การจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า

FACE CLASSIFICATION USING FACE ENCODING DATA

พฤทธิ์ สาริวงค์

โปรแกรมวิชาวิศวกรรมคอมพิวเตอร์

คณะเทคโนโลยีดิจิทัล

มหาวิทยาลัยราชภัฏเชียงราย

ปีการศึกษา 2566

© ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏเชียงราย

การจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า

พฤทธิ์ สาริวงค์
ปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์

โปรแกรมวิชาวิศวกรรมคอมพิวเตอร์
คณะเทคโนโลยีดิจิทัล มหาวิทยาลัยราชภัฏเชียงราย
2566

© ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏเชียงราย

ใบรับรองงานวิจัย
คณะเทคโนโลยีดิจิทัล มหาวิทยาลัยราชภัฏเชียงราย

เรื่อง การจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า
โดย นายพฤทธิ์ สาริวงค์

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ 2566

.....รองคณบดีฝ่ายวิชาการ
(อาจารย์ อนุสรณ์ ใจแก้ว)

คณะกรรมการสอบงานวิจัย

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กฤตกรณ์ ศรีวันนา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.มยูร ใยบัวเทศ)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ธนาวุฒิ ธนวาณิชย์)

.....กรรมการ
(อาจารย์ อธิคม ศิริ)

กิตติกรรมประกาศ

การจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า สำเร็จลุล่วงไปด้วยดีไม่ได้ หากไม่ได้รับความกรุณาจากบุคคลหลาย ๆ ท่าน ข้าพเจ้าขอขอบคุณบุคคลสำคัญดังที่จะกล่าวถึงดังต่อไปนี้เป็นอย่างยิ่งที่ช่วยให้คำแนะนำแก้ไขปัญหา ตลอดจนช่วยให้ข้าพเจ้าทำการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า สำเร็จลุล่วงไปได้ด้วยดี ซึ่งได้แก่ บิดามารดา ที่ให้การสนับสนุน ตลอดจนผู้ช่วยศาสตราจารย์ ดร.กฤตกรณ์ ศรีวันนา อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร.ภูมิพงษ์ ดวงตั้ง ผู้ช่วยศาสตราจารย์ ดร.ธนาวุฒิ ธนวาณิชย์ ผู้ช่วยศาสตราจารย์ กมล บุญล้อม และ อาจารย์อธิคม ศิริ ที่คอยสั่งสอนวิชาเพื่อนำไปใช้ในการทำงาน ให้คำปรึกษาและคำแนะนำในการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า ตั้งแต่เริ่มต้นจนเสร็จสิ้นกระบวนการทำงาน

สำหรับคุณงามความดีใดที่เกิดจากการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีต่อข้าพเจ้า ช่วยเหลือและนำแนะสิ่งต่าง ๆ ให้กับ ข้าพเจ้า

นายพฤทธิ์ สาริวงค์
2 พฤศจิกายน 2566

ชื่องานวิจัย การจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า

ชื่อผู้วิจัย นายพฤทธิ์ สาริวงค์

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ปีการศึกษา 2566

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.กฤตกรณ์ ศรีวันนา

บทคัดย่อ

เนื่องด้วยสถานการณ์การแพร่ระบาดของไวรัสโควิด-19 ในปัจจุบัน ดังนั้นบริษัทและองค์กรหลายแห่งเริ่มหันมาใช้เทคโนโลยีการจดจำใบหน้า ในการยืนยันตัวตนของพนักงานหรือบุคลากรเพิ่มมากขึ้น งานวิจัยนี้ได้จัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า โดยได้ทำการทดลองใช้โมเดลจัดประเภททั้งหมด 3 โมเดลคือ 1)Support Vector Machine(SVM), 2)K-Nearest Neighbors(KNN) และ3)Multi-Layer Perceptron(MLP) เพื่อเปรียบเทียบว่าอัลกอริทึมประเภทใดที่เหมาะสมในการทำนายใบหน้า ด้วยข้อมูลการเข้ารหัสใบหน้ามากกว่ากัน ซึ่งงานวิจัยนี้ได้ใช้รูปใบหน้าทั้งหมด 260 ภาพ จากผู้ทดสอบทั้งหมด 20 คน โดยมีกระบวนการ 2 ขั้นตอนคือ 1)การเข้ารหัสใบหน้า และ 2)การโมเดลและทำนายผล ซึ่งจากผลการทดลอง แสดงค่าความแม่นยำของแต่ละโมเดลดังต่อไปนี้ 1)MLP มีค่าความแม่นยำ 99.62%, 2)KNN มีค่าความแม่นยำ 99.62% และ 3)SVM มีค่าความแม่นยำ 97.69%, ซึ่งแสดงให้เห็นว่าในงานวิจัยนี้โมเดลที่มีประสิทธิภาพมากที่สุด ในการเรียนรู้จากข้อมูลการเข้ารหัสใบหน้าคือโมเดลจากอัลกอริทึม MLP

คำสำคัญ: จดจำใบหน้า, การจัดหมวดหมู่, การเรียนรู้ของเครื่อง

สารบัญ

หน้า

กิตติกรรมประกาศ ก

บทคัดย่อ ข

สารบัญตาราง จ

สารบัญภาพ ฉ

บทที่

1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 1

1.3 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย 2

1.4 ขอบเขตของการวิจัย 2

2 แนวคิด ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง 3

2.1 แนวคิด 3

2.2 ทฤษฎีที่ใช้ที่ใช้ในการวิจัย 4

2.3 งานวิจัยที่เกี่ยวข้อง 12

3 วิธีดำเนินการวิจัย 15

3.1 เครื่องมือที่ใช้ในงานวิจัย 16

3.2 ขั้นตอนการทำงานของระบบ 17

4 ผลการดำเนินงานวิจัย 21

4.1 ผลการทดลองของแต่ละโมเดล 21

4.2 สรุปผลการทดสอบโมเดล 22

5 สรุปและอภิปรายผล 25

สารบัญ (ต่อ)

หน้า

บทที่

5.1 สรุปผลและอภิปรายผล 25

5.2 ข้อจำกัด 25

5.3 ข้อเสนอแนะและงานวิจัยในอนาคต 26

บรรณานุกรม 28

ภาคผนวก 29

ก โค้ดการทำงานของโปรแกรม 30

ประวัติย่อของผู้วิจัย 36

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

ภาพที่ **หน้า**

2.1 ตัวอย่างการทำระบุตำแหน่งใบหน้า 5
2.2 ตัวอย่างการเข้ารหัสใบหน้า 6
2.3 ตัวอย่างเส้นแบ่งชั้น 8
2.4 ตัวอย่างการจำแนก 9
2.5 ตัวอย่างชั้นเรียนรู้ของโมเดล MLP 10
3.1 แผนผังการดำเนินงานวิจัยการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า 16
3.2 ขั้นตอนการทำงาน 17
3.3 ตัวอยางการเก็บภาพ 18
3.4 ตัวอย่างการเก็บข้อมูลเข้ารหัสใบหน้า 20
4.1 ผลการทดสอบโมเดล SVM 22
4.2 ผลการทดสอบโมเดล KNN 23
4.3 ผลการทดสอบโมเดล MLP 23
4.4 ผลการทดสอบเฉลี่ยของแต่ละโมเดล 24

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเนื่องจากสถานการณ์การแพร่ระบาดของไวรัสโควิด 19 ที่ผ่ามมาบริษัทและองค์กรหลายแห่งที่ดำเนินการยืนยันตัวตนของพนักงานหรือบุคลากรโดยใช้เทคโนโลยีการสแกนลายนิ้วมือ ซึ่งเป็นเส้นทางที่เสี่ยงต่อการแพร่ของเชื้อไวรัส จึงทำให้บริษัทและองค์กรเหล่านั้นได้เริ่มปรับเปลี่ยนแนวทางในการยืนยันตัวตน โดยใช้เทคโนโลยีการจดจำใบหน้าแทน ซึ่งจะช่วยลดการสัมผัสและลดความเสี่ยงในการแพร่ของเชื้อไวรัส

