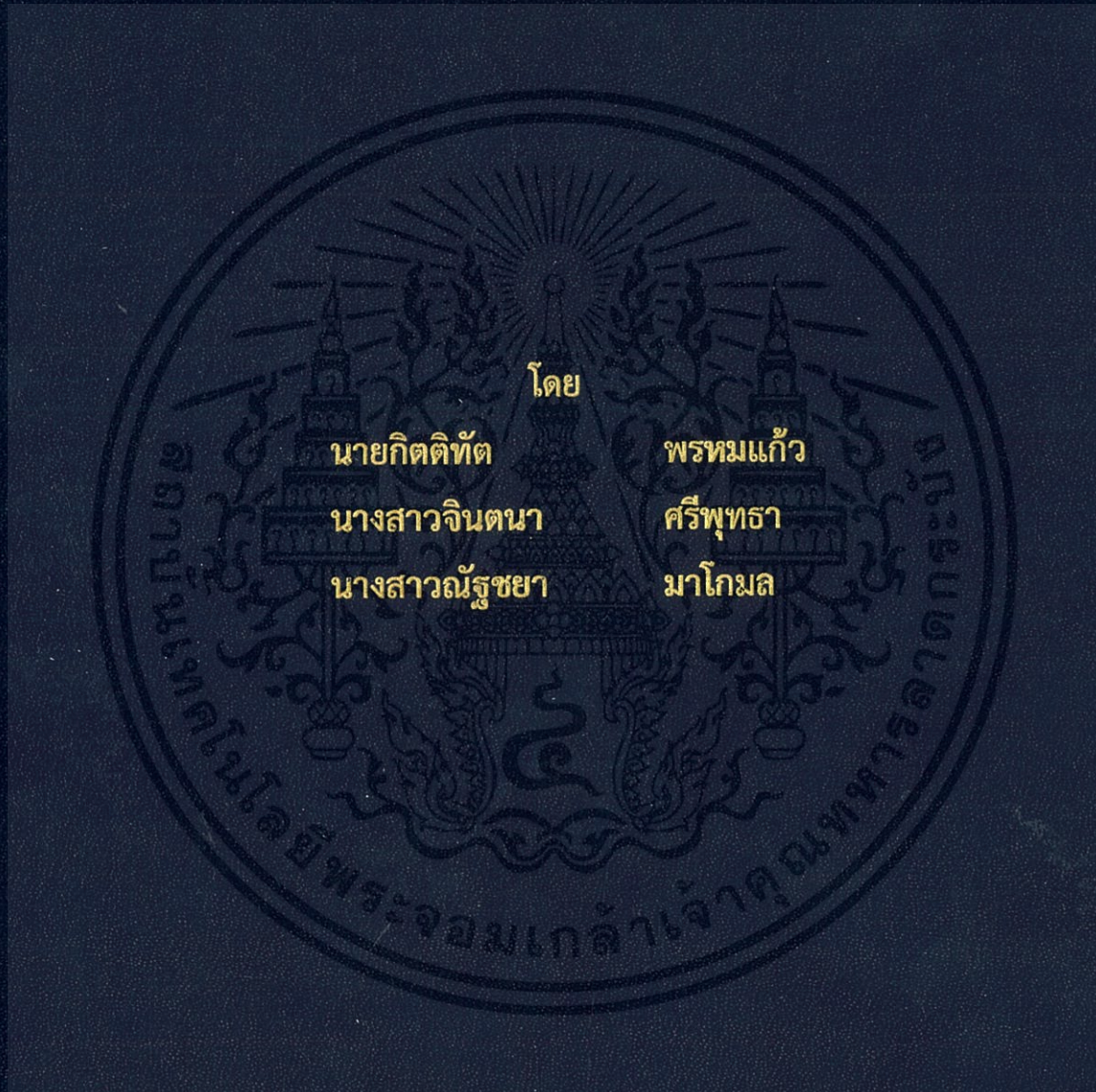


ระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ
AUTOMATIC STUDENT ATTENDANCE SYSTEM
USING FACE RECOGNITION



โดย
นายกิตติทัต พรหมแก้ว
นางสาวจินตนา ศรีพุทธา
นางสาวณัฐชยา มาโกมล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ
AUTOMATIC STUDENT ATTENDANCE SYSTEM
USING FACE RECOGNITION

โดย

| | | |
|--------------|----------|----------|
| นายกิตติทัต | พรหมแก้ว | 58010094 |
| นางสาวจินตนา | ศรีพุทธา | 58010179 |
| นางสาวณัฐชยา | มาโกมล | 58010372 |

อาจารย์ที่ปรึกษา

ดร.สมปอง วิเศษพานิชกิจ

อาจารย์ที่ปรึกษาร่วม

ผศ.ดร.นภัทร สระเอี่ยม

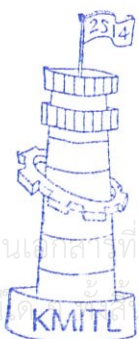
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561



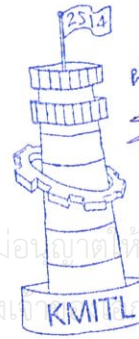
ผ่านการตรวจรูปเล่มแล้ว

(.....
.....)

อาจารย์ที่ปรึกษา

19/12/61

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

.....
.....
19/12/61

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามทำซ้ำหรือดัดแปลงเนื้อหา และต้องอ้างอิงถึง KMITE

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ

AUTOMATIC STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION

ผู้จัดทำ

- | | |
|--------------------------|----------|
| 1. นายกิตติทัต พรหมแก้ว | 58010094 |
| 2. นางสาวจินตนา ศรีพุทธา | 58010179 |
| 3. นางสาวณัฐชยา มาโกมล | 58010372 |


.....
(ดร.สมปอง วิเศษพานิชกิจ)

อาจารย์ที่ปรึกษา


.....
(ผศ.ดร.นภัทร สระเอี่ยม)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินโครงการฉบับนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ หากไม่ได้รับความอนุเคราะห์อย่างดียิ่งจาก ดร.สมปอง วิเศษพานิชกิจ อาจารย์ที่ปรึกษา และผศ.ดร.นภัทร สระเอี่ยม อาจารย์ที่ปรึกษาร่วม อีกทั้ง ผศ.ดร.ธเนศ พัฒนธาดาทพงษ์ อาจารย์ทั้งสามท่านที่ให้คำแนะนำและแนวทางการแก้ไขปัญหาลดตลอดระยะเวลาในการจัดทำโครงการนี้ รวมทั้งสนับสนุนและให้ความช่วยเหลือในด้านสถานที่ เครื่องมือและอุปกรณ์ต่างๆ ที่จำเป็นต้องใช้ในระหว่างการจัดทำโครงการ

ขอขอบคุณคณะอาจารย์ เพื่อนๆ และรุ่นพี่จากภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่คอยให้ความรู้คำแนะนำ อบรมสั่งสอน ชี้แนวทางที่ถูกต้อง ตลอดจนกำลังใจที่ดีแก่ข้าพเจ้าเสมอมา และที่สำคัญที่สุดข้าพเจ้าขอขอบพระคุณบิดา มารดา ครอบครัวของข้าพเจ้า ที่คอยดูแลเอาใจใส่ พร้อมสนับสนุนทางด้านการศึกษาแก่ข้าพเจ้ามาโดยตลอด รวมไปถึงคุณครูและอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาให้แก่ข้าพเจ้าจนมาเป็นข้าพเจ้าในทุกวันนี้

ท้ายที่สุดทางผู้จัดทำคาดหวังโครงการเล่มนี้จะมีประโยชน์แก่ทุกท่านที่สนใจและได้เข้ามาศึกษาไม่มากก็น้อย และถ้าโครงการเล่มนี้เกิดข้อบกพร่อง หรือความไม่สมบูรณ์ในส่วนใดส่วนหนึ่งของเนื้อหา ทางผู้จัดทำขออ้อมรับความผิดนี้แต่เพียงผู้เดียว อีกทั้งพร้อมรับทุกคำแนะนำเพื่อเป็นแนวทางในการพัฒนาโครงการอื่น ๆ ในโอกาสต่อไป

นายกิตติทัต พรหมแก้ว
นางสาวจินตนา ศรีพุทธา
นางสาวณัฐชยา มาโกมล
ผู้จัดทำ

ระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ
 AUTOMATIC STUDENT ATTENDANCE SYSTEM
 USING FACE RECOGNITION

| | | |
|-----|-----------------------|----------|
| โดย | นายกิตติทัต พรหมแก้ว | 58010094 |
| | นางสาวจินตนา ศรีพุทธา | 58010179 |
| | นางสาวณัฐชยา มาโกมล | 58010372 |

อาจารย์ที่ปรึกษา ดร.สมปอง วิเศษพานิชกิจ
 อาจารย์ที่ปรึกษาร่วม ผศ.ดร.นภัทร สระเอี่ยม

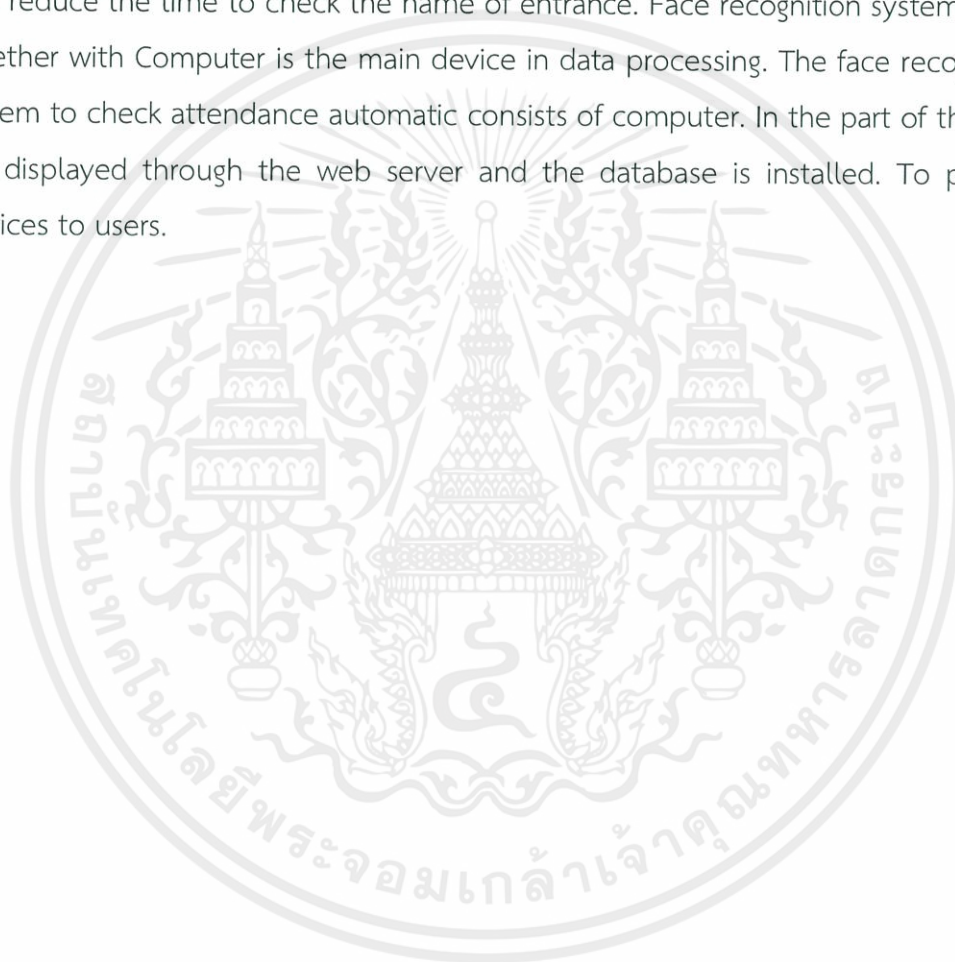
บทคัดย่อ

ในปัจจุบันมีเทคโนโลยีใหม่ๆ มากมายที่ถูกพัฒนาขึ้นเพื่อตอบสนองความต้องการของผู้ใช้งานให้ได้รับความสะดวกสบายมากยิ่งขึ้น โดยเฉพาะอย่างยิ่งทางด้านการศึกษา คงไม่สามารถปฏิเสธได้ว่าเราทุกคนเกิดมาสมควรได้รับการศึกษาทั้งในระดับประถมศึกษา มัธยมศึกษา จนไปถึงระดับมหาวิทยาลัย ซึ่งโครงการนี้ได้เล็งเห็นความสำคัญทางด้านการศึกษา ทางผู้จัดจึงได้ทำการพัฒนาระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ เพื่อเพิ่มความสะดวกและลดระยะเวลาในการเช็คชื่อเข้าเรียนโดยนำระบบรู้จำใบหน้ามาใช้งานร่วมกับคอมพิวเตอร์เป็นอุปกรณ์หลักในการประมวลผลข้อมูล ซึ่งระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติที่ได้ออกแบบนั้นประกอบไปด้วยอุปกรณ์ประมวลผลคอมพิวเตอร์ โดยในส่วนของผู้ใช้งานจะมีการแสดงผลผ่านเว็บเซิร์ฟเวอร์และมีการติดตั้งฐานข้อมูล เพื่อให้บริการแก่ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Nowadays, many new technologies have been developed to satisfy the needs of users to acquire the better conveniences, Especially in education. we could not refuse that We are all born deserve to study both in primary, secondary, and the University. This project has recognized the importance of education thus we have developed Face recognition system for auto attendance to increase the convenience and reduce the time to check the name of entrance. Face recognition system works together with Computer is the main device in data processing. The face recognition system to check attendance automatic consists of computer. In the part of the user are displayed through the web server and the database is installed. To provide services to users.



สารบัญ

| | หน้า |
|---|-----------|
| กิตติกรรมประกาศ | I |
| บทคัดย่อ | II |
| สารบัญ | IV |
| สารบัญรูป | VI |
| สารบัญตาราง | X |
| บทที่ 1 | |
| บทนำ | 1 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา | 1 |
| 1.2 วัตถุประสงค์ | 2 |
| 1.3 ขอบเขตของโครงการ | 2 |
| บทที่ 2 | |
| ทฤษฎีและหลักการที่เกี่ยวข้อง | 3 |
| 2.1 ระบบวิเคราะห์ใบหน้า | 3 |
| 2.2 หน่วยประมวลผลส่วนกลางของระบบ | 15 |
| 2.3 ภาษาคอมพิวเตอร์ส่วนประมวลผล | 17 |
| 2.4 ระบบฐานข้อมูล | 21 |
| บทที่ 3 | |
| การออกแบบและการจัดทำปริญญานิพนธ์ | 23 |
| 3.1 สถาปัตยกรรมการออกแบบ | 24 |
| 3.2 เครื่องมือที่ใช้ในการทดลอง | 37 |
| 3.3 การจัดเก็บผลการทดลอง | 39 |

สารบัญ (ต่อ)

| | หน้า |
|---|------|
| บทที่ 4 | |
| ผลการทดลอง | 41 |
| 4.1 ผลการทดสอบการเรียกใช้งานหน้าเว็บไซต์ | 41 |
| 4.2 ผลการทดสอบการเพิ่มรายวิชาและรายชื่อนักศึกษา | 42 |
| 4.3 ผลการทดสอบการบันทึกใบหน้านักศึกษา | 45 |
| 4.4 ผลการทดสอบการตรวจสอบใบหน้าของนักศึกษา | 59 |
| 4.5 ผลการทดสอบการเข้าเรียนของนักศึกษา | 61 |
| 4.6 ผลการทดสอบการตรวจประวัติการเข้าชั้นเรียนของนักศึกษา | 63 |
| บทที่ 5 | |
| สรุปผลและข้อเสนอแนะ | 66 |
| 5.1 สรุปผล | 66 |
| 5.2 ข้อเสนอแนะ | 67 |
| บรรณานุกรม | 68 |
| ภาคผนวก | 69 |

สารบัญรูป

| รูปที่ | หน้า | |
|--------|--|----|
| 1.1 | บล็อกไดอะแกรมของโครงการ | 2 |
| 2.1 | กล้องโทรศัพท์ที่มีฟังก์ชันการค้นหาใบหน้า | 4 |
| 2.2 | ภาพใบหน้าตัวอย่างถูกแปลงเป็นภาพขาวดำ | 5 |
| 2.3 | ภาพใบหน้าในหน่วยพิกเซลและพิกเซลล้อมรอบที่ต้องการเปรียบเทียบค่าความสว่าง | 5 |
| 2.4 | ลูกศรชี้ขึ้นทางขวาบนแสดงทิศทางพิกเซลที่มีค่าความสว่างที่น้อยกว่าของแต่ละพิกเซล | 5 |
| 2.5 | ภาพต้นฉบับถูกแปลงเป็นภาพแบบ HISTOGRAM OF ORIENTED GRADIENTS หรือ HOG ที่จับภาพคุณสมบัติหลักของภาพโดยไม่คำนึงถึงความสว่างของภาพ | 6 |
| 2.6 | ภาพใบหน้าบุคคลในรูปแบบ HOG ถูกสร้างมาจากภาพใบหน้าของบุคคลคนหลายๆ ภาพ ซึ่งหากมีความคล้ายกับในภาพที่มีใบหน้าบุคคลดังรูป จะทำให้สามารถตรวจจับหน้าภาพที่ต้องการได้ ดังกรอบสี่เหลี่ยม | 6 |
| 2.7 | ผลลัพธ์การค้นหาใบหน้าในภาพใบหน้าบุคคล | 7 |
| 2.8 | ใบหน้าของ WILL FERRELL ที่มีมุมหรือทิศทางที่ต่างกัน | 7 |
| 2.9 | ตัวอย่าง 68 จุดเฉพาะบนใบหน้า | 8 |
| 2.10 | ผลลัพธ์ของการวางตำแหน่งของ 68 จุดเฉพาะบนภาพใบหน้าตัวอย่าง | 8 |
| 2.11 | ภาพใบหน้าที่ถูกแปลงโดยอัลกอริทึม FACE LANDMARK ESTIMATION | 9 |
| 2.12 | ค่าระยะห่างระหว่างจุดต่างๆ บนใบหน้าในหน่วยเซนติเมตรและเปรียบเทียบความใกล้เคียงของค่าระยะห่างระหว่างจุดต่างๆ บนใบหน้าระหว่างสองใบหน้า | 9 |
| 2.13 | ขั้นตอนการทำ TRIPLET TRAINING | 11 |
| 2.14 | ค่าตัวเลขจำนวน 128 ค่าที่ถูกแปลงจากรูปภาพใบหน้า | 12 |
| 2.15 | เส้น GRADIENT DESCENT จากจุดสีน้ำเงินไปจุดสีแดง ซึ่งวิ่งเข้าจุดกลางวงกลม | 13 |

สารบัญญรูป (ต่อ)

| รูปที่ | หน้า |
|--|------|
| 2.16 STOCHASTIC GRADIENT DESCENT (SGD) | 14 |
| 2.15 คอมพิวเตอร์ ASUS K455LD-WX062D | 15 |
| 3.1 บล็อกไดอะแกรมการทำงานของระบบ | 23 |
| 3.2 การทำงานของระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียน | 24 |
| 3.3 ภาพรวมของระบบรู้จำใบหน้า | 25 |
| 3.4 ภาพรวมการใช้งานของระบบ | 27 |
| 3.5 โครงสร้างฐานข้อมูล | 27 |
| 3.6 รายการต่างๆ ที่เรียกใช้ผ่านเว็บเบราว์เซอร์ | 30 |
| 3.7 ขั้นตอนการใช้งานส่วนของการเพิ่มรายชื่อวิชาและรายชื่อนักศึกษา | 31 |
| 3.8 ขั้นตอนการใช้งานส่วนของการบันทึกและทดสอบใบหน้าของนักศึกษา | 32 |
| 3.9 ขั้นตอนการใช้งานส่วนของการเช็คชื่อนักศึกษา | 33 |
| 3.10 ขั้นตอนการใช้งานส่วนเสริมของการเช็คชื่อเข้าชั้นเรียนแบบปกติ | 34 |
| 3.11 ขั้นตอนการเรียกส่วนสรุปการเข้าชั้นเรียนรายคาบ | 35 |
| 3.12 ขั้นตอนการเรียกส่วนสรุปการเข้าชั้นเรียนรายวิชา | 36 |
| 3.13 แผนภาพการเรียกใช้งานระบบ | 37 |
| 4.1 หน้าจอพร้อมที่คำสั่งของการทำงาน | 41 |
| 4.2 หน้าเว็บไซต์หลักของระบบโดยที่ URL คือ HTTPS://202.28.100.128:5000 | 42 |
| 4.3 การเพิ่มรายวิชาและเพิ่มรายชื่อนักศึกษา | 42 |
| 4.4 ข้อมูลภายในไฟล์ EXCEL ที่ต้องการจะเพิ่มรายชื่อนักศึกษา | 43 |
| 4.5 ผลการเพิ่มรายชื่อนักศึกษาภายในรายวิชาใหม่ | 43 |
| 4.6 ก่อนการทดสอบเพิ่มรายชื่อนักศึกษาเข้ารายวิชา ABC02 | 44 |
| 4.7 ผลการเพิ่มรายชื่อนักศึกษาในรายวิชา ABC02 | 44 |
| 4.8 รายชื่อวิชาทั้งหมดที่อยู่ในระบบ | 45 |
| 4.9 เว็บไซต์หน้าจัดการรายชื่อและบันทึกใบหน้า | 46 |

สารบัญรูป (ต่อ)

| รูปที่ | หน้า |
|--|------|
| 4.10 การกรอกรหัสนักศึกษาเพื่อการบันทึกใบหน้า | 46 |
| 4.11 การบันทึกใบหน้าของนักศึกษารหัส 58010044 | 47 |
| 4.12 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010044 | 47 |
| 4.13 การบันทึกใบหน้าของนักศึกษารหัส 58010088 | 48 |
| 4.14 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010088 | 48 |
| 4.15 การบันทึกใบหน้าของนักศึกษารหัส 58010094 | 49 |
| 4.16 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010094 | 49 |
| 4.17 การบันทึกใบหน้าของนักศึกษารหัส 58010238 | 50 |
| 4.18 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010238 | 50 |
| 4.19 การบันทึกใบหน้าของนักศึกษารหัส 58010256 | 51 |
| 4.20 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010256 | 51 |
| 4.21 การบันทึกใบหน้าของนักศึกษารหัส 58010324 | 52 |
| 4.22 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010324 | 52 |
| 4.23 การบันทึกใบหน้าของนักศึกษารหัส 58010372 | 53 |
| 4.24 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010372 | 53 |
| 4.25 การบันทึกใบหน้าของนักศึกษารหัส 58010438 | 54 |
| 4.26 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010438 | 54 |
| 4.27 การบันทึกใบหน้าของนักศึกษารหัส 58010564 | 55 |
| 4.28 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010564 | 55 |
| 4.29 การบันทึกใบหน้าของนักศึกษารหัส 58010956 | 56 |
| 4.30 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010956 | 56 |
| 4.31 สถานะสำเร็จการบันทึกใบหน้าของนักศึกษา | 57 |
| 4.32 การทดสอบใบหน้าของนักศึกษา | 58 |
| 4.33 ตัวเลือกการบันทึกใบหน้าของนักศึกษา | 58 |
| 4.34 การสร้างคาบเรียนของรายวิชา ABC02 | 61 |

สารบัญญรูป (ต่อ)

| รูปที่ | หน้า | |
|--------|---|----|
| 4.35 | การเลือกคาบเรียนและรายวิชาเรียน | 61 |
| 4.36 | การเช็คชื่อเข้าชั้นเรียนของรายวิชา ABC02 | 62 |
| 4.37 | การเช็คชื่อเข้าชั้นเรียนแบบปกติของรายวิชา ABC02 | 62 |
| 4.38 | ผลการเช็คชื่อเข้าชั้นเรียนของรายวิชา ABC02 | 63 |
| 4.39 | ประวัติการเข้าชั้นเรียนรายคาบของรายวิชา ABC01 | 63 |
| 4.40 | ประวัติการเข้าชั้นเรียนรายคาบของรายวิชา ABC02 | 64 |
| 4.41 | การเลือกรายวิชาเพื่อเรียกดูประวัติการเข้าชั้นเรียนรายวิชา | 64 |
| 4.42 | ประวัติการเข้าชั้นเรียนรายวิชา ABC02 | 65 |

สารบัญตาราง

| ตารางที่ | หน้า | |
|----------|---|----|
| 2.1 | คุณสมบัติของคอมพิวเตอร์ ASUS K455LD-WX062D | 16 |
| 3.1 | รายชื่อตารางในฐานข้อมูล | 28 |
| 3.2 | รายละเอียดตารางข้อมูลนักศึกษา | 28 |
| 3.3 | รายละเอียดตารางข้อมูลรายวิชา | 28 |
| 3.4 | รายละเอียดตารางข้อมูลนักศึกษาในรายวิชา | 28 |
| 3.5 | รายละเอียดตารางข้อมูลคาบเรียนในรายวิชา | 29 |
| 3.6 | รายละเอียดตารางข้อมูลการลงบันทึกเวลาของนักศึกษา | 29 |
| 3.7 | การทำงานของระบบในแต่ละกลุ่ม | 30 |
| 3.8 | ผลการทดสอบการตรวจจับใบหน้าของนักศึกษา | 59 |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้อาจกล่าวได้ว่า เทคโนโลยีเข้ามามีบทบาทในชีวิตประจำวันเพิ่มขึ้น สามารถตอบสนองความต้องการในการดำเนินชีวิต มีส่วนช่วยพัฒนาคุณภาพชีวิตของมนุษย์สะดวกสบาย รวดเร็วมากยิ่งขึ้น ซึ่งในช่วงหลายปีที่ผ่านมา การระบุตัวตนเป็นหนึ่งในหัวข้อที่ได้รับการสนใจเป็นอย่างมาก ทั้งในด้านของการศึกษาวิจัย ค้นคว้า และการพัฒนาระบบออกมาใช้งานจริง โดยการระบุตัวตนในองค์กรต่าง ๆ มักจะใช้บัตรประจำตัวหรือผู้ใช้และรหัสผ่านในการระบุตัวตน ซึ่งวิธีเหล่านี้อาจมีข้อบกพร่องในบางประการ เช่น การถูกสวมรอยจากผู้ที่ไม่ใช่เจ้าของบัตร เป็นต้น อย่างไรก็ตามข้อมูลทางกายภาพของมนุษย์จึงถือเป็นอีกทางเลือกหนึ่งที่สามารถนำมาใช้ในการตรวจสอบ และระบุตัวบุคคลได้เป็นอย่างดี ตัวอย่างเช่น ลายนิ้วมือ ม่านตา หรือใบหน้า เป็นต้น โดยใบหน้าของมนุษย์ก็เป็นหนึ่งในลักษณะทางกายภาพที่เหมาะสมสำหรับใช้ตรวจสอบการระบุตัวบุคคล เนื่องจากสามารถมองเห็นได้ง่าย ดังนั้นการรู้จำใบหน้าจึงเป็นอีกหนึ่งวิธีที่สามารถประยุกต์ใช้กับการตรวจสอบและระบุตัวตนบุคคล

ในขณะเดียวกัน พบว่าการตรวจสอบจำนวนผู้เข้าเรียนนักเรียนนักศึกษาในสถานศึกษายังคงใช้รูปแบบของกระดาษ เพื่อเป็นข้อมูลในการให้คะแนนแก่ผู้เรียนที่ได้เข้าเรียนในแต่ละรายวิชาตามที่สถาบันกำหนดไว้หรืออื่นๆ ซึ่งปกติแล้วการบันทึกการเข้าชั้นเรียนมักจะใช้การชื่อนผู้เรียนทีละคน ซึ่งวิธีการนี้จะใช้เวลานานในชั้นเรียนที่มีผู้เรียนจำนวนมากและอาจขาดความแม่นยำ ไม่ว่าจะเป็นการไม่ได้ยินเสียงตอบรับ มองไม่เห็นผู้เรียนยกมือหรือมีการยกมือแทนกัน และการบันทึกการเข้าเรียนด้วยวิธีการเหล่านี้อาจทำให้ระยะเวลาของการเรียนการสอนลดลง และผู้สอนอาจทำการสอนไม่ทันหรือต้องเพิ่มเวลาการสอนมากขึ้น ด้วยเหตุนี้การพัฒนาระบบบันทึกการเข้าเรียนโดยใช้การตรวจจับใบหน้า จึงเป็นส่วนหนึ่งเพื่อช่วยให้การบันทึกเข้าชั้นเรียนมีประสิทธิภาพมากขึ้น

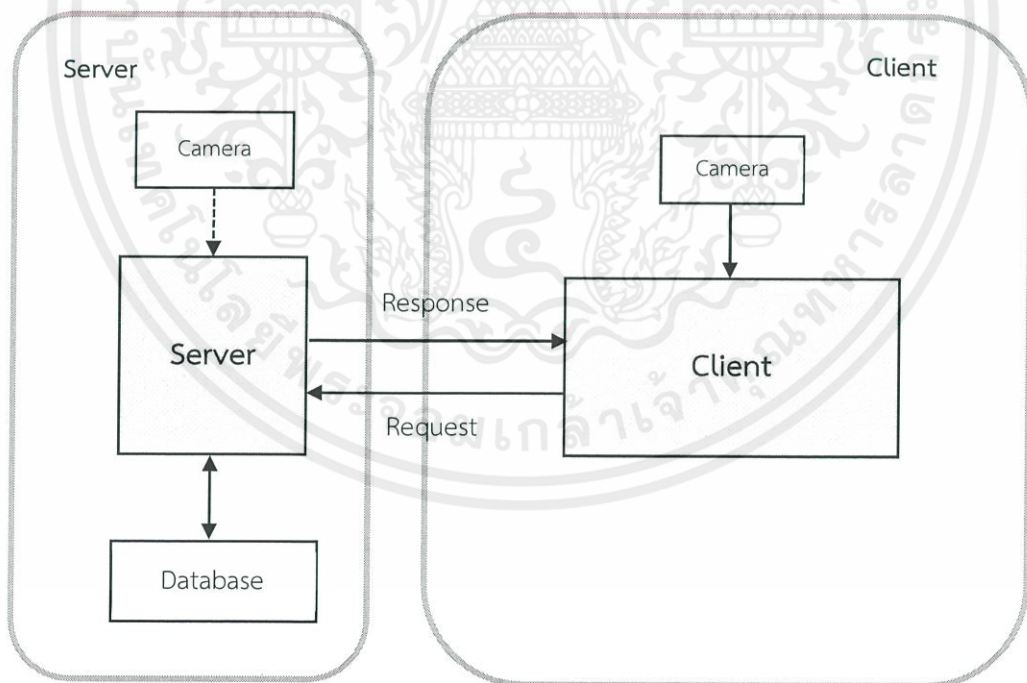
ดังนั้น โครงการนี้ได้เล็งเห็นความสำคัญของการบันทึกเวลาเข้าเรียนในแต่ละรายวิชา จึงทำการสร้างระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ โดยใช้หลักการตรวจจับใบหน้าและรู้จำใบหน้าบุคคล (Face Detection and Recognition) เพื่ออำนวยความสะดวกรวดเร็ว และประหยัดเวลาในการตรวจสอบจำนวนนักเรียนนักศึกษา

1.2 วัตถุประสงค์

- 1) ศึกษาองค์ความรู้ของระบบรู้จำใบหน้า (Face Detection and Recognition)
- 2) ออกแบบและนำระบบรู้จำใบหน้ามาประยุกต์ใช้ เพื่อสร้างระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ โดยใช้กล้องตรวจจับใบหน้า
- 3) ออกแบบระบบที่เพิ่มความสะดวกและลดระยะเวลาในการเช็คชื่อเข้าห้องเรียน
- 4) ศึกษาส่วนประสานกราฟิกกับผู้ใช้งาน และสร้างฐานข้อมูลเพื่อเก็บข้อมูลผู้เรียน
- 5) ออกแบบเว็บไซต์เพื่อแสดงผลข้อมูลที่ได้จากระบบรู้จำใบหน้าในการตรวจสอบการเข้าชั้นเรียนอัตโนมัติบนเว็บไซต์

1.3 ขอบเขตของปัญญานิพนธ์

- 1) สร้างระบบรู้จำใบหน้า (Face Detection and Recognition) เพื่อเช็คชื่อเข้าเรียน
- 2) สร้างระบบซึ่งสามารถตรวจจับใบหน้าได้จำนวนมาก และมีความแม่นยำ
- 3) สร้างเว็บไซต์ให้ผู้ใช้งานสามารถเช็คชื่อนักศึกษา และตรวจเช็คผลการเช็คชื่อได้



รูปที่ 1.1 บล็อกไดอะแกรมของโครงการ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติในครั้งนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่ออำนวยความสะดวก และลดระยะเวลาในการตรวจสอบจำนวนผู้เรียนที่เข้าเรียนในแต่ละรายวิชา ทำให้เวลาสำหรับการเรียนการสอนในคาบเรียนมีประสิทธิภาพยิ่งขึ้น ซึ่งระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติประกอบไปด้วย กล้องตรวจจับใบหน้าบุคคล อุปกรณ์ประมวลผลข้อมูล และชุดประมวลผล โดยเชื่อมโยงกับฐานข้อมูล และเรียกใช้งานผ่านเครือข่ายอินเทอร์เน็ต การทำงานของระบบนี้จะตรวจจับใบหน้าของผู้เรียนผ่านกล้อง และประมวลผลบนเครื่องแม่ข่าย แสดงผลแบบเรียลไทม์ (Real-time) เพื่อยืนยันตัวบุคคลโดยแสดงรหัสผู้เรียนที่กรอบตรวจจับใบหน้าของผู้เรียน และส่งข้อมูลที่บันทึกได้ไปแสดงผลบนเว็บไซต์เกี่ยวกับข้อมูลการเข้าเรียนของผู้เรียน จากที่กล่าวมาข้างต้นมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 ระบบวิเคราะห์ใบหน้า

