

โครงการเลขที่ วศ.คพ. P804-2/2565

เรื่อง

ระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้กล้องวิดีโอการเรียนรู้  
ของเครื่องแบบต่อเนื่อง

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 algorithms**

**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 : ระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้กล้องวิธีการเรียนรู้ของเครื่องแบบต่อเนื่อง  
: Facial authentication system for building access using adaptive machine learning algorithms

โดย : นายนฤศร์ 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  
(อ.ดร. ณัฐนันท์ พรมสุข)

หัวข้อโครงการ : ระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้กล้องวิธีการเรียนรู้ของเครื่องแบบต่อเนื่อง  
: Facial authentication system for building access using adaptive machine learning algorithms

โดย : นายนฤศรัณ กันจินะ รหัส 620612153

ภาควิชา : วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษา : ผศ.ดร. ภานุสกร แซ่บประเสริฐ

ปริญญา : วิศวกรรมศาสตรบัณฑิต

สาขา : วิศวกรรมคอมพิวเตอร์

ปีการศึกษา : 2565

---

## บทคัดย่อ

การระบุตัวตนด้วยการใช้รูปภาพใบหน้าบุคคลเพื่อเข้าสู่สถานที่เป็นหนึ่งในรูปแบบการยืนยันตัวตนที่ช่วยลดการแพร่ระบาดของโรคติดเชื้อไวรัสโคโรนา (COVID-19) การใช้รูปภาพใบหน้าบุคคลนั้นยังมีจุดอ่อนเมื่อใบหน้าบุคคลมีการเปลี่ยนแปลง เช่น ผู้ชายขึ้น สมวัยเด็ก หรือต้องใส่หน้ากาก เป็นต้น จึงมีการใช้กล้องวิธีการเรียนรู้แบบต่อเนื่อง เพื่อนำรูปภาพใบหน้าใหม่ไปทำการเรียนรู้ ซึ่งก็คือ

เพื่อลดจุดอ่อนของการใช้รูปภาพใบหน้าบุคคล โดยได้พัฒนาระบบและทำการทดลองที่ทางเข้าสถานที่นั้นมาพิเศษที่ในการติดตั้งน้อยในโรงงานนี้จึงเลือกใช้ Raspberry Pi และในโปรเจกต์นี้ได้ทำการติดตั้งที่ทางเข้าห้องแลป OASYS เนื่องจาก Raspberry Pi นั้นมีประสิทธิภาพไม่แพ้พอยต์อุปกรณ์ระดับต้น หรือ การเรียนรูปภาพใบหน้าบุคคล ในโปรเจกต์นี้จึงใช้ Raspberry Pi ทำหน้าที่ตรวจจับใบหน้า และใช้เซิร์ฟเวอร์ในการระบุตัวตน และเรียนรู้รูปภาพใบหน้าบุคคล โดยแบบจำลองการตรวจจับใบหน้าบน Raspberry Pi นั้นสามารถที่จะตรวจจับใบหน้าบุคคลได้ดี ซึ่งเวลาเฉลี่ยในการแสดงผลการระบุตัวตนหลังจากการตรวจจับใบหน้านั้นน้อยกว่า 500 มิลลิวินาที และยังมีการยืนยันตัวตนด้วยรหัสอีกชั้นตอนเพื่อลดความผิดพลาดในการระบุตัวตน ทำให้ผู้ใช้งานยังได้รับความสะดวก ปลอดภัยจากโรคติดเชื้อไวรัสโคโรนา (COVID-19) จากการใช้งาน และยังมีความแม่นยำ

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

Facial recognition technology is one form of identity verification that can help reduce the spread of COVID-19 by allowing individuals to access buildings without the need for physical contact. However, using facial images has its weaknesses when a person's face changes, such as by growing a beard or wearing glasses. To improve the accuracy of new facial images, adaptive machine learning algorithms are employed.

Due to limited space, this project utilized Raspberry Pi, which was installed at the entrance of the OASYS lab. Although Raspberry Pi's performance is insufficient for facial recognition and learning images, it was used to detect faces while a server was used for recognizing and training facial models. The facial detection model on the Raspberry Pi works well and takes less than 500 milliseconds on average to verify identity after detecting a face. Additionally, to reduce the likelihood of identity errors, there is a pin code confirmation process that makes it convenient and safe for users to use and helps prevent the spread of COVID-19.

## กิตติกรรมประกาศ

โครงการนี้จะไม่สำเร็จลุล่วงลงได้ ถ้าไม่ได้รับความกรุณาจาก ผศ.ดร.ภาสกร แข่นประเสริฐ อารยที่ปรึกษา ที่ได้สละเวลาให้ความช่วยเหลือทั้งให้คำแนะนำ ให้ความรู้และแนวคิดต่าง ๆ รวมถึง อ.ดร. ณัฐนันท์ พรหมสุข และ ผศ.ดร.กำพล วรดิษฐ์ ที่ให้คำปรึกษาจนทำให้โครงการเล่มนี้เสร็จ สมบูรณ์ไปได้

ขอบคุณห้องวิจัย OASYS ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่เอื้อเฟื้อสถานที่ในการทำโครงการ สนับสนุนอุปกรณ์ต่าง ๆ และ ขอบคุณ นาย กมลพัฒน์ สุนทรพงษ์ และ นางสาวโซชิตาส ประตอน ที่คอยให้ความช่วยเหลือในการทำโครงการมาโดยตลอด

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

นายณัฐรัตน์ กันจินะ

12 ธันวาคม 2565

## สารบัญ

บทคัดย่อ . . . . .	ข
Abstract . . . . .	ค
กิตติกรรมประกาศ . . . . .	ง
สารบัญ . . . . .	จ
<b>1 บทนำ</b>	<b>1</b>
1.1 ที่มาของโครงการ . . . . .	1
1.2 วัตถุประสงค์ของโครงการ . . . . .	1
1.3 ขอบเขตของโครงการ . . . . .	1
1.3.1 ขอบเขตด้านฮาร์ดแวร์ . . . . .	1
1.3.2 ขอบเขตด้านซอฟต์แวร์ . . . . .	2
1.4 ประโยชน์ที่ได้รับ . . . . .	2
1.5 เทคโนโลยีและเครื่องมือที่ใช้ . . . . .	2
1.5.1 เทคโนโลยีด้านฮาร์ดแวร์ . . . . .	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 การส่งข้อมูลโดยใช้โปรโตคอลเอชทีพี (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 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ . . . . .	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 ความรู้นอกหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงการ . . . . .	12
<b>3 โครงสร้างและขั้นตอนการทำงาน</b>	<b>13</b>
3.1 ภาพรวมโครงสร้างและการทำงานของระบบ . . . . .	13
3.1.1 โมดูลกล้อง (Camera Module) . . . . .	14
3.1.2 การส่งข้อมูลไปยังเซิร์ฟเวอร์ . . . . .	14
3.1.3 การแสดงผลการระบุตัวตน . . . . .	14
3.1.4 เซิร์ฟเวอร์ (Server) . . . . .	15

3.1.5 การระบุตัวตน . . . . .	15
3.1.6 การจัดเก็บรูปภาพใบหน้า . . . . .	16
3.1.7 การเรียนรู้รูปภาพ . . . . .	16
3.1.8 การตรวจสอบรหัสยืนยันตัวตน . . . . .	16
<b>4 การทดลองและผลลัพธ์</b>	<b>18</b>
4.1 ความแม่นยำของการตรวจจับใบหน้า . . . . .	18
4.2 ความแม่นยำของการระบุตัวตนด้วยใบหน้าในแต่ละวัน . . . . .	19
4.3 เเวลาในการประมวลผลรูปภาพใบหน้า . . . . .	20
4.4 ความพึงพอใจของการทดลอง . . . . .	21
<b>5 บทสรุปและข้อเสนอแนะ</b>	<b>22</b>
5.1 สรุปผล . . . . .	22
5.2 ปัญหาที่พบและแนวทางการแก้ไข . . . . .	22
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ . . . . .	22
<b>บรรณานุกรม</b>	<b>24</b>
<b>ก อุปกรณ์ต้นแบบและการแสดงผลบนหน้าจอ</b>	<b>26</b>
ก.1 รูปอุปกรณ์ที่ใช้ในการตรวจจับภาพใบหน้าและรับคำสั่งยืนยัน . . . . .	26
ก.2 รูปภาพการแสดงผลบนหน้าจอ . . . . .	28
<b>ข คู่มือการใช้งานระบบ</b>	<b>30</b>
ข.1 คู่มือการติดตั้งโปรแกรมเพื่อตรวจจับใบหน้าบน Raspberry Pi . . . . .	30
ข.1.1 คู่มือการติดตั้ง OpenCV บน Raspberry Pi . . . . .	30
ข.1.2 คู่มือการติดตั้ง TensorFlow lite บน Raspberry Pi . . . . .	30
ข.1.3 คู่มือการติดตั้ง Mediapipe library บน Raspberry Pi . . . . .	30
ข.1.4 คู่มือการติดตั้ง Tkinter library บน Raspberry Pi . . . . .	31
ข.2 คู่มือการติดตั้งโปรแกรมเพื่อระบุตัวตนบน Server . . . . .	31
ข.2.1 คู่มือการติดตั้ง Python บน Windows . . . . .	31
ข.2.2 คู่มือการติดตั้ง Flask framework . . . . .	31
ข.2.3 คู่มือการติดตั้ง OpenCV บน Windows . . . . .	31
ข.2.4 คู่มือการติดตั้ง Scikit learn บน Windows . . . . .	31
ข.3 คู่มือการใช้งานการตรวจจับใบหน้า . . . . .	32
ข.4 คู่มือการใช้งานการระบุตัวตนด้วยใบหน้า . . . . .	32
<b>ข เอกสารเกี่ยวกับข้อมูลส่วนบุคคล</b>	<b>34</b>
ข.1 หนังสือขอความยินยอม เก็บรวบรวม ใช้ และเปิดเผยข้อมูลส่วนบุคคล . . . . .	34
<b>ประวัติผู้เขียน</b>	<b>35</b>