เทคโนโลยีการจดจำใบหน้าเป็นส่วนหนึ่งของการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกเพื่อให้คอมพิวเตอร์เรียนรู้ข้อมูลด้วยอัลกอริทึมซึ่งมีอัลกอริทึมให้เลือกใช้อย่างมากมาย ด้วยเหตุดังกล่าว ผู้วิจัยจึงได้เลือกอัลกอริทึมมา 3 ตัวคือ Support Vector Machine(SVM), K-Nearest Neighbors(KNN) และ Multi-Layer Perceptron(MLP) สำหรับทดสอบและหาประสิทธิภาพของแต่ละอัลกอริทึมว่ามีประสิทธิภาพในการรู้จำใบหน้ามากน้อยเพียงใดเพื่อช่วยตัดสินใจในการเลือกใช้อัลกอริทึมสำรับการนำไปประยุคต์ใช้กับโปรแกรมระบุตัวตน

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและทำความเข้าใจอัลกอริทึมที่ใช้ในการรู้จำใบหน้า
2. เพื่อหาว่าอัลกอริทึมไหนมีประสิทธิภาพในการเรียนรู้ข้อมูลด้วยการเข้ารหัสใบหน้ามากกว่ากัน
3. เพื่อช่วยแนะนำการเลือกใช้อัลกอริทึมที่เหมาะสมในการพัฒนาระบบจดจำใบหน้า

1.3 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. ได้ทราบถึงความแม่นยำของอัลกอริทึมที่เรียนรู้ด้วยการเข้ารหัสใบหน้า
2. ได้ช่วยให้ผู้พัฒนาระบบสามารถเลือกใช้อัลกอริทึมที่เหมาะสมกับงานหรือโครงการได้อย่างมีประสิทธิภาพ

1.4 ขอบเขตของการวิจัย

1. การวิจัยนี้ใช้ข้อมูลการเข้ารหัสใบหน้าในการเรียนรู้เท่านั้น
2. อัลกอริทึมที่ใช้เปรียบเทียบผลประสิทธิภาพมีแค่ SVM, KNN และ MLP เท่านั้น
3. ขอบเขตของการวิจัยจะมุ่งเน้นการวัดและเปรียบเทียบประสิทธิภาพของ SVM, KNN และ MLP ในการรู้จำใบหน้า ไม่ได้ครอบคลุมการศึกษาเกี่ยวกับการทำงานของอัลกอริทึมในระดับลึก

บทที่ 2

แนวคิด ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ผู้วิจัยได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องเพื่อนำมาใช้สำหรับการกำหนดกรอบ ความ คิด หลัก การ ทฤษฎี เครื่อง มือ เช่น Deep Learning, Support Vector Machine(SVM), K-Nearest Neighbors(KNN)และ Multilayer Perceptron (MLP) การรวบรวมข้อมูล การวิเคราะห์และการอภิปรายผลการศึกษา ซึ่งประกอบไปด้วยเนื้อหา ดังต่อไปนี้

2.1 แนวคิด

2.1.1 การจดจำใบหน้า

การจดจำใบหน้า (face recording) เป็นกระบวนการที่ใช้เทคโนโลยีคอมพิวเตอร์ในการระบุและจดจำลักษณะหน้าของบุคคลหรือสิ่งของที่มีใบหน้า เพื่อใช้ในการตรวจสอบและระบุตัวตนของบุคคลนั้น ๆ หรือวัตถุนั้น ๆ ตามลักษณะใบหน้าที่ถูกจดจำไว้ เทคโนโลยีการจดจำใบหน้ามักใช้ในหลายแง่มุมและสามารถประยุกต์ใช้ในหลายสถานการณ์ เช่น ระบบรักษาความปลอดภัย, การตรวจสอบการลงทะเบียน,การจดจำใบหน้าสำหรับการระบุตัวตน

การจดจำใบหน้าใช้เทคโนโลยีการเรียนรู้ของเครื่องจักร (Machine Learning) และการประมวลผลภาพเพื่อสกัดลักษณะที่แยกสิ่งของหรือบุคคลในรูปภาพ เทคโนโลยีนี้มีการใช้งานกับความเป็นส่วนตัวและความปลอดภัยมากขึ้นในชีวิตประจำวันและในอุตสาหกรรมต่าง ๆ ในระหว่างปัจจุบันและอนาคต

2.2 ทฤษฎีที่ใช้ที่ใช้ในการวิจัย

2.2.1 การเรียนรู้ของเครื่อง (Machine Learning)

Machine Learning (ML) คือสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence) ที่เน้นการพัฒนาระบบคอมพิวเตอร์ที่สามารถเรียนรู้และปรับตัวเองจากข้อมูล โดยไม่ต้องโครงหรือโปรแกรมให้มีลิจีนแบบตรงไปตรงมา ในการเรียนรู้, ระบบคอมพิวเตอร์นี้ใช้ข้อมูลเป็นพื้นฐานเพื่อทำนายและทำงานให้ดีขึ้นเรื่อยๆ ตลอดเวลา โดยมีการใช้อัลกอริทึมและการคำนวณทาง คณิตศาสตร์ เพื่อ สร้าง แบบ จำลอง ที่ สามารถ แก้ ปัญหา หรือ ทำงาน แบบ เฉพาะ ทาง ได้ ประเภทหนึ่งของ Machine Learning คือ ”การเรียนรู้แบบจดจำ” (Supervised Learning) ซึ่งมีการสอนระบบโดยให้ข้อมูลอินพุตและผลลัพธ์ที่เฉยๆ หรือข้อมูลเรียนรู้มาให้ระบบเรียนรู้จากข้อมูลดังกล่าวและสร้างแบบจำลองที่สามารถทำนายผลลัพธ์จากข้อมูลใหม่ นอกจากนี้, ยังมีประเภทอื่น ๆ ของ Machine Learning เช่น ”การเรียนรู้แบบไม่จดจำ” (Unsupervised Learning) และ ”การเรียนรู้แบบเสริมซึ่งเรียกว่า Reinforcement Learning ที่ให้ระบบเรียนรู้จากประสบการณ์และการตอบรับจากสิ่งแวดล้อม

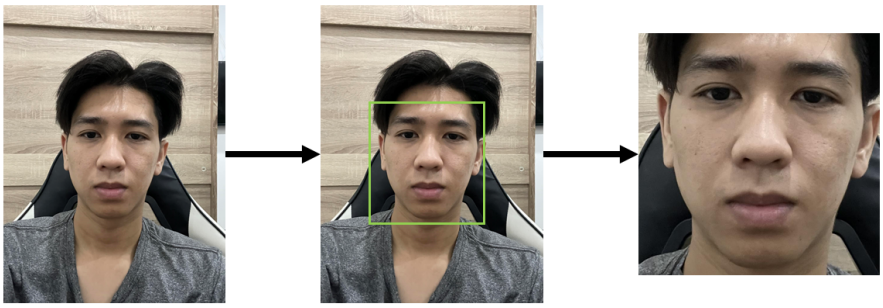
Machine Learning มีการประยุกต์ใช้ในหลายสาขาและงานต่าง ๆ เช่นการจัดการการเรียนรู้เครื่องจักร (Machine Learning for Education), การวิเคราะห์ข้อมูลทางการเงิน (Financial Data Analysis), การวิเคราะห์ข้อมูลทางการแพทย์ (Medical Data Analysis), การคาดการณ์การสูญเสียในธุรกิจ (Business Loss Prediction), การปรับปรุงการควบคุมโรค (Disease Control), การส่งเสริมการขายออนไลน์ (Online Sales Promotion), และหลายงานอื่น ๆ ที่ต้องการการวิเคราะห์ข้อมูลและการคาดการณ์โดยใช้คอมพิวเตอร์