ระบบวิเคราะห์ใบหน้าถือว่าเป็นหนึ่งในระบบที่ใช้ในการพิสูจน์ยืนยันตัวตนบุคคลโดยใช้คุณลักษณะจำเพาะทางสรีระ (BIOMETRIC) หลักการทำงานของระบบรู้จำใบหน้าใน Machine Learning โดยเบื้องต้นจะทำการค้นหาใบหน้าทั้งหมดในภาพ โฟกัสที่ใบหน้าแต่ละหน้าและสามารถเข้าใจได้ว่าเป็นบุคคลเดียวกัน แม้ว่าใบหน้าจะหันไปในทิศทางอื่นๆ หรือในสภาพแสงที่ไม่ดี จากนั้นสามารถเลือกคุณสมบัติที่เป็นเอกลักษณ์ของใบหน้าเพื่อสามารถใช้เพื่อแยกแยะความแตกต่างจากใบหน้าบุคคลอื่นๆ เช่นดวงตามีขนาดใหญ่เท่าไร ใบหน้ายาวเท่าไร เป็นต้น สุดท้ายเปรียบเทียบคุณสมบัติที่เป็นเอกลักษณ์ของใบหน้านั้นกับบุคคลอื่น เพื่อกำหนดชื่อของบุคคลนั้น

ในความเป็นจริงสมองของมนุษย์สามารถทำสิ่งนี้โดยอัตโนมัติและทันทีในประจำวัน โดยขณะที่คอมพิวเตอร์ไม่สามารถทำได้ ดังนั้นมนุษย์จึงใช้วิธีการให้คอมพิวเตอร์เกิดการเรียนรู้และเข้าถึงวิธีการทำแต่ละขั้นตอนในกระบวนการรู้จำหรือจำแนกภาพใบหน้าจากกัน

ระบบรู้จำใบหน้านั้นถูกพัฒนาอย่างต่อเนื่องเป็นเวลาหลายสิบปี ส่งผลให้มีการพัฒนาอัลกอริทึมในการทำงานของระบบออกมาในลักษณะหลายรูปแบบแตกต่างกันไปตามยุคสมัยด้วย อันเนื่องมาจากปัจจัยด้านองค์ความรู้และเทคโนโลยีของอุปกรณ์ต่างๆ ที่พัฒนาขึ้นให้มีความเหมาะสมที่จะนำมาใช้ในระบบจึงทำให้ต้องออกแบบอัลกอริทึมใหม่ให้เหมาะสมกับอุปกรณ์ใหม่ๆ ด้วย ในปัจจุบัน

ระบบรู้จำใบหน้าได้มีการพัฒนาไปอย่างมาก ทำให้ระบบรู้จำใบหน้ามีความน่าเชื่อถือมากขึ้น จนมีการนำระบบรู้จำใบหน้ามาใช้ประโยชน์กันอย่างแพร่หลายในหลายประเทศ โดยได้นำระบบรู้จำใบหน้ามาติดตั้งในสนามบินเพื่อป้องกันคนร้ายหนีเข้าออกนอกประเทศ และมีระบบรู้จำใบหน้าสำหรับการยืนยันตัวคนร้ายในคดีต่างๆ ด้วย

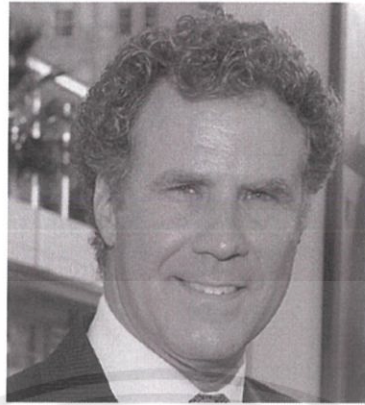
2.1.1 การค้นหาใบหน้าทั้งหมด (Face Detection)

ระบบตรวจจับใบหน้าจะทำการค้นหาใบหน้าของมนุษย์จากกล้องวิดีโอหรือกล้องตรวจจับที่ได้ติดตั้งไว้ โดยหลักการที่นำมาใช้นั้น เป็นหลักการของ Paul Viola และ Michael Jones เป็นการประยุกต์หลักการ Haar Feature สำหรับการตรวจจับใบหน้า ซึ่งได้คิดค้นและตีพิมพ์ในปี ค.ศ. 2001 โดยใช้ชื่อหลักการว่า “Viola-Jones” เป็น Machine Learning ซึ่งจะทำให้คอมพิวเตอร์สามารถเรียนรู้ว่าส่วนใดเป็นใบหน้ามนุษย์ จากการป้อนและจดจำภาพจำนวนมากเข้าสู่คอมพิวเตอร์ โดยภาพที่ป้อนจะเป็น “ภาพเชิงบวก (ภาพที่มีใบหน้า)” และ “ภาพเชิงลบ (ภาพที่ไม่มีใบหน้า)” เพื่อให้ “ตัวจำแนก (Classifier)” สามารถแยกแยะได้ว่าภาพต่อไปภาพใดบ้างที่มีใบหน้าอยู่



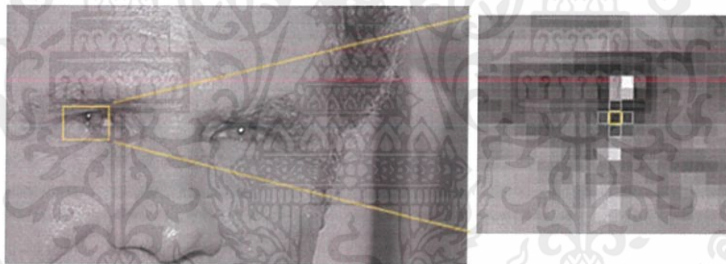
รูปที่ 2.1 กล้องโทรศัพท์ที่มีฟังก์ชันการค้นหาใบหน้า

Paul viola และ Michael J. Jones ได้นำเสนอเทคนิคการตรวจจับใบหน้าที่มีความเร็วและมีความถูกต้องในการตรวจจับสูง ในปี 2001 เทคนิคการตรวจจับใบหน้าของ Viola-Jones เป็นเทคนิคที่ได้รับการยอมรับและรู้จักในงานวิจัยเรื่องการตรวจจับใบหน้าเป็นอย่างมาก อย่างไรก็ตามมีวิธีแก้ไขปัญหาที่เชื่อถือได้มากหลายวิธี เราเลือกได้ใช้วิธีการที่คิดค้นในปี 2005 เรียกว่า Histogram of Oriented Gradients หรือ HOG ในการค้นหาใบหน้าในภาพ จะเริ่มต้นด้วยการทำให้ภาพเป็นขาวดำเพราะข้อมูลสีไม่จำเป็นในการค้นหาใบหน้า

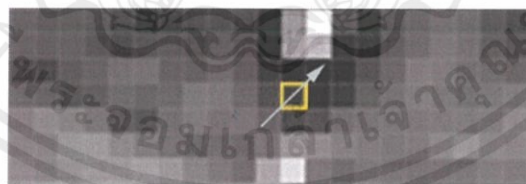


รูปที่ 2.2 ภาพใบหน้าตัวอย่างถูกแปลงเป็นภาพขาวดำ

จากนั้นดูทุกพิกเซลในภาพทีละภาพทุกพิกเซลและพิกเซลที่ล้อมรอบ เป้าหมายคือการหาค่าความสว่างของพิกเซลปัจจุบันเปรียบเทียบกับพิกเซลที่อยู่รอบ ๆ และวาดลูกศรเพื่อแสดงทิศทางความมืดของแต่ละพิกเซลในภาพ สังเกตจากรูปที่ 2.4 ได้ว่าลูกศรแสดงทิศทางไปยังพิกเซลที่มีค่าความสว่างน้อยกว่า



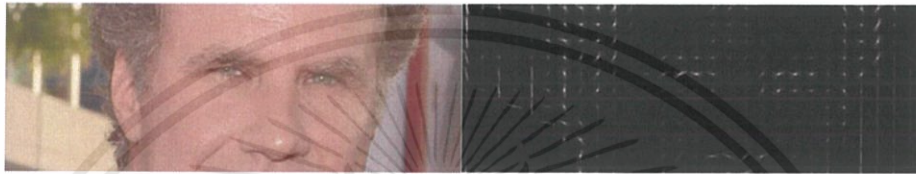
รูปที่ 2.3 ภาพใบหน้าในหน่วยพิกเซลและพิกเซลล้อมรอบที่ต้องการเปรียบเทียบค่าความสว่าง



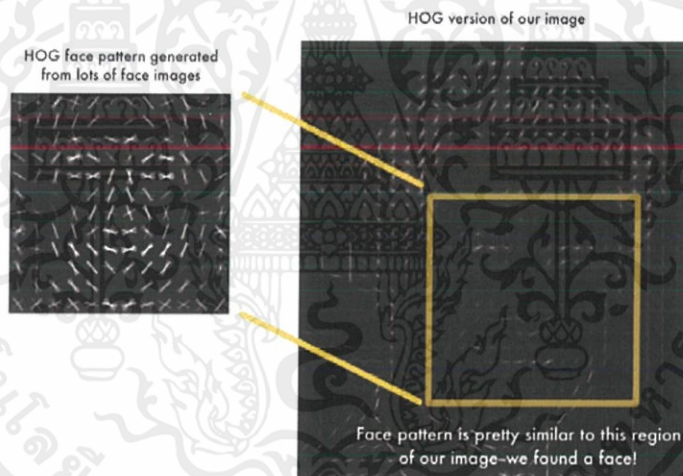
รูปที่ 2.4 ลูกศรชี้ขึ้นทางขวาบนแสดงทิศทางพิกเซลที่มีค่าความสว่างที่น้อยกว่าของแต่ละพิกเซล

หากทำซ้ำขั้นตอนดังกล่าวสำหรับทุกพิกเซลในภาพจะพบว่าทุกพิกเซลถูกแทนที่ด้วยลูกศร ลูกศรเหล่านี้เรียกว่า gradients หรือ การไล่ระดับแสง และมันจะแสดงการไหลของแสงจากแสงที่มีค่าความสว่างมากไปน้อยทั่วทั้งภาพ หากวิเคราะห์ที่พิกเซลของภาพใบหน้าบุคคลเดียวกันที่มีมืดมากและภาพที่มีแสงมาก จะมีค่าพิกเซลที่แตกต่างกันโดยสิ้นเชิง แต่เมื่อพิจารณาจากทิศทางที่ความ

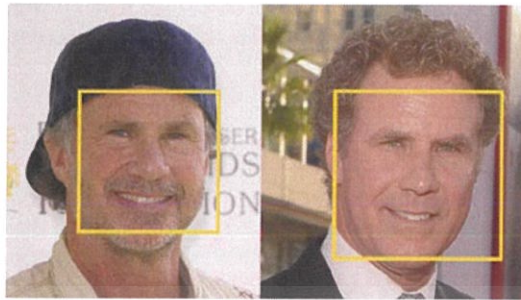
สว่างเปลี่ยนไปทั้งภาพที่มีมืดและภาพที่สว่าง จะเห็นได้ว่าภาพมีลักษณะทิศทางที่เหมือนกัน นั่นทำให้ปัญหาต่างๆ ถูกแก้ไขได้ง่ายมากขึ้น ในการทำเช่นนี้เราจะแยกภาพออกเป็นสี่เหลี่ยมเล็ก ๆ ขนาด 16x16 พิกเซล ในแต่ละช่องจะถูกลับจำนวนลูกศรการไล่ระดับแสงในทิศทางหลัก ได้แก่ ลูกศรที่ชี้ขึ้น ชี้ขึ้นเฉียงทางขวา ชี้ทางขวา ฯลฯ เป็นต้น จากนั้นในช่องสี่เหลี่ยมนั้นจะถูกแทนที่ด้วยลูกศรที่มีแนวโน้มของทิศทางมากที่สุด



รูปที่ 2.5 ภาพต้นฉบับถูกแปลงเป็นภาพแบบ Histogram of Oriented Gradients หรือ HOG ที่จับภาพคุณสมบัติหลักของภาพโดยไม่คำนึงถึงความสว่างของภาพ



รูปที่ 2.6 ภาพใบหน้าบุคคลในรูปแบบ HOG ถูกสร้างมาจากภาพใบหน้าของบุคคลคนหลายๆ ภาพ ซึ่งหากมีความคล้ายกับในภาพที่มีใบหน้าบุคคลดังรูป จะทำให้สามารถตรวจจับหน้าภาพที่ต้องการได้
ดังกรอบสี่เหลี่ยม



รูปที่ 2.7 ผลลัพธ์การค้นหาใบหน้าในภาพใบหน้าบุคคล

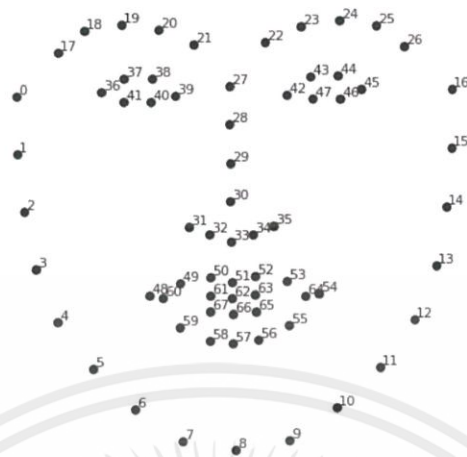
2.1.2 ทิศทางของใบหน้า (Posing and Projecting Faces)

เมื่อสามารถจำแนกใบหน้าจากรูปภาพได้แล้ว ยังมีอีกหนึ่งปัญหาคือใบหน้าที่หันไปในทิศทางที่ต่างกัน

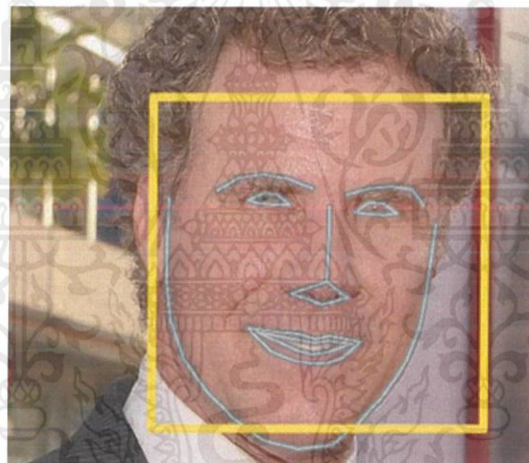


รูปที่ 2.8 ใบหน้าของ Will Ferrell ที่มีมุมหรือทิศทางที่ต่างกัน

สมองของมนุษย์สามารถจดจำได้อย่างง่ายดายว่าภาพทั้งสองนั้นเป็นบุคคลเดียวกัน แต่คอมพิวเตอร์จะเห็นภาพเหล่านี้เป็นคนสองคนที่แตกต่างกันโดยสิ้นเชิง ในการวิเคราะห์ได้เลือกใช้ อัลกอริทึมที่เรียกว่าการประมาณค่าจุดเด่นบนใบหน้าหรือ Face Landmark Estimation ซึ่งถูกคิดค้นในปี 2014 โดย Vahid Kazemi และ Josephine Sullivan ด้วยแนวคิดพื้นฐานคือมี 68 จุดเฉพาะ หรือ 68 landmarks อยู่บนทุกหน้าของมนุษย์เช่น ด้านบนคาง ขอบด้านบนของตา ขอบด้านบนของคิ้วแต่ละข้าง ฯลฯ เป็นต้น จากนั้นจึงทำการเทรน (Train) ให้อัลกอริทึมเกิดการเรียนรู้เพื่อสามารถค้นหา 68 จุดเฉพาะเหล่านี้บนใบหน้าทุกใบหน้าที่เจอได้

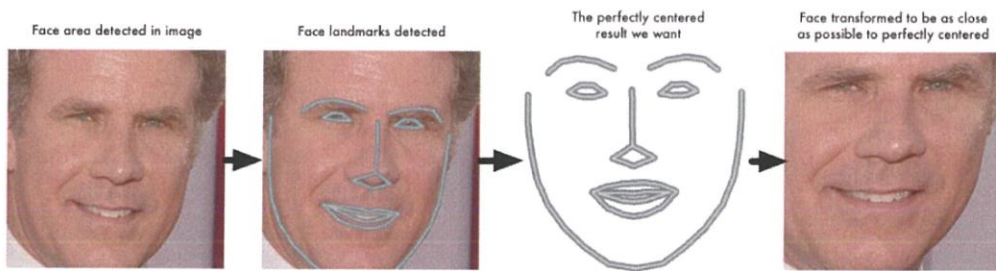


รูปที่ 2.9 ตัวอย่าง 68 จุดเฉพาะบนใบหน้า



รูปที่ 2.10 ผลลัพธ์ของการวางตำแหน่งของ 68 จุดเฉพาะบนภาพใบหน้าตัวอย่าง

เมื่อคอมพิวเตอร์เรียนรู้การค้นหภาพใบหน้าและทำการสร้าง Face Landmarks หรือ 68 จุดเฉพาะ อัลกอริทึมจะทำการแปลงให้ตาและปากจัดอยู่กึ่งกลางที่สุดเท่าที่จะทำได้ ในภาพใบหน้าผลลัพธ์เป็นเพียงการหมุนและปรับขนาดภาพดวงตาและปากเท่านั้น



รูปที่ 2.11 ภาพใบหน้าที่ถูกแปลงโดยอัลกอริทึม Face Landmark Estimation

ไม่ว่าใบหน้าจะเปลี่ยนไปอย่างไร การจัดตำแหน่งดวงตาและปากให้อยู่ในตำแหน่งเดียวกับภาพ จะทำให้ขั้นตอนต่อไปมีความแม่นยำยิ่งขึ้น

2.1.3 การเข้ารหัสภาพใบหน้า (Encoding Faces)

วิธีที่ง่ายที่สุดในการรู้จำใบหน้า (Face Recognition) คือการเปรียบเทียบใบหน้าที่ไม่รู้จักที่ได้จากขั้นตอนการทำ Face Landmarks กับรูปภาพของบุคคลทั้งหมดที่ถูกระบุตัวตนแล้ว เมื่อพบว่าใบหน้าที่ถูกระบุตัวตนแล้ว มีลักษณะคล้ายกับใบหน้าที่ไม่รู้จักมาก จึงสรุปได้ว่าเป็นคนเดียวกัน แต่นั่นเป็นปัญหาใหญ่ อย่างเว็บไซต์ Facebook ที่มีผู้ใช้หลายพันล้านคนและรูปภาพนับล้านภาพ จึงไม่สามารถวนซ้ำขั้นตอนดังกล่าวไปมาได้ทุกใบหน้าที่ติดแท็กหรือถูกระบุตัวตนแล้วก่อนหน้านี้เพื่อเปรียบเทียบกับทุกภาพที่เพิ่งอัปโหลดใหม่ ซึ่งนั่นใช้เวลานานเกินไป พวกเขาต้องการความสามารถในการรู้จำใบหน้าในหน่วยมิลลิวินาทีไม่ใช่ชั่วโมง

สิ่งที่น่าสนใจคือวิธีในการวัดระยะจุดบนใบหน้าแต่ละใบหน้า ตัวอย่างเช่น ขนาดของหู แต่ละข้าง ระยะห่างระหว่างดวงตาทั้งสองข้าง ความยาวของจมูก ฯลฯ โดยทำการวัดใบหน้าที่ไม่รู้จักที่ต้องการระบุตัวตนและค้นหาใบหน้าที่รู้จักด้วยผลลัพธ์ค่าการวัดที่ใกล้เคียงที่สุด



รูปที่ 2.12 ค่าระยะห่างระหว่างจุดต่างๆ บนใบหน้าในหน่วยเซนติเมตรและเปรียบเทียบความใกล้เคียงของค่าระยะห่างระหว่างจุดต่างๆ บนใบหน้าระหว่างสองใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

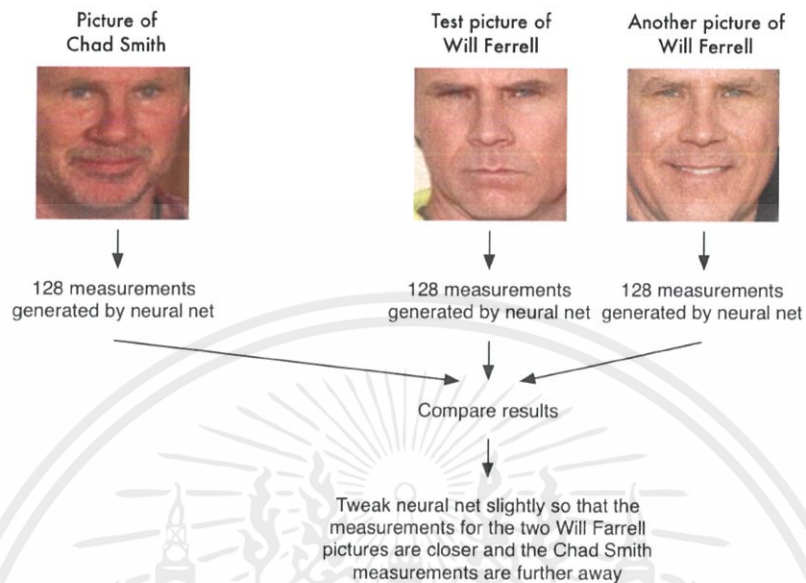
วิธีที่มีความน่าเชื่อถือได้มากที่สุดในการวัดค่าระยะห่างระหว่างจุดต่างๆ บนใบหน้า การวัดระยะห่างรูปแบบใดที่ควรใช้รวบรวมเพื่อสร้างฐานข้อมูลของระบบรู้จำใบหน้าตัวอย่างเช่น ระยะห่างระหว่างหูสองข้าง ความยาวจมูก สีดวงตา หรืออื่น ๆ แต่การเก็บข้อมูลสีดวงตาไม่มีความสมเหตุสมผลกับคอมพิวเตอร์ที่มองแต่ละพิกเซลในภาพ ซึ่งแตกต่างกับสมองของมนุษย์ ผลการวิจัยค้นพบว่าวิธีการที่แม่นยำที่สุดคือการใช้คอมพิวเตอร์หาวิธีการวัดระยะห่างเพื่อรวบรวมด้วยตัวมันเอง เนื่องจากการเรียนรู้อย่างลึกซึ้ง (Deep Learning) นั้นทำได้ดีกว่าสมองมนุษย์ในการหาส่วนของใบหน้าที่สำคัญเพื่อทำการวัดระยะห่างแต่ละจุดเฉพาะ

วิธีการแก้ปัญหาคือใช้วิธีการ Deep Convolutional Neural Network แต่แทนที่จะทำการฝึกระบบ (Train Network) ให้รู้จำภาพเหมือนดังขั้นตอนทั้งหมดที่กล่าวมาข้างต้น จะเป็นการฝึกเพื่อให้เกิดการวัดระยะห่างทั้งหมดจำนวน 128 จุดเฉพาะต่อ 1 ใบหน้า หรือเรียกว่าขั้นตอนการฝึกแบบ Triplet Training โดยทำการดูที่ภาพใบหน้า 3 ภาพในเวลาเดียวกันคือ

- ภาพใบหน้าที่ถูกทำการเทรนและรู้จำใบหน้าสำเร็จแล้ว
- ภาพอีกภาพหนึ่งของบุคคลคนเดียวกับภาพแรก
- ภาพของบุคคลอื่นที่ยังไม่ได้ทำการรู้จำใบหน้า

โดยอัลกอริทึมจะใช้ค่าการวัดระยะห่างระหว่างจุดเฉพาะบนใบหน้าในแต่ละภาพทั้งสาม จากนั้นปรับแต่ง Neural Network เล็กน้อยเพื่อให้แน่ใจว่าค่าระยะห่างที่วัดได้ทำให้ภาพ #1 และ #2 มีความใกล้เคียงกัน และภาพ #2 และ #3 ต่างกัน

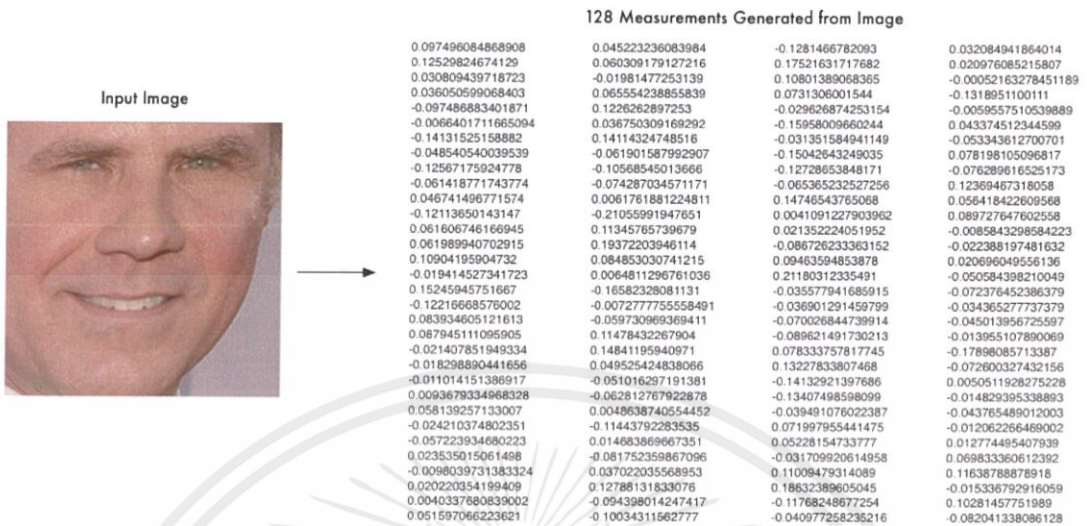
A single 'triplet' training step:



รูปที่ 2.13 ขั้นตอนการทำ Triplet Training

หลังจากทำซ้ำขั้นตอนดังกล่าวเป็นจำนวนมากจากภาพหลายล้านภาพของคนหลายพันคนที่แตกต่างกัน Neural Network จะเกิดการเรียนรู้ที่จะทำการวัดค่าระยะห่างระหว่างจุดบนใบหน้า 128 จุดเฉพาะที่มีความแม่นยำและน่าเชื่อถือสำเร็จ และได้มีแนวคิดในการลดขนาดข้อมูลที่ซับซ้อนอย่างรูปภาพโดยการแปลงเป็นตัวเลขที่สร้างโดยคอมพิวเตอร์ ซึ่งเป็นวิธีที่พบได้ทั่วไปใน Machine Learning โดยวิธีการที่แม่นยำที่ได้เลือกใช้นั้นถูกคิดค้นขึ้นในปี 2558 โดยนักวิจัยจาก Google

การเข้ารหัสรูปภาพใบหน้าเป็นการเลือกใช้วิธีการ Training a Convolutional Neural Network เมื่อระบบได้รับการ Train สำเร็จแล้ว จะสามารถทำการวัดค่าระยะห่างระหว่างจุดต่างๆ สำหรับใบหน้าใด ๆ ได้จำนวน 128 จุดเฉพาะ ซึ่งผลลัพธ์มีจำนวนข้อมูลเป็นจำนวนมากและเปลืองพลังงานคอมพิวเตอร์ เนื่องจากใช้เวลาในการ Train อย่างต่อเนื่องประมาณ 24 ชั่วโมงเพื่อให้มีความแม่นยำที่ดี



รูปที่ 2.14 ค่าตัวเลขจำนวน 128 ค่าที่ถูกแปลงจากรูปภาพใบหน้า

ดังนั้นตัวเลขจำนวน 128 ค่านี้นี้เป็นตัวชี้วัดความเหมือนหรือต่างการของภาพใบหน้าระหว่างสองภาพว่าเป็นบุคคลคนเดียวกันหรือไม่ และค่าตัวเลขนี้ไม่มีความจะเป็นที่จะต้องเป็นค่าตัวเลขเดียวกันจึงจะเป็นบุคคลคนเดียวกัน เมื่อค่าตัวเลขมีความใกล้เคียงกัน ระบบจึงสามารถรู้จำและตัดสินใจให้เป็นบุคคลคนเดียวกันได้

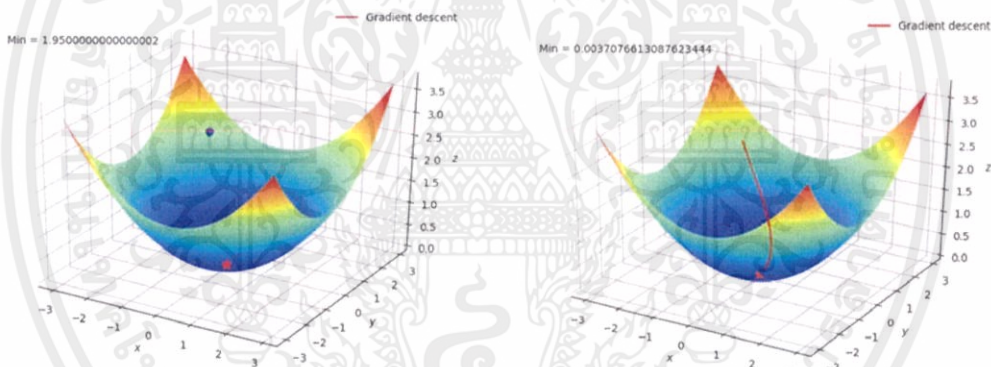
2.1.4 การระบุตัวตนจากการเข้ารหัสภาพใบหน้า (Finding the person's name from the encoding)

หลักการสำคัญคือการค้นหาข้อมูลบุคคลในฐานข้อมูลที่ได้ถูกผ่านขั้นตอนการวัดค่าต่างๆ เพื่อให้ระบบรู้จำภาพของบุคคลนั้นและเปรียบเทียบค่าที่มีความใกล้เคียงที่สุดเพื่อยืนยันตัวบุคคล โดยใช้อัลกอริทึมพื้นฐานของการจัดหมวดหมู่ใน Machine Learning ซึ่งไม่มีความซับซ้อนคือ Support Vector Machine (SVM) เป็นตัวจำแนกเชิงเส้น (Linear Classifier) ข้อได้เปรียบของ SVM คือมีประสิทธิภาพในการจำแนกข้อมูลที่มีมิติจำนวนมากได้ โดยต้องทำการฝึกตัวจำแนก (Train a Classifier) ให้สามารถดึงค่าตัวเลข 128 ค่าที่วัดจากภาพใหม่ที่ต้องการให้ระบบยืนยันตัวบุคคลเปรียบเทียบกับค่าตัวเลขในฐานข้อมูลเพื่อค้นหาค่าตัวเลขที่มีค่าใกล้เคียงที่สุด โดยใช้เวลาดังทั้งหมดเพียงหน่วยมิลลิวินาที

2.1.4.1 Stochastic Gradient Descent (SGD)

เทคนิค Stochastic Gradient Descent (SGD) เป็น Classifier อย่างง่ายแต่มีประสิทธิภาพมากในจำพวกตัวจำแนกเชิงเส้น (Linear Classifier) ภายใต้ loss function รูปกรวยคว่ำหรือ convex อย่าง Support Vector Machines and Logistic Regression เทคนิค SGD มีความประสบความสำเร็จในการนำไปใช้แก้ปัญหาเกี่ยวกับ large-scale and sparse Machine Learning เกี่ยวกับจำแนกข้อความและการประมวลผลทางภาษา เนื่องจากข้อมูลที่เบาบาง classifiers ในโมดูลนี้สามารถปรับแก้ปัญหได้อย่างง่ายด้วยวิธีการเทรน (training) ที่มากกว่า 10^5 ครั้งและมากกว่า 10^5 รูปแบบ โดยข้อดีของ Stochastic Gradient Descent คือมีประสิทธิภาพและความสะดวกในการใช้งาน (มีโอกาสนำนวนมากสำหรับการปรับแต่งโค้ด) ข้อเสียคือต้องการตัวแปรหลายตัวเช่นตัวแปร regularization และจำนวนการวนซ้ำ มีความไวต่อการปรับขนาดรูปแบบ

Gradient Descent หรืออัลกอริทึมในการหามุม θ (cost Function) ที่ต่ำที่สุด จะคล้ายกับ Normal Equation แต่มีประสิทธิภาพมากกว่า



รูปที่ 2.15 เส้น gradient descent จากจุดสีน้ำเงินไปจุดสีแดง ซึ่งวิ่งเข้าจุดกลางวงกลม

คณิตศาสตร์เบื้องหลัง สามารถเขียนได้ดังสูตร

$$\theta(\text{next step}) = \theta - \eta \nabla_{\theta} \text{MSE}(\theta)$$

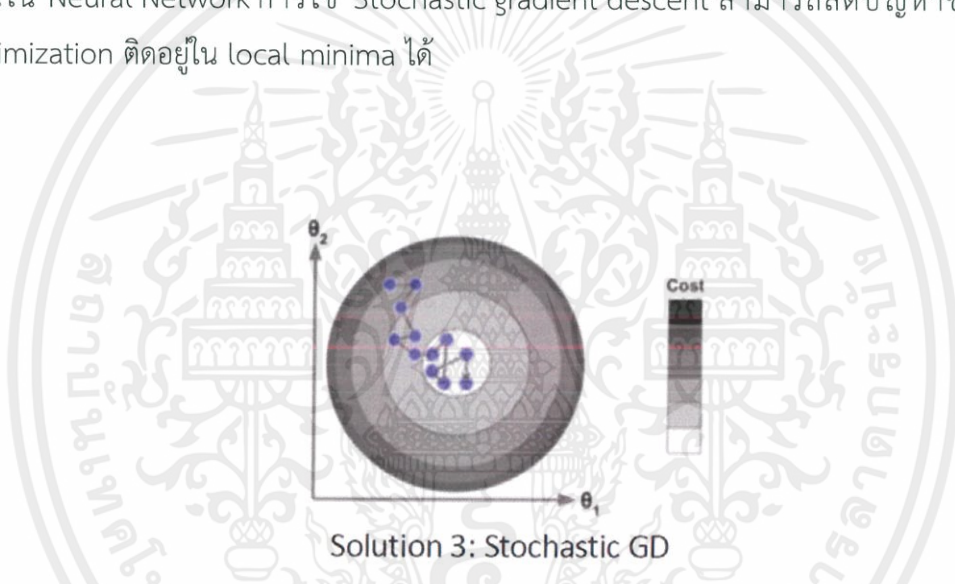
โดยที่ θ เริ่มต้นเป็นค่า random

η : Learning rate ก้าวใหญ่เท่าไร

$$\nabla_{\theta} \text{MSE}(\theta) = \begin{pmatrix} \frac{\partial}{\partial \theta_0} \text{MSE}(\theta) \\ \frac{\partial}{\partial \theta_1} \text{MSE}(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \text{MSE}(\theta) \end{pmatrix} = \frac{2}{m} X^T \cdot (X \cdot \theta - y), \text{ Gradient ของ } \text{MSE}(\theta) \text{ w.r.t. } \theta \text{ เป็นตัวบอก}$$