## บทที่ 1

### บทนำ

#### 1.1 ที่มาของโครงการ

การยืนยันตัวตนในการเข้าสถานที่หลายรูปแบบ เช่น การแสดงลายนิ้วมือ การใช้บัตรประจำตัว การระบุเอกสารลักษณ์ด้วยคลื่นวิทยุ (Radio Frequency Identification: RFID) และอื่น ๆ อีกมาก many ในปัจจุบัน มีสถานการณ์โควิด-19 ยังมีการแพร่ระบาด ทำให้ผู้คนไม่สามารถพบปะระหว่างพนักงานต้อนรับกับผู้ที่เข้าสถานที่ ทำให้การยืนยันตัวตนในการเข้าสถานที่โดยใช้รูปถ่ายใบหน้าจะช่วยลดการแพร่ระบาดของเชื้อโรค และมีความสะดวกในการใช้งานไม่ต่างกับการยืนยันตัวตนแบบอื่น อีกทั้ง เมื่อใบหน้าของบุคคลมีการเปลี่ยนแปลงตลอดในทุกวัน เช่น มีหนวด ไม่มีหนวด ผอมสัน ผอมยาว ใส่แหวน ไม่ใส่แหวน เป็นต้น มีผลทำให้การระบุตัวตนด้วยการใช้ภาพใบหน้าจะเกิดความผิดพลาด เกิดเป็นที่มาของการนำรูปภาพใบหน้าที่ได้รับเข้ามาใหม่ ในแต่ละครั้งของการระบุตัวตนนั้น ไปทำการเรียนรู้ใบหน้าใหม่ ทำให้ระบบสามารถจดจำภาพใบหน้าใหม่ที่มีการเปลี่ยนแปลง และความแม่นยำในการระบุตัวตนจะสูงขึ้นเมื่อมีจำนวนรูปภาพใบหน้าที่เก็บไว้มากขึ้น และหลายครั้งของการระบุตัวตนนี้ ไปทำการเรียนรู้ใบหน้าใหม่ ทำให้ระบบสามารถจดจำภาพใบหน้าใหม่ที่มีความแม่นยำที่ต่ำกว่าค่าที่กำหนดก็จะมีการใช้รหัสในการระบุตัวตน เพื่อเข้ามาช่วยในการยืนยันตัวตนให้มีความปลอดภัยมากยิ่งขึ้น

#### 1.2 วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาระบบตรวจจับใบหน้านุษย์ให้มีความแม่นยำที่สูง
- เพื่อให้ระบบมีความเหมาะสมของอุปกรณ์ที่ติดตั้ง
- เพื่อให้ระบบสามารถนำไปใช้งานได้จริง

#### 1.3 ขอบเขตของโครงการ

โดยระบบตรวจจับใบหน้าจะทำการติดตั้งหน้าทางเข้าห้องกลุ่มวิจัยทฤษฎีและการประยุกต์ใช้การหาค่าที่เหมาะสมที่สุดในระบบทางวิศวกรรม (OASYS Research Group Optimization Theory and Applications for Engineering SYStems Research Group: OASYS) และปรับให้มีความแม่นยำมากที่สุดให้ยังคงความพึงพอใจของผู้ใช้งานได้

##### 1.3.1 ขอบเขตด้านฮาร์ดแวร์

- ระบบจะสามารถค้นหาใบหน้าได้จะต้องมีพื้นที่ที่มีแสงสว่างเพียงพอ
- พื้นที่ที่ทำการติดตั้งต้องมีสัญญาณอินเทอร์เน็ตทั้งไร้สายหรือผ่านสายแลน (ข่ายงานบริเวณเฉพาะที่)

3. พื้นที่ที่ทำการติดตั้งต้องไม่มีผู้คนพลุกพล่าน
4. โปรแกรมการเรียนรู้ของเครื่องที่ไม่เกินกำลังด้านฮาร์ดแวร์ของเซิร์ฟเวอร์ที่ใช้เรียนรู้แบบจำลอง (model)

### 1.3.2 ขอบเขตด้านซอฟต์แวร์

1. สามารถจัดเก็บข้อมูลและรูปภาพใบหน้าได้
2. สามารถที่จะเรียนรู้รูปภาพใหม่ ที่เข้ามายัดเก็บได้
3. ระบบใช้เวลาในการตรวจจับใบหน้า ส่งภาพไปยังเซิร์ฟเวอร์ ระบุตัวตน และส่งผลลัพธ์กลับมาแสดงจะให้เวลาไม่เกิน 20 วินาที

## 1.4 ประโยชน์ที่ได้รับ

1. ผู้ที่เข้าสู่สถานที่ล็อกความเสี่ยงที่จะได้รับเชื่อโรค
2. ระบบสามารถที่จะระบุตัวตนในเวลาที่น้อย ทำให้ยังคงความสะดวกในการเข้าสู่สถานที่ได้ไม่ต่างจาก การเข้าสู่สถานที่รูปแบบอื่น ๆ
3. ระบบสามารถส่งต่อสัญญาณหรือข้อมูลไปยังส่วนอื่น ๆ ได้ เช่น บอกทางไปห้องทำงาน เปิดเครื่องคอมพิวเตอร์ในห้องทำงาน บอกตารางงานของบุคคลนั้น และจดจำเวลาเข้างานหรือออกงาน เป็นต้น

## 1.5 เทคโนโลยีและเครื่องมือที่ใช้

### 1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

1. คอมพิวเตอร์ขนาดเล็ก (Raspberry Pi 4 Model B)
2. กล้องเว็บแคมส์ที่ใช้การเชื่อมต่อผ่านช่องยูเอสบี (บัสคอมพิวเตอร์) (webcam)
3. จอภาพ (monitor)
4. แป้นแป้นอักษร (keyboard)
5. เครื่องบริการ (server)

### 1.5.2 เทคโนโลยีด้านซอฟต์แวร์

1. Python : ภาษาที่ใช้ในการค้นหาภาพใบหน้าบุคคล การส่งรูปภาพใบหน้า การทำเว็บเซิร์ฟเวอร์สำหรับรับรูปภาพ
2. OpenCV : ไลบรารี (Library) ใช้ในการค้นหาใบหน้าบุคคลแบบเรียลไทม์ และใช้ในการระบุตัวตน
3. TensorFlow : ไลบรารี (Library) สำหรับการเรียนรู้รูปภาพใบหน้าอุปกรณ์เป็นโมเดลโดยสามารถใช้งานได้ดีกับภาษา Python
4. Tkinter : ไลบรารี (Library) สำหรับการพัฒนา (Graphical User Interface: GUI) ที่ใช้ภาษา Python

5. Open Face : โมเดลที่ใช้ในการระบุต้นหน้าบุคคล
6. MediaPipe : ไลบรารี (Library) ของ (Machine Learning: ML) หรือ (Deep Learning: DL) ที่พัฒนาโดย Google ใช้ในการตรวจจับใบหน้าบุคคล
7. Flask Framework : เป็นโครงสร้างของ Restful API ที่ใช้ในการทำเว็บเซิร์ฟเวอร์ที่รับรูปภาพโดยเป็นภาษาไพธอน (Python) ทำให้สามารถเรียกใช้งาน OpenCV หรือ TensorFlow เมื่อรับรูปภาพสำเร็จและส่งผลลัพธ์
8. Rest API : ใช้ในการสร้างเว็บเซิร์ฟเวอร์สำหรับรับรูปภาพบนเซิร์ฟเวอร์
9. Application Programming Interface : ใช้ในการส่งรูปภาพผ่านเอชทีทีพี (HyperText Transfer Protocol: HTTP)
10. Virtual Studio Code : ใช้ในการพัฒนาการค้นหาใบหน้าแบบเรียลไทม์และทำเว็บเซิร์ฟเวอร์สำหรับรับรูปภาพ

## 1.6 แผนการดำเนินงาน

ขั้นตอน การดำเนินงาน	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 การตรวจ จับ ใบหน้า และ การ ทำงาน บน raspbian os และการบีบอัดไฟล์ รูปภาพ																
ศึกษา และทดลอง การ ส่ง รูปภาพ ผ่าน RESTful API และ Python Flask framework และเทคนิค การปรับรูปภาพ																

ขั้น ตอน การ ดำเนินงาน	ธ.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. 2566	ก.พ. 2566	มี.ค. 2566
ศึกษา และทดลอง การ ทำงาน ของ DNN และ การ เรียนรู้ภาพใบหน้า บุคคล หรือ Train model															
เก็บ ข้อมูล รูปภาพ ใบหน้า ผู้ใช้งาน ห้องวิจัย OASYS และ ออกรูปแบบ ให้ ระบบ สามารถ ตรวจจับใบหน้าได้ ดีขึ้น และส่งภาพ ใบหน้าเร็วขึ้น															
ติดตั้ง และทดสอบ ระบบ และออกแบบ- แบบ และ พัฒนา โมเดล การ เรียนรู้ ภาพใบหน้าบุคคล บนเซิร์ฟเวอร์ และ การส่งผลลัพธ์															
ออกแบบ และ พัฒนา GUI ตอบ รับ ผลลัพธ์ ส่ง ผลลัพธ์ ไป ยัง เซิร์ฟเวอร์ และ เซิร์ฟเวอร์ จัดการ กับผลลัพธ์ที่ได้รับ <sup>กลับมา</sup>															

ขั้น ตอน การ ดำเนินงาน	ธ.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. 2566	ก.พ. 2566	มี.ค. 2566
ทดสอบ ทั้ง ระบบ ปรับปรุง ระบบ และ ปรับ แต่ง ระบบให้มีประสิทธิภาพขึ้น															
เขียน รายงาน สรุป 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 โดยเมื่อมีเหตุการณ์ก่อการร้ายที่บันทึกมาใช้เป็นหลักฐานได้โดยรูปภาพใบหน้านั้นจะไม่อนุญาตให้ผู้อื่นนำไปใช้ได้ จะสามารถใช้ได้ต่อเมื่อมีการขออนุญาตเรียบร้อยซึ่งจะไม่ขัดกับกฎหมาย รูปภาพที่ส่งไปให้เชิร์ฟเวอร์นั้นมี การเข้ารหัสเพื่อป้องกันการโจกรรมได้ เมื่อยืนยันตัวตนสำเร็จก็สามารถนำข้อมูลหรือสัญญาไปยังระบบอื่น ๆ ได้ เช่นระบบบันทึกการเข้างาน ระบบบอกทางไปยังห้องทำงาน เป็นต้น ทำให้เป็นอีกช่องทางในการยืนยัน ตัวตนเพื่อเข้าสู่สถานที่

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

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

#### 2.1 MediaPipe Holistic

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

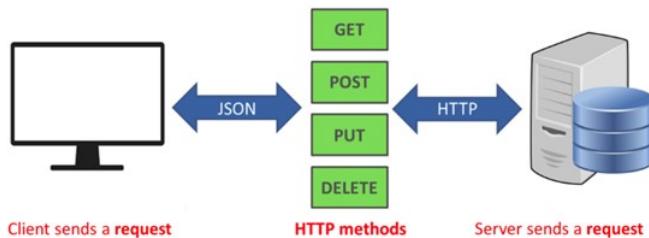
1. การตรวจจับใบหน้า
2. การตรวจจับท่วงท่า
3. การตรวจจับสิ่งของ
4. การตรวจจับเส้นผมบนหัว
5. การตรวจจับท่าทางของมือ