การใช้ Machine Learning มีการเพิ่มขึ้นในตำแหน่งที่สำคัญในวงการเทคโนโลยีและธุรกิจ เนื่องจากความสามารถในการปรับตัวเรียนรู้จากข้อมูลใหญ่ ๆ และซับซ้อน เพื่อช่วยให้คอมพิวเตอร์ทำงานในลักษณะที่มนุษย์มีความยากลำบากในการทำมากขึ้น และการพัฒนาและการ ใช้ งาน Machine Learning ยัง คง เป็น หัวใจ ของ การ ควบคุม หุ่น ยนต์ อัตโนมัติ (Autonomous Robotics), การพัฒนาแอปพลิเคชันอัจฉริยะ (Smart Applications), การวิเคราะห์ข้อมูลอินเทอร์เน็ตของสรรพสิ่งแวดล้อม (IoT Data Analysis) และงานอื่น ๆ ที่เกี่ยวข้องกับการทำนายและการคาดการณ์

2.2.2 ไลบรารี การจดจำใบหน้า (Face Recognition libraries)

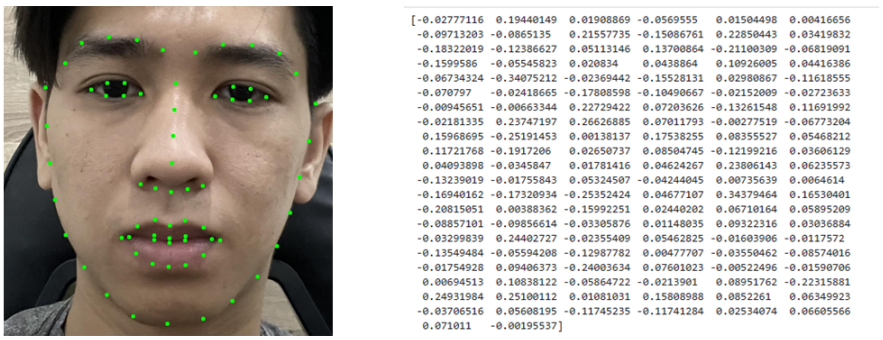
Face recognition เป็นไลบรารี Python ที่ใช้สำหรับงานตรวจจับและระบุใบหน้าในรูปภาพ ไลบรารีนี้เป็นตัวพัฒนาโดย Adam Geitgey และใช้งานง่ายและมีประสิทธิภาพสำหรับการทำ Face Recognition และ Face Detection บนรูปภาพหรือวิดีโอ โดยงานวิจัยนี้ได้ใช้ฟังก์ชันของ Face recognition ดังต่อไปนี้

2.2.2.1 ตำแหน่งใบหน้า(face_locations functions) เป็นฟังก์ชันที่มีในไลบรารี Face Recognition ใน Python ที่ใช้ในการค้นหาตำแหน่งของใบหน้าในรูปภาพ โดยฟังก์ชันนี้จะคืนค่าตำแหน่งของใบหน้าในรูปภาพในรูปแบบของสี่เหลี่ยมผืนผ้า (bounding box) ที่รอบครอบใบหน้าที่ถูกตรวจจับออกมาในรูปภาพ ดังภาพที่ 2.1



ภาพที่ 2.1 ตัวอย่างการทำระบุตำแหน่งใบหน้า

2.2.2.2 การเข้ารหัสใบหน้า(face_encodings) เป็นฟังก์ชันที่ใช้ในไลบรารี Face Recognition ใน Python เพื่อสกัดคุณลักษณะหรือเวกเตอร์การแสดงตัวของใบหน้า (face_encodings) จากรูปภาพของใบหน้าที่ตรวจจับได้ คุณลักษณะหรือเวกเตอร์การแสดงตัวนี้เป็นข้อมูลที่เป็นตัวเลขที่อธิบายลักษณะของใบหน้าในภาพอย่างละเอียด สามารถใช้ในการระบุใบหน้าหรือเปรียบเทียบใบหน้าในภาพต่าง ๆ หรือในการสร้างระบบ Face Recognition ดังภาพที่ 2.2



ภาพที่ 2.2 ตัวอย่างการเข้ารหัสใบหน้า

2.2.3 ไพทอน (Python)

Python เป็นภาษาโปรแกรมคอมพิวเตอร์ที่ถูกออกแบบมาเพื่ออ่านง่ายและมีความเรียบง่าย มันเป็นภาษาโปรแกรมระดับสูง (high-level programming language) ที่ถูกสร้างขึ้นโดย Guido van Rossum และปล่อยตัวเวอร์ชันแรกของมันในปี 1991 ซึ่งทำให้ Python เป็นหนึ่งในภาษาโปรแกรมที่มีอายุมากที่สุดและได้รับความนิยมอย่างแพร่หลายในวงการโปรแกรมนานาประเภท และความนิยมนี้ยังคงเติบโตขึ้นตลอดเวลา

Python มีความแตกต่างจากภาษาโปรแกรมอื่นๆ ในด้านความอ่านง่ายและมีสัญลักษณ์ที่เข้าใจง่าย ซึ่งทำให้มันเป็นภาษาที่เหมาะสำหรับผู้เริ่มต้นในการเรียนรู้การเขียนโปรแกรม นอกจากนี้ Python ยังเป็นภาษาโปรแกรมแบบมาตรฐานที่มีชุมชนสำหรับการพัฒนาและการสนับสนุนที่กว้างขวาง มีไลบรารีและโมดูลที่มีอยู่มากมายให้ใช้งานฟรี ทำให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่มีความหลากหลายและมีประสิทธิภาพได้อย่างรวดเร็ว

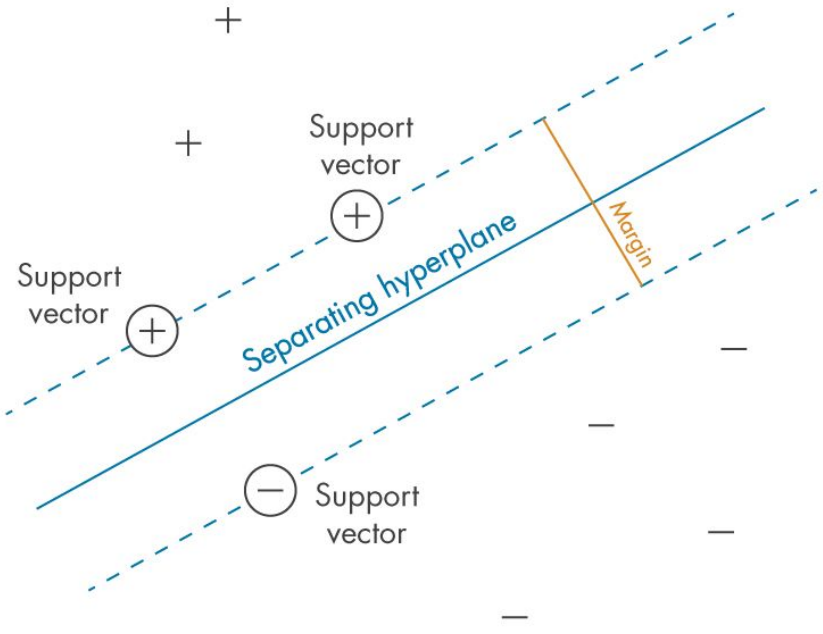
Python ใช้งานในหลายองค์กรและโดเมนต่างๆ เช่น วงการซอฟต์แวร์, การพัฒนาเว็บ, การวิเคราะห์ข้อมูล และการเรียนรู้เครื่องจักร (Machine Learning) และการพัฒนาแอปพลิเคชันสื่อสาร นอกจากนี้ Python ยังเป็นภาษาที่นิยมในการทำงานทางวิทยาศาสตร์ข้อมูลและการทำงานทางวิทยาศาสตร์ชีวภาพ เนื่องจากมีไลบรารีที่มีประสิทธิภาพสำหรับการวิเคราะห์ข้อมูลและการจัดการข้อมูลที่สามารถใช้งานได้อย่างดีในงานดังกล่าว ในสรุป Python เป็นภาษาโปรแกรมที่หลากหลายและเหมาะสำหรับการพัฒนาซอฟต์แวร์ในหลายด้านต่างๆ และมีความนิยมอย่างแพร่หลายในวงการโปรแกรม