ทิศทางลงเนิน

หากสมมติว่ามี data เยอะมากๆ เช่น มีหนึ่งล้านคู่ของ $(x^{(i)}, y^{(i)})$ โดยปกติแล้ว ในการแก้สมการ เมื่อแทนค่าจะได้คำตอบมาในทันที แต่ใน Gradient Descent มันจะอัปเดตค่าแทน ทำไปเรื่อย ๆ จนกว่าจะได้ cost function ที่ต่ำที่สุด การที่จะคำนวณ gradient แต่ละครั้งนั้น เราต้องใช้ data ทั้งหมดเพื่อมาอัปเดตพารามิเตอร์ใหม่ ซึ่งแน่นอนว่าใช้เวลามากกว่าแน่นอน สำหรับ Stochastic gradient descent ในแต่ละการคำนวณ gradient เราจะสุ่ม data เพียงบางส่วนเพื่อใช้อัปเดตเท่านั้น ไม่ได้ใช้ data ทั้งหมด โดยทฤษฎีได้พิสูจน์ว่าเราสามารถสุ่ม data เพียงไม่มากเพื่อใช้ในการอัปเดตพารามิเตอร์ในแต่ละครั้งหรือ iteration โดยไม่ต้องนำ data ทั้งหมดมาทีเดียวก็ได้ แต่ในที่สุดจะลู่เข้าสู่คำตอบใกล้เคียงกัน และยิ่งในกรณีที่พารามิเตอร์ที่เราต้องการประมาณมีเยอะมากๆ เช่นใน Neural Network การใช้ Stochastic gradient descent สามารถลดปัญหาของการที่ optimization ติดอยู่ใน local minima ได้



รูปที่ 2.16 Stochastic Gradient Descent (SGD)

จากรูปเราจะเห็นได้ชัดว่าเส้นการเดินทางของ Stochastic Gradient Descent (SGD) ไม่ได้เป็นเส้นตรงและจุดสุดท้ายไม่ใช่ตรงกลาง แต่ข้อดีคือสามารถรับมือกับข้อมูลที่มีขนาดใหญ่ได้ ซึ่งถ้าเป็น Normal Equation ก็จะใช้เวลานานมากกว่าจะมาถึงตรงกลาง แต่ตัว SGD จะใช้เวลาที่สั้นกว่า เพราะนำข้อมูลแค่บางอันมาทำการเทรนแล้วสุ่มทางจนมาถึงตรงกลาง

จากข้างต้นจึงทำเทคนิค SGD Classifier มาประยุกต์ใช้กับระบบนี้ที่เกี่ยวข้องกับการประมวลผลภาพ เพื่อให้สามารถประมวลผลได้รวดเร็วขึ้น และสามารถประมาณความน่าจะเป็นของการประมวลผลภาพได้ และพารามิเตอร์ที่ใช้จะเป็นพารามิเตอร์ loss ซึ่ง loss เป็นการจำแนก

ในลักษณะความน่าจะเป็น โดยการประมาณค่าความน่าจะเป็นจะประมาณจากชุดข้อมูลใบหน้าที่ได้ผ่านการ Train สำเร็จแล้ว และจะเปรียบเทียบข้อมูลใบหน้ากับชุดข้อมูลที่ได้บันทึกไว้

2.2 หน่วยประมวลผลส่วนกลางของระบบ

ในการพัฒนาระบบจะมีหน่วยประมวลผลกลางที่ควบคุมระบบตรวจจับใบหน้า ในที่นี้ผู้จัดทำได้ใช้คอมพิวเตอร์ ASUS K455LD-WX062D สำหรับประมวลผล ใช้ภาษา Python และใช้ไลบรารี OpenCV ในการเขียนโปรแกรม โดยคอมพิวเตอร์มีความเร็วของซีพียู (สัญญาณนาฬิกา) Core i5 1.7 GHz (3M Cache, up to 2.7 GHz) และใช้ซีพียูช่วยประมวลผลกราฟิก NVIDIA Geforce 820M (N15V-GM) มีขนาดหน่วยความจำ 4GB DDR3 มีลักษณะดังรูปที่ 2.15



รูปที่ 2.15 คอมพิวเตอร์ ASUS K455LD-WX062D

2.2.1 คุณสมบัติของคอมพิวเตอร์ ASUS K455LD-WX062D

คุณสมบัติหลักของคอมพิวเตอร์ ASUS K455LD-WX062D มีหน่วยประมวลผลหลักที่มีความเร็ว (สัญญาณนาฬิกา) Core i5 1.7 GHz (3M Cache, up to 2.7 GHz) หน่วยความจำหลัก (RAM) ขนาด 4GB DDR3 เชื่อมต่อเครือข่ายได้ทั้งการเชื่อมต่อผ่านสาย Ethernet และ Wireless ตามมาตรฐาน IEEE 802.11n พร้อมทั้ง Bluetooth 4.0 พร้อม port USB, สาย HDMI, Audio port

ตารางที่ 2.1 คุณสมบัติของคอมพิวเตอร์ ASUS K455LD-WX062D

| | |
|--|---|
| ขนาดหน้าจอแสดงผล | 14.0" LED Black-lit // Slim 200 nits // HD |
| ความละเอียดหน้าจอ | 1366x768 |
| หน่วยประมวลผลกลาง (CPU : Central Processing Unit) | Core i5 1.7GHz (3M Cache, up to 2.7 GHz) |
| หน่วยประมวลผลภาพ (GPU : Graphics Access Memory) | NVIDIA Geforce 820M (N15V-GM) |
| หน่วยความจำหลัก (RAM : Random Access Memory) | 4GB |
| หน่วยความจำ (Storage) | HDD 500GB |
| การเชื่อมต่อเครือข่าย (Networking) | 10/100/1000 Mbps Ethernet, 802.11 b/g/n wireless |
| Bluetooth | Bluetooth 4.0 (single band) |
| กล้องเว็บแคม (Web Camera) | 0.3 MPixel |
| ระบบปฏิบัติการ | DOS (Disk Operating System) |
| Port | Ethernet LAN, D-Sub/VGA, HDMI, USB 3.0 Type-A 1 Port, USB 2.0 2 Port, Port Card Reader, ช่องต่อหูฟังและไมโครโฟน |
| แบตเตอรี่ | 37 WHrs, 2-cell Li-ion Polymer Battery Pack |
| ขนาด | 2.10 Kg |
| ลำโพง | Built-In Stereo Speakers |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ภาษาคอมพิวเตอร์ส่วนแสดงผล

ในปัจจุบันภาษาคอมพิวเตอร์ที่ใช้ในการออกแบบส่วนติดต่อกับผู้ใช้มีหลากหลายภาษา ซึ่งภาษาคอมพิวเตอร์ที่ใช้ในการออกแบบเว็บไซต์สามารถแบ่งส่วนการทำงานหลักได้เป็น 2 ประเภท ดังนี้

2.3.1 Front End

Front End เป็นส่วนติดต่อกับผู้ใช้ (User interface) ส่วนที่แสดงหน้าตาเว็บไซต์ให้ผู้ใช้ที่เข้ามาชม เช่น หน้าแรกของเว็บไซต์ หน้าเว็บเพจที่แสดงเนื้อหาต่างๆ รูปภาพ สื่อ หรือลิงก์ เป็นต้น เป็นส่วนที่ผู้ใช้งานทั่วไปสามารถเห็น และเข้ามาใช้งานได้ของเว็บไซต์ โดยส่วนมากจะแสดงผลบนเว็บเบราว์เซอร์ เช่น Google Chrome, Internet Explorer หรือ Firefox เป็นต้น ภาษาคอมพิวเตอร์ที่ใช้ในส่วนติดต่อกับผู้ใช้งานนี้ ได้แก่

2.3.1.1 HTML (Hypertext Markup Language)

เป็นภาษาคอมพิวเตอร์หลักที่ใช้ในการที่ใช้ในการเขียนหน้าเว็บไซต์หรือหน้าเว็บเพจเป็นโครงสร้างหลักในการแสดงผลเอกสาร ตัวอักษร ตาราง และสามารถทำการออกแบบแก้ไขได้ด้วยโปรแกรม Text Editor

Hypertext เป็นข้อความที่เชื่อมต่อกันผ่านลิงค์ เป็นภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่างๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้นจึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink

โครงสร้างหลักของ HTML มี Tag เริ่มต้นคือ <html> และปิดท้ายด้วย </html> เสมอ ซึ่งจะมีการแบ่งหลักๆ 2 ส่วน แบ่งได้เป็น

ส่วน head คำสั่งในส่วนนี้จะเริ่มด้วย <head> ปิดด้วย </head> ซึ่งคำสั่งในส่วนนี้ จะเป็นบรรยายรายละเอียดเว็บเพจ เช่น ชื่อหน้าเว็บ (Title) ชื่อผู้ทำ (Author) ซึ่งจะไม่มีผลกับหน้าเว็บเพจโดยตรง

ส่วน body คำสั่งในส่วนนี้จะเริ่มด้วย <body> และปิดด้วย </body> ซึ่งคำสั่งเป็นเนื้อหาหลักของหน้าเว็บใช้ในการจัดรูปแบบหน้าเว็บเพจตามที่ต้องการ

2.3.1.2 Java Script

เป็นภาษาคอมพิวเตอร์สำหรับเขียนโปรแกรมใช้ร่วมกับ HTML และใช้ในการสร้างฟังก์ชันให้เว็บไซต์ตอบสนองต่อผู้ใช้งานได้มากขึ้น เพื่อเพิ่มรูปแบบการทำงานของหน้าเว็บเพจ

ให้มีความหลากหลายมากขึ้น ซึ่ง JavaScript เป็นภาษาประเภท Interpreted Language คือ JavaScript จะทำงานทีละคำสั่ง โดยเป้าหมายหลักของ JavaScript คือ การพัฒนาโปรแกรมในระบบ อินเทอร์เน็ต โดยสามารถทำการออกแบบแก้ไขได้ด้วยโปรแกรมจำพวก Text Editor

2.3.1.3 CSS (Cascading Style Sheets)

เป็นภาษาคอมพิวเตอร์ ซึ่งใช้งานสำหรับกำหนดรูปแบบการแสดงผล องค์ประกอบต่างๆ บนเว็บไซต์ CSS สามารถวางอยู่นอก HTML page ได้ ดังนั้นการใช้งาน CSS จึงช่วยประหยัดเวลาในการกำหนดรูปแบบที่ต้องการจะแสดงในแต่ละหน้าเว็บ ซึ่งมีประโยชน์อย่างมาก ในกรณีที่ต้องพัฒนาเว็บไซต์ขนาดใหญ่ เนื่องจากสามารถกำหนดรูปแบบการแสดงผล สี รูปแบบ ตัวอักษร และอื่นๆ อีกมากมาย โดยการกำหนดคุณลักษณะนั้นให้กับ Tag ที่จะนำไปใช้ ทำให้ CSS สามารถแก้ไขปัญหาที่เคยเกิดกับภาษา HTML รุ่นเก่าๆ ที่ต้องการกำหนดคุณลักษณะทุกอย่างในทุกๆ หน้าเว็บ แม้ว่าจะเป็นคนลักษณะเดียวกันก็ตาม ซึ่งกระบวนการเหล่านี้นับว่าเป็นงานหนักสำหรับผู้พัฒนาเว็บไซต์ การที่ลดกระบวนการนี้ลงไปได้โดยใช้ CSS จึงช่วยให้การพัฒนาเว็บทำได้รวดเร็วและมีประสิทธิภาพยิ่งขึ้น

2.3.1.4 Bootstrap

Bootstrap คือ ส่วนที่แสดงผลให้ผู้ใช้งานทั่วไปเห็น หรืออาจกล่าวในอีกมุมหนึ่งคือหน้าเว็บไซต์ของเว็บไซต์หลัก ซึ่งเป็นสิ่งที่ช่วยกำหนดกรอบของการทำงานการพัฒนาเว็บไซต์ ให้เป็นไปในทางเดียวกัน โดยการผนวก HTML, CSS และ JS มารวมเข้าไว้ด้วยกันสำหรับการพัฒนาเว็บแอปพลิเคชัน เพื่อให้สามารถรองรับการเข้าใช้งานผ่านอุปกรณ์อิเล็กทรอนิกส์ที่สามารถเชื่อมต่อกับอุปกรณ์อื่นๆ หรือเครือข่ายไร้สายผ่านทางโปรโตคอลที่แตกต่างกันได้ เช่น Smart Phone, Laptop หรือเรียกว่า Responsive Web หรือ Mobile First

2.3.2 Back End

ระบบจัดการเว็บไซต์ เช่น จัดการฐานข้อมูล โครงสร้างเว็บไซต์ หรือส่วนประมวลผลข้อมูล ซึ่งมีไว้สำหรับผู้ดูแลระบบ เพื่อทำการเพิ่ม ลบ แก้ไข หรือเปลี่ยนแปลงเว็บไซต์ โดยการที่จะเข้าใช้งานส่วนนี้จะต้องมีการติดตั้งในส่วนของเว็บเซิร์ฟเวอร์ โดยภาษาคอมพิวเตอร์ที่ใช้ในการออกแบบส่วนประมวลผลข้อมูลนั้น ได้แก่

2.3.2.1 ภาษา Python (Python programming language)

Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 Python นั้นเป็นภาษาแบบ Interprete ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้นและโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ทำให้การเขียนโค้ดเข้าใจง่าย ทั้งในโปรแกรมขนาดเล็กไปจนถึงโปรแกรมขนาดใหญ่

Python นั้นมีคุณสมบัติเป็นภาษาเขียนโปรแกรมแบบไดนามิก มีระบบการจัดการหน่วยความจำอัตโนมัติ และสนับสนุนการเขียนโปรแกรมหลายรูปแบบที่ประกอบไปด้วยการเขียนโปรแกรมเชิงวัตถุ imperative การเขียนโปรแกรมแบบฟังก์ชันและการเขียนโปรแกรมแบบขั้นตอน มีไลบรารีที่ครอบคลุมการทำงานอย่างหลากหลาย ตัวแปรในภาษา Python นั้นมีให้ใช้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา Python สามารถประมวลผลในระบบต่างๆ ได้อย่างกว้างขวาง CPython นั้นเป็นการพัฒนาในขั้นต้นของ Python ซึ่งเป็นโปรแกรมแบบ open source และมีชุมชนสำหรับเป็นต้นแบบในการพัฒนา เนื่องจากมันได้มีการนำไปพัฒนากระจายไปอย่างหลากหลาย CPython นั้นจึงถูกจัดการโดยองค์กรไม่แสวงหาผลกำไรอย่าง Python Software Foundation

ภาษา Python นั้นกำเนิดขึ้นในปลายปี 1980 และการพัฒนาของภาษา Python นั้นเริ่มต้นในเดือนธันวาคมปีค.ศ. 1989 โดย Guido van Rossum ที่ Centrum Wiskunde & Informatica (CWI) ในประเทศเนเธอร์แลนด์ เนื่องในผู้ประสบความสำเร็จในการสร้างภาษา ABC ที่มีความสามารถสำหรับการ exception handling และการติดต่อประสานกับระบบปฏิบัติการ Amoeba ซึ่ง Van Rossum นั้นเป็นผู้เขียนหลักการของภาษา Python และเขาทำหน้าที่เป็นกลางในการตัดสินใจสำหรับทิศทางการพัฒนาของภาษา Python

2.3.2.2 OpenCV (Open Source Computer Vision)

เป็นซอฟต์แวร์แบบเปิดเผยแพร่ (Library Open Source) หรือเป็นไลบรารีที่มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพและการประมวลผลภาพพื้นฐาน (Image Processing) เพื่อให้การพัฒนาโปรแกรมทางด้านการมองเห็นของคอมพิวเตอร์ (Computer Vision) สามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่งและภาพเคลื่อนไหว เช่น ภาพจากกล้อง VDO หรือ VDO File

เป็นไปได้โดยง่ายโดยการเรียกใช้งานจะต้องมีการเรียกไฟล์ส่วนหัว (Header file) และลิงค์ (Link) ไบเบรารีต่างๆ รวมถึง DLL (Dynamic Link Library)

OpenCV อยู่ภายใต้ใบอนุญาต BSD (BSD licenses) ซึ่งเราสามารถใช้ได้ฟรี ทั้งทางด้านการศึกษาและทางการค้า นอกจากนี้ OpenCV ยังมีอินเทอร์เฟซที่หลากหลายรองรับการพัฒนาโปรแกรมบนภาษาโปรแกรมต่างๆ เช่น C/C++, Python, Java เป็นต้น และใช้ได้บนระบบปฏิบัติการที่เป็น Linux, Microsoft Windows, Android, และ Mac

2.3.2.3 WebRTC (Web Real-time Communication)

WebRTC (Web real time communication) เป็นมาตรฐานที่เกิดขึ้นโดยมีจุดมุ่งหมายเพื่อช่วยให้เกิดการสื่อสารแบบเรียลไทม์ระหว่างเว็บเบราว์เซอร์เพื่อส่ง และรับข้อมูลแบบมัลติมีเดียรูปแบบ peer-to-peer (P2P) ในลักษณะเรียลไทม์ เทคโนโลยีนี้สามารถลดความซับซ้อนของการพัฒนาระบบเพื่อการสื่อสารเรียลไทม์ระหว่างเบราว์เซอร์และผู้ใช้ได้ ซึ่งในปัจจุบัน WebRTC API มีอยู่ในเว็บเบราว์เซอร์รุ่นใหม่ ๆ เกือบทุกชนิด

สถาปัตยกรรมของ WebRTC มีส่วนประกอบอย่างน้อยสามส่วน หนึ่งคือผู้ให้บริการเว็บแอปพลิเคชัน และอีกสองส่วนคือผู้ใช้งาน (Peers) โดย WebRTC จะช่วยให้ข้อมูลมัลติมีเดียส่งผ่านได้โดยตรงระหว่างเบราว์เซอร์โดยไม่จำเป็นต้องมีการแทรกแซงใดๆ จากเซิร์ฟเวอร์สำหรับความสามารถในการใช้งานเว็บไซต์ (ฝั่งผู้ใช้) ซึ่งการทำงานจะมีการผสมระหว่าง HTML และ JavaScript สามารถมีการสื่อสารระหว่างกันกับเว็บเบราว์เซอร์หรือโปรแกรมบนมือถือ ผ่าน WebRTC API ซึ่ง API นี้จะมีชุดของฟังก์ชันการทำงานหลายประเภท เช่น การจัดการการเชื่อมต่อ ความสามารถในการเข้ารหัสและถอดรหัสข้อมูล การควบคุมสื่อ การจัดการ Firewall และอื่นๆ อีกมากมายที่จะถูกดำเนินการด้วย JavaScript ฯลฯ โดย WebRTC API ช่วยให้เบราว์เซอร์และภาษาสคริปต์สามารถโต้ตอบกับอุปกรณ์สื่อนำเข้า (ไมโครโฟน, กล้องและลำโพง) ได้โดยตรง ทำให้การประมวลผลและส่งผ่านข้อมูลสะดวกรวดเร็วขึ้น

ข้อดีของ WebRTC ประการแรกคือ เป็น Open source API ที่ไม่ต้องเสียค่าใช้จ่ายและเป็นเทคโนโลยีที่มีอยู่มากที่สุดในเบราว์เซอร์ที่ให้บริการอยู่ทั้งหมด ประการที่สองคือ ความเป็นอิสระต่อแพลตฟอร์ม เบราว์เซอร์ใดๆ ก็ตามที่สนับสนุน WebRTC สามารถใช้งานกับระบบปฏิบัติการใด ๆ ก็ได้ ทั้งยังสามารถสร้างการเชื่อมต่อเสียงเรียลไทม์หรือการเชื่อมต่อวิดีโอไปยังอุปกรณ์ WebRTC อื่น หรือเชื่อมต่อไปยังเซิร์ฟเวอร์ได้ ส่วนประการที่สาม WebRTC มีการเข้ารหัสข้อมูลอยู่เสมอ โดยใช้ Secure RTP Protocol (SRTP) ซึ่งเป็นกุญแจสำคัญในการจัดการสำหรับการ

เข้ารหัส และการตรวจสอบของทั้งเสียงและวิดีโอ ประโยชน์และข้อดีสุดท้ายของ WebRTC ก็คือ สามารถที่จะทำงานโดยไม่จำเป็นต้องติดตั้งซอฟต์แวร์เพิ่มเติมหรือปลั๊กอินเสริมใดๆ จากข้อดีข้างต้นนี้ เว็บแอปพลิเคชันกับเทคโนโลยี WebRTC จึงถูกนำมาใช้เพื่อให้ความสะดวกในเรื่องของการเก็บรวบรวมภาพผู้ใช้บนเว็บแอปพลิเคชัน และการพัฒนาระบบตรวจจับใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ เนื่องจากช่วยให้สามารถใช้งานระบบได้กับคอมพิวเตอร์ทุกเครื่อง ซึ่งจะช่วยให้การใช้งานและการศึกษาก็สามารถทำได้อย่างสะดวกรวดเร็วยิ่งขึ้น

2.4 ระบบฐานข้อมูล

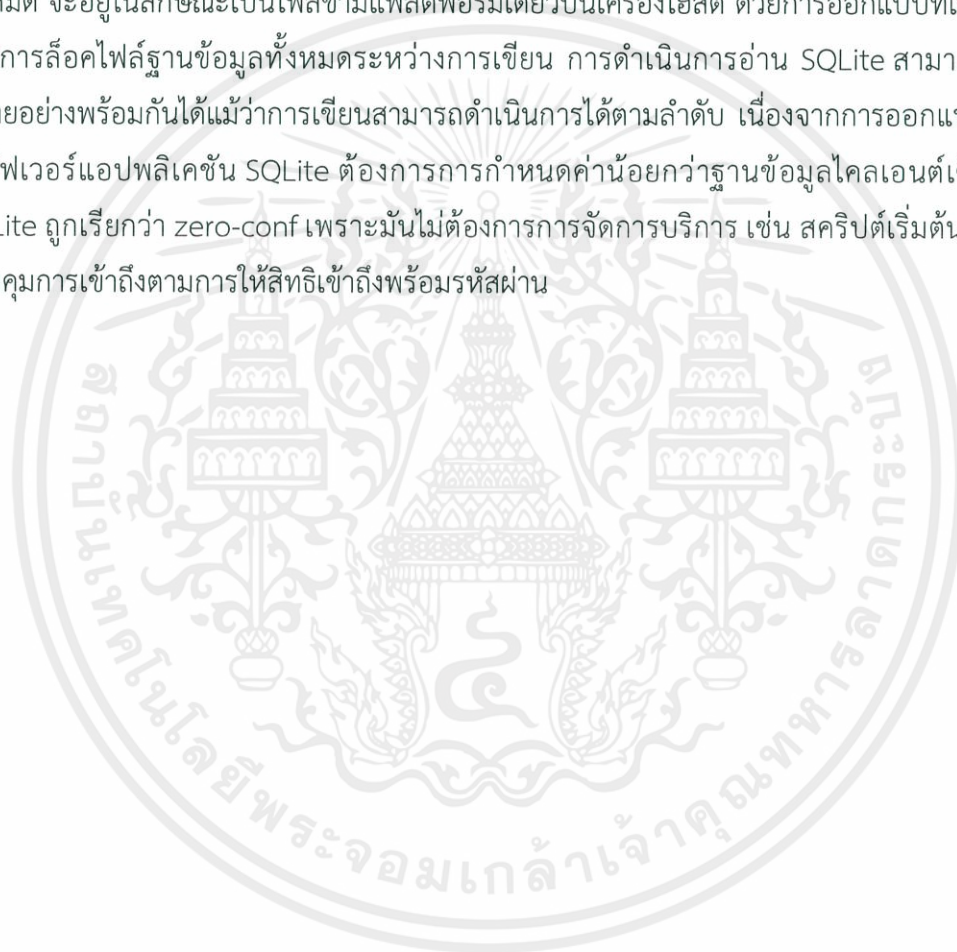
ระบบฐานข้อมูล (Database System) คือระบบที่รวบรวมข้อมูลต่างๆ ที่เกี่ยวข้องกัน เข้าไว้ด้วยกัน ในระบบฐานข้อมูลจะประกอบด้วยหลายแฟ้มข้อมูล และเปิดโอกาสให้ผู้ใช้สามารถใช้งานและป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพซึ่งการเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล หรือการแก้ไขฐานข้อมูล ในการสร้างระบบเช็คชื่อเข้าเรียนอัตโนมัติ และเว็บไซต์นั้นจำเป็นต้องมี Database หรือฐานข้อมูลเพื่อใช้ในการเก็บที่ข้อมูลต่างๆ ที่จำเป็นสำหรับระบบ จึงมีการใช้เครื่องมือ SQLite เข้ามาช่วยจัดการฐานข้อมูล

2.5.1 SQLite

เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ที่มีอยู่ในไลบรารี C มีลักษณะตรงกันข้ามกับระบบการจัดการฐานข้อมูลอื่นๆ ซึ่ง SQLite ไม่ใช่เอนจินฐานข้อมูลโคลเอนด์กับเซิร์ฟเวอร์ แต่มีคุณลักษณะฝังอยู่ในตัวโปรแกรม

SQLite มีการพัฒนาเพื่อให้เป็นไปตามข้อกำหนดของ ACID และใช้มาตรฐาน SQL ส่วนใหญ่ตามด้วยไวยากรณ์ PostgreSQL อย่างไรก็ตาม SQLite ใช้ไวยากรณ์ SQL แบบไดนามิกและความคาดเคลื่อนความถูกต้องของข้อมูลค่อนข้างต่ำ จึงเป็นตัวเลือกละเอียดนิยมในฐานะซอฟต์แวร์ฐานข้อมูลแบบฝังตัวสำหรับที่จัดเก็บภายในเครื่องลูกข่าย ในส่วนแอปพลิเคชันซอฟต์แวร์ ตัวอย่างเช่น เว็บเบราว์เซอร์ ได้มีการนำเอาเอนจินฐานข้อมูลมาส่วนหนึ่ง และใช้อย่างแพร่หลายมากที่สุดในปัจจุบันโดยเบราว์เซอร์ระบบปฏิบัติการและระบบฝังตัว (เช่น โทรศัพท์มือถือ) และอื่นๆ SQLite มีการเชื่อมโยงกับภาษาการเขียนโปรแกรมเป็นจำนวนมาก

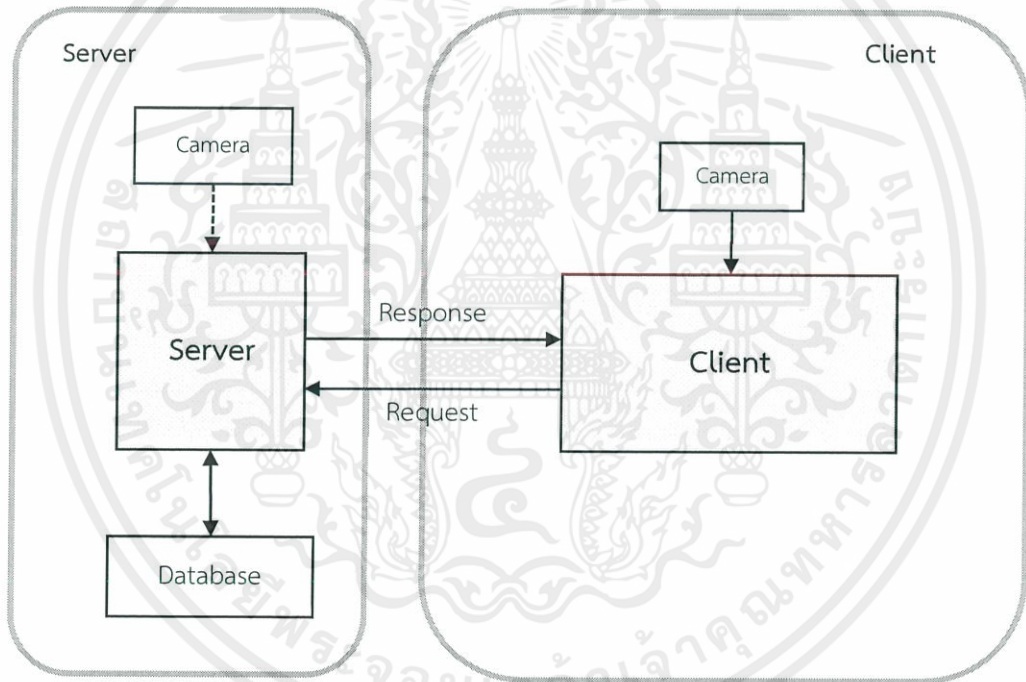
ความแตกต่างระหว่าง SQLite กับระบบจัดการฐานข้อมูลคลเอนท์กับเซิร์ฟเวอร์เอ็นจิน กล่าวคือ SQLite จะไม่มีกระบวนการแบบสแตนด์อโลนที่แอปพลิเคชัน แต่จะเชื่อมโยงผ่านไลบรารี SQLite เข้าด้วยกัน และรวมเป็นส่วนสำคัญของโปรแกรมแอปพลิเคชัน การเชื่อมโยงอาจเป็นแบบสแตติกหรือแบบไดนามิกก็ได้ ซึ่งแอปพลิเคชันโปรแกรมสามารถเรียกใช้ฟังก์ชันการทำงานของ SQLite ผ่านการเรียกใช้ฟังก์ชันอย่างง่ายซึ่งช่วยลดความหน่วงในการเข้าถึงฐานข้อมูล ส่งผลให้การเรียกใช้ฟังก์ชันภายในกระบวนการเดียวกันมีประสิทธิภาพดียิ่งขึ้น ซึ่งรูปแบบการจัดเก็บฐานข้อมูลทั้งหมด จะอยู่ในลักษณะเป็นไฟล์ข้ามแพลตฟอร์มเดียวกันบนเครื่องโฮสต์ ด้วยการออกแบบที่เรียบง่ายนี้ โดยการล็อกไฟล์ฐานข้อมูลทั้งหมดระหว่างการเขียน การดำเนินการอ่าน SQLite สามารถทำงานหลายอย่างพร้อมกันได้แม้ว่าการเขียนสามารถดำเนินการได้ตามลำดับ เนื่องจากการออกแบบที่ไม่ใช้เซิร์ฟเวอร์แอปพลิเคชัน SQLite ต้องการการกำหนดค่าน้อยกว่าฐานข้อมูลคลเอนท์เซิร์ฟเวอร์ SQLite ถูกเรียกว่า zero-conf เพราะมันไม่ต้องการการจัดการบริการ เช่น สคริปต์เริ่มต้น หรือการควบคุมการเข้าถึงตามการให้สิทธิเข้าถึงพร้อมรหัสผ่าน



บทที่ 3

การออกแบบและการจัดทำปฏิญญานิพนธ์

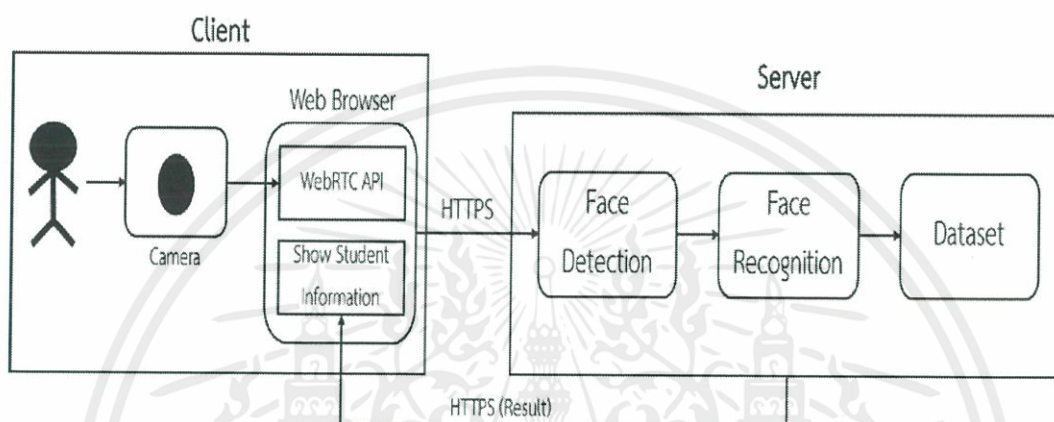
ในการออกแบบระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ สิ่งที่คุณจัดทำมุ่งเน้นคือ ความแม่นยำของระบบ ความสะดวกรวดเร็วในการใช้งาน ลดเวลาในการเช็คชื่อเข้าชั้นเรียน ทั้งนี้ การแสดงผลข้อมูลการเข้าเรียนของผู้เรียนก่อให้เกิดประโยชน์สูงสุดต่อผู้ใช้งาน ในการออกแบบตัวอุปกรณ์นั้น มุ่งเน้นการแสดงผลยืนยันตัวบุคคลให้มีความชัดเจน เข้าใจได้ง่าย เพื่อความรวดเร็วในการเก็บข้อมูลการเช็คชื่อเข้าเรียน แสดงการทำงานดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบ

3.1 สถาปัตยกรรมการออกแบบ

การทำงานในส่วนอุปกรณ์ของระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าห้องเรียนอัตโนมัติสามารถแบ่งออกได้เป็น 2 ส่วน ได้แก่ ส่วนของการประมวลผล (Server) และส่วนของผู้ใช้งาน (Client) แสดงได้ดังรูปที่ 3.2