ซึ่งในโครงการนี้เลือกที่จะเอาการตรวจจับใบหน้ามาใช้งาน

#### 2.2 RESTful API

เป็นแนวทางในการสร้างเว็บเซอร์วิส (Web Service) โดยเรียกใช้ผ่านทางเมธอด GET POST PUT และ DELETE โดย RESTful จะอยู่บนพื้นฐานของเกณฑ์วิธีขึ้นส่งข้อความหลายมิติ (Hypertext Transfer Protocol: HTTP) โดยผู้รับบริการ (Client) จะส่ง คำขอ (Request) ไปยังรหัสสีบดันข้อมูลซึ่งระบุแหล่งที่อยู่ของทรัพยากรที่ต้องการ (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: ผังการทำงานของ RESTful API

## 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)

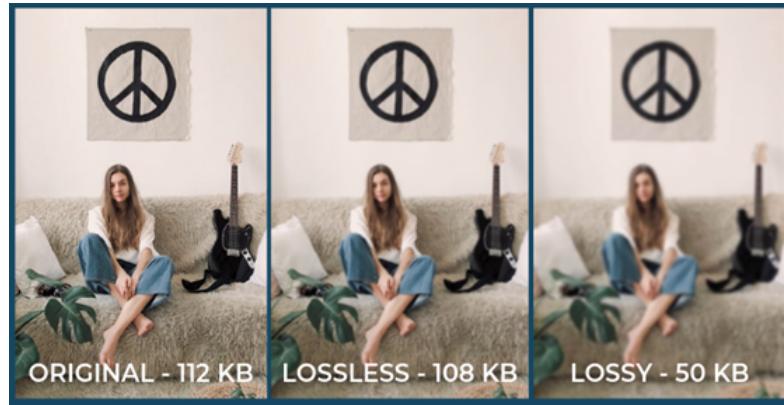
#### 1. การบีบอัดแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless compression)

ค่าความสว่างของแต่ละจุดภาพจะยังคงอยู่เหมือนเดิมทุกประการ หรือไม่มีการเปลี่ยนแปลงค่าของแต่ละจุดภาพ ซึ่งการบีบอัดวิธีนี้จะอาศัยเทคนิคการจัดเก็บข้อมูลเชิงเลขในการลดขนาดของข้อมูล

#### 2. การบีบอัดแบบสูญเสียรายละเอียดข้อมูล (Lossy compression)

วิธีการนี้จะมีการเปลี่ยนแปลงค่าความสว่างของจุดภาพนั้นหมายความว่า วิธีการนี้ไม่เหมาะสมสำหรับข้อมูลภาพที่ต้องมีการจำแนกข้อมูล (Classification)

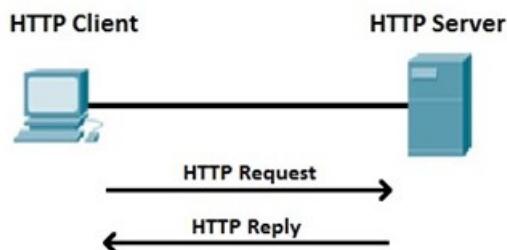
โดยในโครงงานนี้จะใช้การใช้การบีบอัดรูปภาพแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless compression)



รูปที่ 2.2: ความแตกต่างของการบีบอัดข้อมูล

## 2.4 การส่งข้อมูลโดยใช้โปรโตคอลเอชทีพี (HyperText Transfer Protocol: HTTP)

เป็นโปรโตคอลที่ใช้งานในด้านเว็บไซต์และในระบบอินเทอร์เน็ต สามารถสื่อสารกับข้ามแพลตฟอร์มมักรู้จักนิยมใช้งาน HTTP เนื่องจากเป็นโปรโตคอลมาตรฐานที่มีมาให้ใช้งานในทุกวากาชา และทุกอุปกรณ์ที่เชื่อมต่ออินเทอร์เน็ตได้ 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 และสามารถเชื่อมต่ออินเทอร์เน็ตผ่านพอร์ตแลนหรือผ่านเครือข่าย [10] เช่น ระบบแลนไร้สาย (WiFi) ในโครงงานนี้ได้เลือกใช้ Raspberry Pi มาเป็นอุปกรณ์ในการรับรูปภาพและค้นห้าใบหน้าในรูปภาพแบบเรียลไทม์ ส่งรูปภาพไป

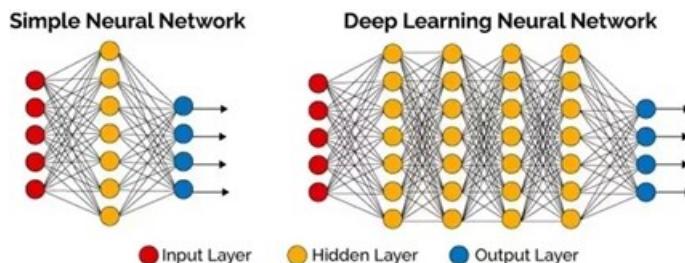
ยังเชิร์ฟเวอร์ รองรับผลลัพธ์กลับมาแสดงผล ซึ่งใช้พลังงานกระเส้าไฟฟ้าไม่เกิน 2 แอม培ร์ในสภาวะการทำงานปกติ และสามารถเชื่อมต่ออินเทอร์เน็ตผ่านระบบแลนรีสาย (Wi-Fi) เพื่อในการรับส่งข้อมูลจากเซิร์ฟเวอร์

## 2.6 Opensource Computer Vision (OpenCV)

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

## 2.7 Deep Learning

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



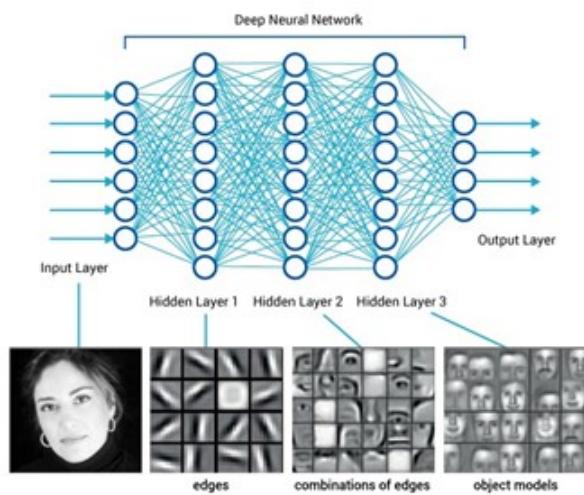
รูปที่ 2.4: โครงสร้าง Deep Learning

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

1. ชั้นรับข้อมูล (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: Deep Learning ที่มีหลาย Hidden layer

## 2.8 ส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface: GUI)

การติดต่อกับผู้ใช้โดยใช้ภาษาสัญลักษณ์ เป็นการออกแบบส่วนของโปรแกรมคอมพิวเตอร์ให้มีการโต้ตอบกับผู้ใช้ โดยการใช้สัญลักษณ์ (Icon) รูปภาพ และสัญลักษณ์อื่น เพื่อแทนลักษณะต่าง ๆ ของโปรแกรม แทนที่ผู้ใช้จะพิมพ์คำสั่งต่าง ๆ ในการทำงาน ช่วยทำให้ผู้ใช้งานสามารถทำงานได้ง่าย และรวดเร็วขึ้น ไม่ต้องจดจำคำสั่งต่าง ๆ ของโปรแกรม เป็นวิธีการให้ความสะดวกแก่ผู้ใช้คอมพิวเตอร์ ให้ติดต่อสื่อสารกับระบบโดยผ่านทางภาพ เช่น ใช้มาสก์คลิปเลือกสัญลักษณ์ (Icon) แทนการพิมพ์คำสั่งดังแต่ก่อน โดยเฉพาะในบางโปรแกรมที่มีคำสั่งจำนวนมาก ซึ่งทำให้ไม่ต้องพิมพ์คำสั่งต่าง ๆ ทางแป้นพิมพ์ ช่วยทำให้เกิดความรวดเร็วในการทำงาน และไม่ต้องเสียเวลาในการเรียนรู้และจำคำสั่งที่ต้องการมากนัก เพียงดูจากไอคอนที่ปรากฏในโปรแกรมก็สามารถใช้งานได้ทันที [9]

## 2.9 Tkinter

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

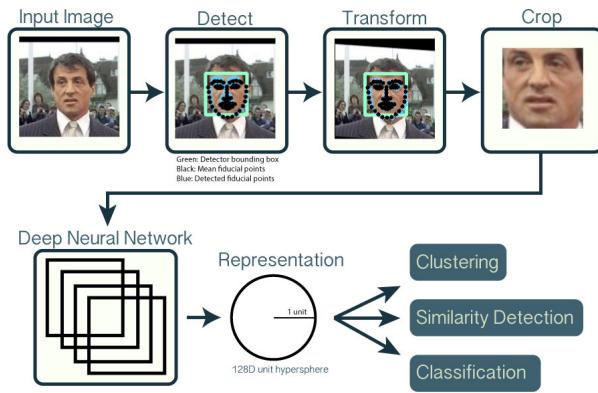
3 ส่วนที่สำคัญคือ วิดเจ็ต (Widgets) การจัดการรูปทรงเรขาคณิตให้กับวิดเจ็ต (Geometry management) และ การจัดการกับเหตุการณ์ต่าง ๆ (Event Handling) [2] ซึ่งมีรายละเอียดดังนี้

1. **วิดเจ็ต (Widgets)** คือ สิ่งต่าง ๆ หรือเรียกว่า อ็อปเจ็กต์ (Object) ที่ประกอบอยู่บนจอภาพ 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 ที่ผู้ใช้งานกระทำการกับวิดเจ็ต (Widgets) บน GUI เช่น การกดปุ่ม การกดปุ่มใด ๆ บนแป้นพิมพ์ การเคลื่อนเมาส์ การปรับขนาดของหน้าต่างวินโดว์ เป็นต้น เหตุการณ์ต่าง ๆ เหล่านี้จะถูกจัดการโดย Tk ซึ่งเรียกว่าวนรอบเหตุการณ์ (Event loop) โดยจะทำงานร่วมกับระบบปฏิบัติการโดยตรง เช่น เมื่อเคลื่อนเมาส์ไปยังปุ่มจะส่งผลให้ปุ่มดังกล่าวจะเปลี่ยนสี และเมื่อเคลื่อนเมาส์ออกจากปุ่มจะทำให้สีของปุ่มกลับไปเป็นสีเดิม 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) เพื่อแสดงหรือฝังใบหน้าบนหน่วยไซเบอร์สเปียร์ 128 มิติ การฝังเป็นการแสดงความแตกต่างสำหรับใบหน้าของครึ่งตัว ซึ่งแตกต่างจากใบหน้าอื่น ๆ การฝังนี้ทำให้เห็นคุณสมบัติที่ต่างกันมากขึ้นระหว่างใบหน้าสองใบหน้า ซึ่งทำให้ทราบว่าใบหน้านั้นไม่ใช่คนเดียวกันหรือเป็นคนคนเดียวกัน คุณสมบัตินี้ทำให้การจัดกลุ่ม การตรวจจับความคล้ายคลึงกัน และการจัดหมวดหมู่ทำได้ง่ายกว่าเทคนิคการจำจำใบหน้าอื่น ๆ โดยที่ระยะห่างแบบยุคลิดระหว่างคุณลักษณะต่าง ๆ ไม่มีความหมาย
4. ใช้เทคนิคการจัดกลุ่ม หรือการจัดหมวดหมู่ที่สนใจกับคุณสมบัติต่าง ๆ เพื่อทำงานการจัดจำของใบหน้าบุคคล