2.2.4 อัลกอริทึมที่ใช้ในงานวิจัยนี้

2.2.4.1 Support Vector Machine (SVM) เป็นอัลกอริทึมการเรียนรู้ที่ใช้ในการแก้ปัญหาการจำแนกและการพยากรณ์ในหลายด้านงาน เช่น การประมวลผลสัญญาณและการแอปพลิเคชันทางการแพทย์ การประมวลผลภาษาธรรมชาติ (natural language processing) และการรู้จำเสียงและการรู้จำรูปภาพ (speech and image recognition) รวมถึงงานการประมวลผลสัญญาณและการรักษาทางการแพทย์ เป็นต้น อัลกอริทึม SVM เป็นอัลกอริทึมการเรียนรู้แบบควบคุม (supervised learning) ที่มีความสามารถในการจัดกลุ่มข้อมูลและทำนายค่าต่างๆ ในแง่กว้างของงานที่แตกต่างกันในสาขาต่างๆ

วัตถุประสงค์ของอัลกอริทึม SVM คือการค้นหาเส้นแบ่งชั้น (hyperplane) ที่สามารถแบ่งข้อมูลจากคลาสหนึ่งออกจากคลาสอื่นอย่างดีที่สุดเท่าที่จะเป็นไปได้ คำว่า ”ดีที่สุด” ถูกนิยามว่าเส้นแบ่งชั้นที่มีระยะห่าง (margin) ที่ใหญ่ที่สุดระหว่างคลาสสองคลาส แทนด้วยสัญลักษณ์บวกกับลบ ดังภาพที่ 2.3 ระยะห่างหมายถึงความกว้างสูงสุดของแผ่นกระดานขนานกับเส้นแบ่งชั้นที่ไม่มีจุดข้อมูลอยู่ข้างใน อัลกอริทึมนี้สามารถค้นหาเส้นแบ่งชั้นแบบเชิงเส้นได้เฉพาะในกรณีที่ข้อมูลสามารถแบ่งเชิงเส้น สำหรับปัญหาทางการใช้งานจริงแทบทุกครั้ง อัลกอริทึมจะพยายามให้ระยะห่างระหว่างแผ่นกระดานแบบนุ่ม (soft margin) ให้มากที่สุด โดยอนุญาตให้มีจำนวนข้อมูลที่ตีความผิดพลาดน้อยๆ บางจำนวน

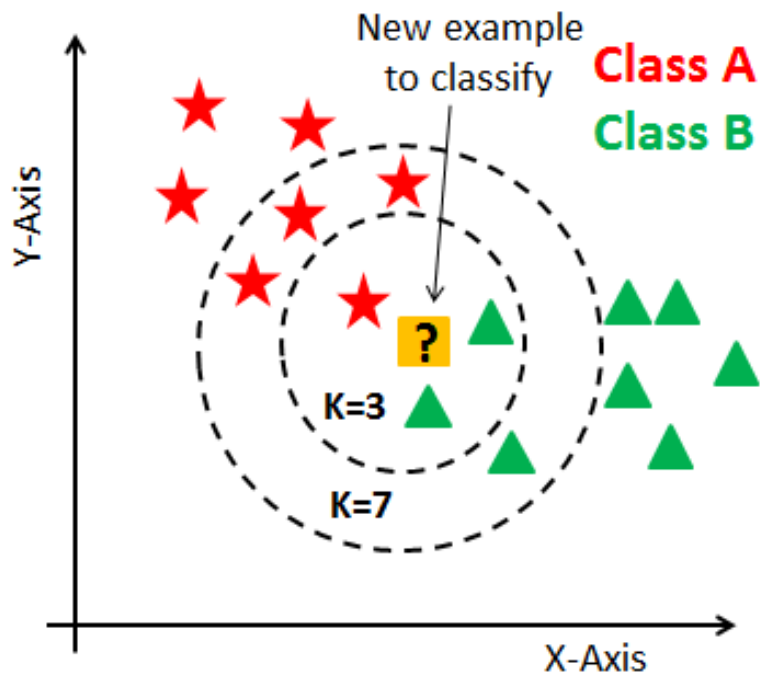
นอกจากการแยกแยกแบบเส้นตรงแล้ว SVM ยังสามารถใช้ในการแยกแยกข้อมูลแบบไม่เส้นตรง (non-linear separation) โดยใช้เคิร์ฟ (kernel) ที่เปลี่ยนแนวกั้นเส้นตรงให้เป็นแนวกั้นแบบฟังก์ชันโค้ง (curved) เพื่อรองรับข้อมูลที่มีความซับซ้อนมากขึ้น แนวกั้นแบบไม่เส้นตรงนี้ช่วยให้ SVM สามารถใช้ในงานที่มีข้อมูลที่มีความซับซ้อนหรือไม่สามารถแยกแยกด้วยแนวกั้นเส้นตรงได้



ภาพที่ 2.3 ตัวอย่างเส้นแบ่งชั้น

2.2.4.2 k-nearest neighbors (KNN) เป็นอัลกอริทึมในการจำแนกข้อมูลและการจำแนกประเภท (classification) ในเครื่องมือเรียนรู้เครื่องจักรซึ่งใช้หลักการค้นหาข้อมูลใกล้เคียง (nearest neighbors) เพื่อตัดสินใจว่าข้อมูลที่ต้องการจำแนกอยู่ในกลุ่มหรือประเภทใด อัลกอริทึมนี้เป็นอัลกอริทึมการจำแนกแบบการเรียนรู้เครื่องจักรโดยมีความง่ายต่อการเรียนรู้และใช้งาน

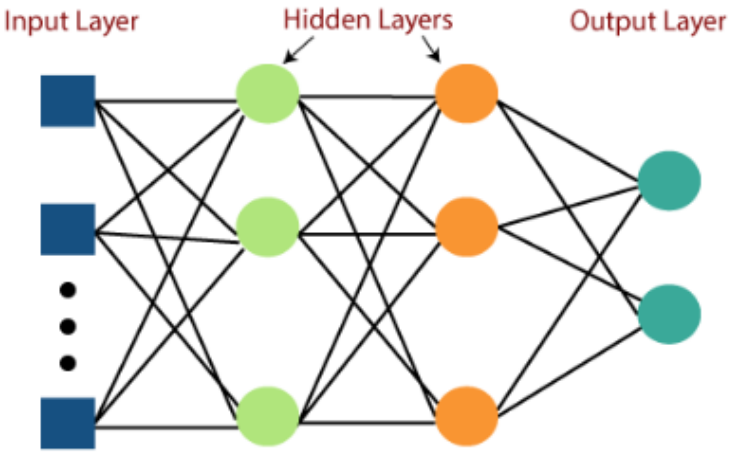
หลักการทำงานของ KNN คือการค้นหาตัวอย่างที่ใกล้ที่สุด K ตัวอย่างจากชุดข้อมูลสำหรับข้อมูลที่ต้องการจำแนก จากนั้น KNN จะนับจำนวนตัวอย่างในแต่ละกลุ่มหรือประเภทที่ใกล้ที่สุดและตัดสินใจให้ข้อมูลที่ต้องการจำแนกอยู่ในกลุ่มที่มีจำนวนตัวอย่างมากที่สุดใน K ตัวอย่างนั้น ๆ โดยตัวอย่าง K ที่ใกล้ที่สุดเหล่านี้เรียกว่า ”เพื่อนบ้าน” หรือ ”Neighbors” ตัวอย่างการจำแนกดังภาพที่ 2.4



