผลลัพธ์การระบุตัวตนผิด

## ภาคผนวก ข

### คู่มือการใช้งานระบบ

โปรแกรมของโครงการนี้สามารถดาวน์โหลดได้จาก [https://www.example.com/test\\_ทดสอบ\\_url](https://www.example.com/test_ทดสอบ_url) ให้ระบบปฏิบัติการของ Raspberry Pi คือ Raspbian buster และให้ระบบปฏิบัติการของ Windows คือ Windows 10

#### ข.1 คู่มือการติดตั้งโปรแกรมเพื่อตรวจสอบใบหน้าบน Raspberry Pi

##### ข.1.1 คู่มือการติดตั้ง OpenCV บน Raspberry Pi

1. เปิด Termenal
2. พิมพ์คำสั่ง `sudo git clone https://github.com/freedomwebtech/raspbianlegacy.git`
3. พิมพ์คำสั่ง `cd raspbianlegacy`
4. พิมพ์คำสั่ง `sudo chmod 775 install.sh`
5. พิมพ์คำสั่ง `sudo ./install.sh` จากนั้นรอนานกว่าจะเสร็จ ใช้เวลาประมาณ 2 ชั่วโมง
6. เมื่อทำการติดตั้งเสร็จแล้วให้ทดลองใช้คำสั่ง `python3`
7. แล้วพิมพ์ `import cv2`
8. แล้วพิมพ์ `cv2.__version__` หากติดตั้งสำเร็จจะแสดงเลข `version`

##### ข.1.2 คู่มือการติดตั้ง TensorFlow lite บน Raspberry Pi

1. เปิด Termenal
2. พิมพ์คำสั่ง `sudo git clone https://github.com/freedomwebtech/raspbianlegacy.git`
3. พิมพ์คำสั่ง `cd raspbianlegacy`
4. พิมพ์คำสั่ง `sudo chmod 775 tensorflow-lite.sh`
5. พิมพ์คำสั่ง `sudo ./tensorflow-lite.sh` จากนั้นรอนานกว่าจะเสร็จ
6. เมื่อทำการติดตั้งเสร็จแล้วให้ทดลองใช้คำสั่ง `pip show tensorflow` หากติดตั้งสำเร็จจะแสดงข้อมูลของ tensorflow

##### ข.1.3 คู่มือการติดตั้ง Mediapipe library บน Raspberry Pi

1. เปิด Termenal
2. พิมพ์คำสั่ง `sudo apt update`
3. พิมพ์คำสั่ง `sudo pip3 install mediapipe-rpi4` จากนั้นรอนานกว่าจะเสร็จ

#### **ข.1.4 คู่มือการติดตั้ง Tkinter library บน Raspberry Pi**

1. เปิด Termenal
2. พิมพ์คำสั่ง sudo apt update
3. พิมพ์คำสั่ง sudo pip3 install tk จากนั้นรอนานกว่าจะเสร็จ

#### **ข.2 คู่มือการติดตั้งโปรแกรมเพื่อระบุตัวตนใบหน้าบน Server**

##### **ข.2.1 คู่มือการติดตั้ง Python บน Windows**

1. ไปยังเว็บไซต์ <https://www.python.org/downloads/> เพื่อดาวน์โหลดตัว setup ของ Python
2. เปิดไฟล์ setup
3. คลิก Install Now รอจนติดตั้งเสร็จ จากนั้นสามารถปิดหน้าต่างการ setup ได้
4. เมื่อติดตั้ง Python เสร็จแล้ว ให้ทดลองตรวจสอบว่า Python นั้นติดตั้งสำเร็จ โดยการเปิด cmd หรือ powershell และใช้คำสั่ง python -V จะแสดง version ของ Python ที่ทำการติดตั้ง

##### **ข.2.2 คู่มือการติดตั้ง Flask framework**

1. เปิด cmd หรือ powershell หรือ windows terminal
2. พิมพ์คำสั่ง pip install Flask รอจนติดตั้งเสร็จ

##### **ข.2.3 คู่มือการติดตั้ง OpenCV บน Windows**

1. เปิด cmd หรือ powershell หรือ windows terminal
2. พิมพ์คำสั่ง pip install opencv-python รอจนติดตั้งเสร็จ
3. เมื่อทำการติดตั้งเสร็จแล้วให้ทดลองใช้คำสั่ง python
4. แล้วพิมพ์ import cv2
5. แล้วพิมพ์ cv2.\_\_version\_\_ หากติดตั้งสำเร็จจะแสดงเลข version