รูปที่ 2.6: การทำงานของ OpenFace

## 2.11 ความรู้้ตามหลักสูตรชีงถูกนำมาใช้หรือบูรณาการในโครงการ

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

ใช้ความรู้้จากส่องวิชานี้ในการออกแบบระบบการทำงานของอุปกรณ์ของโครงสร้างชิ้นงานในแต่ละส่วน และการเขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์

### 2.11.2 Digital Image Processing

ใช้ความรู้้จากส่องวิชานี้ในการออกแบบการบีบอัดรูปภาพ การจัดเก็บรูปภาพ การทำให้รูปภาพมีคุณภาพที่ดีขึ้น เพื่อให้ความแม่นยำของการทำงานやすูงขึ้น

### 2.11.3 Deep Learning

เรียนรู้้โครงสร้าง และการทำงานของ Neuron Network เพื่อหาโมเดลการรีบุตตันที่เหมาะสมกับงาน

### 2.11.4 CPE Lab

ใช้ในการทำ Web service และออกแบบ GUI

## 2.12 ความรู้้นอกหลักสูตรชีงถูกนำมาใช้หรือบูรณาการในโครงการ

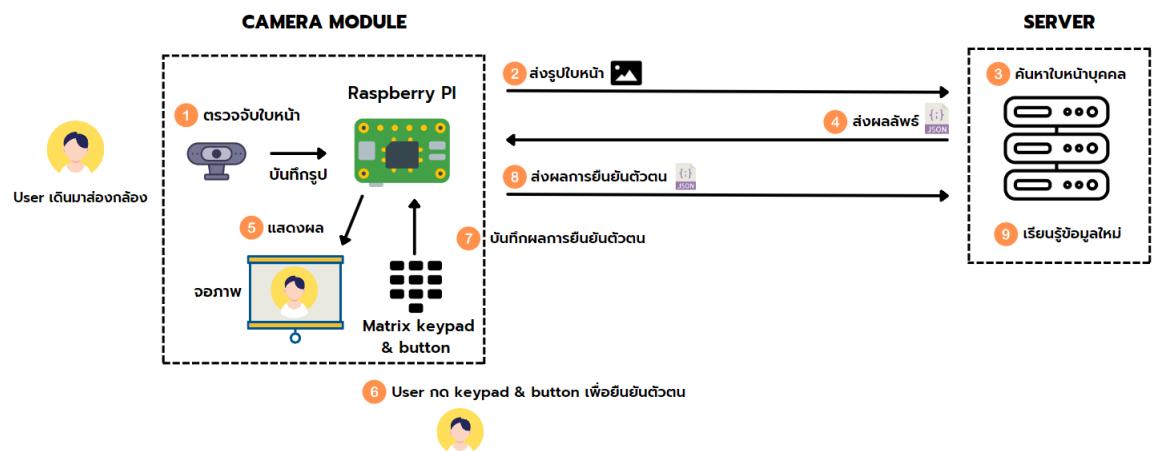
- การใช้งาน Raspbian OS ของ Raspberry Pi
- การตั้งค่ากล้องให้กับ Raspberry Pi และระบบ GPIO ของ Raspberry Pi เพื่อติดตั้งชุดระบบายความร้อน

## บทที่ 3

### โครงสร้างและขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงโครงสร้างของระบบในภาพรวม ขั้นตอนการทำงานของระบบ อุปกรณ์ หน้าที่ของอุปกรณ์ กลไกการต่าง ๆ และซอฟต์แวร์ที่ใช้ในระบบ โดยขั้นตอนการทำงานจะมี 9 ขั้นตอน โดยจะแบ่งระบบออก เป็นเป็น 2 ส่วนใหญ่ ๆ คือส่วนของมอดูลกล้อง และเซิร์ฟเวอร์

#### 3.1 ภาพรวมโครงสร้างและการทำงานของระบบ

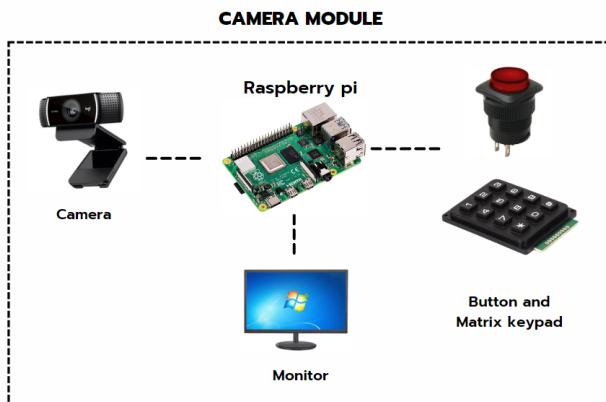


รูปที่ 3.1: ภาพรวมของระบบ

#### ขั้นตอนการทำงานของระบบมีดังนี้

- เมื่อผู้ใช้งานเดินมาส่องกล้องแล้วมอดูลกล้องจะทำการตรวจจับใบหน้า
- บันทึกและส่งรูปภาพใบหน้าไปยังเซิร์ฟเวอร์ในรูปแบบแฟ้มข้อมูลภาพกราฟิกส์สีเครื่องข่ายใช้ได้หลายระบบ (Portable Network Graphics: PNG) ผ่านโปรโตคอลเชิงทีทีพี (HTTP)
- เซิร์ฟเวอร์รับรูปภาพแล้วบันทึกรูปภาพเพื่อนำไประบุตัวตนของรูปภาพกับแบบจำลอง (model)
- เซิร์ฟเวอร์ส่งผลลัพธ์ของการระบุตัวตนกลับไปยังมอดูลกล้อง (Camera module) ในรูปแบบแฟ้มข้อมูลเจชัน (JSON)
- แสดงผลการระบุตัวตนทางหน้าจอ
- ผู้ใช้งานกดແປ່ນພິເສດ (Matrix keypad) หรือ ປຸ່ມກົດເພື່ອຍືນຢັນຕົວຕາມ
- บันทึกผลการกดຢືນຢັນຕົວຕາມ
- ส่งผลการຢືນຢັນຕົວຕາມไปยังเซิร์ฟเวอร์
- นำรูปภาพใบหน้าที่มีการຢືນຢັນຕົວຕາມไปจัดเก็บในฐานข้อมูลของบุคคลนั้น ๆ และจึงทำการเรียนรู้รูปภาพใบหน้าที่ได้รับเข้ามาใหม่ในเวลาลางคืน หรือช่วงเวลาที่มีผู้ใช้งานน้อย

### 3.1.1 มอดูลกล้อง (Camera Module)



รูปที่ 3.2: ส่วนประกอบต่าง ๆ ที่ใช้ในมอดูลกล้อง (Camera Module)

อุปกรณ์ที่ใช้ในมอดูลกล้องสำหรับการตรวจจับใบหน้ามีดังนี้

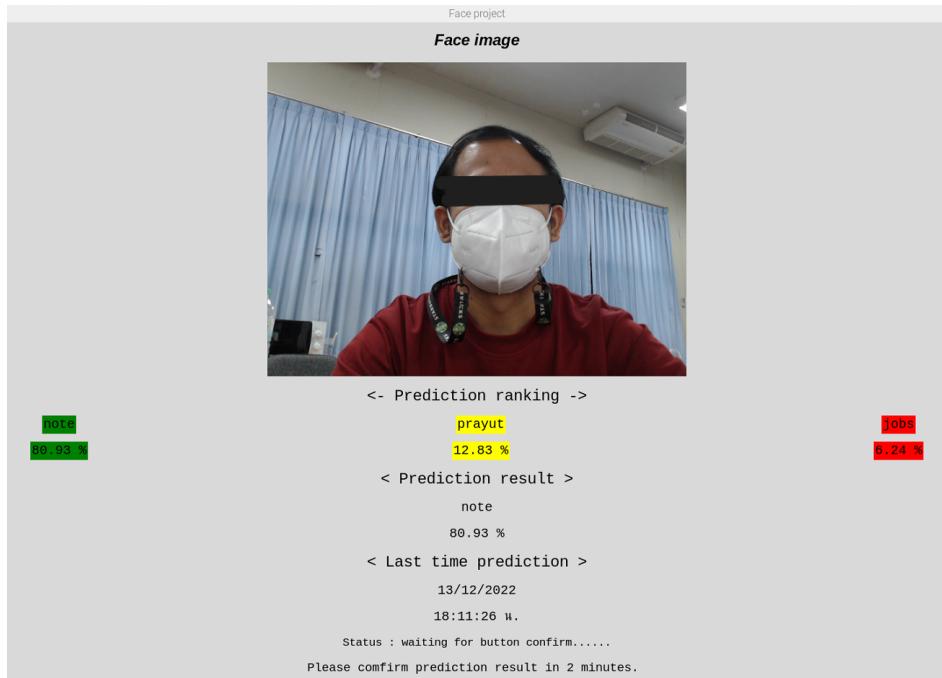
1. Raspberry Pi 4 Model B : แพลตฟอร์มที่ใช้ในการค้นหาใบหน้าบุคคล ซึ่งคุณสมบัติที่จำเป็นได้แก่ มีขนาดเล็ก สามารถส่งข้อมูลผ่านเครื่องข่ายไร้สายไวไฟ (WI-FI) หรือผ่านเครือข่ายที่ใช้สาย (LAN) สามารถอ่านข้อมูลภาพจากกล้องถ่ายภาพ และส่งรูปภาพไปยังเซิร์ฟเวอร์และรอรับผลลัพธ์ และส่งผลลัพธ์จากปุ่มกลับไปยังเซิร์ฟเวอร์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

### 3.1.2 การส่งข้อมูลไปยังเซิร์ฟเวอร์

การส่งรูปภาพจากมอดูลกล้องไปยังเซิร์ฟเวอร์นั้นในมอดูลกล้องใช้คำสั่ง ภาษาไพธอน (Python) ในการใช้สั่งคำสั่งของระบบ (System) คือเคิร์ล (Client for URLs: cURL) ในการส่งรูปภาพผ่าน HTTP ไปยังเซิร์ฟเวอร์ที่เป็น (RESTful Web Services: RWS) ผ่านเลขที่อยู่ไอพี (IP Address) ของเซิร์ฟเวอร์ โดย RWS นั้นใช้ Flask Framework ในการสร้างเนื้องจาก Flask Framework นั้นใช้ภาษาไพธอน (Python) ในการเขียนทำให้มีความสะดวกในการเรียก OpenCV มาใช้งาน