ภาพที่ 2.4 ตัวอย่างการจำแนก

2.2.4.3 MLP (Multilayer Perceptron) เป็นโมเดลเครื่องเรียนรู้เชิงลึก (Deep Learning) ในด้านของการประมวลผลข้อมูลแบบกลุ่ม (feedforward neural network) และมีโครงสร้างที่ประกอบด้วยชั้น (layers) หลาย ๆ ชั้น ที่เชื่อมต่อกัน โดยมีหน้าที่ในการแก้ปัญหาทาง คณิตศาสตร์ และ สถิติ ใน หลาย งาน และ แอปพลิเคชัน ที่ เกี่ยวข้อง กับ การ จัด กลุ่ม ข้อมูล (classification) และการทำนาย (prediction)

MLP เป็นหนึ่งในโมเดลเครื่องเรียนรู้ที่มีความยืดหยุ่นมากและสามารถปรับปรุงตัวเองเพื่อเรียนรู้รูปแบบซับซ้อน โดยประกอบด้วยชั้นอย่างน้อย 3 ชั้นหรือ input layer, hidden layers, และ output layer ส่วน hidden layers นั้นอาจมีหลายชั้นและมีจำนวนโหนสามารถปรับเปลี่ยนได้ตามความต้องการของงาน ตัวอย่างชั้นเรียนรู้ดังภาพที่ 2.5



ภาพที่ 2.5 ตัวอย่างชั้นเรียนรู้ของโมเดล MLP

2.2.5 Scikit-Learn library

Scikit-Learn เป็นไลบรารี (library) สำหรับ Python ที่ใช้ในการทำ Machine Learning และ Data Mining อย่างมีประสิทธิภาพและเป็นที่นิยมอย่างแพร่หลายในวงการวิทยาศาสตร์ข้อมูลและการพัฒนาแอปพลิเคชันแบบ Machine Learning ใน Python ไลบรารีนี้ถูกสร้างขึ้นโดยนักพัฒนา Python ชุมชน และมีความเน้นที่ความง่ายในการใช้งาน, ประสิทธิภาพในการทำงาน, และความยืดหยุ่นในการใช้งานหลายประเภทของแบบจำลอง Machine Learning.

Scikit-Learn มีความสามารถในการรองรับการเรียนรู้แบบจดจำ (Supervised Learning) และการเรียนรู้แบบไม่จดจำ (Unsupervised Learning) รวมถึงการจัดการข้อมูล (data preprocessing), การเลือกคุณลักษณะ (feature selection), การประเมินแบบจำลอง (model evaluation), และ การ ทำความ เข้าใจ ผลลัพธ์ ของ แบบ จำลอง (model interpretation) ซึ่งทำให้มันเป็นเครื่องมือที่สมบูรณ์และเหมาะสำหรับงานการพัฒนาและทดสอบแบบจำลอง Machine Learning ต่างๆ โดยงานวิจัยนี้ได้ใช้ฟังก์ชันของ scikit-Learn ดังต่อไปนี้

2.2.5.1 cross_val_score functions cross_val_score เป็นฟังก์ชันหนึ่งในไลบรารี Scikit-Learn ที่ใช้ในการทำ Cross-Validation ของแบบจำลอง Machine Learning ใน Python. Cross-Validation เป็นเทคนิคที่ใช้ในการประเมินประสิทธิภาพของแบบจำลอง Machine Learning โดยการแบ่งข้อมูลออกเป็นส่วนๆ และทดสอบแบบจำลองบนชุดข้อมูลส่วนหนึ่ง จากนั้นจะเปลี่ยนส่วนที่ถูกทดสอบและส่วนที่ใช้ในการฝึกแบบจำลอง และทำกระบวนการนี้ซ้ำจนกว่าทุกส่วนของข้อมูลจะถูกทดสอบอย่างน้อยครั้งหนึ่ง ซึ่งช่วยในการประเมินความประสิทธิภาพของแบบจำลองอย่างเป็นอิสระและลดความเสี่ยงที่อาจเกิดจากการแบ่งข้อมูลออกเพียงครั้งเดียว (single split) เท่านั้น

cross_val_score จะคำนวณคะแนนประสิทธิภาพของแบบจำลองบนแต่ละชุดข้อมูลทดสอบแล้วคืนคะแนนเหล่านี้ออกมาเป็นรายการ (list) ซึ่งนักพัฒนาสามารถใช้ค่าเฉลี่ยหรือค่ามัธยฐานของคะแนนเหล่านี้เพื่อประเมินความประสิทธิภาพของแบบจำลอง โดยทั่วไปแล้ว, Cross-Validation ช่วยในการปรับแต่งแบบจำลอง, การเลือกคุณลักษณะ (feature selection), และการประเมินความประสิทธิภาพของแบบจำลองอย่างเป็นระบบและถูกใช้งานอย่างกว้างขวางในการพัฒนาและทดสอบแบบจำลอง Machine Learning

2.2.5.2 svm.SVC functions svm.SVC คือคลาสในไลบรารี Scikit-Learn (sklearn) ที่ใช้ในการสร้างแบบจำลอง Support Vector Machine (SVM) สำหรับงานการจำแนกข้อมูล (classification) ใน Machine Learning. SVM เป็นอัลกอริทึมที่ใช้ในการจำแนกข้อมูลด้วยการสร้างเส้นแบ่งคลาสที่แยกแยะระหว่างคลาสต่าง ๆ โดยใช้หลักการของขอบเขตของคลาสและเวกเตอร์สนับระหว่างเส้นแบ่ง และมีความสามารถในการจำแนกข้อมูลแบบเชิงเชิงเป็นเชิงเส้น (linear) และแบบเชิงอย่างมีประสิทธิภาพ (non-linear) โดยการใช้ฟังก์ชันหนึ่งที่เรียกว่า ”kernel.”

2.2.5.3 KNeighborsClassifier functions KNeighborsClassifier เป็นคลาสในไลบรารี Scikit-Learn (sklearn) ที่ใช้ในการสร้างแบบจำลอง K-Nearest Neighbors (KNN) สำหรับงานการจำแนกข้อมูล (classification) ใน Machine Learning. แบบจำลอง KNN นี้ใช้หลักการของการจำแนกข้อมูลด้วยการหาค่า K ตัวอย่างที่ใกล้ที่สุดจากข้อมูลฝึก (training data) เพื่อตัดสินใจว่าแต่ละตัวอย่างในข้อมูลทดสอบ (test data) ควรจำแนกเป็นคลาสใด ๆ โดยคำนวณโดยการคิดคะแนนหรือความคล้ายคลึงระหว่างตัวอย่างในเชิงเส้น.

2.2.5.4 MLPClassifier functions MLPClassifier เป็นคลาสในไลบรารี Scikit-Learn (sklearn) ที่ใช้ในการสร้างแบบจำลอง Multi-layer Perceptron (MLP) สำหรับงานการจำแนกข้อมูล (classification) ใน Machine Learning. MLP เป็นประเภทหนึ่งของ Artificial Neural Network (ANN) ที่มีชั้นการเรียนรู้หลายชั้น (hidden layers) ระหว่างชั้นข้อมูลนำเข้า (input layer) และชั้นข้อมูลเอาต์พุต (output layer) แต่ละชั้นมีโหนด (neurons) หลายๆ โหนดและมีการเชื่อมต่อกันหลายๆ ชั้น ทำให้ MLP มีความซับซ้อนและสามารถจำแนกข้อมูลที่มีความซับซ้อนได้.