##### **ข.2.4 คู่มือการติดตั้ง Scikit learn บน Windows**

1. เปิด cmd หรือ powershell หรือ windows terminal
2. พิมพ์คำสั่ง pip install scikit-learn รอจนติดตั้งเสร็จ

### ข.3 คู่มือการใช้งานการตรวจจับใบหน้า

1. เปิด Termenal
2. พิมพ์คำสั่ง git clone <https://github.com/freedomwebtech/raspbianlegacy.git>
3. หลังจากนั้นพิมพ์คำสั่ง cd face detection
4. เปิดไฟล์ request\_response.py
5. แก้ url ให้เป็นเลข IP Address ของ server รับรูปภาพ
6. พิมพ์คำสั่ง python main\_ui.py

หลังจากพิมพ์คำสั่ง python main\_ui.py จะแสดงหน้าตา UI ดังรูป

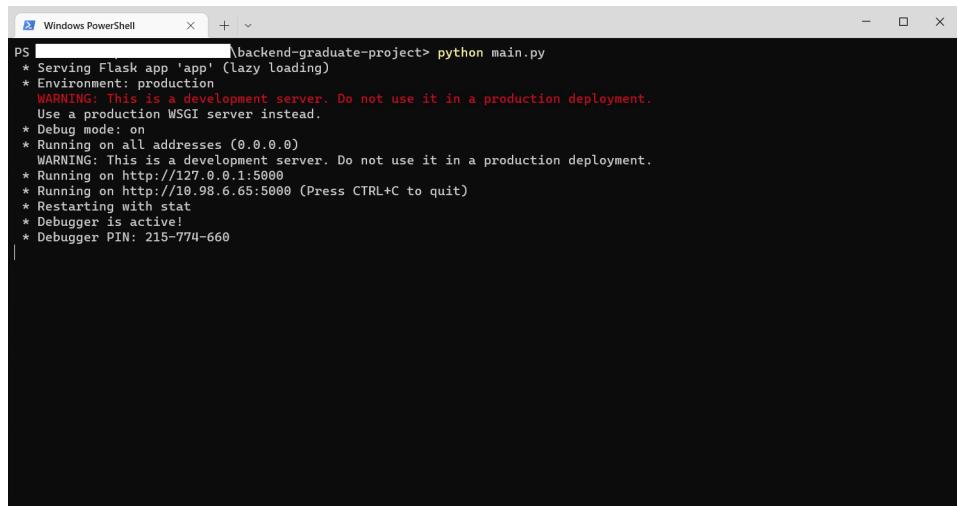


รูปที่ ข.1: แสดงหน้า GUI เมื่อโปรแกรมเริ่มทำงาน

### ข.4 คู่มือการใช้งานการระบุตัวตนด้วยใบหน้า

1. เปิด cmd หรือ powershell หรือ windows terminal
2. พิมพ์คำสั่ง git clone <https://github.com/protonnote/backend-graduate-project.git>
3. หลังจากนั้นพิมพ์คำสั่ง cd backend-graduate-project
4. หลังจากนั้นเปิดไฟล์ app.py เพื่อแก้ที่จัดเก็บรูปภาพ (UPLOAD\_FOLDER) ตามเส้นทาง (path) ที่อยู่ปัจจุบันให้ถูกต้องแล้วบันทึก
5. พิมพ์คำสั่ง python main.py

หลังจากพิมพ์คำสั่ง python main.py จะแสดงหน้าตา Termenal ดังรูป



```
PS C:\Users\...\backend-graduate-project> python main.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses (0.0.0.0)
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:5000
* Running on http://10.98.6.65:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 215-774-660
```

รูปที่ ข.2: แสดงหน้า Terminal เมื่อโปรแกรมระบุตัวตนด้วยไฟล์เริ่มทำงาน

## ประวัติผู้เขียน



ชื่อ-นามสกุล : นาย นฤสรณ์ กันจินะ

ระดับการศึกษา : ปริญญาตรี สาขา วิศวกรรมคอมพิวเตอร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ คณะ วิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

E-mail : naruson.kan@outlook.com

### กิจกรรมที่เคยเข้าร่วม

- สมาชิกทีม OASYS Data Analytics ตั้งแต่ปี พ.ศ. 2563
- ร่วมฝึกงานที่สำนักบริการเทคโนโลยีสารสนเทศ มหาวิทยาลัยเชียงใหม่ (Information Technology Service Center, Chiang Mai University)