### 3.1.3 การแสดงผลการระบุตัวตน

การแสดงผลทางหน้าจอที่เชื่อมต่อกับ Raspberry Pi โดยรับข้อมูลมาจากเซิร์ฟเวอร์ในรูปแบบ JSON ที่มีข้อมูลผลลัพธ์การระบุตัวตน แล้วนำข้อมูลมาวิเคราะห์ ผลลัพธ์การระบุตัวตนจะต้องมีความน่าจะเป็นมากกว่า 80 เปอร์เซ็นต์จะแสดงผลลัพธ์ของบุคคลที่มีความน่าจะเป็นมากที่สุด แต่ถ้ามีความใกล้เคียงน้อยกว่า 80 เปอร์เซ็นต์จะแสดงผลว่า “ไม่รู้จัก” และเมื่อผลลัพธ์มีความน่าจะเป็นของการระบุตัวตนระหว่าง 60 ถึง 80 เปอร์เซ็นต์จะให้ผู้ใช้งานป้อนรหัสยืนยันตัวตน



รูปที่ 3.3: แสดงผลลัพธ์การระบุตัวตน

### 3.1.4 เชิร์ฟเวอร์ (Server)

ทำหน้าที่ในการเป็นเว็บเซอร์วิส (Web service) ในการรับรูปภาพเพื่อนำรูปภาพมาระบุตัวตนโดยใช้ OpenCV เพื่อบอกว่ารูปภาพใบหน้าที่ได้รับเข้ามามีความใกล้เคียงกับบุคคลในฐานข้อมูล โดยแบบจำลอง (model) ที่ใช้ในการระบุตัวตนนั้นมาจากการเรียนรู้รูปภาพใบหน้าจาก Storage ที่จัดเก็บ (Storage) จะได้แบบจำลอง (model) ไปใช้งานในการระบุตัวตน และเมื่อ OpenCV บอกผลลัพธ์ได้แล้วจึงทำการส่งการตอบสนอง (Response) กลับไปยังมอเดลกล้อง แล้วทำการรอรับข้อมูลการยืนยันตัวตนเพื่อนำรูปภาพที่ได้รับเข้ามาใหม่ นั้นย้ายไปยังตำแหน่งที่จัดเก็บของบุคคลนั้น ๆ ในเวลาปกติคือ 1 วินาที หรือเสลาที่มีผู้ใช้งานน้อยของทุก วัน เชิร์ฟเวอร์จะทำการสั่ง OpenFace เรียนรู้รูปภาพใหม่ และนำแบบจำลอง (model) ใหม่ไปใช้งาน โดยเชิร์ฟเวอร์จะมีความต้องการด้านฮาร์ดแวร์คือต้องมีความจุมากกว่า 1 เ特拉ไบต์ (Terabyte) หน่วยความจำขนาด 16 กิกะไบต์ (Gigabyte) ในการประมวลผล

### 3.1.5 การระบุตัวตน

การระบุตัวตนจะใช้ภาพถ่ายใบหน้าที่ได้รับมาจากมือถือกล้อง โดยใช้ OpenFace ในการระบุตัวตน และใช้แบบจำลอง (model) ที่มีการเรียนรู้จากรูปภาพที่ได้รับเข้ามาก่อนหน้า และนำมาใช้ในการระบุตัวตนของรูปภาพใบหน้าที่ได้รับเข้ามาใหม่ และนำผลลัพธ์การระบุตัวตนส่งข้อมูลไปให้เชิร์ฟเวอร์เพื่อทำการส่งผลลัพธ์การระบุตัวตนกลับไปยังมือถือกล้อง