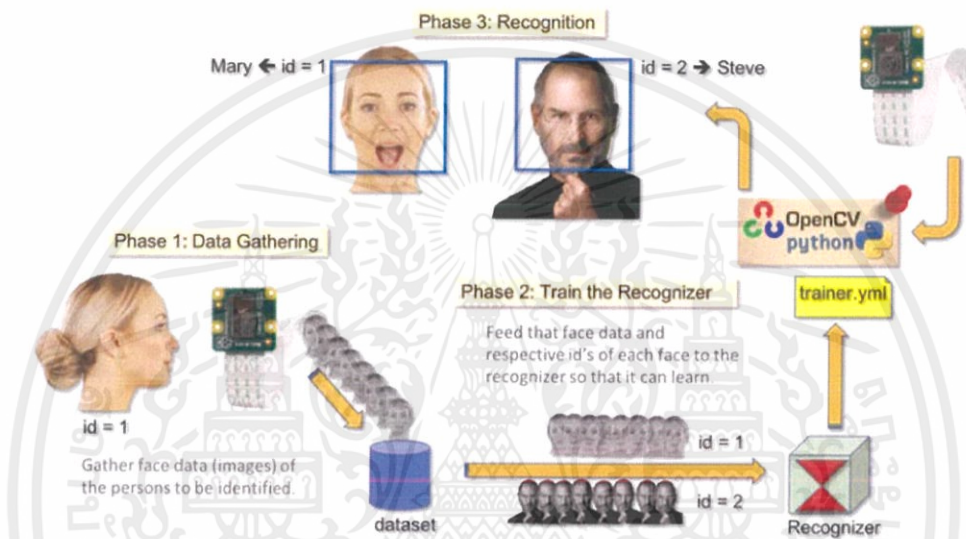
รูปที่ 3.2 การทำงานของระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียน

จากรูปที่ 3.2 แสดงการอธิบายการทำงานของระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียน โดยผู้จัดทำได้แบ่งการทำงานของระบบเป็น 2 ส่วน กล่าวคือ จัดเตรียมคอมพิวเตอร์จำนวน 2 ชุด โดยใช้คอมพิวเตอร์ 1 ชุดจำลองเป็นเครื่องแม่ข่าย (Server) ในการประมวลผลข้อมูล และเครื่องคอมพิวเตอร์ 1 ชุดสำหรับผู้ใช้งาน (Client)

ในการออกแบบ ผู้จัดทำได้ดำเนินการพัฒนาระบบเมื่อผู้เข้าใช้งานเข้าสู่ระบบผ่านเว็บเบราว์เซอร์และเรียกใช้งานระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียน ระบบจะส่งข้อมูลภาพจากกล้องจากเครื่องผู้ใช้งาน (Client) โดยใช้ WebRTC API เป็นตัวรับและส่งข้อมูลไปเข้ามาประมวลผลที่เครื่องแม่ข่าย (Server) ผ่านกระบวนการตรวจสอบใบหน้า ประมวลผล และส่งข้อมูลกลับยังผู้ใช้งาน โดยระบบจะแสดงผลผ่านเว็บเบราว์เซอร์บนเครื่องผู้ใช้งาน โดยผู้ใช้งานสามารถเข้าไปทำการตรวจสอบการเข้าเรียนของนักศึกษา อีกทั้งสามารถตรวจสอบข้อมูลการเข้าเรียนของนักศึกษาได้ ทั้งนี้การออกแบบจะใช้งานผ่านโปรโตคอล https (secure hyper-text transfer protocol)

3.1.1 การออกแบบโปรแกรมระบบรู้จำใบหน้า

ในการออกแบบระบบรู้จำใบหน้านั้น ผู้พัฒนาระบบได้ออกแบบไว้ในส่วนของเซิร์ฟเวอร์ โดยแบ่งการทำงานออกเป็น 3 ส่วน ประกอบด้วย ส่วนแรกส่วนข้อมูลใบหน้าสำหรับผู้ใช้งานรายใหม่ ส่วนที่สองส่วนการเทรนข้อมูล (Train the Recognizer) และส่วนที่สามส่วนการรู้จำใบหน้าเพื่อจัดการการเข้าถึง (Provide Access) ให้แก่ผู้ใช้เก่าที่เคยลงทะเบียนแล้ว



รูปที่ 3.3 ภาพรวมของระบบรู้จำใบหน้า

3.1.1.1 ส่วนข้อมูลใบหน้าผู้ใช้งานรายใหม่ (Data Gathering)

เมื่อโปรแกรมเริ่มทำงาน สำหรับผู้ใช้งานรายใหม่จะถูกขอให้ถ่ายภาพใบหน้าตรงอย่างน้อย 5 ภาพแล้วเก็บไว้ใน Dataset หรือ Directory ที่สร้างไว้ ก่อนจะทำการ Train data ให้ระบบรู้จำใบหน้าต่อไป

3.1.1.2 การออกแบบส่วนการ Train data ให้ Recognizer

การทำงานส่วนข้อมูลใบหน้าสำหรับผู้ใช้งานรายใหม่ โปรแกรมส่วน Trainer จะทำการดึงรูปภาพที่ถูกเก็บไว้มาตรวจจับใบหน้า โดยใช้ Stochastic Gradient Descent Classifier (SGD Classifier) ซึ่งเป็นเทคนิคการตรวจจับใบหน้าที่ผ่านการ Train ด้วยข้อมูลจำนวนมากเพื่อทำการจำแนกใบหน้าบุคคลหรือสิ่งต่าง ๆ

3.1.1.3 ออกแบบระบบรู้จำใบหน้า

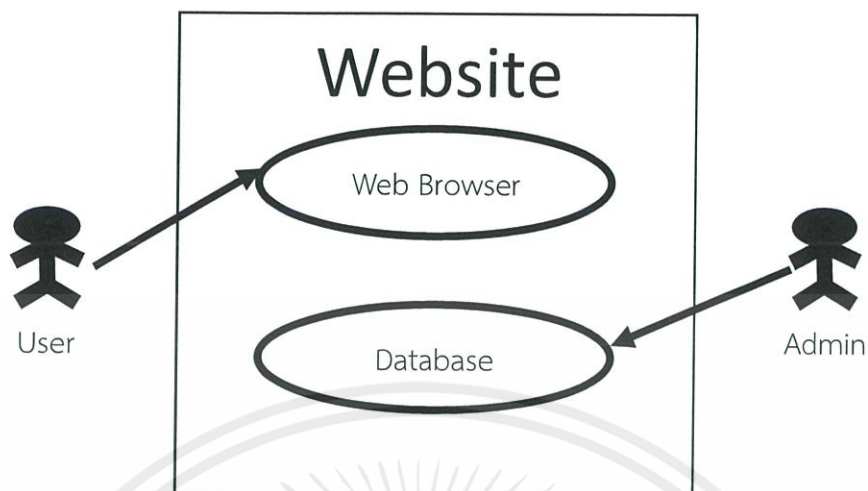
ออกแบบเพื่อจัดการการเข้าถึง (provide access) ให้แก่ผู้ใช้ที่มีข้อมูลใบหน้าแล้ว เมื่อโปรแกรมเริ่มทำงาน โดยกล้อง webcam จะถูกเปิดการใช้งานเพื่อค้นหาใบหน้า โดยใช้การตรวจจับใบหน้า (Face detection) โดยอาศัย Haar-like feature cascade classifier เช่นกัน เมื่อพบใบหน้าของผู้ใช้ที่มีข้อมูลใบหน้าแล้ว ระบบจะทำการระบุตัวตนของผู้ใช้ผู้นั้นว่าเป็นใครโดยการแสดงชื่อที่หน้าต่างของภาพที่กล้องจับ

3.1.2 การออกแบบส่วนของผู้ใช้งาน (Client)

การออกแบบในฝั่งผู้ใช้ประกอบด้วย 2 องค์ประกอบคือ กล้องสำหรับถ่ายภาพและเว็บเบราว์เซอร์ โดยกล้องถ่ายภาพที่ใช้จะใช้กล้องเว็บแคมที่ติดตั้งมากับเครื่องคอมพิวเตอร์อยู่แล้ว เป็นอุปกรณ์หลักในการนำเข้าสู่รูปภาพ และหัวใจสำคัญของเว็บเบราว์เซอร์คือ การใช้งาน WebRTC API ซึ่งเป็นเทคโนโลยีที่มีอยู่ในเบราว์เซอร์ส่วนใหญ่ในขณะนี้ ซึ่งฟังก์ชันหลักของ WebRTC API ที่นำมาใช้คือ getUserMedia() โดยฟังก์ชัน getUserMedia() เป็นฟังก์ชันที่ใช้สำหรับการร้องขอการเข้าถึงไปยังอุปกรณ์สื่อเข้า (กล้องเว็บแคม) โดยตรง และส่งข้อมูลภาพผ่านโพรโตคอล HTTPS ไปยังฝั่งเซิร์ฟเวอร์ ดังโครงสร้างของระบบซึ่งแสดงดังรูปที่ 3.2

3.1.3 การออกแบบการใช้งานระบบ (User Case Diagram)

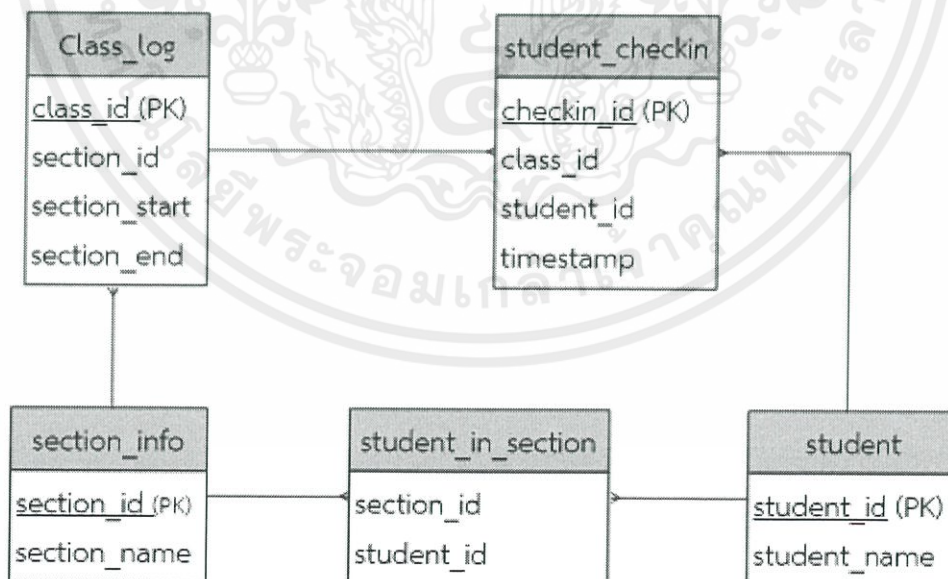
การออกแบบการใช้งานระบบ ได้ออกแบบให้อยู่บนหน้าเว็บไซต์ เพื่อความสะดวกต่อการใช้งานรูปแบบต่างๆ ได้แก่ การเพิ่มรายวิชาใหม่ รายชื่อนักศึกษาในรายวิชา รวมถึงการเก็บข้อมูลใบหน้าและตรวจสอบใบหน้าของนักศึกษา ระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติได้ เชื่อมเข้ากับฐานข้อมูลที่สร้างไว้เพื่อเก็บข้อมูลของผู้ที่ลงทะเบียนหรือดึงข้อมูลในฐานข้อมูลมาแสดงบนหน้าเว็บไซต์ ภาพรวมการใช้งานของระบบจัดถูกออกแบบแสดงดังรูปที่ 3.4



รูปที่ 3.4 ภาพรวมการใช้งานของระบบ

3.1.3.1 การออกแบบฐานข้อมูล (Database Management System)

ฐานข้อมูลเป็นส่วนสำคัญในการใช้งานระบบ หากข้อมูลที่สมควรที่จะอยู่ในฐานข้อมูลไม่ถูกจัดเก็บไว้อย่างเหมาะสม อาจทำให้ระบบไร้ประสิทธิภาพ ฐานข้อมูลจึงเปรียบเสมือนรากฐานที่สำคัญของระบบ การออกแบบโครงสร้างฐานข้อมูลให้เหมาะสมกับการใช้งาน ระบบฐานข้อมูลจึงเป็นส่วนสำคัญในการพัฒนาระบบ เนื่องจากข้อมูลในแต่ละตารางภายในฐานข้อมูลจะต้องมีความสัมพันธ์กันอย่างมีเหตุผล แสดงดังรูปที่ 3.5



รูปที่ 3.5 โครงสร้างฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในฐานข้อมูล ประกอบไปด้วยตารางต่างๆ โดยแต่ละตารางจัดเก็บรายละเอียดต่างๆ ตามคำอธิบาย และมีความสัมพันธ์กันดังตารางที่ 3.1 ถึง 3.6

ตารางที่ 3.1 รายชื่อตารางในฐานข้อมูล

| ลำดับ | ชื่อตาราง | คำอธิบาย |
|-------|--------------------|---|
| 1 | student | จัดเก็บข้อมูลนักศึกษา |
| 2 | section_info | จัดเก็บข้อมูลรายวิชา |
| 3 | student_in_section | จัดเก็บข้อมูลนักศึกษาในรายวิชา |
| 4 | class_log | จัดเก็บข้อมูลคาบเรียนของรายวิชา |
| 5 | student_checkin | จัดเก็บข้อมูลการลงบันทึกเวลาของนักศึกษา |

ตารางที่ 3.2 รายละเอียดตารางข้อมูลนักศึกษา

| Table Name : student | | | | |
|----------------------|--------------|------------|--------------|----------|
| ลำดับ | รายละเอียด | ชนิดข้อมูล | คำอธิบาย | หมายเหตุ |
| 1 | student_id | TEXT | รหัสนักศึกษา | PK |
| 2 | studebt_name | TEXT | ชื่อนักศึกษา | |

ตารางที่ 3.3 รายละเอียดตารางข้อมูลรายวิชา

| Table Name : section_info | | | | |
|---------------------------|--------------|------------|-------------|----------|
| ลำดับ | รายละเอียด | ชนิดข้อมูล | คำอธิบาย | หมายเหตุ |
| 1 | section_id | INTEGER | รหัสรายวิชา | PK |
| 2 | section_name | TEXT | ชื่อรายวิชา | |

ตารางที่ 3.4 รายละเอียดตารางข้อมูลนักศึกษาในรายวิชา

| Table name : student_in_section | | | | |
|---------------------------------|------------|------------|--------------|----------|
| ลำดับ | รายละเอียด | ชนิดข้อมูล | คำอธิบาย | หมายเหตุ |
| 1 | section_id | INTEGER | รหัสนักศึกษา | |
| 2 | student_id | TEXT | รหัสรายวิชา | |

ตารางที่ 3.5 รายละเอียดตารางข้อมูลคาบเรียนในรายวิชา

| Table Name : class_log | | | | |
|------------------------|---------------|------------|------------------------------------|----------|
| ลำดับ | รายละเอียด | ชนิดข้อมูล | คำอธิบาย | หมายเหตุ |
| 1 | class_id | INTEGER | รหัสคาบเรียน | PK |
| 2 | section_id | INTEGER | รหัสรายวิชา | |
| 3 | section_start | TEXT | วันและเวลาที่เริ่มเรียนของคาบเรียน | |
| 4 | section_end | TEXT | วันและเวลาที่เลิกเรียนของคาบเรียน | |

ตารางที่ 3.6 รายละเอียดตารางข้อมูลการลงบันทึกเวลาของนักศึกษา

| Table Name : student_checkin | | | | |
|------------------------------|------------|------------|-----------------------------------|----------|
| ลำดับ | รายละเอียด | ชนิดข้อมูล | คำอธิบาย | หมายเหตุ |
| 1 | checkin_id | INTEGER | รหัสเช็คชื่อ | PK |
| 2 | class_id | INTEGER | รหัสคาบเรียน | |
| 3 | student_id | TEXT | รหัสนักศึกษา | |
| 4 | timestamp | TEXT | วันและเวลาที่เข้าเรียนของนักศึกษา | |

3.1.3.2 การออกแบบส่วนต่อประสาน (Interface Design)

การออกแบบในส่วนของผู้ใช้ ผู้จัดทำได้ออกแบบหน้าเว็บไซต์เพื่อให้ผู้ใช้สามารถใช้งานได้สะดวกขึ้น โดยผู้ใช้งานสามารถเปิดระบบผ่านทางเว็บเบราว์เซอร์ (Web Browser) เช่น Google Chrome เป็นต้น หลังจากนั้นจะเข้าสู่การใช้งานระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ โดยแบ่งการใช้งานฟังก์ชันการใช้งานได้ดังรูปที่ 3.6



รูปที่ 3.6 รายการต่างๆ ที่เรียกใช้ผ่านเว็บเบราว์เซอร์

จากรูปที่ 3.6 เมื่อเปิดใช้งานระบบขึ้น จะพบกับการเรียกใช้งานในแบบต่างๆ โดยการทำงานเหล่านี้จัดเป็นการทำงานหลักๆ เพื่อให้ง่ายต่อการเข้าถึง และได้แบ่งการทำงานหลักเป็นการทำงานกลุ่มย่อยๆ โดยผู้ใช้งานสามารถเลือกการใช้งานที่ต้องการเรียกใช้ได้ แสดงการทำงานย่อยดังตารางที่ 3.7

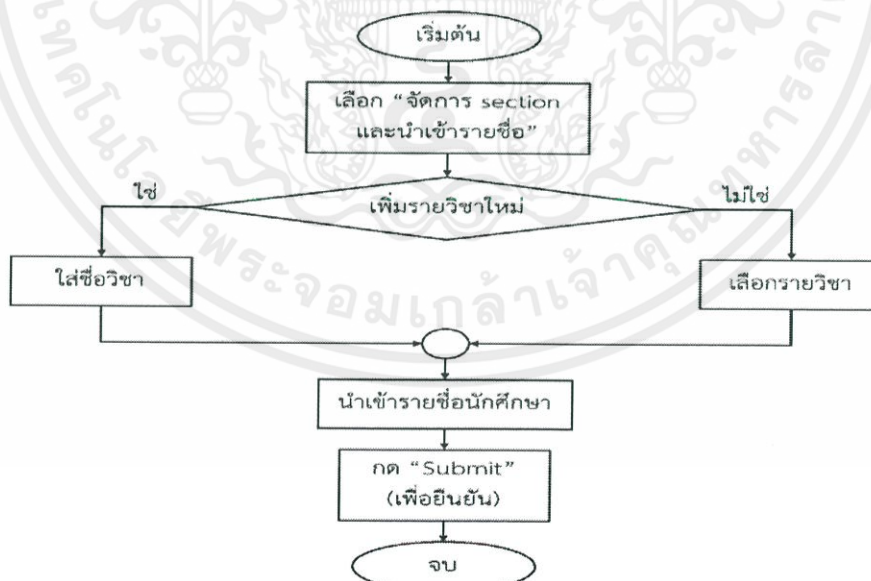
ตารางที่ 3.7 การทำงานของระบบในแต่ละกลุ่ม

| การทำงาน | การทำงานย่อย | คำอธิบาย |
|-------------------|--|--|
| 1. เช็คชื่อ | เลือกรายวิชา | เลือกวิชาในการเช็คชื่อ |
| | เปิดกล่อง | ชื่อนักศึกษา |
| | ผลการเช็คชื่อ | ระบุสถานะของนักศึกษา (มาเรียน, สาย, ขาดเรียน) และวันที่ทำการเช็คชื่อ |
| 2. จัดการคาบเรียน | สร้างคาบเรียน | กำหนดวันและเวลาของคาบเรียน |
| | รายการคาบเรียน และผลการเข้าเรียนรายคาบ | รายการคาบเรียนในแต่ละคาบ และผลการเข้าเรียนของนักศึกษาในแต่ละคาบเรียน |
| | เข้าชั้นเรียนแบบปกติ | จัดการกับปัญหาการตรวจจับใบหน้าไม่พบ |

ตารางที่ 3.7 การทำงานของระบบในแต่ละกลุ่ม (ต่อ)

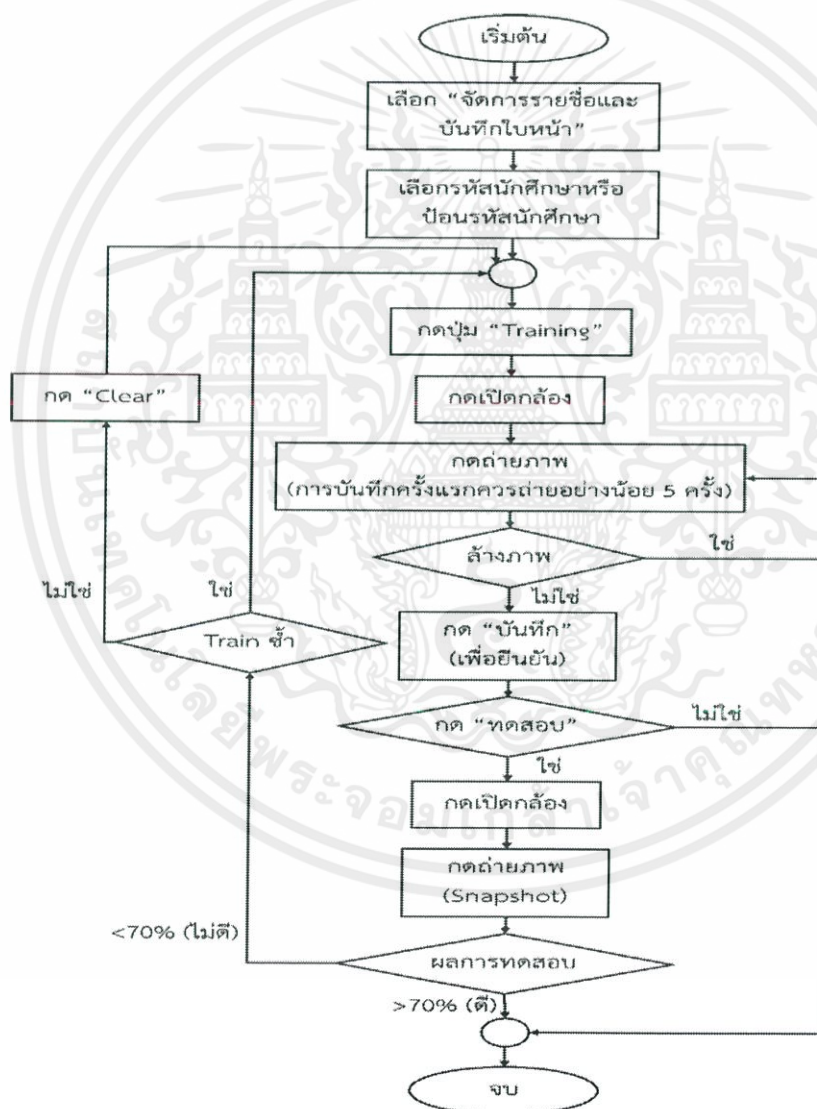
| การทำงาน | การทำงานย่อย | คำอธิบาย |
|---------------------------------------|---------------------------|--|
| 3. จัดการรายชื่อ และบันทึกใบหน้า | รายชื่อนักศึกษา | ตรวจสอบข้อมูลนักศึกษาเกี่ยวกับการ บันทึกใบหน้า |
| | บันทึกใบหน้า | บันทึกใบหน้านักศึกษา |
| | ทดสอบ | ทดสอบการบันทึกใบหน้าของนักศึกษา |
| 4. จัดการ section และนำเข้ารายชื่อ | บันทึกใบหน้า | บันทึกใบหน้านักศึกษา |
| | ทดสอบ | ทดสอบการบันทึกใบหน้าของนักศึกษา |
| | รายชื่อนักศึกษาใน section | ตรวจสอบรายชื่อนักศึกษาในรายวิชา และสามารถลบรายชื่อนักศึกษาได้ |
| 5. ออกรายงาน | รายงานการเข้าเรียนรายวิชา | สรุปผลการเข้าเรียนของนักศึกษาในแต่ละ รายวิชา |

จากตารางที่ 3.7 การใช้งานการทำงานต่างๆ ได้ถูกแบ่งออกเป็นส่วนใหญ่ทั้งหมด 3 ส่วน โดยส่วนแรกคือส่วนของการเพิ่มรายชื่อนักศึกษาและรายชื่อนักศึกษา ส่วนที่สองคือส่วนของการบันทึกและทดสอบใบหน้าของนักศึกษา และส่วนสุดท้ายคือส่วนของการตรวจสอบรายชื่อ การเข้าชั้นเรียนของนักศึกษา แสดงการทำงานในแต่ละส่วนดังรูปที่ 3.7 ถึง 3.9



รูปที่ 3.7 ขั้นตอนการใช้งานส่วนของการเพิ่มรายชื่อนักศึกษาและรายชื่อนักศึกษา

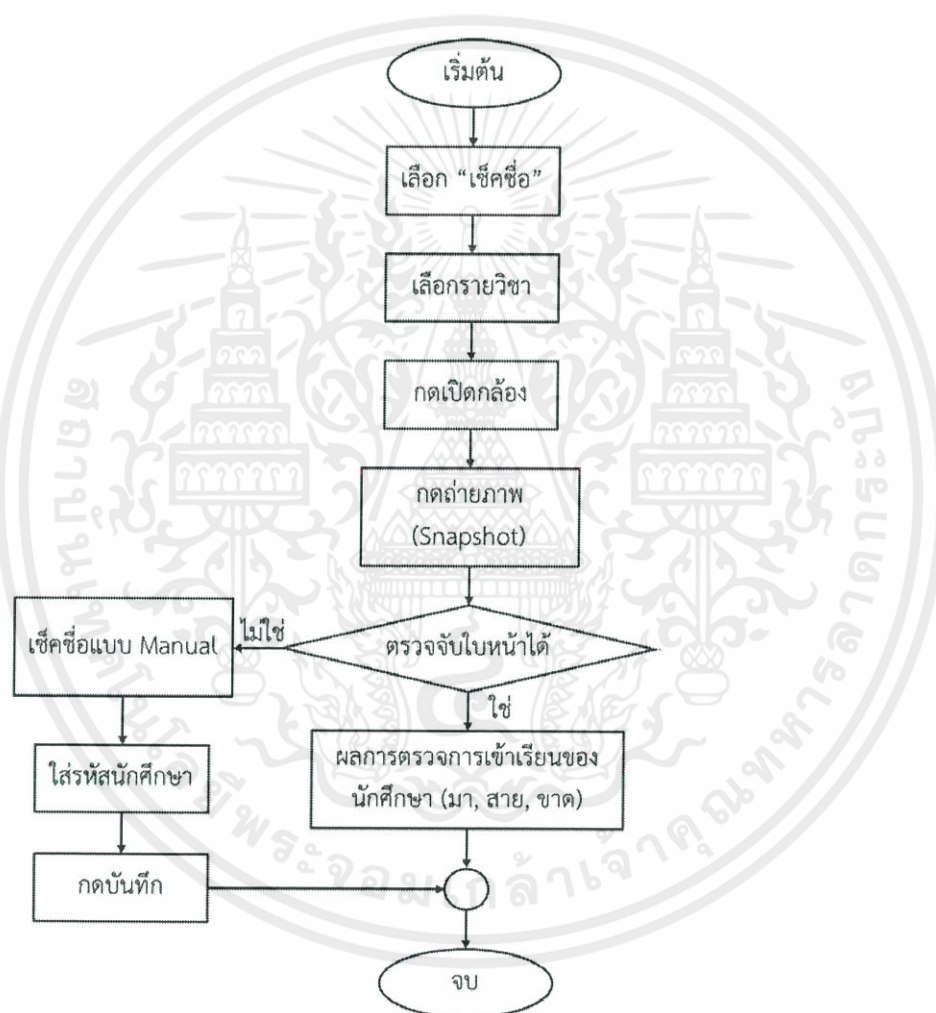
จากรูปที่ 3.7 เป็นส่วนที่จำเป็นอย่างยิ่ง หากไม่มีการเพิ่มรายชื่อวิชาและรายชื่อนักศึกษาจะทำให้ระบบขาดข้อมูลสำคัญในการตรวจสอบการเข้าชั้นเรียนของนักศึกษา โดยการเพิ่มรายชื่อของนักศึกษาได้ถูกออกแบบให้สามารถนำเขารายชื่อด้วยไฟล์ Excel โดยใช้โค้ด `pd.read_excel(section_file,encoding='TIS-620')` เป็นการอ่านไฟล์ Excel ที่ได้นำเข้าซึ่งช่วยให้สะดวกต่อการใช้งาน ในกรณีที่มีนักศึกษาจำนวนมาก การเพิ่มรายชื่อนักศึกษาแบบรายบุคคลจะทำให้สิ้นเปลืองระยะเวลาการใช้งานระบบ ส่วนนี้จึงเป็นการทำงานที่สำคัญที่จะช่วยอำนวยความสะดวกรวดเร็วแก่ผู้ใช้งานระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ



รูปที่ 3.8 ขั้นตอนการใช้งานส่วนของการบันทึกและทดสอบใบหน้าของนักศึกษา

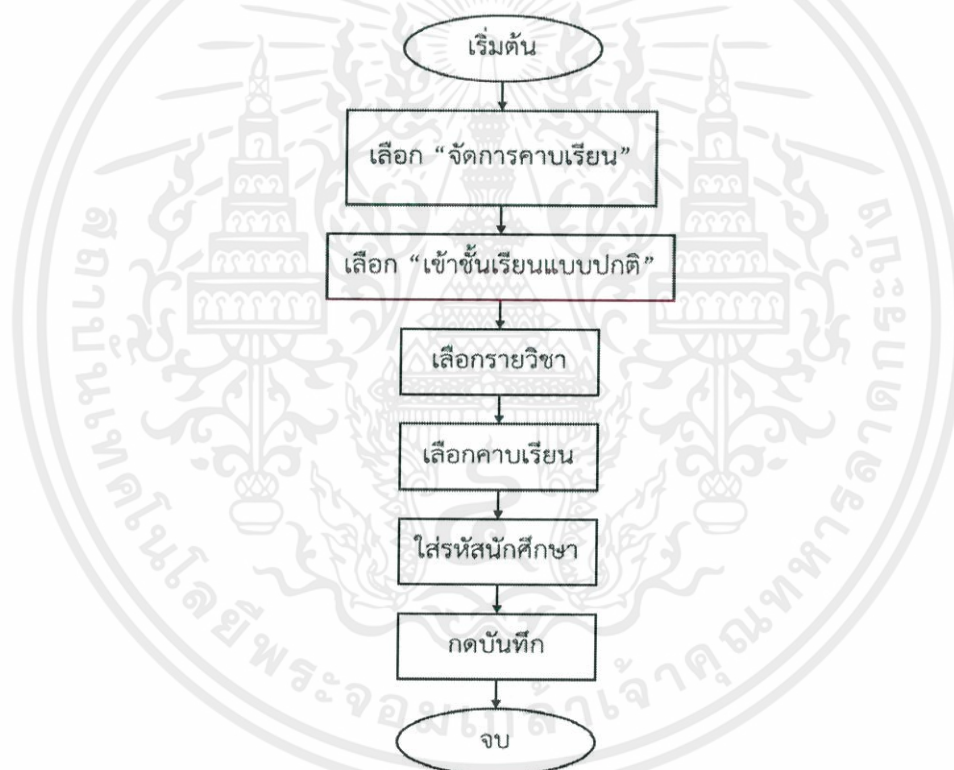
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 เป็นส่วนที่สองของระบบ เมื่อทำการเพิ่มรายชื่อวิชาและรายชื่อนักศึกษาแล้ว ขั้นตอนใช้งานต่อมาจะทำการบันทึกใบหน้านักศึกษา โดยการเปิดกล้องจะใช้ฟังก์ชัน getUserMedia() และการบันทึกใบหน้านักศึกษาจะทำการถ่ายภาพอย่างน้อย 5 รูป ซึ่งใช้ฟังก์ชัน SGDClassification อีกทั้งระบบมีการทดสอบใบหน้านักศึกษาหลังจากการบันทึกใบหน้า เพื่อความถูกต้องแม่นยำของการบันทึกใบหน้า หากการบันทึกใบหน้าไม่มีประสิทธิภาพเพียงพอ (ผลการทดสอบ <70%) สามารถทำการบันทึกใบหน้าใหม่ได้ทั้งบันทึกซ้ำหรือบันทึกใหม่ทั้งหมด



รูปที่ 3.9 ขั้นตอนการใช้งานส่วนของการเช็คชื่อนักศึกษา

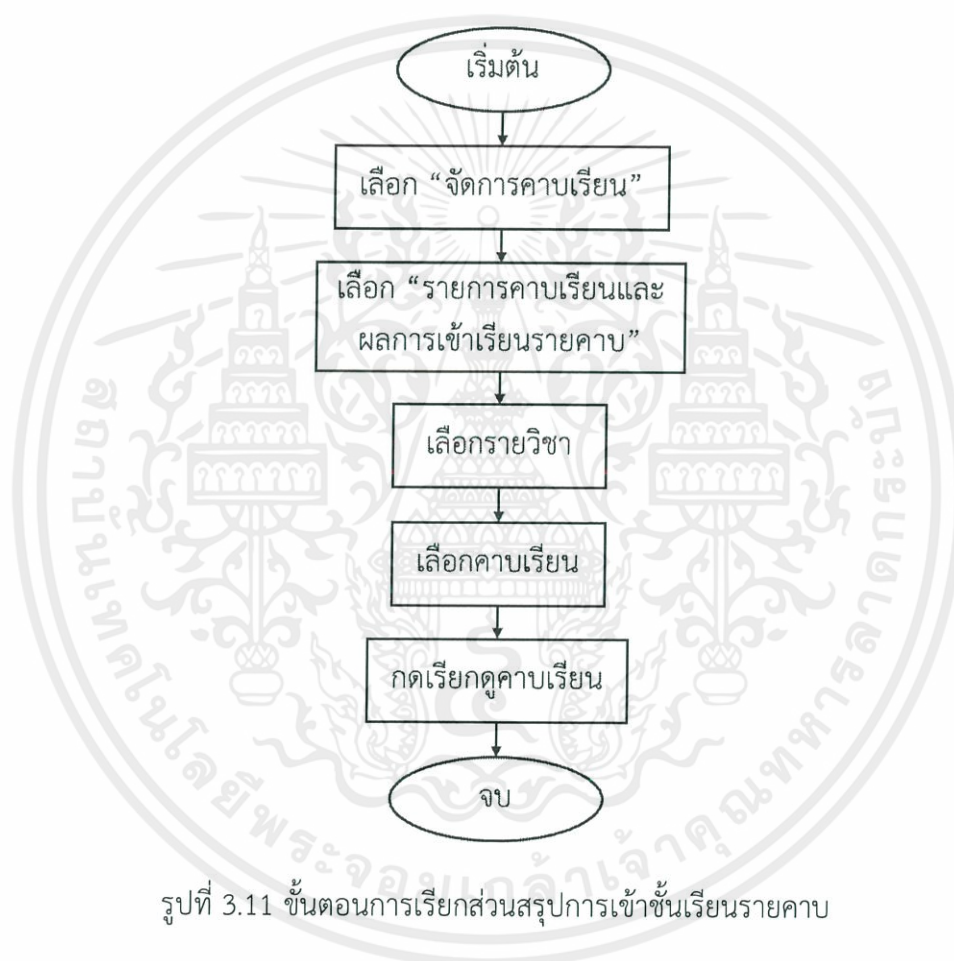
จากรูปที่ 3.9 เป็นส่วนของการตรวจสอบการเข้าเรียนของนักศึกษา ซึ่งประกอบไปด้วยระบบย่อยสองส่วนคือ ส่วนแรกจะเป็นการเช็คชื่อด้วยกล้อง หากทำการถ่ายภาพแล้วสามารถตรวจจับใบหน้าและระบุตัวบุคคลได้ โดยใช้ `fr.face_encodings()` จากนั้นจะถูกลงบันทึกการเข้าเรียนของนักศึกษาคนนั้น และส่วนที่สองเป็นส่วนเสริม กรณีระบบตรวจจับใบหน้าจากกล้องถ่ายภาพไม่ได้ ในส่วนนี้สามารถเช็คชื่อด้วยการระบุรหัสนักศึกษา เพื่อให้สามารถตรวจสอบรายชื่อการเข้าชั้นเรียนได้ประสิทธิภาพมากขึ้น หากกรณีไม่มีส่วนที่สอง การตรวจสอบการเข้าเรียนของนักศึกษาอาจเกิดการตกค้างได้ เช่นนักศึกษาคนหนึ่งมาเรียนแต่ไม่ผ่านการตรวจจับใบหน้า จะทำให้นักศึกษาคนนั้นขาดเรียนในคาบนั้นได้ จึงเป็นส่วนหนึ่งที่สำคัญเป็นอย่างยิ่งของระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ การใช้งานส่วนเสริมการเช็คชื่อเข้าเรียนแบบปกติ ดังรูปที่ 3.10