2.2.6 5-fold cross-validation

5-fold cross-validation เป็นวิธีหนึ่งในการทดสอบและประเมินประสิทธิภาพของแบบจำลองในงาน การเรียนรู้ของเครื่องหรือ วิทยาศาสตร์ข้อมูลโดยใช้ชุดข้อมูลที่มีอยู่ โดยการแบ่งชุดข้อมูลออกเป็น 5 ส่วนเท่า ๆ กัน หรือ 5 กลุ่ม ที่ถูกสร้างขึ้นจากชุดข้อมูลหลัก โดยแต่ละกลุ่มจะถูกใช้เป็นชุดทดสอบในรอบหนึ่งและข้อมูลที่เหลือในชุดข้อมูลหลักจะถูกใช้เป็นชุดฝึกฝนในรอบนั้นด้วย กระบวนการนี้จะทำซ้ำไปเรื่อย ๆ จนครบ 5 รอบ เพื่อให้ทุกกลุ่มมีโอกาสเป็นชุดทดสอบ ที่ เป็น ไป ได้ โดย ทั้งหมด จะ ถูก ใช้ เพื่อ ประเมิน และ ทดสอบ ความ สามารถ และประสิทธิภาพของแบบจำลองที่สร้างขึ้น

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 การศึกษาเปรียบเทียบอัลกอริทึมการจดจำใบหน้า (Paul & Acharya, 2020)

การศึกษาเปรียบเทียบอัลกอริทึมการจดจำใบหน้า เพื่อหาวิธีการระบุใบหน้าที่แตกต่างกันและเปรียบเทียบความถูกต้องในการระบุของพวกเขา ใบหน้าถูกตรวจพบผ่านอัลกอริทึม Haar Cascades ซึ่งถูกบันทึกลงในฐานข้อมูล หลังจากนั้นการศึกษาเป้าหมายที่จะเปรียบเทียบความถูกต้องในการระบุใบหน้าของอัลกอริทึมที่รู้จักอย่างดี Eigen faces ร่วมกับ PCA, SVM, KNN, และ CNN ผลลัพธ์แสดงให้เห็นว่าจากอัลกอริทึม 4 ตัวที่เราใช้ CNN ให้ความถูกต้องสูงสุด

2.3.2 การเปรียบเทียบอัลกอริธึมการจดจำใบหน้า (Delbiaggio, 2017)

การเปรียบเทียบอัลกอริธึมการจดจำใบหน้า เพื่อการสำรวจอัลกอริทึมการระบุใบหน้าหลายตัวและทำการเปรียบเทียบเรื่องความถูกต้องของพวกเขา อัลกอริทึมการระบุใบหน้าที่เปรียบเทียบกันได้รับการนำไปใช้งานอย่างกว้างขวางโดยอุตสาหกรรม โดยในฐานะของอัลกอริทึมได้รวมถึง Eigenfaces, Fisher-faces, Local Binary Pattern Histogram และ อัลกอริทึมของเครือข่ายประสาทคอนโวลูชันแนลลัล OpenFace

2.3.3 ระบบ จดจำ ใบหน้า เชิง ลึก โดย ใช้ อัล กอ ริธึ มอัจฉ ริ ยะ ทาง คอมพิวเตอร์ (Salama AbdELminaam, Almansori, Taha, & Badr, 2020)

ระบบจดจำใบหน้าเชิงลึกโดยใช้อัลกอริธึมอัจฉริยะทางคอมพิวเตอร์ เพื่อการพัฒนาระบบการรู้จำใบหน้า (Facial Recognition) ที่ใช้การโอนความรู้ (Transfer Learning) ในระบบ Fog Computing และ Cloud Computing ที่มีความสำคัญในเมืองสมาร์ท ในปัจจุบันระบบนี้ใช้เครือข่ายปรับสัญญาณลึก (Deep Convolutional Neural Networks - DCNN) เนื่องจาก DCNN เป็นวิธีการที่ครอบคลุมและมีประสิทธิภาพสูงในการดึงคุณสมบัติใบหน้าที่สำคัญ เป็นเหตุผลหลักที่ DCNN ถูกใช้ในระบบนี้ เหมือนกัน ยังมีเงื่อนไขต่าง ๆ เช่น การปิดบัง (occlusions) การแสดงอารมณ์ (expressions) แสงสว่าง (illuminations) และการเปลี่ยนท่าตัว (pose) ซึ่งอาจมีผลกับประสิทธิภาพของการรู้จำใบหน้าลึก ระบบสามารถฝึกให้รู้จำชุดคนและเรียนรู้ผ่านวิธีออนไลน์ โดยรวมผู้คนใหม่ที่ระบบนี้มีการประมวลผลและปรับปรุงการทำนายบนผู้คนที่ระบบรู้จำไว้แล้ว วิธีการการรู้จำที่เสนอนี้ได้รับการทดสอบด้วยอัลกอริทึมเรียนเครื่องที่มาตรฐานสามแบบ (Decision Tree (DT), K Nearest Neighbor (KNN), Support Vector Machine (SVM)) ระบบที่เสนอได้รับการประเมินโดยใช้ชุดข้อมูลใบหน้าสามชุด (SDUMLA-HMT, 113, และ CASIA) โดยใช้ตัวชี้วัดประสิทธิภาพ เช่น ความแม่นยำ (accuracy), ความแม่นยำ (precision), ความไวในการตอบสนอง (sensitivity), ความไวในการตอบสนองเฉพาะ (specificity), และเวลา ผลการทดสอบแสดงให้เห็นว่าวิธีการที่เสนอมีประสิทธิภาพมากกว่าอัลกอริทึมอื่น ๆ ตามทุกพารามิเตอร์ วิธีการที่เสนอนี้ให้ผลการรู้จำที่มีความแม่นยำสูง (99.06%) ความแม่นยำสูง (99.12%) ความสามารถในการระลึกสูง (99.07%) และความไวในการตอบสนองเฉพาะสูง (99.10%) กว่าอัลกอริทึมที่เปรียบเทียบ

2.3.4 การวิเคราะห์ภาพสำหรับการจดจำใบหน้า (Lu, 2003)

การวิเคราะห์ภาพสำหรับการจดจำใบหน้า เพื่อการวิเคราะห์พื้นที่ในรูปแบบพื้นที่เชิงเส้นสามวิธี และรายละเอียดของหลายวิธีการวิเคราะห์โฮลด์อินเนียร์ที่ไม่เชิงเส้นสำหรับการรู้จำใบหน้าถูกกล่าวถึงอย่างสั้น ๆ วิธีการตามรูปแบบถูกนำเสนอ รวมถึง Elastic Bunch Graph matching, Active Appearance Model และ 3D Morphable Model และกล่าวถึงฐานข้อมูลใบหน้าหลายรายการที่มีให้ในสาธารณะและผลการประเมินประสิทธิภาพที่เผยแพร่

2.3.5 การจดจำใบหน้าโดยใช้อัลกอริธึม IPCA-ICA (Dagher & Nachar, 2006)