### 3.1.6 การจัดเก็บรูปภาพใบหน้า

จัดเก็บรูปภาพใบหน้าลงในที่เก็บข้อมูลในเครื่อง (Local storage) บนเซิร์ฟเวอร์โดยแบ่งเป็นแฟ้มข้อมูล (Folder) ในแต่ละแฟ้มก็จะเป็นรายชื่อของบุคคลที่ลงทะเบียนหรือเป็นผู้ที่ใช้งานห้อง โดยเมื่อได้รับรหัสการยืนยันตัวตนในกรณีที่ความแม่นยำการระบุตัวตนระหว่าง 60 ถึง 80 เปอร์เซ็นต์จากการระบุตัวตน จะทำการเทียบรหัสที่ได้รับเข้ามา กับรหัสที่จัดเก็บบนฐานข้อมูล เมื่อผลลัพธ์การเทียบมีความถูกต้องจะย้ายรูปภาพไปยังแฟ้มของบุคคลนั้น แต่ถ้าผลลัพธ์การเทียบไม่ถูกต้อง จะย้ายรูปภาพไปยังแฟ้มของอื่น ๆ เพื่อให้ผู้ดูแลมาทำการเทียบด้วยตัวเอง

```
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

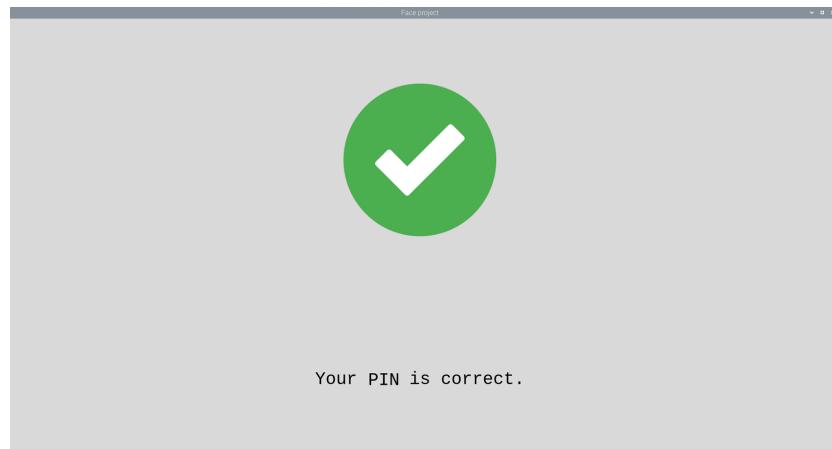
### 3.1.7 การเรียนรู้รูปภาพ

ใช้ OpenFace ในการเรียนรู้รูปภาพ โดย OpenFace นั้นใช้ภาษาไพธอน (Python) ในการเขียนโปรแกรมเพื่อเรียนรู้รูปภาพใบหน้า เริ่มจากการนำรูปภาพของบุคคลที่บันทึกไว้ในที่จัดเก็บ (Storage) และรายชื่อของบุคคลที่มีรูปภาพใบหน้าในที่จัดเก็บ (Storage) แล้วทำการเรียนรู้ด้วย OpenFace จะได้แบบจำลองการเรียนรู้เพื่อนำไปใช้ในการระบุตัวตนของบุคคล โดยขั้นตอนการเรียนรู้จะใช้เวลาขึ้นอยู่กับจำนวนของรูปภาพที่จัดเก็บไว้และใช้ทรัพยากรในการเรียนรู้รูปภาพใบหน้าที่สูงจึงนำ OpenFace ไปทำการเรียนรู้ที่เซิร์ฟเวอร์

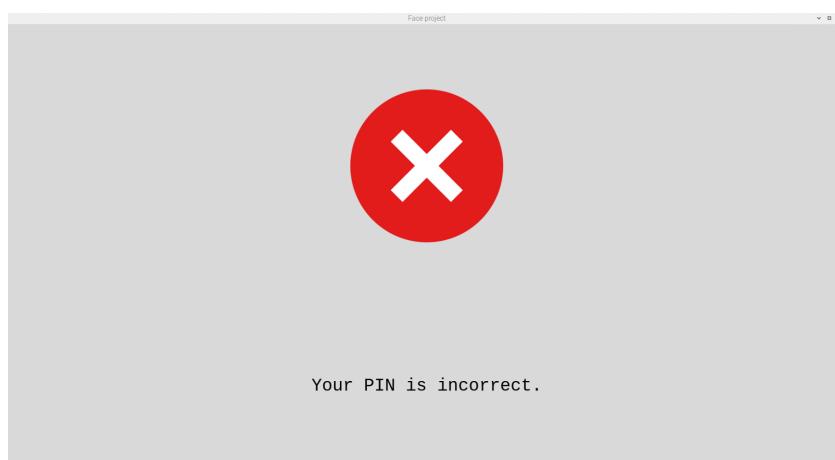
### 3.1.8 การตรวจสอบรหัสยืนยันตัวตน

การแสดงผลทางหน้าจอที่เชื่อมต่อกับ Raspberry Pi โดยรับข้อมูลรหัสยืนยันตัวตนจากผู้ใช้งานที่กรอกรหัสยืนยันตัวตนทาง Keypad เป็นเลขจำนวน 6 หลัก และส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อทำการตรวจสอบรหัสกับฐานข้อมูล และจะส่งผลลัพธ์การตรวจเช็คกลับไปยังมอเดลกล้อง โดยเซิร์ฟเวอร์ส่งข้อมูลในรูปแบบ JSON

กลับมาให้ Raspberry Pi แบบการสนอง (Response) เมื่อรับข้อมูลจากเซิร์ฟเวอร์จะนำไปแสดงผลที่หน้าจอ



รูปที่ 3.5: แสดงผลลัพธ์การกรอกรหัส เมื่อผลลัพธ์ถูกต้อง



รูปที่ 3.6: แสดงผลลัพธ์การกรอกรหัส เมื่อผลลัพธ์ไม่ถูกต้อง

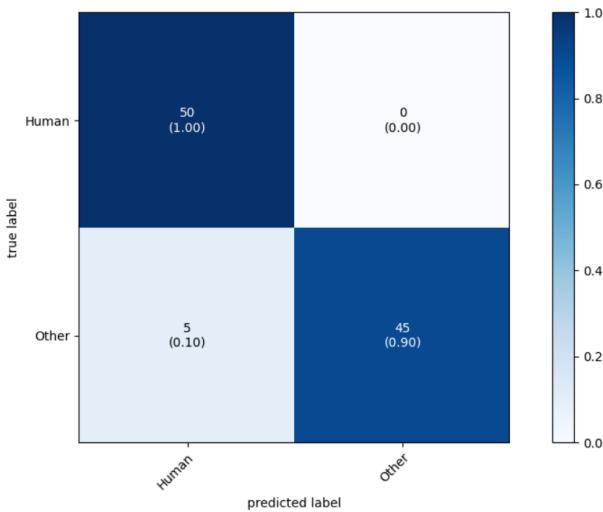
## บทที่ 4

### การทดลองและผลลัพธ์

ในบทนี้จะทดสอบระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าบุคคล ซึ่งมีการเก็บข้อมูลรูปภาพใบหน้าบุคคล เพื่อทำการทดลองเป็นเวลา 5 วัน และวัดผลการทดลองในแต่ละวัน ได้แก่ ความแม่นยำในการตรวจจับรูปภาพใบหน้าบุคคล ความแม่นยำของการระบุตัวตนด้วยรูปภาพใบหน้าบุคคล เวลาในการประมวลผลรูปภาพใบหน้าบุคคลตลอดถึงการแสดงผล และความพึงพอใจของผู้ทดลองต่อระบบ

#### 4.1 ความแม่นยำของการตรวจจับใบหน้า

การทดสอบนี้จะเป็นการทดสอบเพื่อวัดผลความแม่นยำในการตรวจจับภาพใบหน้าบุคคล โดยมีรูปภาพใบหน้าบุคคลจำนวน 50 รูป และรูปภาพสัตว์จำนวน 50 รูป ทำการทดลองกับแบบจำลองการตรวจจับใบหน้าบุคคล ได้ผลลัพธ์ดังกราฟด้านล่าง



รูปที่ 4.1: กราฟแสดงความแม่นยำของการตรวจจับใบหน้า

จากราฟจะสามารถคำนวณความแม่นยำได้ดังสูตร

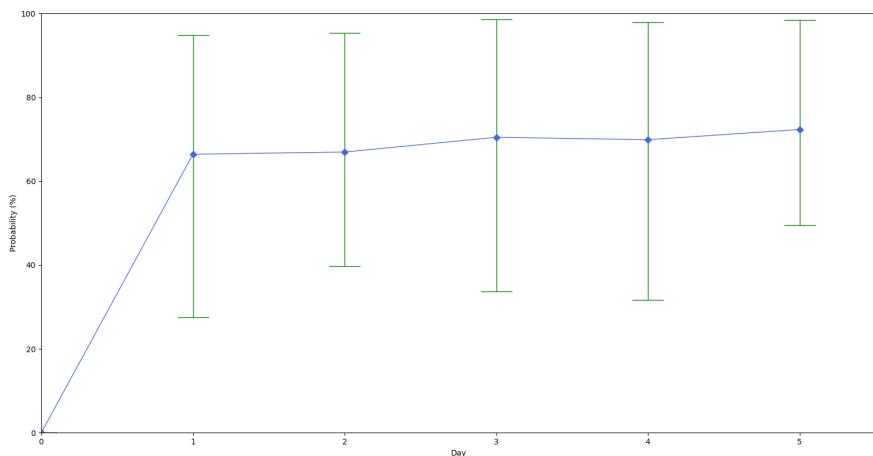
$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} = \frac{50}{50 + 0} = 1.0 \quad (4.1)$$

$$Accuracy = \frac{TruePositive + TrueNegative}{Positive + Negative} = \frac{50 + 45}{50 + 50} = 0.95 \quad (4.2)$$

$$F1Score = \frac{2 \cdot TruePositive}{2 \cdot TruePositive + FalsePositive + FalseNegative} = \frac{2 \cdot 50}{2 \cdot 50 + 0 + 5} = 0.9524 \quad (4.3)$$

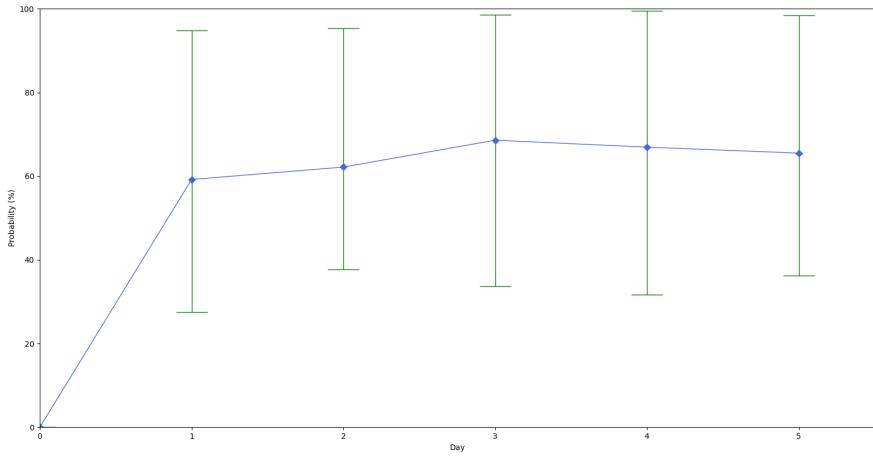
## 4.2 ความแม่นยำของการระบุตัวตนด้วยใบหน้าในแต่ละวัน

การทดสอบนี้จะเป็นการทดสอบเพื่อวัดผลความแม่นยำในการระบุตัวตน โดยความแม่นยำจะมีแนวโน้มที่เพิ่มขึ้นหรือลดลงตามจำนวนวันที่ตรวจจับภาพใบหน้าบุคคล เนื่องจากใช้กลวิธีการเรียนรู้ของเครื่องแบบต่อเนื่องซึ่งการทดลองนี้จะบันทึกผลลัพธ์การระบุตัวตนในทุกครั้งที่มีการตรวจจับภาพใบหน้าบุคคลไว้ที่เซิร์ฟเวอร์และ Raspberry Pi ในรูปแบบของ JSON และส่งผลลัพธ์การระบุตัวตนกลับไปยังมอเดลกล้องเพื่อแสดงผลลัพธ์การระบุตัวตนให้ผู้ใช้งานได้ทราบ โดยรูปที่ 4.2 จะแสดงกราฟผลลัพธ์ความแม่นยำของการระบุตัวตนหนึ่งคนจากผู้ทดลองทั้งหมด โดยจะนำข้อมูลที่ได้บันทึกจากทั้งสองแหล่งมาเปรียบเทียบความพิเศษของข้อมูลการระบุตัวตน



รูปที่ 4.2: กราฟแสดงความแม่นยำเฉลี่ยของการระบุตัวตนของ 1 บุคคลในแต่ละวัน

ความแม่นยำเฉลี่ยของการระบุตัวตน 1 บุคคลในแต่ละวันคือ 66.42, 66.95, 70.46, 69.89 และ 72.32 เปอร์เซ็นต์ ตามลำดับ ความแม่นยำต่ำสุดของการระบุตัวตนในแต่ละวันคือ 27.4, 39.68, 33.65, 31.70 และ 49.51 เปอร์เซ็นต์ ตามลำดับ และความแม่นยำสูงสุดของการระบุตัวตนคือ 94.81, 95.38, 98.6, 97.94 และ 98.37 เปอร์เซ็นต์ ตามลำดับ