รูปที่ 3.10 ขั้นตอนการใช้งานส่วนเสริมของการเช็คชื่อเข้าชั้นเรียนแบบปกติ

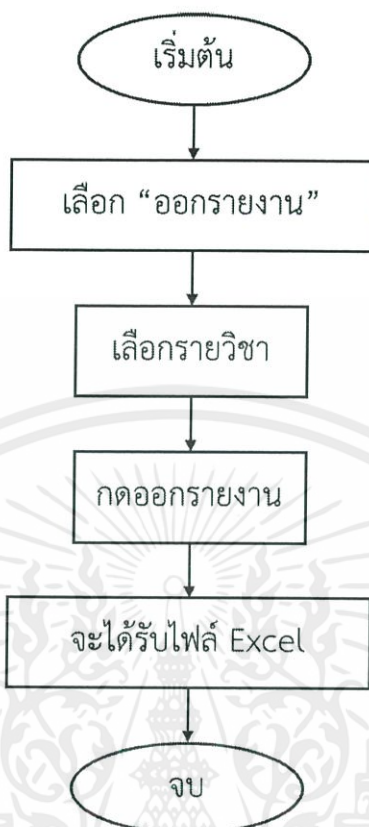
3.1.3.3 การออกแบบส่วนการออกรายงาน (History Report)

เป็นส่วนที่ผู้จัดทำออกแบบเพื่อสรุปการเข้าชั้นเรียนของนักศึกษาทั้งรายคาบและทั้งรายวิชา ซึ่งผู้ใช้สามารถตรวจสอบได้ว่านักศึกษามาเรียนกี่ครั้ง ตรงเวลา, สาย หรือขาดเรียนเป็นจำนวนกี่ครั้ง ซึ่งการใช้งานถูกแบ่งออกเป็นสองส่วนได้แก่ ส่วนแรกคือส่วนสรุปการเข้าชั้นเรียนรายคาบ และส่วนที่สองคือส่วนสรุปการเข้าชั้นเรียนรายวิชา การใช้งานแสดงดังรูปที่ 3.11 และ 3.12



รูปที่ 3.11 ขั้นตอนการเรียกส่วนสรุปการเข้าชั้นเรียนรายคาบ

จากรูปที่ 3.11 เป็นขั้นตอนของการเรียกดูผลการเช็คชื่อการเข้าชั้นเรียนของนักศึกษารายคาบ ซึ่งระบบการเข้าชั้นเรียนของนักศึกษาในแต่ละคาบ โดยการตรวจสอบนั้นจะเรียกดูในรายวิชา ต่อจากนั้นเข้าดูในคาบเรียนของวิชานั้น

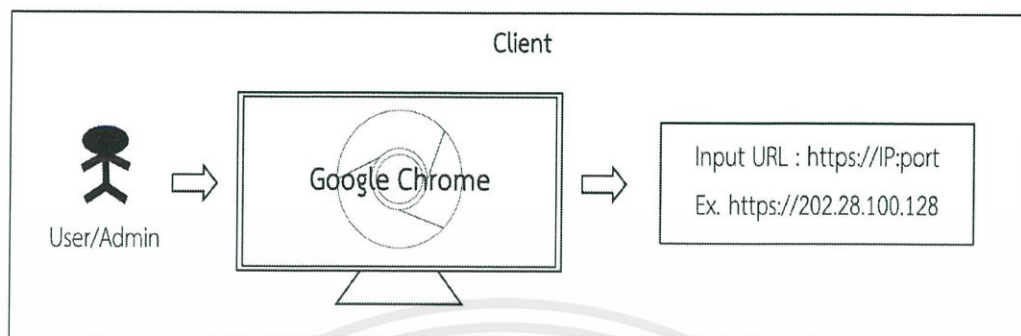


รูปที่ 3.12 ขั้นตอนการเรียกส่วนสรุปการเข้าชั้นเรียนรายวิชา

จากรูปที่ 3.12 เป็นขั้นตอนของการเรียกใช้งานส่วนสรุปการเข้าชั้นเรียนของนักศึกษารายวิชา ซึ่งการสรุปผลในส่วนจะถูกออกแบบให้มีการรายงานด้วยไฟล์ Excel ด้วยฟังก์ชัน `exportexcel(data)` โดยระบบมีการเลือกวิชาและกดออกรายงาน ผู้ใช้งานจะได้รับไฟล์ Excel ภายในอุปกรณ์ที่เรียกใช้งานระบบนี้

3.1.3.4 การออกแบบส่วนการเรียกใช้งานระบบ

ในส่วนการเรียกใช้งานของระบบผู้ใช้งานรวมทั้งผู้ดูแลระบบ สามารถเรียกใช้งานได้จากเว็บเบราว์เซอร์ได้ ด้วยฟังก์ชัน `app.run()` โดยใส่ข้อมูลที่อยู่ของเครื่องเซิร์ฟเวอร์ (IP Address) ที่จะเรียกใช้งาน เท่านั้นก็สามารถใช้งานระบบได้ โดยขั้นตอนการเรียกใช้งาน แสดงดังรูปที่ 3.13



รูปที่ 3.13 แผนภาพการเรียกใช้งานระบบ

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Stochastic Gradient Descent Classifier (SGD Classifier)

เทคนิคการตัดแยกใบหน้าจะใช้ไลบรารี `sklearn.linear_model` ซึ่งเป็น Machine Learning และใช้เทคนิค SGD Classifier เป็นเทคนิคการเรียนรู้ข้อมูลใบหน้า ตัวอย่างการนำเข้าไลบรารีคือ `from sklearn.linear_model import SGDClassifier` และตัวอย่างการใช้งานคือ `SGDClassifier(loss='modified_huber')` โดย `loss='modified_huber'` เป็นการประมาณค่าความน่าจะเป็น ซึ่งนำมาใช้ในการตัดแยกใบหน้า อีกทั้งยังสามารถพยากรณ์หรือระบุตัวตนของภาพใบหน้านักศึกษาที่ได้จากการเก็บข้อมูลในรูปแบบของ `BitArray` มีจำนวนทั้งหมด 128 ชุดต่อหนึ่งอาเรย์ต่อหนึ่งภาพใบหน้า และเปรียบเทียบชุดอาเรย์เหล่านี้เพื่อระบุตัวตนของภาพใบหน้า ซึ่ง `BitArray` ทั้ง 128 ชุด ได้มาจากการวัดระยะห่างระหว่างจุดเฉพาะบนใบหน้า

3.2.2 ไลบรารีที่เกี่ยวข้อง

3.2.1.1 ไลบรารี Pandas

Pandas เป็นไลบรารีใน python ที่เกี่ยวกับการจัดการ Data frame สำหรับการประมวลผล ใช้ในการนำเข้รายชื่อนักศึกษาด้วยไฟล์ Excel ซึ่งการใช้งาน เช่น `pd.read_excel(section_file,encoding='TIS-620')`

3.2.1.2 ไลบรารี Flask และ CORS

เป็นไลบรารีสำหรับสร้าง Web server ตัวอย่างการใช้งาน เช่น `app = Flask(__name__, static_url_path='/www2'); CORS(app)`

3.2.1.3 ไลบรารี os

เป็นไลบรารีสำหรับจัดการกับระบบปฏิบัติการ ตัวอย่างการใช้งาน เช่น `os.path.split()`

3.2.1.4 ไลบรารี pickle

เป็นไลบรารีสำหรับจัดการกับไฟล์ ซึ่งเป็นการแปลงวัตถุให้เป็นข้อมูลในรูปแบบบิต ใช้งานไลบรารี pickle ในการเก็บข้อมูล โดยมีรูปแบบ `pickle.dump(วัตถุของข้อมูล,open("ชื่อไฟล์ที่ต้องการบันทึก.นามสกุล","wb"))` ตัวอย่างการใช้งานในการเก็บข้อมูล เช่น `pickle.dump([encodings,names], open(datasets, 'wb'))`

3.2.1.5 ไลบรารี xlrd

เป็นไลบรารีสำหรับอ่านข้อมูลจากไฟล์ Excel

3.2.1.5 ไลบรารี BytesIO และ base64

เป็นไลบรารีสำหรับอ่านหรือเขียนไฟล์ และเข้ารหัสข้อมูล

3.2.3 การเข้าถึงอุปกรณ์กล้องถ่ายภาพ

การออกแบบระบบให้สามารถใช้งานอยู่บนเว็บเบราว์เซอร์นั้น ได้นำฟีเจอร์ getUserMedia API ที่เป็นส่วนหนึ่งของมาตรฐาน WebRTC ทำให้เบราว์เซอร์สามารถเข้าถึงกล้องเว็บแคมของคอมพิวเตอร์ได้ ยกตัวอย่างการใช้งาน เช่น getUserMedia(video: true)

3.2.4 การออกรายงาน

การออกรายงานในระบบนี้ เป็นการออกรายงานในรูปแบบไฟล์ excel ซึ่งใช้ฟังก์ชัน exportexcel(data)

3.2.5 การทำงานของ Server

การเริ่มการทำงานของ Server จะใช้ฟังก์ชันการทำงาน คือ app.run() ซึ่งการใช้งานในระบบนี้เป็น app.run(debug=True, host='0.0.0.0', ssl_context='adhoc') ได้กำหนด host='0.0.0.0' หมายถึงอนุญาตให้เข้าถึงได้จากทุก IP Address และได้กำหนดการเปิดใช้โปรโตคอล SSL/HTTPS โดยมี Certificate แบบ Self-Sign คือ ssl_context='adhoc'

3.2.6 การเชื่อมต่อฐานข้อมูล

ในการเชื่อมต่อฐานข้อมูล จะใช้ไลบรารี sqlite3 ซึ่งเป็นไลบรารีของระบบฐานข้อมูล SQLite โดยมีการเชื่อมต่อ คือ con = sqlite3.connect('testDB6.db') testDB6 คือชื่อของฐานข้อมูลที่ใช้

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบในส่วนระบบรู้จำใบหน้า (Face Detection and Recognition)

- 1) ในส่วนของการลงทะเบียนสำหรับนักศึกษารายใหม่และการตรวจจับใบหน้า (Face Detection) โดยเริ่มใช้งานโปรแกรม เพื่อทดลองสแกนใบหน้าผ่านกล้อง Webcam
- 2) ในส่วนของนักศึกษาที่เคยลงทะเบียนแล้ว ทดลองใช้โปรแกรมเพื่อให้ระบบสามารถรู้จำใบหน้า (Face Recognition) จากภาพที่เก็บไว้ในฐานข้อมูล
- 3) สังเกตผลที่ได้จากกระบวนการของระบบรู้จำใบหน้า

- 4) บันทึกผลที่ได้ ทำการเปลี่ยนใบหน้า โดยการทดลองสแกนใบหน้ากับผู้ใช้คนอื่น ๆ เพื่อดูผลการทดลอง และเช็คความผิดพลาดของระบบเพื่อทำการแก้ไขต่อไป
- 5) ส่งค่าที่ได้จากการตรวจจับใบหน้าไปแสดงผลบนหน้าเว็บไซต์
- 6) บันทึกประวัติการเข้าชั้นเรียนไปแสดงผลบนหน้าเว็บไซต์



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

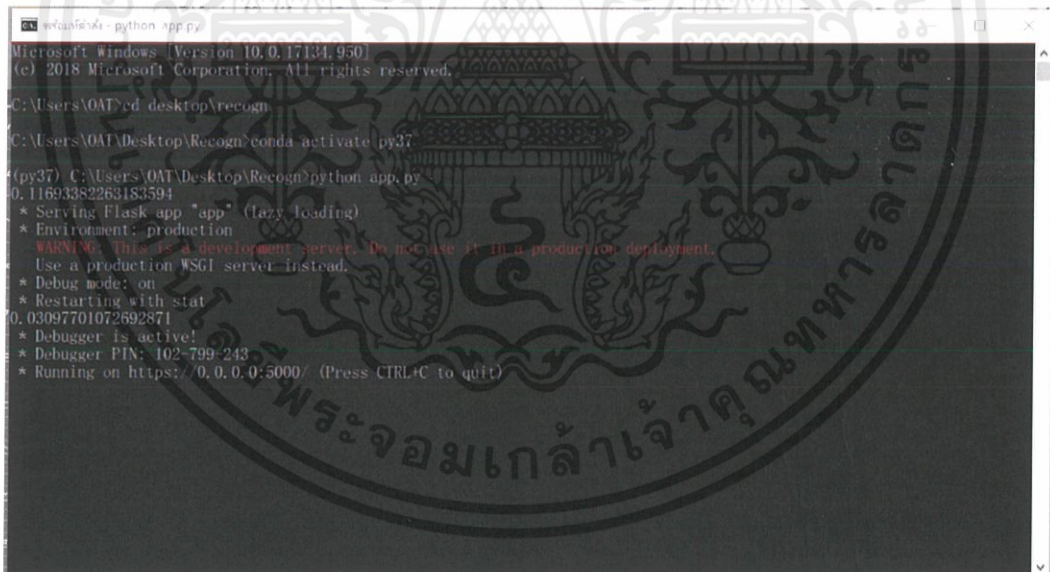
บทที่ 4

ผลการทดลอง

ในการทดลองระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียน ได้ทำการสังเกตและบันทึกผลการทดลองต่าง ๆ โดยมีการแบ่งการทดสอบและจัดเก็บผลเป็นส่วนต่าง ๆ ดังต่อไปนี้

4.1 ผลการทดสอบการเรียกใช้งานหน้าเว็บไซต์

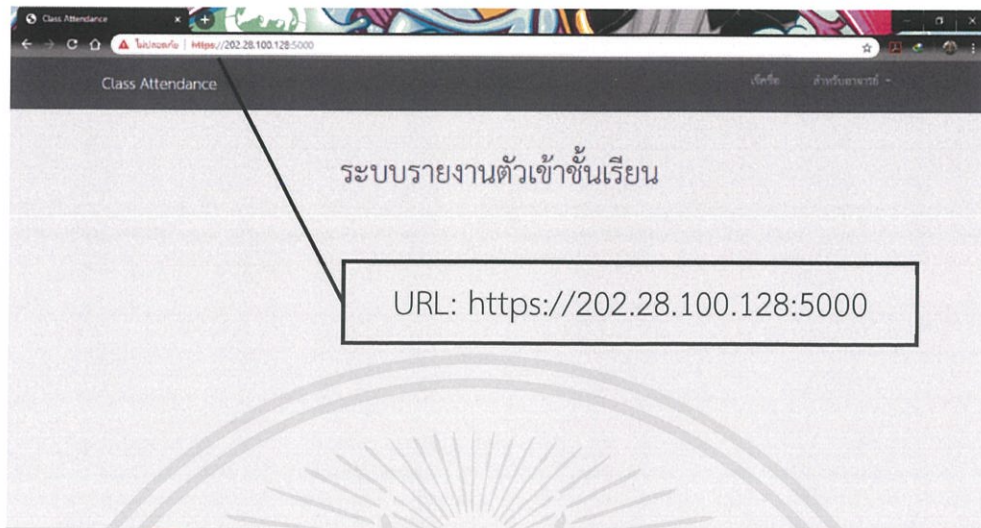
จากการออกแบบระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ ผู้ใช้งานสามารถเรียกใช้งานระบบได้ โดยผู้ใช้งานจะต้องเรียกใช้งานระบบผ่านเว็บเบราว์เซอร์เป็นหลัก เมื่อระบบเริ่มทำงาน แสดงดังรูปที่ 4.1 และผู้ใช้งานเรียกใช้งานระบบ จำเป็นต้องใช้เว็บเบราว์เซอร์ในการเรียกใช้งาน โดยผู้ใช้เปิดเว็บเบราว์เซอร์ ต่อจากนั้นใส่ URL ในช่องที่ใส่ URL ของเว็บเบราว์เซอร์นั้น ซึ่งการทดสอบนี้จะใช้ URL คือ <https://202.28.100.128:5000> จากนั้นเราก็จะเข้าสู่หน้าเว็บไซต์หลักของระบบรู้จำใบหน้า แสดงดังรูปที่ 4.2



```
python app.py
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\OAT>cd desktop\recogn
C:\Users\OAT\Desktop\Recogn>conda activate py37
(py37) C:\Users\OAT\Desktop\Recogn>python app.py
0.11693382263183594
* 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
* Restarting with stat
0.03097701072692871
* Debugger is active!
* Debugger PIN: 102-799-243
* Running on https://0.0.0.0:5000/ (Press CTRL+C to quit)
```

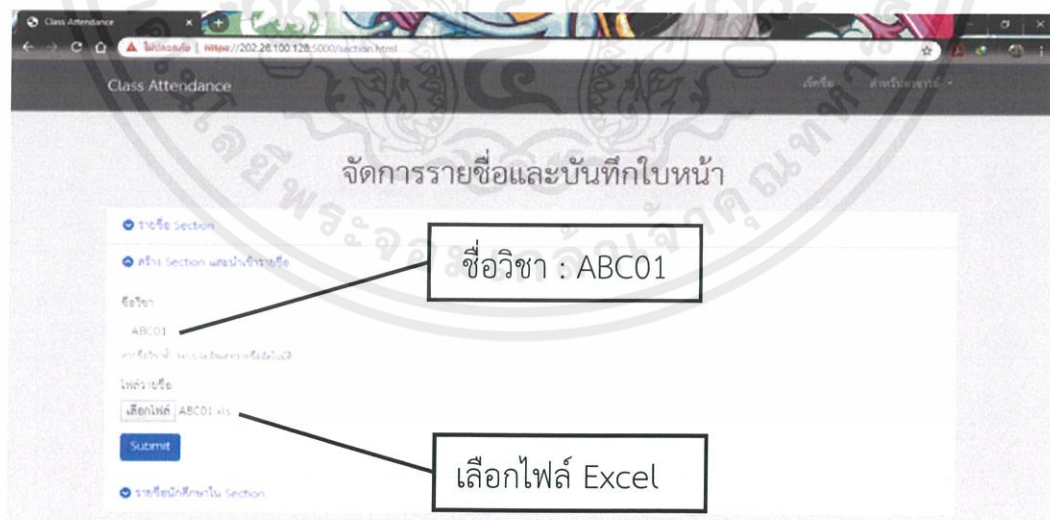
รูปที่ 4.1 หน้าจอพร้อมที่คำสั่งของการทำงาน



รูปที่ 4.2 หน้าเว็บไซต์หลักของระบบโดยที่ URL คือ <https://202.28.100.128:5000>

4.2 ผลการทดสอบการเพิ่มรายวิชาและรายชื่อนักศึกษา

กรณีที่ไม่มีรายวิชาในระบบที่เปิดใช้งาน ผู้ใช้สามารถเพิ่มรายวิชาใหม่ได้ โดยการทดสอบนี้จะใช้ชื่อวิชา ABC01 แสดงดังรูปที่ 4.3 จากนั้นทำการเพิ่มรายชื่อนักศึกษาเข้าไปในวิชา ABC01 โดยการเพิ่มสามารถเพิ่มได้ด้วยไฟล์ Excel ซึ่งตัวอย่างข้อมูลภายในไฟล์ Excel แสดงดังรูปที่ 4.4 ผลการเพิ่มรายวิชาและเพิ่มรายชื่อนักศึกษาภายในวิชา แสดงดังรูปที่ 4.5



รูปที่ 4.3 การเพิ่มรายวิชาและเพิ่มรายชื่อนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | A | B | C | D | E | F | G | H | I | J |
|----|----|----------|------------------------------|---|---|---|---|---|---|---|
| 8 | 1 | 58011448 | นายธีรพงศ์ ภูอกเงิน | | | | | | | |
| 9 | 2 | 58010022 | นายกรวิชญ์ เขื่อนทอง | | | | | | | |
| 10 | 3 | 58010040 | นางสาวกฤติยาภรณ์ เหมือตขุนทด | | | | | | | |
| 11 | 4 | 58010044 | นายกฤษฏีชานนท์ ทองเนาวรัตน์ | | | | | | | |
| 12 | 5 | 58010062 | นางสาวกัญญาพร ฐูเรียนทอง | | | | | | | |
| 13 | 6 | 58010065 | นางสาวกัญญารัตน์ คำราม | | | | | | | |
| 14 | 7 | 58010067 | นายกัณฐ์ เตโชภักดี | | | | | | | |
| 15 | 8 | 58010070 | นายกันตธร ประทีนพงษ์ | | | | | | | |
| 16 | 9 | 58010072 | นายกัปตัน ประทุมศรี | | | | | | | |
| 17 | 10 | 58010088 | นายกิตติกรีน วามวรรัตน์ | | | | | | | |
| 18 | 11 | 58010094 | นายกิตติกริต พชรแก้ว | | | | | | | |
| 19 | 12 | 58010103 | นายกิตติพงษ์ ทองคงหาญ | | | | | | | |
| 20 | 13 | 58010115 | นางสาวกมลลภรี สีราห์ใหม่ | | | | | | | |
| 21 | 14 | 58010179 | นางสาวจินตนา ศรีพุทธา | | | | | | | |
| 22 | 15 | 58010187 | นายจิรภัทร กิจท่า | | | | | | | |
| 23 | 16 | 58010190 | นายจิรภัทร ฐูเรียน | | | | | | | |
| 24 | 17 | 58010194 | นายจิรภัฏ สีทองธรรม | | | | | | | |
| 25 | 18 | 58010211 | นายเกษฎา สรรสวัสดิ์ | | | | | | | |
| 26 | 19 | 58010256 | นายโรเจอร์ เทวีวิมลภ | | | | | | | |

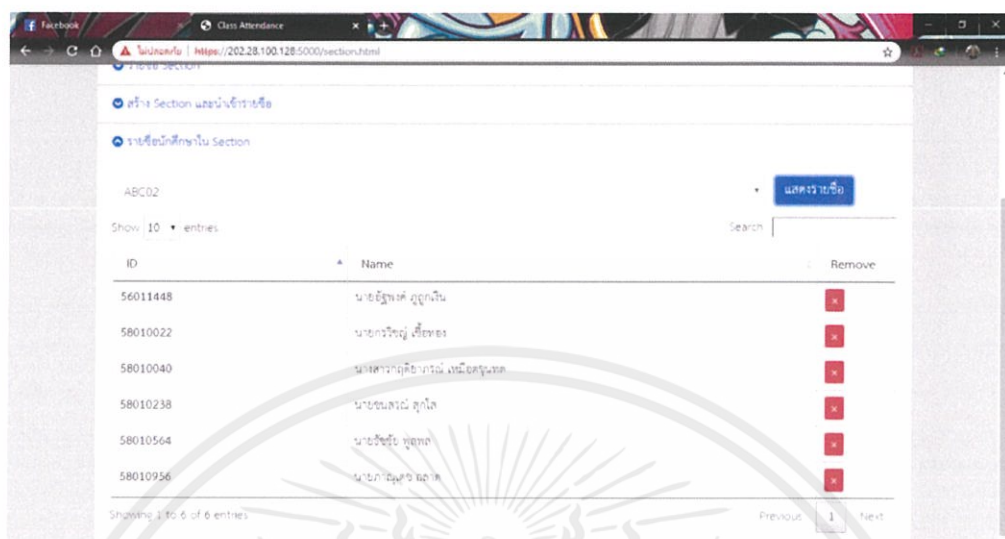
รูปที่ 4.4 ข้อมูลภายในไฟล์ Excel ที่ต้องการจะเพิ่มรายชื่อนักศึกษา



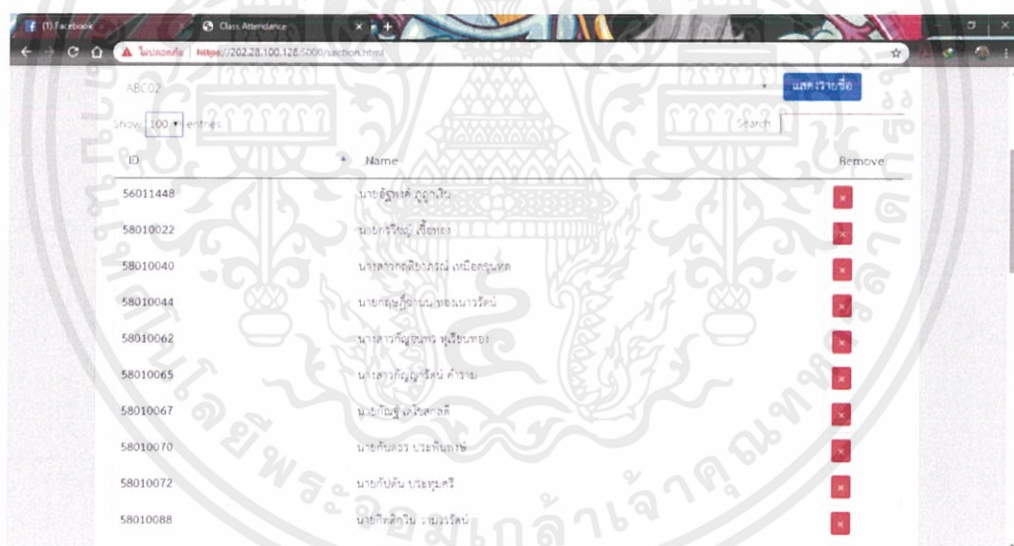
รูปที่ 4.5 ผลการเพิ่มรายชื่อนักศึกษาภายในรายวิชาใหม่

กรณีที่มีรายวิชาอยู่ในระบบแล้ว หากต้องการเพิ่มรายชื่อนักศึกษาภายในวิชานั้น สามารถเพิ่มได้ โดยรหัสนักศึกษาที่ถูกเพิ่มใหม่ จะไม่สามารถเพิ่มซ้ำได้ แต่ถ้ารหัสนักศึกษาที่ไม่ซ้ำกับรหัสนักศึกษาที่มีอยู่ในระบบ จะถูกเพิ่มลงในระบบ ตัวอย่างการเพิ่มรายชื่อนักศึกษาในรายวิชา ABC02 โดยใช้รายชื่อจากรายวิชา ABC01 แสดงดังรูปที่ 4.6 และ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



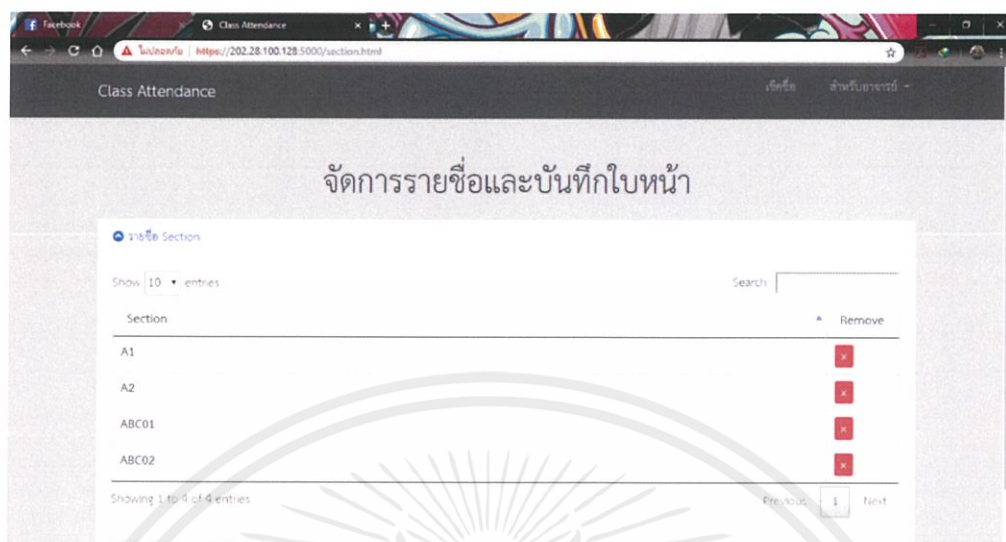
รูปที่ 4.6 ก่อนการทดสอบเพิ่มรายชื่อนักศึกษาเข้ารายวิชา ABC02



รูปที่ 4.7 ผลการเพิ่มรายชื่อนักศึกษาในรายวิชา ABC02

เมื่อเพิ่มรายวิชาเรียบร้อยแล้ว สามารถตรวจสอบรายวิชาที่ถูกเพิ่มเข้าระบบได้ โดยผู้ใช้เข้าไปหน้าเว็บไซต์หน้ารายชื่อ section ในหน้าเว็บไซต์นี้แสดงรายวิชาทั้งหมดที่อยู่ในระบบ แสดงดังรูปที่ 4.8

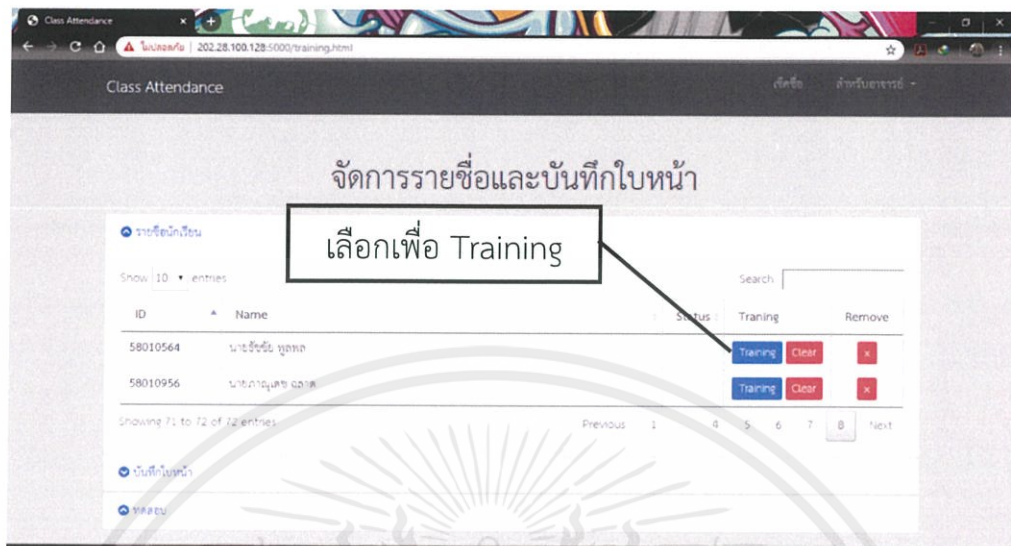
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 รายชื่อวิชาทั้งหมดที่อยู่ในระบบ

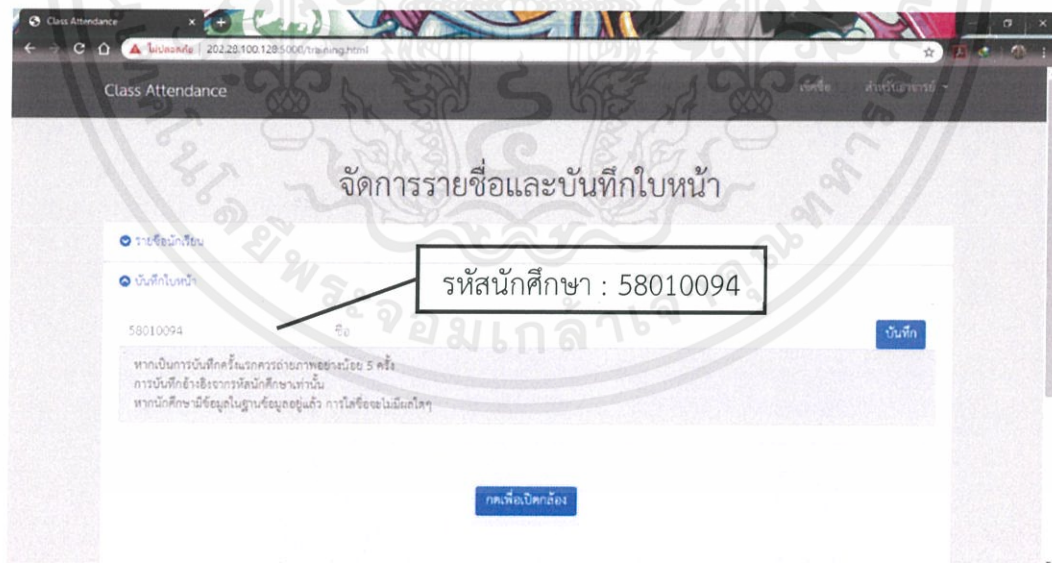
4.3 ผลการทดสอบการบันทึกใบหน้านักศึกษา

การบันทึกข้อมูลใบหน้านักศึกษา ทำการเปิดเว็บไซต์ในหน้าจัดการรายชื่อและบันทึกใบหน้า แสดงดังรูปที่ 4.9 จากนั้นทำการเลือกนักศึกษาคนที่ต้องการจะบันทึกใบหน้า และกดปุ่ม Training หรือกรอกรหัสนักศึกษาในช่องบันทึกใบหน้า แสดงดังรูปที่ 4.10 จากนั้นกดเปิดกล้อง Webcam ให้ทำการนำใบหน้าของนักศึกษามาอยู่หน้ากล้อง และกดถ่ายภาพใบหน้า ในการทดลองได้ทำการถ่ายภาพไว้อย่างน้อย 5 ภาพ แสดงดังรูปที่ 4.11 ถึง 4.30 หลังการบันทึกใบหน้าจะแสดงรูปที่ถ่ายไว้ หากพบว่ารูปไม่ชัดหรือการถ่ายภาพเกิดข้อผิดพลาด สามารถลบรูปเหล่านั้นได้ และระบบขึ้นสถานะของนักศึกษาที่ทำการบันทึกใบหน้าแล้ว แสดงดังรูปที่ 4.31 หลังจากนั้นระบบจะทำการทดสอบการระบุตัวตนของนักศึกษาที่ได้ทำการบันทึกข้อมูลใบหน้าเพื่อตรวจสอบว่าระบบได้ระบุตัวตนได้ถูกต้อง แสดงดังรูป 4.32 และ 4.33