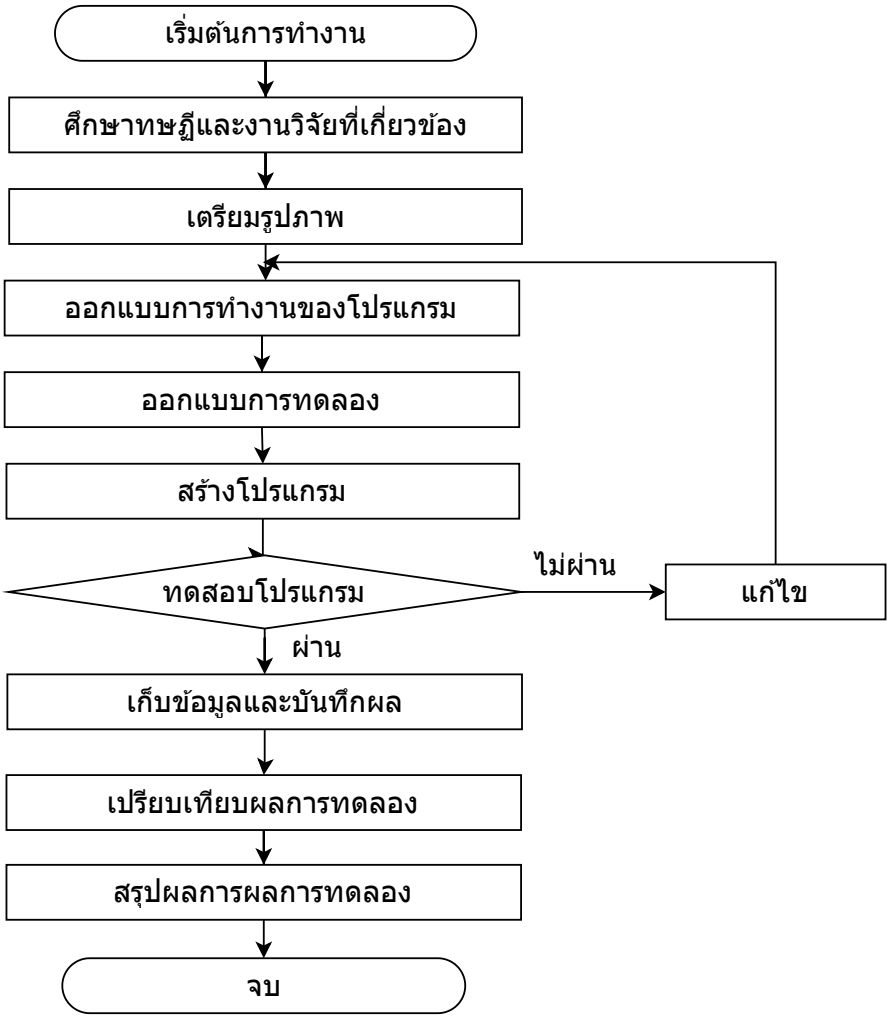
การจดจำใบหน้าโดยใช้อัลกอริธึม IPCA-ICA เพื่อการแนะนำอัลกอริธึมการวิเคราะห์ทิศทางที่ไม่ใช่เกาส์หลักที่เพิ่มขึ้นอย่างรวดเร็วที่เรียกว่า IPCA-ICA อัลกอริธึมนี้คำนวณส่วนประกอบหลักของลำดับเวกเตอร์ภาพทีละน้อยโดยไม่ต้องประเมินเมทริกซ์ความแปรปรวนร่วม (ปราศจากความแปรปรวนร่วม) และในขณะเดียวกันก็เปลี่ยนส่วนประกอบหลักเหล่านี้เป็นทิศทางอิสระที่เพิ่มความไม่ใช่เกาส์ของแหล่งที่มา เทคนิคหลักสองอย่างถูกนำมาใช้ตามลำดับแบบเรียลไทม์เพื่อให้ได้ส่วนประกอบที่มีประสิทธิภาพและเป็นอิสระที่สุดซึ่งอธิบายฐานข้อมูลใบหน้ามนุษย์ทั้งชุด ขั้นตอนนี้ทำได้โดยการรวมการรันของอัลกอริธึมสองตัวตามการวิเคราะห์ส่วนประกอบหลัก (PCA) และการวิเคราะห์ส่วนประกอบอิสระ (ICA) ที่ทำงานตามลำดับ อัลกอริทึมนี้ใช้กับปัญหาการจดจำใบหน้า ผลการจำลองบนฐานข้อมูลที่แตกต่างกันแสดงอัตราความสำเร็จเฉลี่ยสูงของอัลกอริทึมนี้เมื่อเทียบกับคนอื่น ๆ

บทที่ 3

วิธีดำเนินการวิจัย

การดำเนินงานวิจัยนี้ผู้วิจัยได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องที่มีความจำเป็นต่องานวิจัย โดยนำแนวคิด และปัญหามาทำการทดลองการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า โดยเริ่มต้นจากการศึกษาหาอัลกอริทึม ที่เหมาะสมกับการสร้างโมเดลการรู้จำใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า ผู้วิจัยได้นำฟังก์ชัน ระบุตำแหน่งใบหน้า และ การเข้ารหัสใบหน้า จากไลบรารี การจดจำใบหน้ามาใช้สำหรับการเข้ารหัสใบหน้าของรูปภาพ แล้วจึงทำการสร้างโมเดลจากอัลกอริทึม SVM,KNN และ MLP เพื่อฝึกและทดสอบหาความแม่นยำของแต่ละโมเดล โดยใช้เทคนิค 5-fold cross-validation

ในการดำเนินงานวิจัยนี้ ผู้วิจัยได้สร้างโปรแกรมสำหรับการทำงาน เพื่อให้ได้ประสิทธิภาพ สูงสุดในการดำเนินงานวิจัย ซึ่งผู้วิจัยได้แบ่งขั้นตอน เพื่อศึกษาหาวิธีการที่เหมาะสมกับการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า โดยมีขั้นตอน แสดงในแผนผังการดำเนินงานวิจัยดังภาพที่ 3.1



ภาพที่ 3.1 แผนผังการดำเนินงานวิจัยการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า

3.1 เครื่องมือที่ใช้ในงานวิจัย

การพัฒนาระบบเพื่อการเปรียบเทียบผลการประเมินอัลกอริทึมการจดจำใบหน้า อาศัยเครื่องมือและซอฟต์แวร์ดังต่อไปนี้

3.1.1 ซอฟต์แวร์ และเครื่องมือทางสถิติ

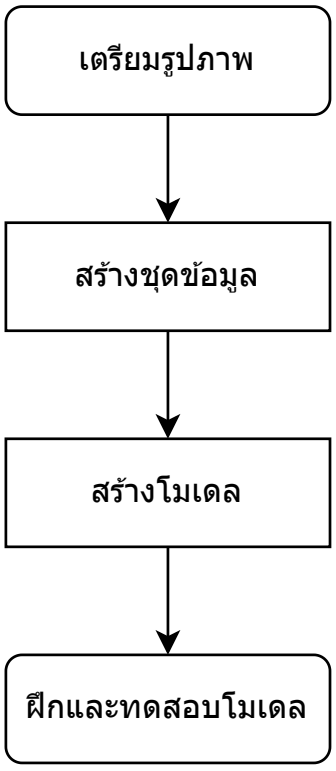
- Python ใช้เป็นภาษาโปรแกรมในการพัฒนา
- jupyter notebook ใช้สำหรับเป็นเครื่องมือ IDE ที่ช่วยเขียนภาษา Python