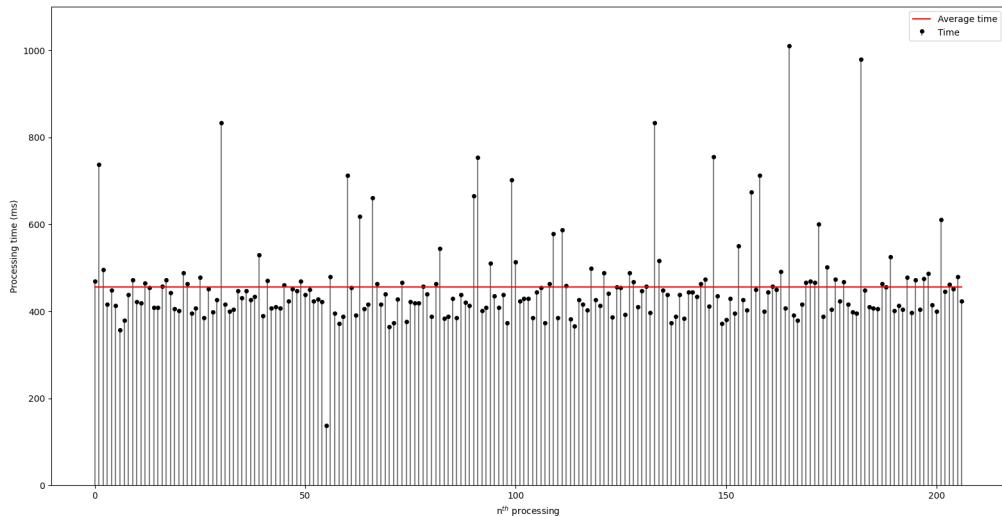


รูปที่ 4.3: กราฟแสดงความแม่นยำเฉลี่ยของการระบุตัวตนของแบบจำลองในแต่ละวัน

ความแม่นยำเฉลี่ยของการระบุตัวตนจากแบบจำลองในวันที่ 1 คือ 59.21 เปอร์เซ็นต์ และในวันที่ 5 คือ 65.51 เปอร์เซ็นต์ จะเห็นว่าความแม่นยำของการระบุตัวตนเฉลี่ยเพิ่มขึ้น 6.3 เปอร์เซ็นต์ ซึ่งเป็นไปตามหลักการเรียนรู้ของเครื่องแบบต่อเนื่อง

### 4.3 เวลาในการประมวลผลรูปภาพใบหน้า

การทดสอบต่อไปนี้เป็นการทดสอบเพื่อวัดเวลาทั้งหมดของการประมวลผลรูปภาพใบหน้า ส่งรูปภาพไปยังเซิร์ฟเวอร์ ระบุตัวตน และการส่งผลลัพธ์กลับมาแสดงผลในหน้าจอ โดยจะวัดเวลาในหน่วยของมิลลิวินาที (Millisecond) และบันทึกผลที่ Raspberry Pi โดยจะเปรียบเทียบเวลาที่ได้ได้ในทุกครั้งที่มีการตรวจจับภาพใบหน้า ซึ่งจำนวนครั้งที่ตรวจจับใบหน้าบุคคลคือ 207 ครั้งจากทั้งหมด 5 วันที่ทำการทดลองแล้วสรุปผลออกมาเป็นกราฟ

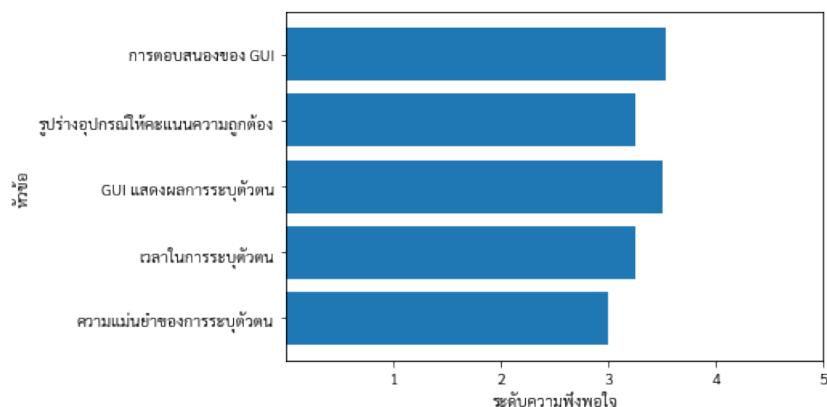


รูปที่ 4.4: กราฟแสดงเวลาในการประมวลผลของระบบในแต่ละวัน

จากการไฟเข็นได้ว่าเวลาในการประมวลผลรูปภาพใบหน้า ส่งรูปภาพไปยังเซิร์ฟเวอร์ การระบุตัวตน การส่งผลลัพธ์กลับมาแสดงผลในหน้าจอหนึ่งมีเวลามากที่สุด 1010 มิลลิวินาที น้อยที่สุด 138 มิลลิวินาที เฉลี่ย 455.62 มิลลิวินาที และค่าเบี่ยงเบนมาตรฐานของเวลาในการประมวลผล คือ 99.19

#### 4.4 ความพึงพอใจของการทดลอง

จากการทดลองที่ห้องวิจัย OASYS ได้ทำการบันทึกผลความพึงพอใจของผู้ทดลองต่อระบบระบุตัวตนด้วยรูปภาพใบหน้าบุคคล ในด้านต่าง ๆ ได้แก่ ความแม่นยำของกระบวนการระบุตัวตน เวลาในการระบุตัวตน GUI แสดงผลการระบุตัวตน รูปร่างอุปกรณ์ให้คะแนนความถูกต้อง และการตอบสนองของ GUI โดยจะได้ผลสรุปดังนี้



รูปที่ 4.5: กราฟแสดงความพึงพอใจของผู้ทดลอง

ผลความพึงพอใจของผู้ทดลองต่อระบบระบุตัวตนด้วยรูปภาพใบหน้าบุคคลในด้านความแม่นยำของการระบุตัวตนอยู่ที่ 3 (ปานกลาง) ในด้านเวลาในการระบุตัวตน GUI อยู่ที่ 3.25 (ปานกลาง) ในด้านแสดงผลการระบุตัวตนอยู่ที่ 3.5 (ปานกลาง) ในด้านรูปร่างอุปกรณ์ให้คะแนนความถูกต้องอยู่ที่ 3.25 (ปานกลาง) และในด้านการตอบสนองของ GUI อยู่ที่ 3.5 (ปานกลาง) และความคิดเห็นของผู้ใช้งานต่อระบบมีดังนี้

- มีความแม่นยามากกว่า นี่ ควรระบุได้ว่าเราเป็นใคร
- ในขณะใช้งานระบบ พบร่วมกับต้องยืนยันค้างเป็นระยะเวลาหนึ่ง ระบบจึงสามารถตรวจสอบได้ หากจะนำไปใช้งานจริง ก็ควรจะพัฒนาให้การตรวจจับใบหน้าใช้เวลาที่น้อยลง

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 สรุปผล

ในการทำโครงการระบบการยืนยันตัวตนด้วยการใช้ภาพใบหน้าเพื่อเข้าสู่สถานที่ด้วยการใช้กล้องวิธีการเรียนรู้ของเครื่องแบบต่อเนื่องสามารถพัฒนาให้มีคุณสมบัติที่ดีขึ้น สามารถตรวจสอบใบหน้านั้นตรวจสอบใบหน้าได้แม่นยำถึง 100 เปอร์เซ็นต์ เมื่อนำไปติดตั้งทดสอบสามารถทำงานได้อย่างถูกต้อง ใช้พื้นที่ติดตั้งน้อย และเวลาเฉลี่ยในการทำงานระหว่างเซิร์ฟเวอร์กับมอดูลกล้องคือ 455.618 มิลลิวินาที

#### 5.2 ปัญหาที่พบและแนวทางการแก้ไข

ในการทำโครงการนี้ พบร่วมกับปัญหาหลัก ๆ ดังนี้

- เมื่อรูปภาพใบหน้ามีความคมชัดน้อย จะทำให้แบบจำลองการตรวจจับใบหน้าไม่สามารถนำรูปภาพใบหน้าไปเรียนรู้รูปภาพใบหน้าเป็นแบบจำลองการระบุตัวตน จึงต้องมีการใช้การปรับรูปภาพให้ดีขึ้น เช่น การลดสัญญาณรบกวนที่เกิดขึ้นในภาพ การเพิ่มความคมชัด และการบีบอัดคงสัญญาณ เป็นต้น
- ในการตรวจจับรูปภาพใบหน้าบนมอดูลกล้อง เมื่อผู้ใช้งานขยับตัวในขณะที่กล้องกำลังໂฟกัสใบหน้าบุคคล จะทำให้รูปภาพใบหน้าบุคคลไม่ชัดส่งผลให้ผลลัพธ์ของรูปภาพที่ส่งไประบุตัวตนมีความผิดพลาดสูง จึงมีการกำหนดข้อตกลงให้ผู้ใช้อยู่ในตอนที่กล้องกำลังໂฟกัส
- เมื่อมอดูลกล้องเปิดใช้งานเป็นเวลากานจะเกิดปัญหาน่าวัยความจำข้าวคราวเต็ม ซึ่งจะต้องทำการเปิดโปรแกรมใหม่
- เมื่ออินเทอร์เน็ตมีปัญหา เช่น หลุด หรือ ไฟดับ จะทำให้โปรแกรมบนมอดูลกล้องนั้นค้างต้องมีการตรวจสอบอินเทอร์เน็ตทุกวัน
- การจ่ายไฟให้กับมอดูลกล้องจะต้องใช้ตัวปรับกระแสไฟที่มีกำลังการจ่ายไฟที่เพียงพอ เนื่องจากกำลังไฟที่น้อยไปจะทำให้ประสิทธิภาพของ Raspberry Pi ลดลง
- ในขั้นตอนการกดปุ่มกดตัวเลขเพื่อยืนยันตัวตน เมื่อผู้ใช้กดปุ่มค้างจะทำให้เลยืนยันตัวตนจะผิดพลาดไปด้วย และเมื่อหน่วยเวลาของปุ่มกดมากไปจะทำให้ผู้ใช้กรอกรหัสการยืนยันตัวตนผิดพลาด

#### 5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะเพื่อพัฒนาโครงการนี้ต่อไป มีดังนี้

- สามารถที่จะพัฒนาแบบจำลองการเรียนรู้รูปภาพใบหน้า โดยใช้กล้องวิธีการเรียนรู้ของเครื่องในรูปแบบอื่น ๆ เพื่อเพิ่มความแม่นยำในการระบุตัวตนขณะที่ใส่หน้ากากอนามัยสูงขึ้น
- ในการออกแบบบนมอดูลกล้องสามารถออกแบบให้กล่องเก็บ Raspberry Pi นั้นมีขนาดเล็กลงก็จะใช้พื้นที่ลดลงได้