รูปที่ 4.9 เว็บไซต์หน้าจัดการรายชื่อและบันทึกใบหน้า

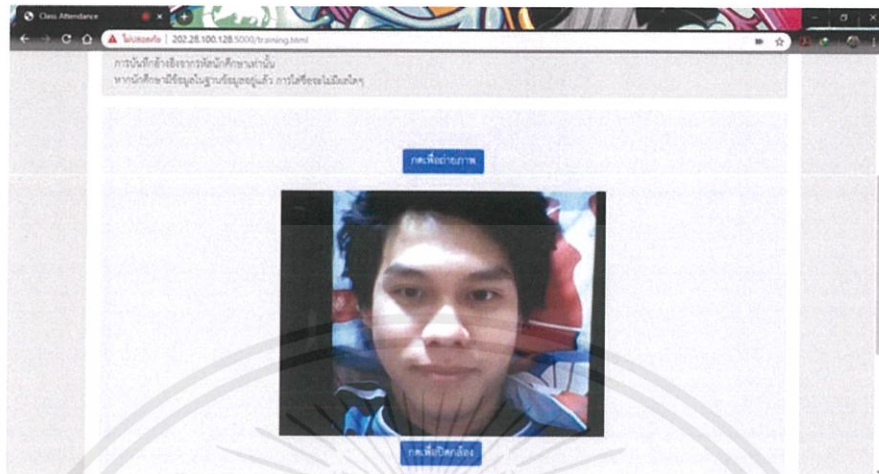
กรณีที่ผู้ต้องการเลือกกรอกรหัสนักศึกษาเพื่อบันทึกใบหน้านักศึกษา สามารถทำได้ โดยเปิดแถบการบันทึกใบหน้า และกรอกรหัสนักศึกษา แสดงดังรูปที่ 4.10



รูปที่ 4.10 การกรอกรหัสนักศึกษาเพื่อการบันทึกใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010044



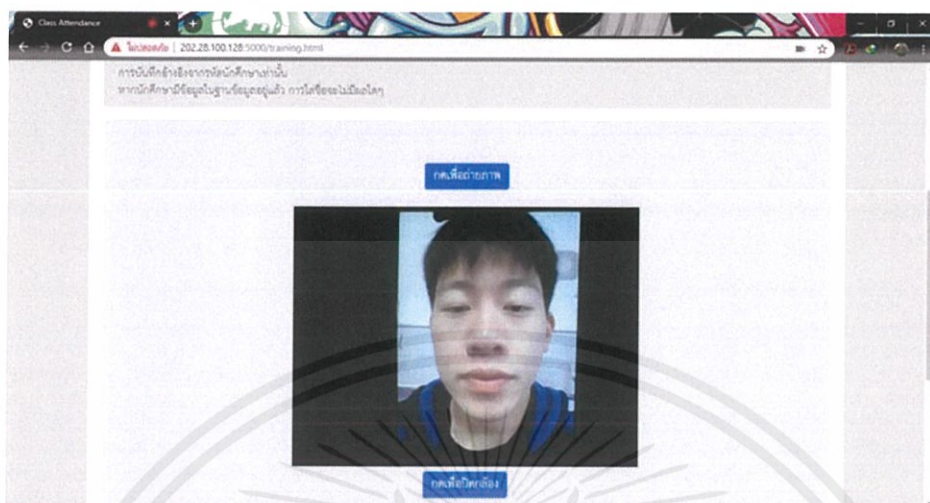
รูปที่ 4.11 การบันทึกใบหน้าของนักศึกษารหัส 58010044

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.08228613 | 0.11812214 | 0.00625804 | -0.05568115 | -0.05235356 | -0.0038129 |
| -0.02317148 | -0.14090501 | 0.15313834 | -0.0916836 | 0.29457161 | -0.0802711 |
| -0.25534177 | -0.11846285 | -0.05206042 | 0.15277345 | -0.17774409 | -0.14025609 |
| 0.01966929 | -0.03136818 | 0.10158467 | -0.08885203 | 0.05848981 | 0.10717908 |
| -0.08938231 | -0.34739766 | 0.13533689 | -0.08168744 | 0.02141201 | -0.04598615 |
| 0.0712411 | 0.02086236 | 0.21201941 | 0.04415194 | 0.02085534 | 0.04267056 |
| -0.05607793 | -0.05473616 | 0.20390888 | -0.06962498 | -0.20927481 | 0.02787533 |
| 0.04686857 | 0.23069297 | 0.1357055 | 0.03732274 | 0.01274693 | -0.11508063 |
| 0.0808837 | -0.14523177 | 0.10223465 | 0.18175381 | 0.14405899 | 0.02686751 |
| -0.0190852 | -0.12410412 | -0.03452521 | 0.12435593 | -0.15623872 | 0.04586466 |
| 0.02312536 | -0.06394921 | -0.02298672 | -0.08467919 | 0.20771675 | 0.1632245 |
| -0.12210248 | -0.22385554 | 0.14750238 | -0.14815608 | -0.04818648 | 0.11818606 |
| -0.16809359 | -0.25091112 | -0.28164372 | 0.08364792 | 0.38553473 | 0.10152228 |
| -0.14429353 | 0.05682839 | -0.04805097 | -0.02455491 | 0.12094944 | 0.16174373 |
| -0.03906605 | 0.0334446 | -0.11867549 | 0.05032605 | 0.18403751 | -0.1119795 |
| -0.11622878 | 0.23294148 | -0.0334762 | 0.12205443 | 0.01556523 | 0.07116496 |
| -0.0276518 | 0.09721185 | -0.15728159 | 0.03764775 | 0.04420587 | -0.00835547 |
| -0.03691162 | 0.06172547 | -0.09139322 | 0.02754887 | 0.02268745 | 0.02487736 |
| -0.01786834 | 0.01287384 | -0.15490428 | -0.11264712 | 0.09801654 | -0.30540821 |
| 0.22923309 | 0.1566391 | 0.06829729 | 0.1134986 | 0.13455799 | 0.05639851 |
| 0.1005548 | -0.10965537 | -0.18488266 | 0.01817664 | 0.15956439 | -0.00645527 |
| 0.07901745 | -0.00500085 | 58010044 | | | |

รูปที่ 4.12 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010044

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010088



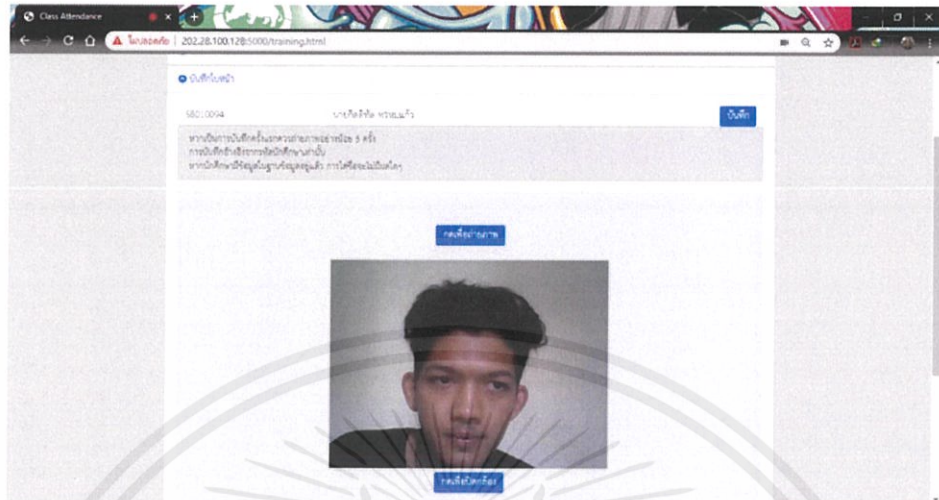
รูปที่ 4.13 การบันทึกใบหน้าของนักศึกษารหัส 58010088

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.00336887 | 0.08102737 | -0.00518458 | 0.02063056 | -0.06855802 | -0.06025418 |
| -0.03291048 | -0.11567919 | 0.11743043 | -0.09083122 | 0.29736573 | -0.05122384 |
| -0.1876651 | 0.09584801 | 0.01784733 | 0.1323238 | -0.16449901 | -0.08034968 |
| 0.00537529 | -0.04339559 | 0.05861536 | -0.01383194 | 0.07430177 | 0.11852349 |
| -0.05704341 | -0.2525039 | -0.15609737 | -0.08436742 | -0.09936544 | -0.06675709 |
| -0.06184464 | 0.03694262 | -0.1316721 | -0.02665672 | 0.04461583 | -0.00165967 |
| -0.03138823 | -0.06462476 | 0.22278471 | -0.02803666 | -0.17033216 | -0.01041206 |
| 0.01990088 | 0.26196399 | 0.15076195 | 0.05220705 | 0.01817447 | -0.09142759 |
| 0.05316619 | -0.09123093 | 0.08897565 | 0.17710251 | 0.17748874 | 0.03634067 |
| -0.07402603 | -0.16072536 | -0.0323198 | 0.12831201 | -0.08433924 | 0.03378224 |
| 0.07654714 | -0.03914509 | 0.01375936 | -0.05722796 | 0.2768802 | 0.09755772 |
| -0.14863807 | -0.16319387 | 0.12036095 | -0.16551603 | -0.09451895 | 0.06641461 |
| -0.19946772 | -0.18033186 | -0.21775989 | 0.02495927 | 0.37485278 | 0.07242066 |
| -0.18274437 | 0.03343688 | -0.09239829 | 0.00724946 | 0.07085573 | 0.17016265 |
| -0.06603947 | -0.01366534 | -0.09117888 | -0.01949921 | 0.21408169 | -0.10592384 |
| -0.05636956 | 0.20947415 | -0.00501809 | 0.0942307 | 0.01682938 | -0.05384384 |
| 0.02472181 | 0.04291298 | -0.07959051 | 0.02768433 | -0.05728498 | -0.09666199 |
| -0.02836491 | 0.09199663 | -0.18820651 | 0.12452117 | 0.06420951 | 0.03819554 |
| 0.04121532 | 0.09130754 | -0.13744578 | -0.11369581 | 0.14430048 | -0.2718491 |
| 0.22860515 | 0.16508818 | 0.13955918 | 0.12856916 | 0.08109026 | 0.17769256 |
| 0.02863637 | -0.05333255 | -0.16327885 | 0.01479611 | 0.0951054 | -0.05798641 |
| 0.06308959 | 0.01245065 | | | | 58010088 |

รูปที่ 4.14 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010088

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010094



รูปที่ 4.15 การบันทึกใบหน้าของนักศึกษารหัส 58010094

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.18809742 | 0.00812377 | 0.01667492 | -0.03164467 | -0.07080788 | -0.04667572 |
| -0.00066381 | -0.08402891 | 0.06975542 | -0.08603125 | 0.26068875 | -0.02566359 |
| -0.17368422 | -0.21399403 | 0.0264445 | 0.16573328 | -0.11361632 | -0.1620001 |
| -0.04987351 | 0.01736188 | 0.07169788 | 0.00404541 | -0.0119832 | 0.13112837 |
| -0.18639955 | -0.43226323 | -0.06649854 | -0.10718519 | -0.0015315 | -0.10145709 |
| -0.04672927 | 0.08120747 | -0.21614076 | -0.05030886 | 0.01099302 | 0.13099325 |
| 0.0199088 | -0.00216339 | 0.16113557 | -0.00893744 | -0.23416261 | -0.02196023 |
| 0.02626017 | 0.27999154 | 0.21484318 | 0.01862955 | 0.02924812 | 0.01528299 |
| 0.10853367 | -0.22844124 | 0.0546564 | 0.07575663 | 0.13211569 | 0.01257548 |
| -0.01112156 | -0.10759406 | -0.02926293 | 0.18203634 | -0.28760406 | 0.05424234 |
| -0.04225888 | -0.16918974 | -0.02549001 | -0.02830392 | 0.23377301 | 0.13014194 |
| -0.15216991 | -0.11337833 | 0.17616168 | -0.19726399 | 0.00647064 | 0.01166139 |
| -0.12373663 | -0.22953102 | -0.36163935 | 0.08966511 | 0.30241567 | 0.15320696 |
| -0.22298665 | 0.04331908 | -0.0181148 | 0.04280913 | 0.11413274 | 0.11498281 |
| -0.00917582 | 0.00988067 | -0.09297743 | 0.03640249 | 0.16287637 | 0.0466212 |
| -0.0290133 | 0.25803027 | -0.05280512 | 0.05111192 | 0.07199718 | 0.05638158 |
| -0.0697929 | 0.00098371 | -0.06606656 | 0.05983776 | 0.00266628 | -0.04129875 |
| 0.06853965 | 0.11372356 | -0.10169227 | 0.13911372 | -0.05689936 | 0.06764958 |
| -0.00472776 | -0.00541299 | -0.09528907 | -0.10962103 | 0.05886415 | -0.23626609 |
| 0.21328874 | 0.17956109 | 0.00389627 | 0.1857577 | 0.08515702 | 0.01865238 |
| 0.01381934 | -0.04718114 | -0.20346698 | 0.02182583 | 0.15999378 | -0.1101944 |
| 0.12814941 | 0.00570592 | 58010094 | | | |

รูปที่ 4.16 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010094

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010238



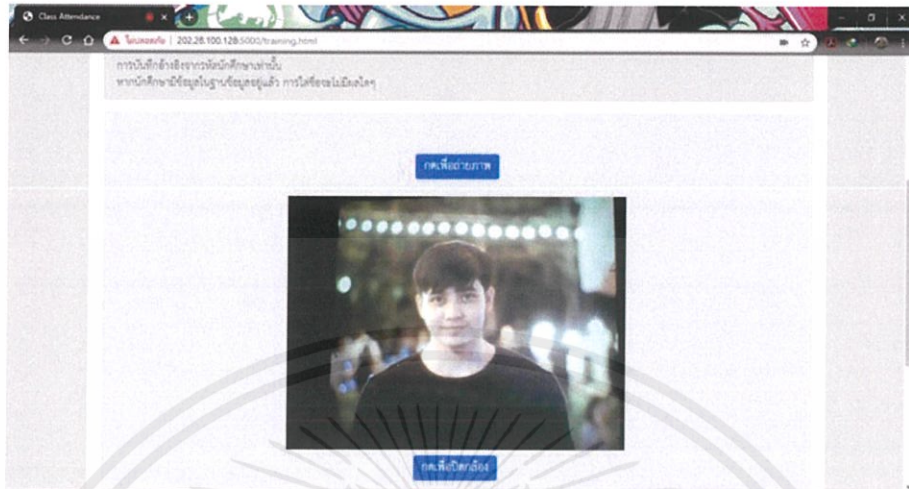
รูปที่ 4.17 การบันทึกใบหน้าของนักศึกษารหัส 58010238

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.04570986 | 0.07657693 | 0.06940318 | 0.01417426 | 0.0019049 | -0.04247964 |
| -0.06986122 | -0.07905928 | 0.1060366 | -0.08398346 | 0.20521542 | -0.01168865 |
| -0.10959988 | -0.05243042 | -0.02046511 | 0.13586971 | -0.16358048 | -0.08365606 |
| -0.07676999 | -0.01355487 | 0.03714781 | -0.02175276 | 0.05649675 | -0.08741585 |
| -0.11024512 | -0.34020635 | -0.12680145 | -0.18251386 | -0.01761059 | -0.12316 |
| -0.08776071 | -0.04281433 | -0.14358099 | 0.02292888 | -0.07806872 | -0.09366937 |
| 0.05658495 | -0.03651078 | 0.1772792 | 0.01299292 | -0.179524 | 0.03981608 |
| -0.01448542 | 0.1995904 | 0.1660749 | 0.06074284 | 0.03047761 | -0.08471537 |
| 0.12326238 | -0.16365893 | 0.14687499 | 0.0719502 | 0.10775699 | 0.05679375 |
| 0.0134188 | -0.130364 | -0.01229125 | 0.0394697 | -0.16085847 | 0.08713616 |
| 0.0315755 | 0.02872446 | -0.04195447 | -0.03177511 | 0.23261146 | 0.14326875 |
| -0.12903702 | -0.12683146 | 0.10527609 | -0.12679398 | -0.00726785 | 0.00062107 |
| -0.13344079 | -0.20144813 | -0.2759746 | 0.11176718 | 0.36424151 | 0.15338384 |
| -0.24004395 | 0.0175338 | -0.15012659 | -0.03282851 | 0.07403887 | 0.09696264 |
| -0.07581142 | 0.0163374 | -0.07260052 | 0.03405795 | 0.13277957 | -0.02804784 |
| -0.04791768 | 0.21702251 | -0.04688006 | 0.08273964 | -0.02693347 | 0.05078124 |
| -0.1111949 | 0.05806466 | -0.07558114 | 0.04422031 | 0.03920633 | 0.04984504 |
| -0.03339752 | 0.09667253 | -0.12801827 | 0.1564047 | 0.00893399 | -0.01257967 |
| 0.02972981 | 0.11295161 | -0.15721835 | -0.12752192 | 0.09842189 | -0.19599673 |
| 0.16245928 | 0.12929569 | 0.09614357 | 0.12681317 | 0.0676641 | 0.11546145 |
| 0.00057522 | 0.00987543 | -0.22261548 | 0.03687294 | 0.16641267 | -0.03069326 |
| 0.02251415 | 0.0515953 | 58010238 | | | |

รูปที่ 4.18 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010238

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010256



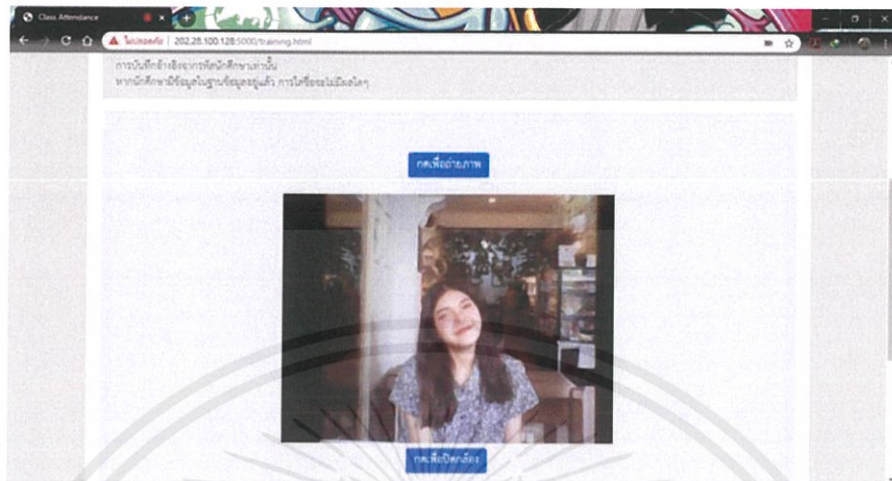
รูปที่ 4.19 การบันทึกใบหน้าของนักศึกษารหัส 58010256

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.03754063 | 0.09477726 | -0.02955882 | -0.05955577 | -0.0975803 | -0.08277979 |
| -0.03810979 | -0.04420595 | 0.14673664 | -0.08365489 | 0.20448871 | -0.13279839 |
| -0.20671557 | -0.04630376 | -0.0106471 | 0.15829061 | -0.18038869 | -0.09221864 |
| -0.070559 | 0.00631779 | 0.07976594 | 0.01697785 | 0.04072401 | 0.12729642 |
| -0.08408473 | -0.37267214 | -0.07067225 | -0.12493676 | 0.00497366 | -0.10053042 |
| -0.04439269 | -0.0084706 | -0.21605043 | -0.12578218 | 0.02807609 | 0.01474708 |
| -0.03052693 | -0.07189652 | 0.21015008 | 0.02657868 | -0.23095965 | 0.03867307 |
| 0.04800764 | 0.26089105 | 0.15197174 | 0.08643379 | -0.00873699 | -0.12829952 |
| 0.13154823 | -0.19824016 | 0.06225862 | 0.10328879 | 0.044111 | 0.05852452 |
| -0.01502684 | -0.1300303 | 0.07501701 | 0.13526109 | -0.21929429 | 0.04184577 |
| 0.07843295 | -0.0033798 | 0.0333608 | -0.00687135 | 0.2967687 | 0.03672671 |
| -0.08723873 | -0.13351226 | 0.11425802 | -0.15256023 | -0.03525817 | 0.09738091 |
| -0.10325532 | -0.22361018 | -0.2535615 | 0.05172048 | 0.43352556 | 0.16396701 |
| -0.18760115 | 0.02926744 | -0.10962613 | 0.05284126 | 0.15050165 | 0.11687294 |
| -0.00630041 | 0.02898236 | -0.10978931 | 0.03659765 | 0.24240145 | -0.04227174 |
| -0.05804178 | 0.28653127 | 0.03583388 | 0.09278317 | 0.04534899 | -0.02243619 |
| -0.12096294 | 0.02728165 | -0.17726527 | -0.04958603 | 0.05146562 | 0.01299556 |
| -0.03099148 | 0.12193326 | -0.17526415 | 0.13094786 | 0.0147496 | -0.03718046 |
| -0.02163255 | 0.05109348 | -0.10085974 | -0.10326336 | 0.11466999 | -0.2083298 |
| 0.18105628 | 0.14720979 | 0.16630317 | 0.12832658 | 0.08995178 | 0.00469505 |
| 0.04309303 | -0.05336981 | -0.22665286 | 0.01357886 | 0.15733659 | 0.00719459 |
| 0.14741831 | 0.00409947 | 58010256 | | | |

รูปที่ 4.20 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010256

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010324



รูปที่ 4.21 การบันทึกใบหน้าของนักศึกษารหัส 58010324

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.12471925 | 0.08116552 | -0.01219002 | -0.00157273 | -0.14185764 | -0.02129184 |
| -0.08377999 | 0.14882988 | 0.18688005 | -0.11669677 | 0.13335039 | -0.02388497 |
| -0.19306329 | 0.07122221 | -0.05343966 | 0.15695909 | -0.23486607 | -0.19931158 |
| 0.0440699 | 0.00985898 | 0.09521594 | 0.06646422 | 0.05538451 | 0.10190039 |
| -0.16574436 | -0.40260541 | -0.10258903 | -0.08398246 | -0.00650543 | 0.005992 |
| -0.0661127 | 0.11388114 | -0.12723631 | -0.05375131 | 0.06013382 | 0.17574409 |
| 0.04647041 | -0.05036949 | 0.23208016 | 0.02561738 | -0.27276513 | -0.03129566 |
| 0.06539926 | 0.31402639 | 0.18098077 | -0.05223766 | 0.0052193 | -0.05197722 |
| 0.11505676 | -0.25879976 | 0.04352972 | 0.1929682 | 0.07023758 | 0.03993791 |
| 0.01315387 | -0.13234235 | -0.00163118 | 0.09722805 | -0.17751491 | -0.04172835 |
| 0.05995006 | -0.10779159 | 0.03962541 | -0.04881603 | 0.15521526 | 0.09459193 |
| -0.06813724 | -0.17224416 | 0.09482352 | -0.22136696 | -0.0907677 | 0.03795791 |
| -0.12165233 | -0.2131706 | -0.34019619 | -0.02394316 | 0.36678419 | 0.15603332 |
| -0.16621055 | 0.09185641 | -0.02275313 | -0.05350024 | 0.10032427 | 0.1445398 |
| 0.01161111 | 0.116554 | -0.14845107 | 0.03373697 | 0.2033632 | -0.04255857 |
| -0.01250613 | 0.26570514 | 0.06012277 | 0.08309551 | 0.14226216 | 0.12098032 |
| -0.05875316 | 0.01669701 | -0.20395911 | 0.04332082 | 0.00369776 | -0.02508104 |
| 0.011426 | 0.08757122 | -0.18387587 | 0.1291275 | -0.0104876 | -0.05227879 |
| -0.05830274 | -0.06127812 | -0.11463032 | -0.08055808 | 0.17753854 | -0.21907069 |
| 0.07399686 | 0.18121827 | 0.01250606 | 0.24884643 | 0.07420618 | 0.04599439 |
| -0.03799264 | -0.08379626 | -0.14848846 | -0.03603017 | 0.06004314 | -0.03102621 |
| 0.01303527 | 0.04386201 | 58010324 | | | |

รูปที่ 4.22 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010324

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010372



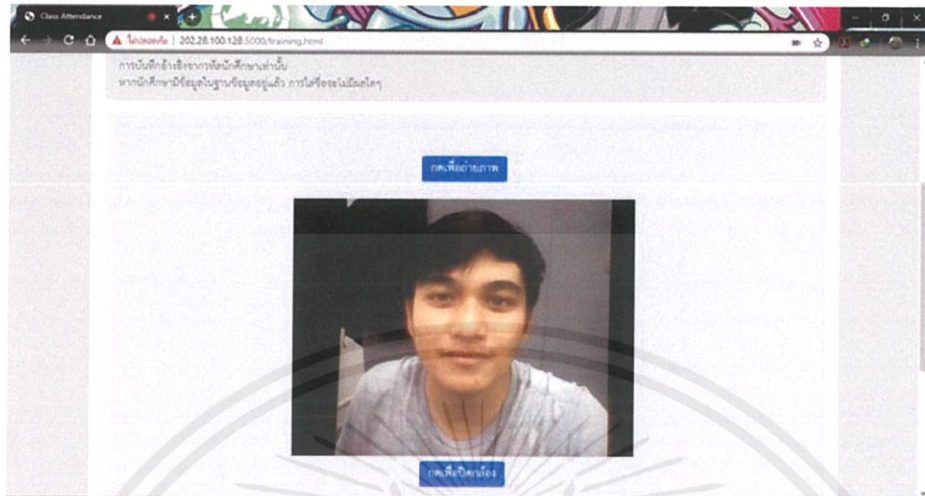
รูปที่ 4.23 การบันทึกใบหน้าของนักศึกษารหัส 58010372

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.0767519 | 0.0838119 | 0.05396979 | -0.02677216 | -0.10393689 | -0.06236991 |
| -0.05090424 | -0.10798442 | 0.13031094 | -0.13520977 | 0.23453596 | -0.06282611 |
| -0.21344647 | -0.08273627 | -0.07953153 | 0.23555903 | -0.17405029 | -0.13256902 |
| -0.08755969 | 0.00556048 | 0.09951726 | 0.02818097 | -0.03844093 | 0.05539331 |
| -0.10978924 | -0.38032809 | -0.06718611 | -0.02961296 | -0.02886645 | -0.06019055 |
| -0.06526171 | 0.0101749 | -0.21752483 | -0.06091499 | 0.04711781 | 0.01010144 |
| -0.02209274 | -0.08276749 | 0.20169432 | 0.01326088 | -0.29631793 | 0.07597059 |
| 0.07454315 | 0.219594 | 0.17021923 | -0.04948487 | 0.0169993 | -0.15959126 |
| 0.17623466 | -0.16555126 | 0.00372261 | 0.16907167 | 0.09023038 | 0.06236669 |
| -0.04303413 | -0.12683158 | -0.00745947 | 0.13616014 | -0.18048525 | -0.02477417 |
| 0.07326044 | -0.10190828 | -0.02391758 | -0.11601616 | 0.20513025 | 0.03066434 |
| -0.14020923 | -0.18203713 | 0.14430137 | -0.17582603 | -0.03141386 | 0.024353 |
| -0.12583223 | -0.20841651 | -0.31410792 | 0.02933671 | 0.34189451 | 0.11686182 |
| -0.19028622 | 0.04818455 | -0.07305899 | -0.00113256 | 0.07056274 | 0.17193455 |
| 0.02428149 | 0.04360999 | -0.03675081 | -0.00899153 | 0.25176144 | -0.04975383 |
| 0.01345075 | 0.28042307 | 0.0093513 | 0.03608912 | -0.0011729 | 0.00172538 |
| -0.12634975 | 0.01261525 | -0.07910037 | 0.02162221 | -0.0082403 | -0.02073901 |
| 0.02645635 | 0.11456873 | -0.17694646 | 0.16595739 | -0.04627056 | 0.05385134 |
| 0.02748349 | 0.0112944 | -0.07629855 | -0.12305174 | 0.08633758 | -0.19145957 |
| 0.21289018 | 0.15840006 | 0.09344459 | 0.12696187 | 0.10855485 | 0.08179208 |
| -0.02194198 | -0.0055598 | -0.22799064 | -0.03950138 | 0.13978001 | -0.07401445 |
| 0.12353946 | -0.01823009 | 58010372 | | | |

รูปที่ 4.24 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010372

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.8 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010438



รูปที่ 4.25 การบันทึกใบหน้าของนักศึกษารหัส 58010438

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.13952893 | 0.06636094 | 0.01784778 | -0.07577151 | -0.09329814 | -0.03015581 |
| -0.02572699 | -0.048119 | 0.11149923 | -0.01844801 | -0.24146479 | -0.02560639 |
| -0.19509707 | -0.14186919 | -0.02949285 | 0.14697665 | -0.20120147 | -0.16445412 |
| 0.01245901 | 0.02407355 | 0.0775434 | 0.03988813 | 0.07468615 | 0.04421818 |
| -0.11545981 | -0.44616708 | -0.03376361 | -0.08914933 | -0.03304421 | 0.00412969 |
| -0.03294301 | 0.09801458 | -0.22321245 | 0.06611072 | 0.1036602 | 0.12906966 |
| 0.01080402 | -0.03549393 | 0.21644321 | 0.01928901 | -0.29051957 | 0.01664602 |
| 0.1272141 | 0.27491426 | 0.13588738 | 0.06475719 | 0.01387667 | -0.1222951 |
| 0.08733658 | -0.16195098 | 0.08736282 | 0.1542218 | 0.10654978 | 0.00549987 |
| 0.02964834 | -0.17730978 | 0.04473916 | 0.08283248 | -0.20145127 | -0.04476033 |
| 0.02898842 | -0.00286043 | 0.03639088 | -0.09693895 | 0.24562716 | 0.05980607 |
| -0.12041452 | -0.17012465 | 0.10851438 | -0.09489252 | -0.05591512 | 0.13696626 |
| -0.16014209 | -0.17837973 | -0.33382744 | 0.03319753 | 0.40854424 | 0.11633668 |
| -0.14409311 | 0.02309805 | -0.05351616 | -0.04829017 | 0.10096609 | 0.1480182 |
| -0.09611967 | 0.07527377 | -0.10173447 | 0.02800555 | 0.2001916 | 0.01465655 |
| -0.10501362 | 0.16502038 | 0.02067713 | 0.11633106 | 0.02290239 | 0.03790224 |
| -0.08419573 | 0.05353445 | -0.21384011 | -0.03395229 | -0.00190892 | 0.00942331 |
| 0.01113688 | 0.17975789 | -0.18044589 | 0.02340785 | 0.01493713 | -0.04863672 |
| -0.02288371 | 0.03355654 | -0.06826997 | -0.08506442 | 0.02632172 | -0.25631741 |
| 0.20656483 | 0.18259442 | 0.13837618 | 0.13807797 | 0.15948336 | -0.03472608 |
| 0.01106515 | -0.05722632 | -0.20159349 | -0.00054616 | 0.13367325 | -0.03171238 |
| 0.15614673 | 0.00786096 | 58010438 | | | |

รูปที่ 4.26 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010438

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.9 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010564



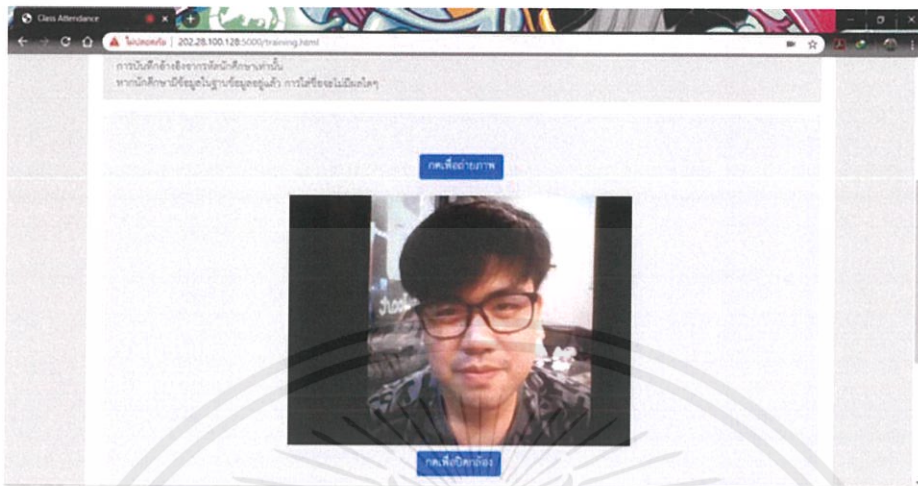
รูปที่ 4.27 การบันทึกใบหน้าของนักศึกษารหัส 58010564