3. บุ่มกดยืนยันตัวตนสามารถที่จะเปลี่ยนไปใช้บุ่มที่ใช้ระบบอินฟราเรด หรือ ระบบบีนยันแบบการซูจานวนนิวเพื่อปงบอกร่องรอยผ่านกล้อง (Hand Gesture Recognition) ก็จะลดการสัมผัสทำให้ลดการแพร่กระจายของโรคติดต่อ
4. ควรพัฒนาแบบจำลองการตรวจจับใบหน้าให้มีประสิทธิภาพมากขึ้น เนื่องจากเมื่อประสิทธิภาพมากขึ้นจะทำให้การตรวจจับใบหน้านั้นรวดเร็วยิ่งขึ้น
5. ควรพัฒนาวิธีการเพิ่มความชัดของรูปภาพใบหน้า เนื่องจากมีผลต่อความแม่นยำในการระบุตัวตน
6. ควรสำรวจพื้นที่ติดตั้งมอเตอร์กล้อง เนื่องจากต้องพิจารณาสภาพแสงของพื้นที่ติดตั้งเพื่อพัฒนาการปรับแต่งรูปภาพให้ส่วน หรือ มีด ซึ่งมีผลต่อความแม่นยำในการระบุตัวตน
7. สามารถใช้กล้องแบบไฟกัลลูมที่ เพื่อกำหนดระยะของการตรวจจับใบหน้าที่ชัดเจน ทำให้สามารถลดการความพร่ามัวของรูปภาพใบหน้าส่งผลให้ความแม่นยำในการระบุตัวตนสูงขึ้น

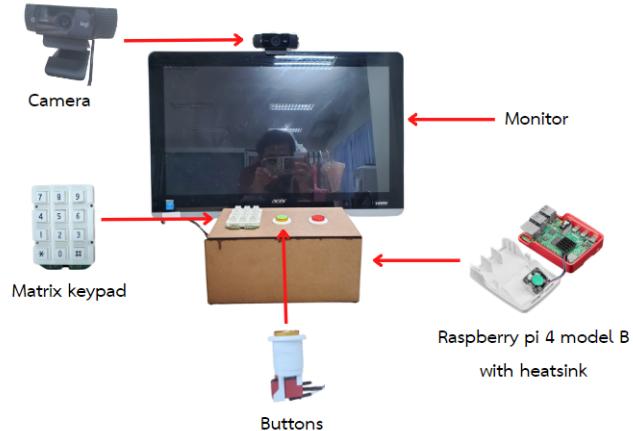
## บรรณานุกรม

- [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



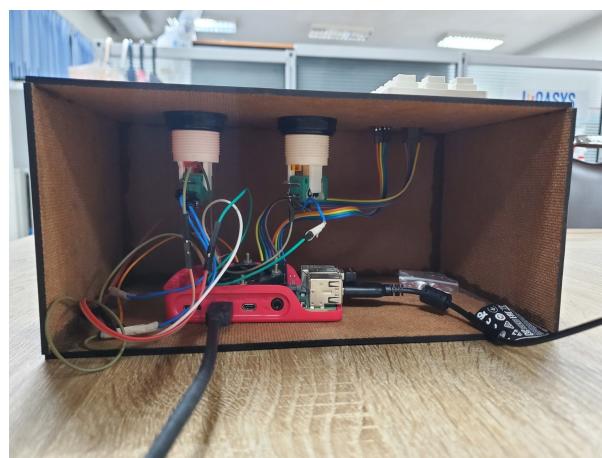
รูปที่ ก.1: อุปกรณ์ตรวจจับใบหน้า แสดงผลและรับผล



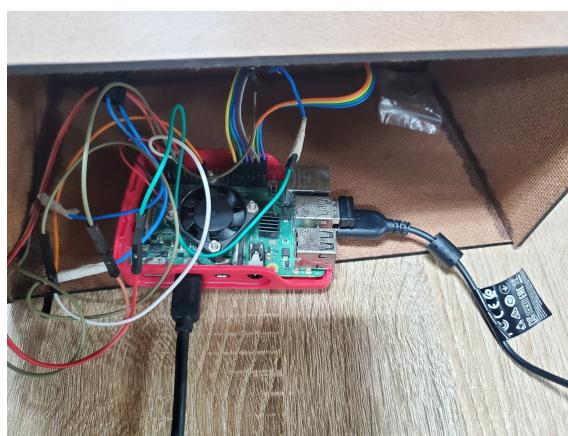
รูปที่ ก.2: กล้องที่ใช้ในการตรวจจับใบหน้าบุคคล



รูปที่ ก.3: ปุ่มกดยืนยันความถูกต้องของการระบุตัวตน

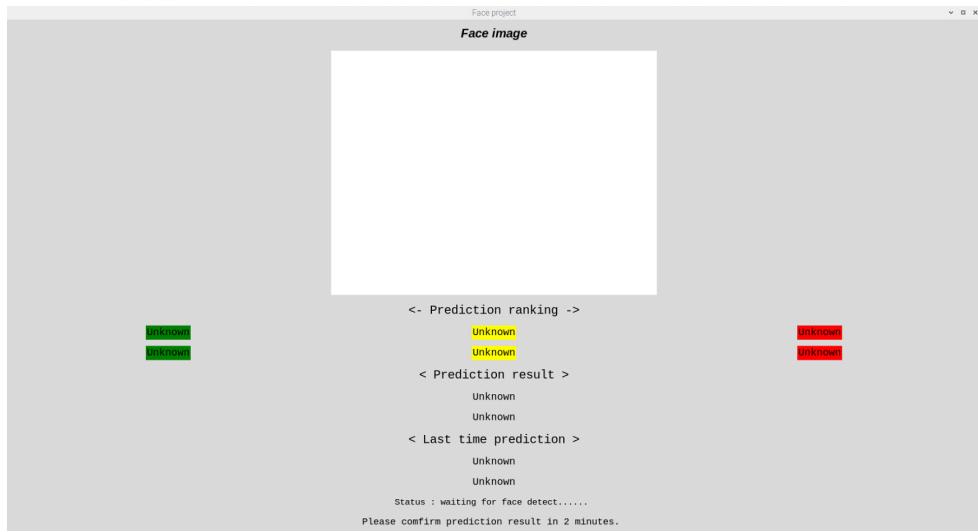


รูปที่ ก.4: ภายในมอดูลการตรวจจับใบหน้า และแสดงผล



รูปที่ ก.5: การเชื่อมต่อ Raspberry Pi

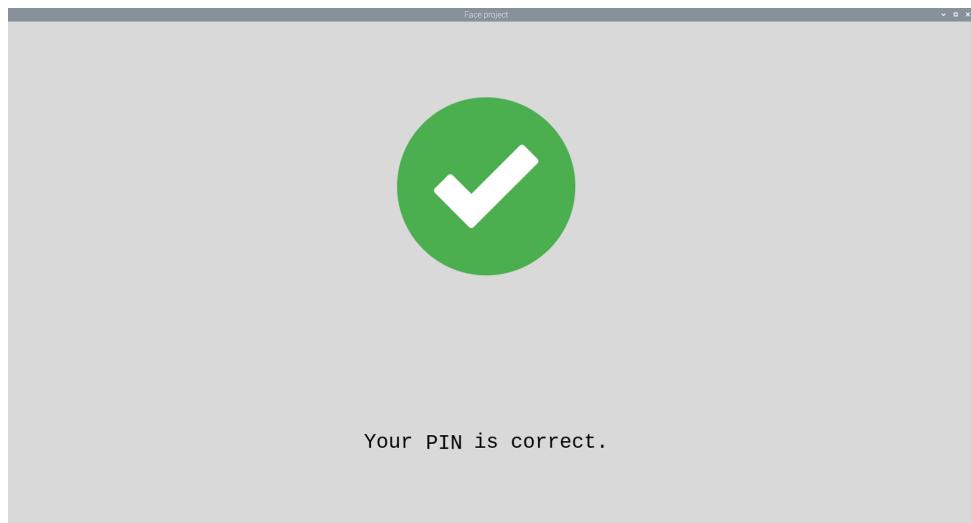
## ก.2 รูปภาพการแสดงผลบนหน้าจอ



รูปที่ ก.6: หน้า GUI เมื่อโปรแกรมเริ่มทำงาน



รูปที่ ก.7: หน้า GUI เมื่อผลลัพธ์การระบุตัวตนผิด



รูปที่ ก.8: หน้า GUI เมื่อผลลัพธ์รหัสยืนยันตัวตนถูกต้อง



รูปที่ ก.9: หน้า GUI เมื่อผลลัพธ์รหัสยืนยันตัวตนไม่ถูกต้อง

## ภาคผนวก ข

### คู่มือการใช้งานระบบ

โปรแกรมของโครงงานนี้ใช้ระบบปฏิบัติการของ Raspberry Pi คือ Raspbian buster และใช้ระบบปฏิบัติการของ Windows คือ Windows 10

#### ข.1 คู่มือการติดตั้งโปรแกรมเพื่อตรวจจับใบหน้าบน 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` จากนั้นรอจนกว่าจะเสร็จ 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` จากนั้นรอจนกว่าจะเสร็จ
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` จากนั้นรอจนกว่าจะเสร็จ

#### **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

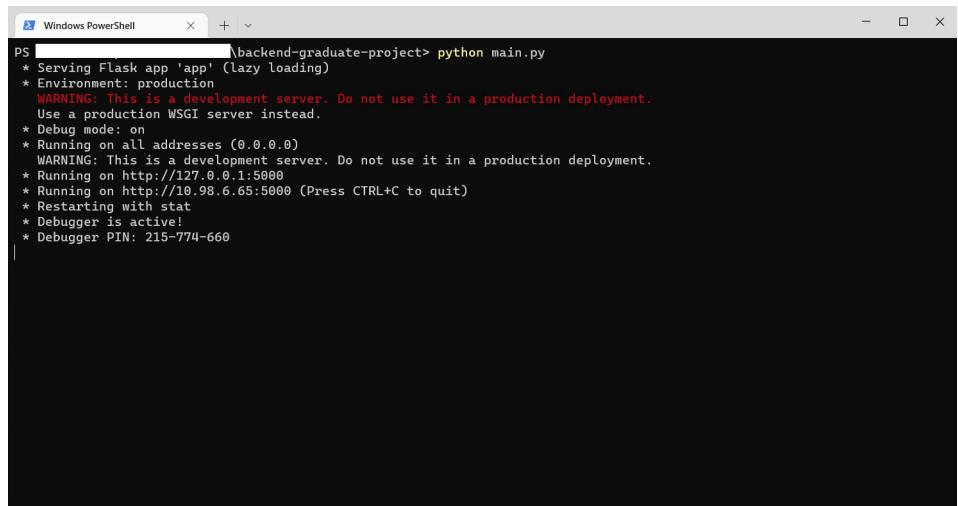


รูปที่ ข.1: หน้า GUI เมื่อโปรแกรมเริ่มทำงาน

### 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: หน้า Terminal เมื่อโปรแกรมระบุตัวตนด้วยใบหน้าเริ่มทำงาน

## ภาคผนวก ข

### เอกสารเกี่ยวกับข้อมูลส่วนบุคคล

ในโครงการนี้ได้มีการเก็บข้อมูลส่วนบุคคลของผู้ทดลอง จึงมีหนังสือขอความยินยอม เก็บรวบรวม ใช้ และเปิดเผยข้อมูลส่วนบุคคล เพื่อนำข้อมูลรูปภาพใบหน้า และข้อมาทำการเรียนรู้เพื่อสร้างแบบจำลอง (model)

#### ข.1 หนังสือขอความยินยอม เก็บรวบรวม ใช้ และเปิดเผยข้อมูลส่วนบุคคล

##### หนังสือขอความยินยอม เก็บรวบรวม ใช้ และเปิดเผยข้อมูลส่วนบุคคล

วันที่ .....  
เขียนที่ .....

ข้าพเจ้า นาย/นาง/นางสาว ..... เลขบัตรประจำตัวประชาชน ..... ให้ความยินยอมในการเก็บรวบรวม ใช้ และเปิดเผยข้อมูลส่วนบุคคลของข้าพเจ้า เพื่อนำไปใช้ในการระบุตัวตนเพื่อเข้าสู่ห้องกลุ่มวิจัยทฤษฎีและการประยุกต์ใช้การหาค่าที่เหมาะสมที่สุดในระบบทางวิศวกรรม (OASYS Research Group Optimization Theory and Applications for Engineering SYStems Research Group) หรือ (OASYS) โดยมีจุดประสงค์เพื่อใช้ในการเรียนรู้และถอดการระบุตัวตน (Face Recognition)

ทั้งนี้ ก่อนการแสดงเจตนา ข้าพเจ้าได้อ่านรายละเอียดจากเอกสารข้างข้อมูล และได้รับคำอธิบายจากผู้ขอเก็บข้อมูล ถึงวัตถุประสงค์ในการเก็บรวบรวม ใช้ เปิดเผย และ “ประมวลผล” ข้อมูลส่วนบุคคล และมีความเข้าใจดีแล้ว

ข้าพเจ้าให้ความยินยอมด้วยความสมัครใจ ปราศจากการบังคับ ช่มชู ล่อลง ชูชัก ซักจุ่ง และข้าพเจ้าทราบว่าข้าพเจ้าสามารถถอนความยินยอมนี้ตามกฎหมาย เว้นแต่ในกรณีมีข้อจำกัดสิทธิตามกฎหมายหรือตามเงื่อนไขข้อตกลงระหว่างข้าพเจ้ากับ (“ผู้ขอข้อมูล”) ในส่วนที่ยังเป็นประโยชน์กับข้าพเจ้า

กรณีที่ข้าพเจ้าประสงค์ที่จะถอนความยินยอม ข้าพเจ้าทราบว่าการถอนความยินยอมจะมีผลทำให้ข้าพเจ้าอาจไม่ได้รับความสะดวกในการใช้งานห้องกลุ่มงานวิจัยและไม่มีหลักฐานในการยืนยันการเข้าใช้งานห้องกลุ่มงานวิจัยได้ และข้าพเจ้าทราบว่าการถอนความยินยอมดังกล่าว ไม่มีผลต่อการประมวลผลข้อมูลส่วนบุคคลที่ได้ดำเนินการเสร็จสิ้นไปแล้วก่อนการถอนความยินยอม

ลงชื่อ .....  
(.....)

รูปที่ ข.1: หนังสือขอความยินยอม เก็บรวบรวม ใช้ และเปิดเผยข้อมูลส่วนบุคคล

## 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)