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| -0.15962574 | 0.08506372 | 0.0540791 | 0.11112718 | -0.0880508 | 0.03039126 |
| -0.12299691 | -0.08409914 | 0.1792368 | -0.15283248 | 0.16732851 | 0.01193758 |
| -0.25301176 | 0.05450986 | -0.04765656 | 0.14603648 | -0.13301633 | 0.08810462 |
| -0.02788116 | -0.00762086 | 0.06511604 | 0.11526169 | 0.01236477 | 0.08221332 |
| -0.07966708 | -0.34112024 | -0.09128747 | -0.00390386 | -0.03040315 | 0.06222596 |
| -0.0870643 | 0.04866043 | -0.17032883 | -0.00828455 | -0.00044306 | 0.12467408 |
| -0.00101579 | 0.00594589 | 0.14601229 | 0.05443166 | -0.22358793 | 0.01022308 |
| 0.03876227 | 0.25039157 | 0.2621738 | -0.03274028 | -0.01336526 | -0.17990962 |
| 0.09582884 | -0.18928277 | 0.07215571 | 0.19059838 | 0.00621234 | 0.05422671 |
| 0.05830812 | -0.15883575 | -0.00823945 | 0.13888103 | -0.06521131 | 0.03053091 |
| 0.08170681 | -0.04984608 | 0.00198882 | -0.05944751 | 0.22291809 | 0.07742953 |
| -0.13494208 | -0.23808204 | 0.12709931 | -0.15522031 | -0.12814467 | 0.09557981 |
| -0.14773057 | -0.16660419 | -0.29788527 | 0.07710843 | 0.40555918 | 0.19359729 |
| -0.22092092 | 0.08096235 | -0.07285763 | -0.05466075 | 0.16916265 | 0.16086069 |
| -0.03988796 | -0.00577731 | -0.07043004 | 0.01214055 | 0.20656802 | 0.07017061 |
| -0.03861336 | 0.2456383 | 0.04437752 | 0.10431173 | 0.06914926 | 0.06411934 |
| -0.10976354 | 0.02682315 | -0.20885754 | -0.0802411 | 0.04054321 | -0.02967223 |
| 0.03711163 | 0.19216265 | -0.13678618 | 0.22350061 | -0.00395194 | 0.03358326 |
| 0.04505985 | 0.05569374 | -0.09856686 | -0.02561518 | 0.13598111 | -0.20417945 |
| 0.18817177 | 0.09816272 | 0.11579764 | 0.13097306 | 0.17739241 | 0.16599214 |
| -0.00748243 | 0.01014362 | -0.17300601 | -0.00959431 | 0.08948915 | -0.04128884 |
| 0.15831636 | 0.03995166 | 58010564 | | | |

รูปที่ 4.28 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010564

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.10 ผลการทดสอบการบันทึกใบหน้าของนักศึกษารหัส 58010956



รูปที่ 4.29 การบันทึกใบหน้าของนักศึกษารหัส 58010956

```
[ -0.09206124 0.07868998 0.07792243 0.04383225 -0.09885656 -0.04078216
-0.04601016 -0.113884 0.07614859 -0.0704212 0.20944625 -0.02636197
-0.20353386 -0.16777073 -0.05254874 0.18157427 -0.1376148 -0.11683869
-0.04599141 0.01876552 0.1143766 0.02702797 0.06225226 0.01864902
-0.08530752 -0.36165762 -0.08710939 -0.02749596 0.0060622 -0.03471664
-0.04646317 0.01381946 -0.22497877 -0.07482792 -0.00999839 0.0656305
-0.01991972 -0.09163918 0.13642818 0.01838054 -0.22364955 0.08913532
-0.02467434 0.26354387 0.20870015 0.04563443 0.03502076 -0.09444992
0.11953824 -0.18820822 0.08943304 0.13221627 0.07456708 0.01508042
-0.00459216 -0.07588103 0.04734037 0.11441541 -0.12632729 -0.00877603
0.10822717 -0.00963229 0.00058056 -0.09098109 0.17333527 0.04207019
-0.11974509 -0.18365462 0.08145961 -0.1086861 -0.08163039 0.01335478
-0.19725619 -0.15025118 -0.34465766 0.10539234 0.35111356 0.11896975
-0.23799501 -0.02109208 -0.06340486 0.10659596 0.15655312 0.09806739
0.00668233 -0.02025897 -0.14295398 0.01970994 0.18727405 0.06709943
-0.11860619 0.25620636 0.03767522 0.04692862 0.01902384 0.04089123
-0.04321329 0.09779581 -0.12439939 0.04996634 0.11638985 -0.00854053
-0.01398914 0.16846515 -0.10972818 0.15077974 -0.03240817 0.0917094
0.06380589 -0.0190754 -0.19461761 -0.08329353 0.06855467 -0.17035048
0.22652741 0.20042439 0.08284881 0.06550795 0.16599666 0.0725728
0.00284732 0.02780701 -0.19274025 -0.05879548 0.15008198 -0.01219836
0.07885642 0.02898257] 58010956
```

รูปที่ 4.30 ผลการบันทึกใบหน้าของนักศึกษารหัส 58010956

จากรูปที่ 4.11 ถึง 4.30 (อ้างอิงถึงข้อมูลจากหัวข้อในบทที่ 2.1.3) จะเห็นได้ว่าการบันทึกข้อมูลใบหน้าของนักศึกษาแต่ละรายบุคคล จะมีการเข้ารหัสใบหน้าไว้ โดยการเข้ารหัสใบหน้าจะอยู่ในรูปแบบของ BitArray ซึ่งผลจากการทดสอบการบันทึกใบหน้าของนักศึกษาจำนวน 10 คน

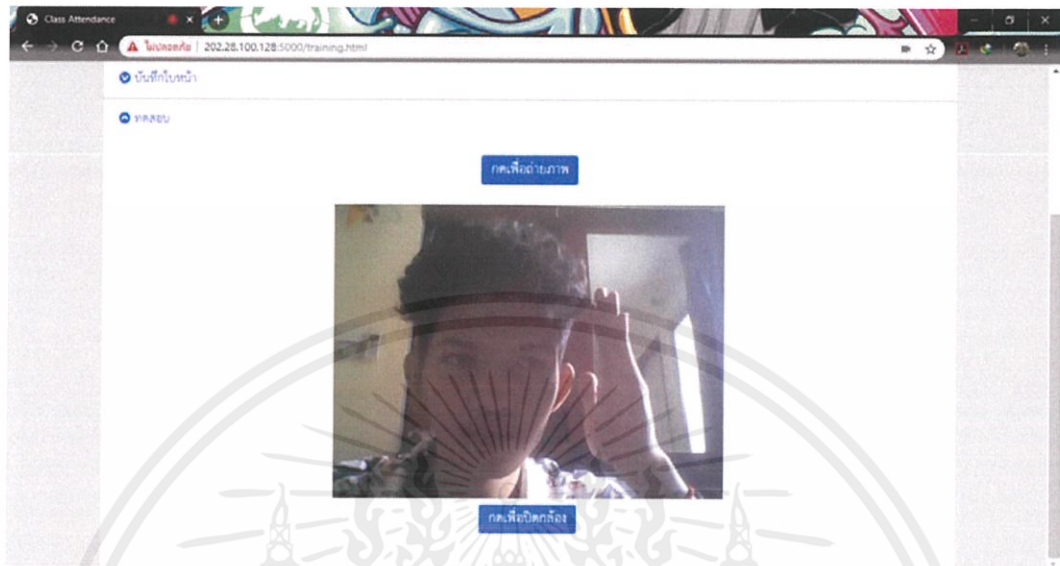
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(รูปที่ 4.12, 4.14, ... , 4.30) การเทรนใบหน้า 1 รูป ระบบจะวิเคราะห์แล้วนำรูปไปเข้ารหัส BitArray ของรูปแต่ละรูป ยกตัวอย่าง ถ้าบันทึกใบหน้านักศึกษา 1 คน จำนวน 100 ใบหน้า เหมือนๆกัน จะมี BitArray 100 ชุดที่แทบจะไม่แตกต่างกัน และจะบันทึกว่า BitArray ชุดนี้เป็นของ นักศึกษาคนใด เมื่อมีการเช็คชื่อเข้าชั้นเรียนหรือการทดสอบ ระบบจะเปรียบเทียบจากใบหน้าที่ทำ การเช็คชื่อว่าตรงหรือใกล้เคียงกับ BitArray ชุดไหนที่ได้บันทึกไว้ก่อนหน้านี้ แล้วระบุว่าใบหน้าที่ได้ ทำการตรวจจับนั้นเป็นของใคร ซึ่งค่า Bitarray เหล่านี้มีทั้งค่าบวกและค่าลบเป็นผลมาจากการวัด ระยะห่างระหว่างจุดเฉพาะบนใบหน้า

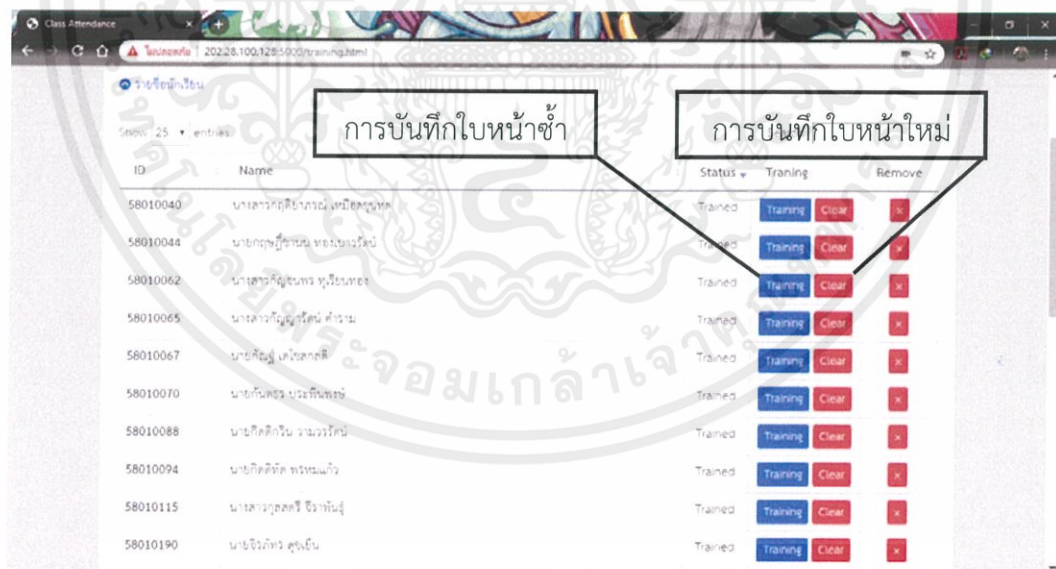
| ID | Name | Status | Training | Remove |
|----------|--------------------------------|---------|----------|--------|
| 58010040 | นางสาวณิชาธิยาพรณ์ เหมอิตตบุตร | Trained | Training | Clear |
| 58010044 | นายอดุลย์ธวัช อดุลย์ธวัช | Trained | Training | Clear |
| 58010062 | นางสาววิภาดา วัฒนศิริ | Trained | Training | Clear |
| 58010065 | นายสุวิทย์ วัฒนศิริ | Trained | Training | Clear |
| 58010067 | นายวิวัฒน์ เกษมศักดิ์ | Trained | Training | Clear |
| 58010070 | นายกันตว บวรสินทรัพย์ | Trained | Training | Clear |
| 58010088 | นายวิวัฒน์ วัฒนศิริ | Trained | Training | Clear |
| 58010094 | นายวิวัฒน์ พรหมแก้ว | Trained | Training | Clear |
| 58010115 | นายวิวัฒน์ วิวัฒน์ | Trained | Training | Clear |
| 58010190 | นายวิวัฒน์ สุวัฒน์ | Trained | Training | Clear |

รูปที่ 4.31 สถานะสำเร็จการบันทึกใบหน้าของนักศึกษา

การทดสอบการบันทึกใบหน้า ให้เปิดเข้าไปในแถบทดสอบ และกดเปิดกล้อง ต่อจากนั้นก็เข้าสู่การทดสอบ แสดงผลการทดสอบดังรูปที่ 4.14 ซึ่งส่วนนี้เป็นเพียงการทดสอบ ไม่มี ผลต่อการตรวจสอบรายชื่อการเข้าชั้นเรียน หากผลจากการบันทึกใบหน้าไม่ดีพอ สามารถกลับไป บันทึกใบหน้าใหม่ได้ โดยบันทึกซ้ำหรือบันทึกใหม่ได้ แสดงดังรูปที่ 4.15



รูปที่ 4.32 การทดสอบใบหน้าของนักศึกษา



รูปที่ 4.33 ตัวเลือกการบันทึกใบหน้าของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบการตรวจสอบใบหน้าของนักศึกษา

เมื่อนักศึกษาได้ทำการบันทึกใบหน้าแล้ว ถือเป็นบุคคลที่มีรูปใบหน้าอยู่ในฐานข้อมูล ซึ่งการทดสอบความถูกต้องของใบหน้า นักศึกษาจะใช้ฟังก์ชันการทดสอบการบันทึกใบหน้า เพื่อไม่ให้กระทบต่อการตรวจสอบการเข้าชั้นเรียน ซึ่งนักศึกษาทั้งหมดที่มาทดสอบจำนวน 10 คน แต่ละคนจะทำการทดสอบจำนวน 20 ครั้ง โดยมีเกณฑ์ที่กำหนดขึ้นได้แก่ การตรวจจับใบหน้า และการระบุตัวบุคคลจากใบหน้า ซึ่งการตรวจจับใบหน้าของนักศึกษาเป็นการเก็บผลของการทดสอบของระบบ ในกรณีที่ระบบสามารถค้นหาใบหน้าได้หรือไม่ได้ เพื่อประสิทธิภาพของระบบ และได้สร้างฟังก์ชันการเช็คชื่อเข้าเรียนแบบปกติไว้แก้ปัญหาการที่ระบบไม่สามารถค้นหาใบหน้าได้ แสดงดังรูปที่ 4.37 หากระบบสามารถค้นหาใบหน้าได้จะนำไปสู่การระบุตัวบุคคลจากใบหน้า โดยมีการระบุตัวบุคคลได้ถูกต้อง การระบุตัวบุคคลได้ไม่ถูกต้อง และไม่สามารถระบุตัวบุคคลได้ มีเกณฑ์การระบุตัวบุคคลที่ใช้ในการเปรียบเทียบภาพใบหน้าของนักศึกษาได้กำหนดไว้คือ 70% หากใบหน้าของนักศึกษาที่ใช้เปรียบเทียบกับชุดข้อมูลที่ได้บันทึกไว้มีค่ามากกว่าหรือเท่ากับ 70% ระบบจะระบุชื่อภาพใบหน้าของนักศึกษานั้นเป็นของใคร หากใบหน้าของนักศึกษาที่ใช้เปรียบเทียบกับชุดข้อมูลที่ได้บันทึกไว้มีค่าน้อยกว่า 70% ระบบจะไม่ระบุชื่อเจ้าของภาพใบหน้า จากผลการทดลองการระบุตัวบุคคลจากใบหน้าของนักศึกษา นำไปคิดเป็นร้อยละความถูกต้องของผลการระบุตัวบุคคลถูกต้องกับการตรวจจับใบหน้าได้ ซึ่งเก็บผลดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบการตรวจจับใบหน้าของนักศึกษา

| ลำดับ | รหัสนักศึกษา | การตรวจจับใบหน้า | | การระบุตัวบุคคลจากใบหน้า | | | เปอร์เซ็นต์ความถูกต้อง |
|-------|--------------|------------------|--------|--------------------------|------------|---------|------------------------|
| | | ได้ | ไม่ได้ | ถูกต้อง | ไม่ถูกต้อง | ไม่ระบุ | |
| 1 | 58010044 | 19 | 1 | 17 | 0 | 2 | 89.47% |
| 2 | 58010088 | 20 | 0 | 17 | 0 | 3 | 85% |
| 3 | 58010094 | 20 | 0 | 20 | 0 | 0 | 100% |
| 4 | 58010238 | 20 | 0 | 18 | 0 | 2 | 90% |
| 5 | 58010256 | 19 | 1 | 18 | 0 | 1 | 94.74% |
| 6 | 58010324 | 20 | 0 | 19 | 0 | 1 | 95% |
| 7 | 58010372 | 20 | 0 | 18 | 0 | 2 | 90% |

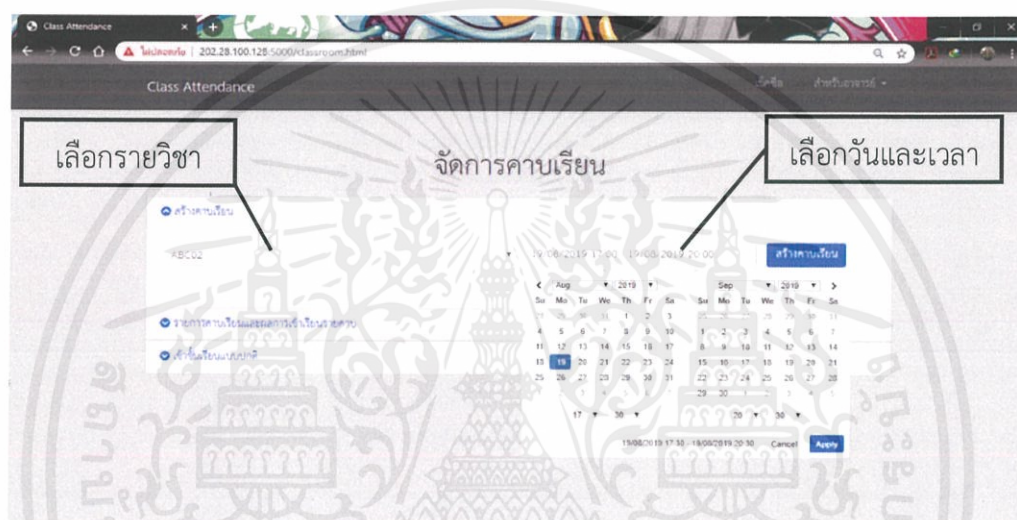
ตารางที่ 4.1 ผลการทดสอบการตรวจจับใบหน้าของนักศึกษา (ต่อ)

| ลำดับ | รหัสนักศึกษา | การตรวจจับใบหน้า | | การระบุตัวบุคคลจากใบหน้า | | | เปอร์เซ็นต์ ความถูกต้อง |
|--------|--------------|------------------|--------|--------------------------|------------|---------|----------------------------|
| | | ได้ | ไม่ได้ | ถูกต้อง | ไม่ถูกต้อง | ไม่ระบุ | |
| 8 | 58010438 | 20 | 0 | 19 | 0 | 1 | 95% |
| 9 | 58010564 | 19 | 1 | 17 | 0 | 2 | 89.47% |
| 10 | 58010956 | 19 | 1 | 18 | 0 | 1 | 94.74% |
| สรุปผล | | 196 | 4 | 181 | 0 | 15 | 92.35% |

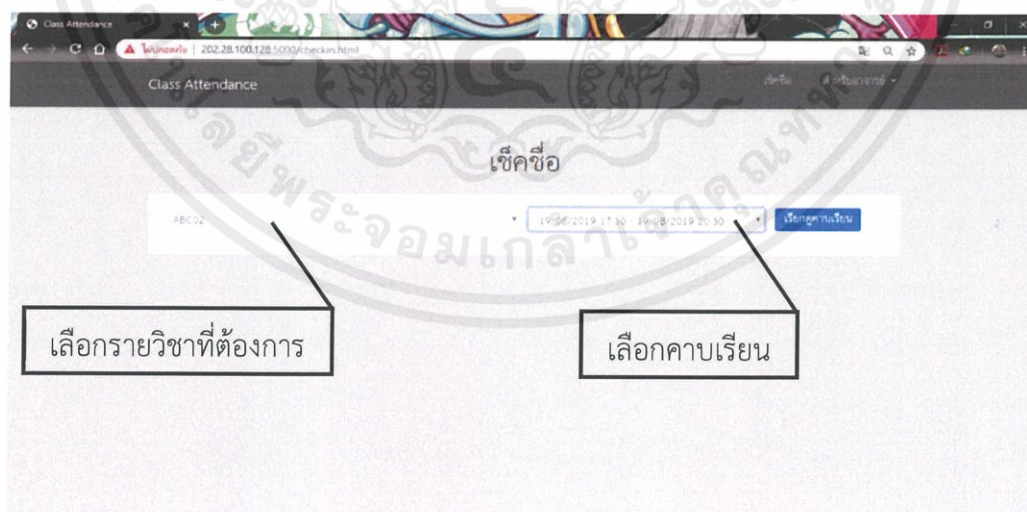
จากตารางที่ 4.1 ระบบได้มีการเก็บผลการทดสอบการตรวจจับใบหน้า ซึ่งจากการทดสอบทั้งหมด 10 คน คนละ 20 ครั้ง รวม 200 ครั้ง มีจำนวนการตรวจจับใบหน้าได้ 196 ครั้ง คิดเป็นร้อยละ 98 ของจำนวนในการทดสอบทั้งหมด ส่วนการตรวจจับใบหน้าไม่ได้คิดเป็นร้อยละ 2 ของจำนวนใบการทดสอบ ซึ่งส่วนการตรวจจับใบหน้าไม่ได้ ได้สร้างฟังก์ชันในการเช็คชื่อเข้าเรียนของนักศึกษา เพื่อให้ระบบนี้สามารถใช้งานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น แสดงดังรูปที่ 4.37

4.5 ผลการทดสอบการเข้าเรียนของนักศึกษา

การทดสอบการเข้าชั้นเรียนของนักศึกษา ระบบจะมีการสร้างคาบเรียนก่อนการเช็คชื่อเข้าเรียนโดยการสร้างคาบเรียนแสดงดังรูปที่ 4.34 เมื่อสร้างคาบเรียนแล้ว ทำการเลือกคาบเรียนที่ต้องการ แสดงดังรูปที่ 4.35 ต่อจากนั้นจะเป็นการเช็คชื่อเข้าชั้นเรียน แสดงดังรูปที่ 4.36 กรณีการเช็คชื่อไม่สำเร็จ สามารถทำการเช็คชื่อเข้าชั้นเรียนแบบปกติได้ แสดงดังรูปที่ 4.37 เมื่อเช็คชื่อสำเร็จ จะแสดงผลการเช็คชื่อดังรูปที่ 4.38

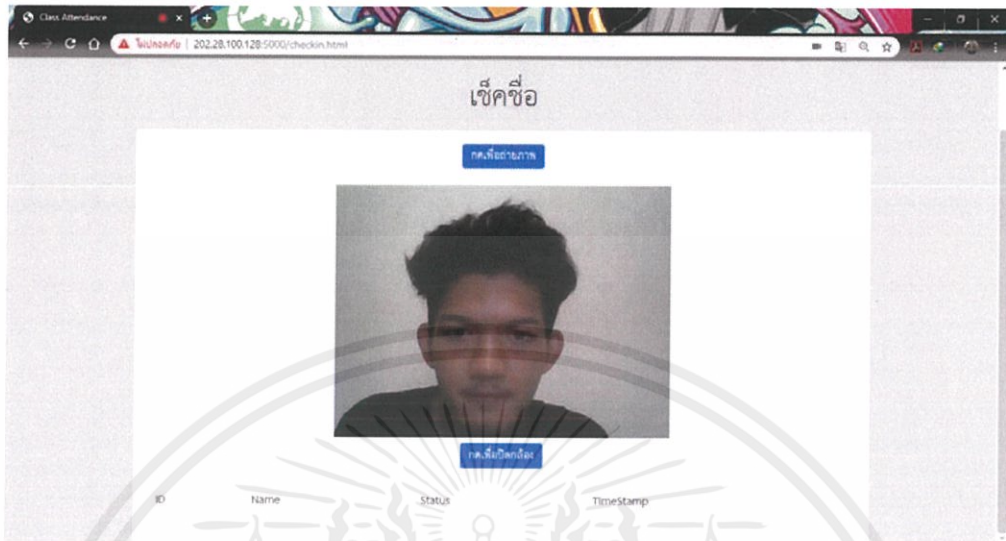


รูปที่ 4.34 การสร้างคาบเรียนของรายวิชา ABC02

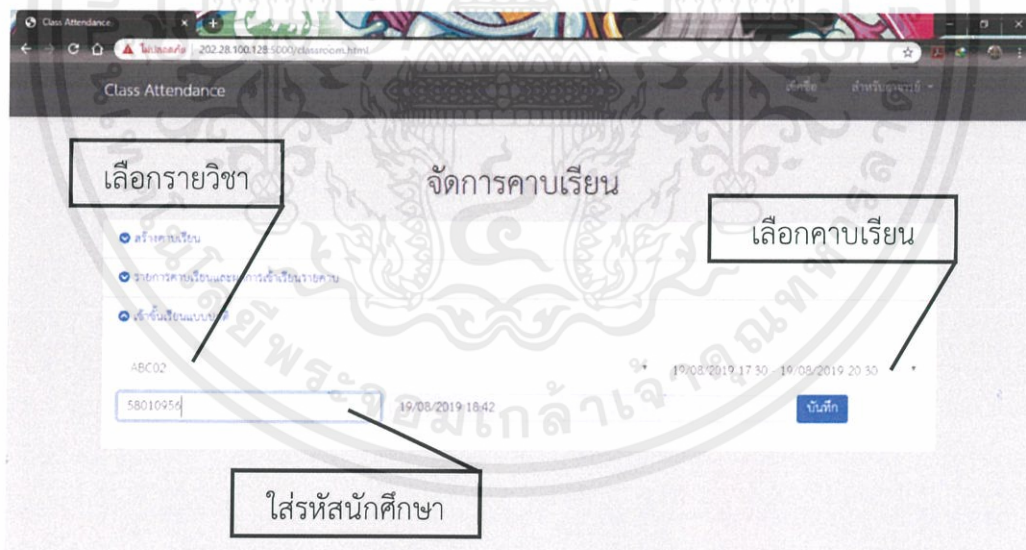


รูปที่ 4.35 การเลือกคาบเรียนและรายวิชาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.36 การเช็คชื่อเข้าชั้นเรียนของรายวิชา ABC02



รูปที่ 4.37 การเช็คชื่อเข้าชั้นเรียนแบบปกติของรายวิชา ABC02

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class Attendance

เช็คชื่อ สถานะการเข้าชั้นเรียน : สาย

สถานะการเข้าชั้นเรียน : ตรงเวลา

| ID | Name | Status | TimeStamp |
|----------|---------------------------|---------|------------------|
| 58010238 | นายชนวัฒน์ สุภกิจ | สาย | 19/08/2019 18:01 |
| 58010956 | นายภาณุเดช ฉลาด | สาย | 19/08/2019 18:42 |
| 58010564 | นายธีรชัย สุขทอง | สาย | 19/08/2019 19:50 |
| 58010094 | นายวิวัฒน์ พงษ์มนตรี | ตรงเวลา | 19/08/2019 17:46 |
| 58010088 | นายศักดิ์วิทย์ รามวรวิทย์ | ตรงเวลา | 19/08/2019 17:51 |
| 58010372 | นางสาวณัฐชยา นาโหม | ตรงเวลา | 19/08/2019 17:52 |
| 58010324 | นางสาวสุวิภา ใจประดิษฐ์ | ตรงเวลา | 19/08/2019 17:52 |

รูปที่ 4.38 ผลการเช็คชื่อเข้าชั้นเรียนของรายวิชา ABC02

4.6 ผลการทดสอบการตรวจสอบประวัติการเข้าชั้นเรียนของนักศึกษา

ระบบเช็คชื่อการเข้าชั้นเรียนอัตโนมัติ สามารถเรียกดูประวัติการเข้าชั้นเรียนได้ ซึ่งการที่จะเรียกดูการเข้าชั้นเรียนมี 2 รูปแบบดังนี้

4.6.1 การเรียกดูประวัติการเข้าชั้นเรียนรายคาบ

หน้าเว็บไซต์แสดงการทดสอบการใช้งานส่วนของการเรียกดูประวัติการเข้าชั้นเรียนรายคาบ แสดงดังรูปที่ 4.39 และ 4.40

Class Attendance

เลือกรายวิชา จัดการคาบเรียน เลือกคาบเรียน

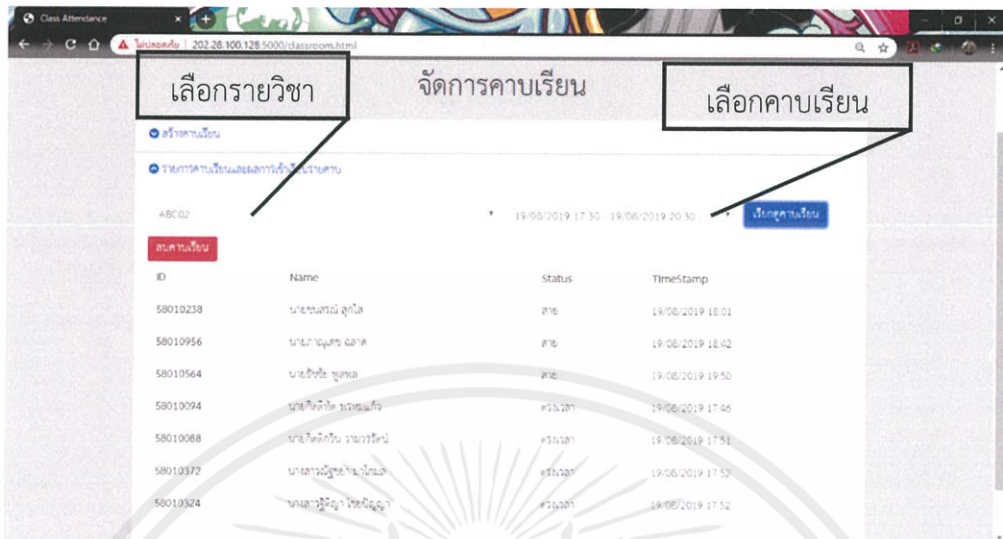
ABC01 19/08/2019 17:30 - 19/08/2019 20:30

สถานะการเรียน

| ID | Name | Status | TimeStamp |
|----------|----------------------|---------|------------------|
| 58010094 | นายวิวัฒน์ พงษ์มนตรี | ตรงเวลา | 18/08/2019 17:38 |

รูปที่ 4.39 ประวัติการเข้าชั้นเรียนรายคาบของรายวิชา ABC01

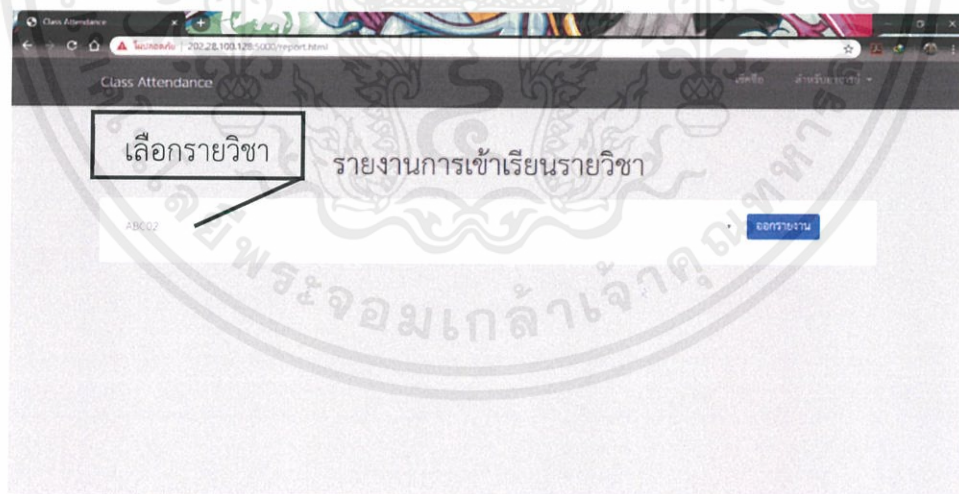
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 ประวัติการเข้าชั้นเรียนรายคาบของรายวิชา ABC02

4.6.2 การเรียกดูประวัติการเข้าชั้นเรียนรายวิชา

หน้าเว็บไซต์แสดงการทดสอบการใช้งานส่วนของการเรียกดูประวัติการเข้าชั้นเรียนรายวิชา และผลการออกรายงาน แสดงดังรูปที่ 4.39 และ 4.40



รูปที่ 4.41 การเลือกรายวิชาเพื่อเรียกดูประวัติการเข้าชั้นเรียนรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ID | ชื่อ | ตรงเวลา | สาย | ขาด | ##### 19/08/2019 17:30 |
|----------|----------------|---------|-----|-----|------------------------|
| 56011448 | นายอัฐพงศ์ | 0 | 0 | 0 x | x |
| 58010022 | นางกรวิมล | 0 | 0 | 0 x | x |
| 58010040 | นางสาวกฤดี | 0 | 0 | 0 x | x |
| 58010044 | นายกฤษณัฐ | 0 | 0 | 0 x | x |
| 58010062 | นางสาวกัญญา | 0 | 0 | 0 x | x |
| 58010065 | นางสาวกัญญา | 0 | 0 | 0 x | x |
| 58010067 | นายกฤษณ์ ไร่ | 0 | 0 | 0 x | x |
| 58010070 | นายกันตธร | 0 | 0 | 0 x | x |
| 58010072 | นายกฤษณ์ ร | 0 | 0 | 0 x | x |
| 58010088 | นายศักดิ์วิภา | 1 | 0 | 0 x | / |
| 58010094 | นายศักดิ์พิศ | 1 | 0 | 0 x | / |
| 58010103 | นายศักดิ์สิงห์ | 0 | 0 | 0 x | / |
| 58010115 | นางสาวกมล | 0 | 0 | 0 x | x |
| 58010179 | นางสาวจันทิมา | 0 | 0 | 0 x | x |
| 58010187 | นายธีรภัทร | 0 | 0 | 0 x | x |
| 58010190 | นายธีรภัทร | 0 | 0 | 0 x | x |
| 58010194 | นายธีรภัทร ธี | 0 | 0 | 0 x | x |
| 58010211 | นายเจษฎา | 0 | 0 | 0 x | x |

รูปที่ 4.42 ประวัติการเข้าชั้นเรียนรายวิชา ABC02

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงการเรื่องระบบรู้จำใบหน้าเพื่อเช็คชื่อเข้าเรียนอัตโนมัติ ทางผู้จัดทำมุ่งเน้นการพัฒนากระบวนการอำนวยความสะดวก ความรวดเร็ว และแม่นยำสำหรับการเช็คชื่อเข้าเรียนในแต่ละรายวิชา โดยการทำงานของระบบถูกแบ่งเป็น 2 ส่วน ได้แก่ ส่วนระบบรู้จำใบหน้าของผู้เรียนและส่วนของการแสดงผลข้อมูลการเข้าเรียนของผู้เรียน ในส่วนของระบบรู้จำใบหน้า ใช้คอมพิวเตอร์ ASUS K455LD-WX062D ในการประมวลผลหลัก ซึ่งสามารถออกแบบชุดคำสั่งโดยใช้ภาษา Python โดยมีหลักการ SGD Classifier เป็นหลักการที่มีฟังก์ชันอย่างง่ายสำหรับจัดการข้อมูลภาพ การประมวลผลภาพพื้นฐาน และใช้การจำลองรูปแบบการเปรียบเทียบภาพใบหน้าที่ผ่านมาการ Train สำเร็จแล้ว จะวิเคราะห์ข้อมูลจำนวนมากได้รวดเร็วยิ่งขึ้น

การทำงานของระบบรู้จำใบหน้า เริ่มจากมีสัญญาณภาพอินพุตจากกล้อง Webcam จากคอมพิวเตอร์ ซึ่งเป็นภาพใบหน้าของนักศึกษาเข้าสู่หน่วยประมวลผล แล้วบันทึกไว้ในฐานข้อมูล โดยหลักการของการ Train ข้อมูล เมื่อมีสัญญาณภาพอินพุตมากกว่า 2 ภาพขึ้นไป ระบบจะสามารถเรียนรู้ได้ด้วยตนเอง (Machine Learning) โดยใช้หลักการของ Viola-Jones ทำให้สามารถแยกแยะส่วนของภาพที่มีใบหน้าและไม่มีใบหน้าอยู่ และจำแนกภาพใบหน้าบุคคลได้ตั้งผลการทดลอง เมื่อระบบสามารถรู้จำใบหน้าได้แล้ว ในส่วน Face Recognition เมื่อกล้อง Webcam จากคอมพิวเตอร์ทำงานและทำการตรวจจับใบหน้า จะเป็นการทำงานในส่วนของหน่วยประมวลผล ซึ่งได้ออกแบบชุดคำสั่งในการทำงานไว้โดยใช้ภาษา Python ในการประมวลผลภาพพื้นฐาน ทำให้ระบบสามารถระบุตัวบุคคลได้โดยแสดงชื่อของนักศึกษาที่จอแสดงผล

ในส่วนของการใช้งานจริงออกแบบให้เมื่อโปรแกรมทำงานบนเว็บไซต์ทั้งหมด เพื่อความสะดวกสบายของผู้ใช้งาน หากผู้ใช้ต้องการเพิ่มรายชื่อนักศึกษาใหม่ เพียงแค่กดปุ่ม Training ที่ชื่อนักศึกษาคนที่ต้องการบันทึกภาพใบหน้า หรือกรอกข้อมูลของนักศึกษาในช่องบันทึกใบหน้า และทำการบันทึกใบหน้านักศึกษาใหม่ด้วยการกดปุ่ม “กดเพื่อเพื่อถ่ายภาพ” กล้องจะเปิดขึ้นรับใบหน้าของนักศึกษาและการบันทึกภาพใบหน้านักศึกษา ระบบจะทำการบันทึกข้อมูลนักศึกษาใหม่สำหรับการทำงานต่อไป เป็นการใช้งานของนักศึกษาที่มีข้อมูลแล้ว ให้เลือกวิชาเรียนและเวลาเรียนที่เรียน แล้วกดปุ่ม “เรียกดูคาบเรียน” ระบบจะแสดงข้อมูลภายในคาบเรียน และกดเปิดกล้อง

กล้องจะเปิดขึ้นรับใบหน้าของนักศึกษา เพื่อทำการเช็คชื่อของนักศึกษา เมื่อใบหน้านักศึกษาผ่านการเช็คชื่อ สถานะของนักศึกษาจะเปลี่ยนเป็นการมาเรียน โดยจะเป็นการมาเรียนตรงเวลาหรือมาเรียนสาย และตรวจสอบประวัติสถานะการมาเรียนของนักศึกษาแต่ละครั้งได้ โดยจะบันทึกเป็นจำนวนการมาเรียนตรงเวลา การมาเรียนสาย และการขาดเรียน ในหน้าเว็บไซต์หรือรูปแบบของไฟล์ Excel

การทดสอบระบบรู้จักใบหน้าของนักศึกษา 10 คน คนละ 20 ครั้ง รวม 200 ครั้ง ได้ผลการทดสอบการตรวจจับใบหน้าของนักศึกษาได้ 196 ครั้ง คิดเป็นร้อยละ 98 ของจำนวนในการทดสอบทั้งหมด ซึ่งระบบการระบุตัวตนจากภาพใบหน้าที่ตรวจจับได้ทั้งหมด 196 ครั้ง สามารถระบุตัวตนได้ถูกต้องจำนวน 181 ครั้ง คิดเป็นร้อยละ 92.35 ของจำนวนที่ระบบสามารถตรวจจับภาพใบหน้าได้ ซึ่งระบบการระบุตัวตนของใบหน้านักศึกษาผิดพลาดมีจำนวน 0 ครั้ง และไม่สามารถระบุตัวตนของใบหน้านักศึกษามีจำนวน 15 ครั้ง

ทั้งนี้ระบบมีความแม่นยำเพียงพอที่จะระบุตัวตนของใบหน้านักศึกษาได้ หากระบบไม่สามารถระบุตัวตนของภาพใบหน้านักศึกษาได้ จะมีฟังก์ชันการเช็คชื่อเข้าเรียนแบบปกติ โดยการใส่รหัสนักศึกษาเพื่อทำการเช็คชื่อเข้าเรียน เพื่อระบบรู้จักใบหน้ามีประสิทธิภาพในการเช็คชื่อเข้าเรียนของนักศึกษา และสามารถเรียกดูผลการเข้าชั้นเรียนในแต่ละรายวิชาได้ด้วยไฟล์ Excel เพื่อความสะดวกสบายของผู้ใช้งานระบบ

5.2 ข้อเสนอแนะ

1. การใช้งานระบบตรวจจับและรู้จักใบหน้านั้น ผู้ใช้งานต้องมีทิศทางตรงกับภาพที่เก็บไว้ในฐานข้อมูล และระหว่างการตรวจจับใบหน้านั้นต้องมีแสงสว่างเพียงพอ เพื่อประสิทธิภาพในการรู้จักใบหน้าที่สูงขึ้น
2. ส่วนของเว็บไซต์สามารถทำให้มีความน่าสนใจมากยิ่งขึ้นได้และสามารถเพิ่มฟังก์ชันต่าง ๆ ที่มีประโยชน์ เช่น การเพิ่มให้สามารถคะแนนการสอบต่าง ๆ เอกสารประกอบการเรียน
3. ระบบการประมวลผลค่อนข้างช้า อันเนื่องมาจากหน่วยประมวลผลมีประสิทธิภาพต่ำ หากผู้ใช้งานต้องการใช้งานควรที่จะมีหน่วยประมวลผลที่มีประสิทธิภาพการทำงานที่ดีกว่า

บรรณานุกรม

- [1] รุสลี สุทธิวีรกุล และ วิไลพร แซ่ลี. “การตรวจจับใบหน้าด้วยวิธีพื้นฐานของการจำลองรูปแบบ Haar-lie (Face Detection bases-on Haar-like Features)”. วารสาร วิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ. 6 (ธันวาคม 2554)
- [2] ชาญชัย ศุภอรธกร. สร้างเว็บแอปพลิเคชัน PHP MySQL + AJAX jQuery ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. กรุงเทพฯ : รีไควว่า, 2560.
- [3] อ.ลิขิต ยืนบุญ. “คู่มือการใช้งาน PHPMyAdmin”. มหาวิทยาลัยราชภัฏเชียงใหม่ คณะ วิทยาการจัดการ
- [4] นพ มหิษานนท์. การติดตั้งและการใช้งาน Raspberry Pi. พิมพ์ครั้งที่ 1. นนทบุรี : สำนักพิมพ์ คอร์ฟิงก์ซัน, 2560
- [5] Pattapong J. “Rasberry Pi3”.
<https://blog.pattapongj.com/raspberry-pi-3-day-1-setup-pi-c23a430e8427>
- [6] microcode. “Ionic 3 Tutorial #31 Firebase Real Time Database CRUD”
<https://www.youtube.com/watch?v=5e7k8T8D4Lo&fbclid=IwAR2MRoaHO4hTITlN0MAKN-slhHPMpq6-TOdSinO7B-l35-aw9Eg-s1J1Wiw>
- [7] Adam Geitgey. “Modern Face Recognition with Deep Learning”.
<https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
- [8] Scikit-learn developers. “Stochastic gradient descent”.
<https://scikit-learn.org/stable/modules/sgd.html>
- [9] Scikit-learn developers. “sklearn.linear_model.SGDClassifier”.
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายโค้ดไฟล์ app.py

```

# Full App
from flask import Flask, request, Response, send_from_directory, jsonify
from flask_cors import CORS, cross_origin // นำเข้า Flask เป็น Library
                                         สำหรับสร้าง Web Server

import pandas as pd // นำเข้า Library เกี่ยวกับการจัดการ Data frame สำหรับการประมวลผล

import os
import io } // นำเข้า Library สำหรับจัดการกับระบบปฏิบัติการและไฟล์
import sys

import xlrd // นำเข้า Library สำหรับอ่านข้อมูลจาก Excel
import sqlite3 // นำเข้า Library สำหรับฐานข้อมูล SQLite
from thradecontrol import threadcontrol as Lock // นำเข้า Library สำหรับควบคุม
                                                Multi-Threading เป็น Library ที่
                                                พัฒนาเพิ่มเติม

import cv2
import shutil
from PIL import Image } // นำเข้า Library ที่เกี่ยวกับการประมวลผลภาพ
import face_recognition as fr } // และระบุตัวตนใบหน้า

import numpy as np
from sklearn.linear_model import SGDClassifier // นำเข้า Library Machine Learning
                                                เพื่อใช้ในการคัดแยกใบหน้า

import pickle
import time // นำเข้า Library สำหรับจัดการเวลา
from io import BytesIO } // นำเข้า Library สำหรับเขียน/อ่านไฟล์และเข้ารหัสข้อมูล
import base64

import datetime // นำเข้า Library สำหรับจัดการเวลา

conn = sqlite3.connect('testDB6.db', check_same_thread=False) // เชื่อมต่อฐานข้อมูล
                                                                หากไม่มีข้อมูล
                                                                จะถูกสร้างขึ้นใหม่

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

datasets = 'face_dataset.sav' // ระบุชื่อไฟล์ที่จัดเก็บข้อมูลใบหน้า
mutex_db = Lock()
mutex_face = Lock() } // เปิดใช้ Mutex เพื่อควบคุมการทำงานแบบ Muti-Threading
global encodings
global names
#mutex.acquire()
#mutex.release()
app = Flask(__name__, static_url_path='/www2') } // ระบุตำแหน่งเก็บไฟล์ HTML
CORS(app)

#ตรวจสอบตัวแปรที่ได้จาก API
@app.route('/uploadsection', methods=['POST'])
def process_section():
    if 'section_name' not in request.form:
        return {'status':'failed', 'message':
            'Class name not found'}
    if 'section_file' not in request.files:
        return {'status':'failed', 'message':'File not found'}
    section_name = request.form.get('section_name')
    section_file = request.files['section_file']
    if(section_name is None or section_name == ""):
        return {'status':'failed', 'message':'Class name not found'}
    if(section_file is None or section_file.filename == ""):
        return {'status':'failed', 'message':'File not found'}
    df = pd.read_excel(section_file,encoding='TIS-620')
    section_name = section_name.strip()
    students = []
    students_id = []
    try:
        for i in range(6,df.shape[0]):
            val = int(df.iloc[i][0])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

students.append( (str(df.iloc[i][1]).strip(),str(df.iloc[i][2]).strip()) )
students_id.append(str(df.iloc[i][1]).strip())
except:
    #exc_type, exc_obj, exc_tb = sys.exc_info()
    #fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    #print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    pass
try:
    mutex_db.acquire()
    for student in students:
        c.execute("insert OR IGNORE into students (student_id, student_name) values
('%s','%s') " % (student))
        conn.commit()
        c.execute("SELECT section_id FROM section_info WHERE section_name=" + section_name + "'")
        result = c.fetchone()
        if result is None:
            print("insert into section_info (section_name) values (" + section_name + ")",
file=sys.stdout)
            sys.stdout.flush()
            c.execute("insert into section_info (section_name) values (" + section_name
+ "'")")
            conn.commit()
            c.execute("SELECT section_id FROM section_info WHERE section_name=" + section_name + "'")
            conn.commit()
            result = c.fetchone()
        result = result[0]
    for member in students_id:
        c.execute("SELECT section_id FROM student_in_section WHERE section_id =
%d and student_id='%s'" % (result, member) )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

result2 = c.fetchone()
if result2 is None:
    c.execute("insert into student_in_section values (%d, '%s') " % (result, member) )
    conn.commit()
    mutex_db.release()
    return {'status':'success', 'message':'Save complete'}
except:
    mutex_db.release()
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    return {'status':'failed', 'message':'Save failed'}

@app.route('/sections')
def section_list():
    try:
        mutex_db.acquire()
        result = c.execute("select section_id, section_name from section_info").fetchall()
        mutex_db.release()
        sections = []
        for row in result:
            sections.append({"section_id":row[0], "section_name":row[1]})
        return {'status':'success', 'sections':sections}
    except:
        mutex_db.release()
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
        print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
        return {'status':'failed'}

```

```

@app.route('/deletesection/<int:sectionid>')
def deletesections(sectionid):
    try:
        mutex_db.acquire()
        c.execute("delete from section_info where section_id=" + str(sectionid))
        conn.commit()
        mutex_db.release()
        return {'status':'success'}
    except Exception as ex:
        mutex_db.release()
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
        print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
        return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

@app.route('/delete_student_section/<string:studentid>/<string:sectionid>')
def delete_student_section(studentid,sectionid):
    try:
        if studentid is None or sectionid is None:
            return jsonify({'status':'failed', 'message': 'Error: Invalid Parameter'})
        mutex_db.acquire()
        c.execute("delete from student_in_section where section_id=" + str(sectionid)+
and student_id=' + str(studentid))
        conn.commit()
        mutex_db.release()
        return {'status':'success'}
    except Exception as ex:
        mutex_db.release()
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

```

```
@app.route('/delete_student/<string:studentid>')
```

```
def delete_student(studentid):
```

```
    global names,encodings
```

```
    try:
```

```
        if studentid is None:
```

```
            return jsonify({'status':'failed', 'message': 'Error: Invalid Parameter'})
```

```
        mutex_db.acquire()
```

```
        conn.commit()
```

```
        c.execute("delete from students where student_id=" + str(studentid) + "")
```

```
        mutex_face.acquire()
```

```
        std_index = list(np.where(np.array(names)==studentid))[0]
```

```
        if len(std_index) > 0:
```

```
            names = [i for j, i in enumerate(names) if j not in std_index]
```

```
            encodings = [i for j, i in enumerate(encodings) if j not in std_index]
```

```
            pickle.dump([encodings,names], open(datasets, 'wb'))
```

```
        face_model = SGDClassifier(loss='modified_huber')
```

```
        face_model = face_model.fit(encodings,names)
```

```
        face_model.sparsify()
```

```
    // ทำการ Train ข้อมูล
```

```
    ใบหน้าให้แก่
```

```
    Machine Learning ซึ่ง
```

```
    ใช้เทคนิค SGD
```

```
    Classification
```

```
        mutex_face.release()
```

```
        conn.commit()
```

```
        mutex_db.release()
```

```
        return {'status':'success'}
```

```
    except Exception as ex:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

conn.rollback()
mutex_db.release()
mutex_face.release()
exc_type, exc_obj, exc_tb = sys.exc_info()
fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

@app.route('/clear_trained_student/<string:studentid>')
def clear_trained_student(studentid):
    global names,encodings
    try:
        if studentid is None:
            return jsonify({'status':'failed', 'message': 'Error: Invalid Parameter'})
        mutex_face.acquire()
        std_index = list(np.where(np.array(names)!=studentid))[0]
        if len(std_index) > 0:
            names = [i for j, i in enumerate(names) if j not in std_index]
            encodings = [i for j, i in enumerate(encodings) if j not in std_index]
            pickle.dump([encodings,names], open(datasets, 'wb'))

            face_model = SGDClassifier(loss='modified_huber')
            face_model = face_model.fit(encodings,names)
            face_model.sparsify()
            // ทำการ Train ข้อมูล
            // ใบหน้าให้แก่
            // MachineLearning
            // ซึ่งใช้เทคนิค SGD
            // Classification

            mutex_face.release()
            return {'status':'success'}
    except Exception as ex:
        conn.rollback()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mutex_face.release()
exc_type, exc_obj, exc_tb = sys.exc_info()
fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

```

```

@app.route('/studentinsection/<int:sectionid>')
def studentinsection(sectionid):
    try:
        mutex_db.acquire()
        result = c.execute("select a.student_id,student_name from student_in_section
as a left join students as b on a.student_id = b.student_id where section_id=" + str(se
ctionid)).fetchall()
        conn.commit()
        mutex_db.release()
        sections = []
        for row in result:
            sections.append({"student_id":row[0], "student_name":row[1]})
        return {'status':'success', 'students':sections}
    except Exception as ex:
        mutex_db.release()
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
        print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
        return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

```

```

@app.route('/students')

```

```

def studentslist():

```

```

    try:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mutex_db.acquire()
result = c.execute("select student_id,student_name from students order by stu
dent_id").fetchall()
conn.commit()
mutex_db.release()
sections = []
for row in result:
    trained = False
    if str(row[0]).strip() in np.unique(names):
        trained = True
    sections.append({"student_id":row[0], "student_name":row[1], "trained":trained
})
return {'status':'success', 'students':sections}
except Exception as ex:
    mutex_db.release()
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lin
eno)+')'})

@app.route('/create_class', methods=['POST'])
def create_class():
    content = request.json
    print(content)
    try:
        try:
            mutex_db.acquire()
            sql="select class_id from class_info where section_id = %d and section_start
= '%s' and section_end = '%s'" % (int(content['section_id']),content['time_start'],conten
t['time_end']);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class_id = c.execute(sql).fetchone()
if class_id is None:
    c.execute("insert into class_info (section_id,section_start,section_end) values (%d, '%s', '%s') " % (int(content['section_id']),content['time_start'],content['time_end']))
    conn.commit()
    #class_id = c.lastrowid
    #total_student = c.execute("select count(student_id) from student_in_section where section_id = %d" % (int(content['section_id']))).fetchone()
    #sql = "select a.student_id as studentid, student_name, timestamp from (select student_id from student_in_section where section_id=%d) as c left join student_checkout as a on a.student_id=c.student_id left join students as b on a.student_id=b.student_id where class_id = %d order by timestamp asc" % (int(content['section_id']),int(class_id[0]))
    #student_list = c.execute(sql).fetchall()
    #checkedin_student = len(student_list)
    mutex_db.release()
    return {'status':'success'}
except:
    mutex_db.release()
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno)
    return {'status':'failed', 'message':exc_tb.tb_lineno}
return {'status':'failed'}
except:
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    return {'status':'failed'}

```

```

@app.route('/checkin', methods=['POST'])
def checkin():
    if 'classid' not in request.form:
        return {'status':'failed', 'message':'classid name not found'}
    if 'image' not in request.files:
        return {'status':'failed', 'message':'image_file not found'}
    timestamp = datetime.datetime.now().isoformat()
    classid = request.form.get('classid')
    image_file = request.files['image']
    if(classid is None or classid == "" or classid == "undefined"):
        return {'status':'failed', 'message':'classid name not found'}
    if(image_file is None or image_file.filename == ""):
        return {'status':'failed', 'message':'image_file not found' + image_file}
    try:
        im_pil = Image.open(image_file.stream) } // ทำการประมวลผลภาพ
        face = np.array(im_pil.convert('RGB')) } // และตรวจหาใบหน้า
        face_rec = fr.face_encodings(face) }
        checkin_count = 0
        if(len(face_rec) >= 1): } // ถ้ามีใบหน้าอย่างน้อย 1 คน ให้ทำการตรวจสอบ
            student_id_list = "" } // ที่ละใบหน้าแล้วบันทึกผลลงตัวแปร
        student_id_list
        for face_enc in face_rec:
            if face_enc is not None:
                mutex_face.acquire()
                result = face_model.predict_proba([face_enc]) //ทำการเปรียบเทียบใบหน้า

#ถ้าผลการเปรียบเทียบมีความใกล้เคียงต่ำกว่า 70% ให้ % ความใกล้เคียง หากมากกว่า 70% ให้ทำ
การเปรียบเทียบอีกครั้งเพื่อระบุตัวบุคคลแล้วบันทึกผล
        if (np.amax(result) < 0.7):
            print(result[0] + " == " + np.amax(result))
            mutex_face.release()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else:
    result = face_model.predict([face_enc])
    mutex_face.release()
    std_id = result[0]

    sectionid = c.execute("select section_id
from class_info where class_id=" + str(classid)).fetchone()[0]
    is_student_in_section = c.execute("select
count(student_id) from student_in_section where section_id="
+ str(sectionid) + " and student_id="+std_id+""").fetchone()[0]
    is_student_in_class = c.execute("select
count(student_id) from student_checkin where class_id="
+ str(classid) + " and student_id="+std_id+""").fetchone()[0]

    if is_student_in_section >=1 and
is_student_in_class <=0:
        mutex_db.acquire()
        sql = "insert into student_checkin
(class_id,student_id,timestamp) values (" + classid + ", "
+ result[0] + ", " + timestamp + """)
        student_id_list += std_id+", "
        checkin_count += 1
        c.execute(sql)
        conn.commit()
        mutex_db.release()

    if checkin_count > 0 :
        return {'status':'success', 'message':
'Checkin student ' + student_id_list}
    else:
        return {'status':'failed', 'message':
'Cannot recognize any people'}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return {'status':'failed', 'message':'Cannot detect face'}
except:
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    mutex_face.release()
    mutex_db.release()

return {'status':'failed', 'message':'System error'}

@app.route('/uploadaface', methods=['POST'])
def uploadaface():
    content = request.json
    if 'studentid' not in content or 'studentname' not in content:
        return {'status':'failed', 'message':'ID or Name not found'}
    if content['studentid'].strip()==" or content['studentname'].strip()=="":
        return {'status':'failed', 'message':'ID or Name not found'}
    try:
        mutex_db.acquire()
        c.execute("insert OR IGNORE into students (student_id, student_name) values ('
%s','%s') " % (content['studentid'].strip(), content['studentname'].strip()))
        conn.commit()
        mutex_db.release()

    imgcount = 0;
    if 'img' in content:
        for image_data in content['img']:
            im_pil = Image.open(BytesIO(base64.b64decode(image_data['img'])))
            face = np.array(im_pil.convert("RGB"))
            face_rec = fr.face_encodings(face)
            if face_rec is not None:

```

```

        face_rec = face_rec[0]
        encodings.append(face_rec)
        names.append(content['studentid'].strip())
        imgcount += 1
    mutex_face.acquire()
    pickle.dump([encodings,names], open(datasets, 'wb'))

    face_model = SGDClassifier(loss='modified_huber')
    face_model = face_model.fit(encodings,names)
    face_model.sparsify()
    mutex_face.release()
    return {'imgcount':imgcount,'status':'success'}
except:
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    mutex_face.release()
    mutex_db.release()
    return {'status':'failed', 'message':'System error'}
return {'status':'failed', 'message':'System error'}

@app.route('/getclasstime/<int:sectionid>')
def getclasstime(sectionid):
    try:
        mutex_db.acquire()
        result = c.execute("select class_id,section_start,section_end from class_info where section_id=" + str(sectionid)).fetchall()
        mutex_db.release()
        classinfo = []
        for row in result:
            classinfo.append({"time_id":row[0], "time_start":row[1], "time_end":row[2]})

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return {'status':'success', 'timelist':classinfo}
except Exception as ex:
    mutex_db.release()
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno)
    return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

@app.route('/getclassinfo/<int:classid>')
def getclassinfo(classid):
    try:
        total_student = c.execute("select count(student_id) from student_checkin where class_id = %d" % (classid) ).fetchone()
        sectionid = c.execute("select section_id from class_info where class_id = %d" % (classid) ).fetchone()

        sql = "select a.student_id as studentid, student_name, timestamp from (select student_id from student_in_section where section_id=%d) as c left join student_checkin as a on a.student_id=c.student_id left join students as b on a.student_id=b.student_id where class_id = %d order by timestamp asc" % (int(sectionid[0]),int(classid))
        student_list = c.execute(sql).fetchall()
        result = c.execute("select class_id,section_start,section_end from class_info where class_id=" + str(classid)).fetchone()

        return {'status':'success', 'student_list':student_list, 'total_student':total_student, "time_id":result[0], "time_start":result[1], "time_end":result[2]}
    except Exception as ex:
        mutex_db.release()
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
        print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

```

```

@app.route('/removeclass/<string:classid>')

```

```

def removeclass(classid):

```

```

    global names,encodings

```

```

    try:

```

```

        if classid is None:

```

```

            return jsonify({'status':'failed', 'message': 'Error: Invalid Parameter'})

```

```

        mutex_db.acquire()

```

```

        conn.commit()

```

```

        c.execute("delete from class_info where class_id=" + str(classid))

```

```

        conn.commit()

```

```

        mutex_db.release()

```

```

        return {'status':'success'}

```

```

    except Exception as ex:

```

```

        conn.rollback()

```

```

        mutex_db.release()

```

```

        exc_type, exc_obj, exc_tb = sys.exc_info()

```

```

        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]

```

```

        print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)

```

```

        return jsonify({'status':'failed', 'message': 'Error: '+ str(ex) + ' (' + repr(exc_tb.tb_lineno)+')'})

```

```

@app.route('/testface', methods=['POST'])

```

```

def testface():

```

```

    if 'image' not in request.files:

```

```

        return {'status':'failed', 'message':'image_file not found'}

```

```

    image_file = request.files['image']

```

```

    if(image_file is None or image_file.filename == ""):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return {'status':'failed', 'message':'image_file not found' + image_file}
try:
    im_pil = Image.open(image_file.stream)
    face = np.array(im_pil.convert('RGB'))
    face_rec = fr.face_encodings(face)
    if(len(face_rec) >= 1):
        student_id_list = ""
        for face_enc in face_rec:
            if face_enc is not None:
                mutex_face.acquire()
                result = face_model.predict_proba([face_enc])
                if (np.amax(result) < 0.7):
                    mutex_face.release()
                else:
                    result = face_model.predict([face_enc])
                    mutex_face.release()
                    std_id = result[0]
                    student_id_list += std_id+", "
            return {'status':'success', 'message':'Found student ' + student_id_list}
        return {'status':'failed', 'message':'Cannot detect face'}
except:
    exc_type, exc_obj, exc_tb = sys.exc_info()
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
    mutex_face.release()
    mutex_db.release()

return {'status':'failed', 'message':'System error'}

@app.route('/getreport/<int:sectionid>')
def getreport(sectionid):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    section_info = c.execute("select section_name from section_info where section_id
="+str(sectionid)).fetchone()
    """ List Class """
    sql = "select class_id, section_start, section_end from class_info where section_id=
" + str(sectionid) + " order by section_start ASC"
    class_info = c.execute(sql).fetchall()
    classlist = []
    for each_class in class_info:
        classlist.append({"cid": each_class[0], "tstart": each_class[1], "tend": each_class[2]}
)
    checkin_data = {}
    sql = "select a.student_id,timestamp from student_in_section as a left join student
_checkin as b on a.student_id=b.student_id where a.section_id=%d and b.class_id=%
d"
    for classinfo in classlist:
        checkin_class = []
        checkin_info = c.execute(sql % (sectionid,classinfo['cid'])).fetchall()
        for student in checkin_info:
            checkin_class.append({"sid":student[0],"ctime":student[1]})
        checkin_data.update({classinfo['cid']:checkin_class})
    student_list = []
    result = c.execute("select a.student_id,student_name from student_in_section as
a left join students as b on a.student_id = b.student_id where section_id=" + str(secti
onid) + " order by a.student_id").fetchall()
    for row in result:
        student_list.append({"student_id":row[0], "student_name":row[1]})

    return jsonify({"status":"success","classlist": classlist, "checkin_data":checkin_data, "st
udent_list":student_list, "section":section_info[0]})

@app.route('/manual_check', methods=['POST'])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def manual_check():
    content = request.json
    if(content["sid"] is None or content["sid"].strip() == "" or content["sid"] == "undefined
"):
        return {'status':'failed', 'message':'Student id not found'}
    try:
        classid = content["cid"]
        std_id = content["sid"]
        timestamp = content["timestamp"]
        sectionid = c.execute("select section_id from class_info where class_id=" + str(classid)).fetchone()[0]
        is_student_in_section = c.execute("select count(student_id) from student_in_section where section_id=" + str(sectionid) + " and student_id="+std_id+""").fetchone()[0]
        is_student_in_class = c.execute("select count(student_id) from student_checkin where class_id=" + str(classid) + " and student_id="+std_id+""").fetchone()[0]
        if is_student_in_section >=1 and is_student_in_class <=0:
            mutex_db.acquire()
            sql = "insert into student_checkin (class_id,student_id,timestamp) values (" + classid + ", " + std_id + ", " + timestamp + """)
            c.execute(sql)
            conn.commit()
            mutex_db.release()
            return {'status':'success', 'message':'Checkin student complete'}
        else:
            return {'status':'failed', 'message':'Student not found in section'}
    except:
        exc_type, exc_obj, exc_tb = sys.exc_info()
        fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
        print(exc_type, fname, exc_tb.tb_lineno, file=sys.stderr)
        mutex_db.release()
    return {'status':'failed', 'message':'System error'}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@app.route('/')
def index():
    return send_from_directory('www2', 'index.html')

@app.route('/<path:path>')
def www(path):
    return send_from_directory('www2', path)

# กำหนด Response Header ไม่ให้ Client เก็บ Cache เนื่องจากไฟล์แสดงผลเป็น Static HTML
# ทำให้ Browser เก็บ Cache และข้อมูลจะไม่อัปเดตตลอดเวลา (สามารถยกเลิกการทำงานของ
# ฟังก์ชันนี้ได้เมื่อเข้าสู่ระดับ Production)
@app.after_request
def add_header(r):
    """
    Add headers to both force latest IE rendering engine or Chrome Frame,
    and also to cache the rendered page for 10 minutes.
    """
    r.headers["Cache-Control"] = "no-cache, no-store, must-revalidate"
    r.headers["Pragma"] = "no-cache"
    r.headers["Expires"] = "0"
    r.headers['Cache-Control'] = 'public, max-age=0'
    return r

# ทำการตรวจสอบฐานข้อมูล หากฐานข้อมูลถูกสร้างขึ้นใหม่หรือมีบางตารางขาดหาย โปรแกรมจะสั่ง
# สร้างตารางและฐานข้อมูลใหม่ทันที
if __name__ == '__main__': // ส่วนเริ่มต้นการทำงาน
    conn.execute("PRAGMA foreign_keys = on")
    c = conn.cursor()

    sql = """CREATE TABLE IF NOT EXISTS "students" (
        "student_id" TEXT NOT NULL,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"student_name" TEXT NOT NULL,
PRIMARY KEY("student_id")
);"";
```

```
c.execute(sql)
```

```
sql = """CREATE TABLE IF NOT EXISTS "section_info" (
"section_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
"section_name" TEXT NOT NULL UNIQUE
);""";
```

```
c.execute(sql)
```

```
sql = """CREATE TABLE IF NOT EXISTS "student_in_section" (
"section_id" INTEGER NOT NULL,
"student_id" TEXT NOT NULL,
CONSTRAINT fk_section
FOREIGN KEY (section_id)
REFERENCES section_info (section_id)
ON DELETE CASCADE
CONSTRAINT fk_student
FOREIGN KEY (student_id)
REFERENCES students (student_id)
ON DELETE CASCADE
);""";
```

```
c.execute(sql)
```

```
sql = """CREATE TABLE IF NOT EXISTS "class_info" (
"class_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
"section_id" INTEGER NOT NULL,
"section_start" TEXT NOT NULL,
"section_end" TEXT NOT NULL,
CONSTRAINT fk_section
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FOREIGN KEY (section_id)
REFERENCES section_info (section_id)
ON DELETE CASCADE
);""";

```

```
c.execute(sql)
```

```

sql = """CREATE TABLE IF NOT EXISTS "student_checkin" (
"checkin_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
"class_id" INTEGER NOT NULL,
"student_id" TEXT NOT NULL,
"timestamp" TEXT NOT NULL,
CONSTRAINT fk_class
FOREIGN KEY (class_id)
REFERENCES class_info (class_id)
ON DELETE CASCADE
);""";

```

```
c.execute(sql)
```

```

sql = """CREATE UNIQUE INDEX IF NOT EXISTS "index_student" ON
"students" ( "student_id"
);""";

```

```
c.execute(sql)
```

```
conn.commit()
```

```
[encodings,names] = pickle.load(open(datasets, 'rb')) // ทำการอ่านข้อมูลใบหน้าตัวแปร
```

encoding เก็บพิกัดสำคัญบน

ในหน้าและ names เก็บชื่อ

ของใบหน้านั้น ๆ ลำดับข้อมูล

ของทั้งสองตัวแปรสอดคล้องกัน

```
t = time.time() // เริ่มนับเวลา
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

face_model = SGDClassifier(loss='modified_huber')
face_model = face_model.fit(encodings,names)
face_model.sparsify()

```

} // ทำการ Train Machine Learning ใหม่

```

elapsed = time.time() - t // สรุปเวลาที่ใช้ในการ Train ข้อมูล (เพื่อให้ทราบระยะเวลา
โดยประมาณในการ Train)

```

```

print(elapsed) // แสดงเวลาที่ใช้ในการ Train

```

```

#print(face_model.classes_)

```

//เริ่มการทำงานของ Server โดย 0.0.0.0 หมายถึงอนุญาตให้เข้าถึงได้จากทุก IP Address และ
ssl_context='adhoc' หมายถึงเปิดใช้โพรโทคอล SSL/HTTPS โดยมี Certificate แบบ Self-Sign

```

app.run(debug=True, host='0.0.0.0', ssl_context='adhoc')

```

```

#app.run(debug=True, host='0.0.0.0')

```