3.1.2 ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์

3.2 ขั้นตอนการทำงานของระบบ

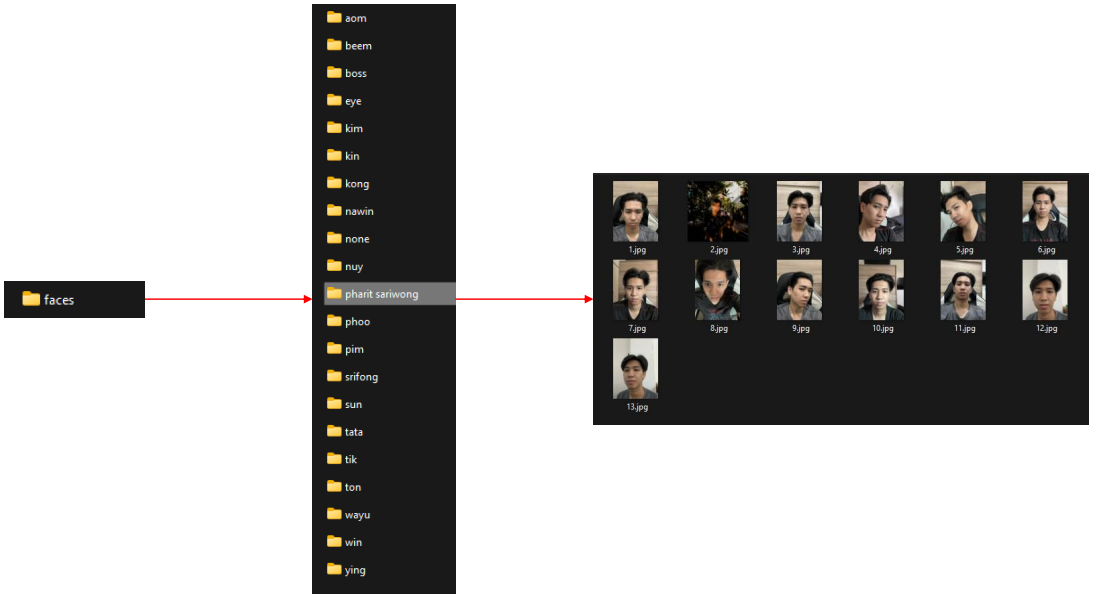
ขั้นตอนการทำงานของระบบการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า จากการที่ผู้วิจัยได้วางแผนการดำเนินงานผู้วิจัยได้ออกแบบภาพรวมของการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า โดยมีขั้นตอนในการทำงานดังภาพที่ 3.2



ภาพที่ 3.2 ขั้นตอนการทำงาน

3.2.1 เตรียมรูปภาพ

ในขั้นตอนเตรียมรูปภาพนี้ ผู้วิจัยได้ทำการรวบรวมภาพจากการถ่ายด้วยมือถือมาใช้ในงานวิจัยนี้ โดยมีจำนวนทั้งหมด 260 ภาพ แบ่งเป็น 20 คน แต่ละคนมีจำนวนภาพอยู่ที่ 13 ภาพ ดังภาพที่ 3.3



ภาพที่ 3.3 ตัวอย่างการเก็บภาพ

3.2.2 สร้างชุดข้อมูล

ในขั้นตอนสร้างชุดข้อมูลนี้ ผู้วิจัยได้นำรูปภาพจากการเตรียมรูปภาพทำการสร้างชุดข้อมูลด้วยฟังก์ชันตำแหน่งใบหน้า (อ่านรายละเอียดเพิ่มเติมที่หัวข้อ 2.2.2.1) และฟังก์ชันการเข้ารหัสใบหน้า (อ่านรายละเอียดเพิ่มเติมที่หัวข้อ 2.2.2.2) โดยใช้ฟังก์ชันตำแหน่งใบหน้า เพื่อตรวจจับใบหน้าในรูปภาพและใช้ฟังก์ชันการเข้ารหัสใบหน้า เพื่อทำการสร้างเวกเตอร์การเข้ารหัสใบหน้า 128 มิติแล้วบันทึกพร้อมทั้งเก็บชื่อบุคคลที่เป็นเจ้าของใบหน้า ตัวอย่างการเก็บข้อมูลเข้ารหัสใบหน้าดังภาพที่ 3.4

3.2.3 สร้างโมเดล

ในขั้นตอนสร้างโมเดลนี้ผู้วิจัยได้ทดสอบสร้างโมเดลทั้งหมด 3 โมเดลดังต่อไปนี้

1. Support Vector Machine (SVM) ผู้วิจัยได้ทำการสร้างโมเดล Support Vector Machine โดยใช้ฟังก์ชัน svm.SVC (อ่านรายละเอียดเพิ่มเติมที่หัวข้อ 2.2.5.2) ของไลบรารี scikit-learn โดยได้มีการกำหนดค่าพารามิเตอร์ kernel='linear' และ C=1.0

2. K-Nearest Neighbors (KNN) ผู้วิจัยได้ทำการสร้างโมเดล K-Nearest Neighbors โดยใช้ฟังก์ชัน KNeighborsClassifier (อ่านรายละเอียดเพิ่มเติมที่หัวข้อ 2.2.5.3) ของไลบรารี scikit-learn โดยได้มีการกำหนดค่าพารามิเตอร์ n_neighbors=13

3. Multilayer Perceptron (MLP) ผู้วิจัยได้ทำการสร้างโมเดล Multilayer Perceptron โดยใช้ฟังก์ชัน MLPClassifier (อ่านรายละเอียดเพิ่มเติมที่หัวข้อ 2.2.5.4) ของไลบรารี scikit-learn โดยได้มีการกำหนดค่าพารามิเตอร์ hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42

3.2.4 ฝึกและทดสอบโมเดล

ในขั้นตอนการฝึกและทดสอบโมเดลนี้ผู้วิจัยได้ฝึกและทดสอบด้วยเทคนิค 5-fold cross-validation โดยนำข้อมูลมาฝึกและทดสอบทั้ง 3 โมเดลด้วยวิธีการฝึกและทดสอบแบบเดียวกันคือการใช้ฟังก์ชัน Cross_val_score (อ่านรายละเอียดเพิ่มเติมที่หัวข้อ 2.2.5.1) ของไลบรารี scikit-learn

```
[ [-0.088714    0.17180818  0.07094651 -0.03868919  0.01406526 -0.02777415
 -0.08564029 -0.10269818  0.20594722 -0.15652768  0.254814    0.05048282
 -0.17949045 -0.12382928  0.07902551  0.13048005 -0.21261092 -0.06303444
 -0.13513914 -0.02934021  0.02601832  0.00633642  0.13425605  0.03451255
 -0.10190764 -0.36423033 -0.06068733 -0.1434326   0.03752499 -0.14276057
 -0.10428689 -0.0556677  -0.21450576 -0.10956226 -0.06157243 -0.00782351
 -0.03623524  0.01484457  0.21370912  0.03293862 -0.18010974  0.09640578
 -0.06887515  0.22709164  0.25897136  0.10862653  0.00564782 -0.08197976
  0.15237902 -0.22208419  0.05920453  0.14849894  0.06037315  0.02662973
  0.07893033 -0.16153625 -0.01288263  0.05729698 -0.10602481  0.02799062
  0.08666776 -0.06801182  0.00594226  0.01129408  0.21076991  0.05626627
 -0.112939   -0.06341899  0.07561962 -0.05181171  0.00411406  0.03028627
 -0.14764644 -0.18768038 -0.27301872  0.04665429  0.34939468  0.14406815
 -0.19340928 -0.00279953 -0.19435737 -0.00293572  0.05308245  0.06404451
 -0.04830956 -0.03750512 -0.07698688  0.0311969   0.11772831 -0.00609136
 -0.02398854  0.2218208  -0.02200313  0.06854195 -0.01913123  0.01059941
 -0.17011334 -0.01635078 -0.14087246 -0.01931228  0.00501063 -0.03285195
 -0.02569978  0.10275601 -0.21256594  0.16458216  0.00128366 -0.00047683
  0.0171465   0.08221824 -0.08451498 -0.053265    0.07350663 -0.21632168
  0.24190475  0.24544537  0.02563318  0.15428735  0.08558679  0.07140108
 -0.05066324  0.07377335 -0.18427491 -0.10029154  0.06408376  0.05941249
  0.11187739  0.01874191] ,
[-0.07546042  0.10884102  0.01820965 -0.06256932  0.01238308  0.00108031
 -0.0869963  -0.08658794  0.19445807 -0.09988614  0.24618471  0.08836973
 -0.21514088 -0.14409998  0.04567104  0.13575552 -0.18451059 -0.07171042
 -0.10766366 -0.12154544  0.03059869 -0.0041246   0.11233218  0.03440855
 -0.12648645 -0.35149547 -0.06411266 -0.17257744  0.0027295  -0.10655167
 -0.08780177  0.01092672 -0.18007073 -0.11318414  0.0223209  -0.02252459
  0.01075476 -0.00193357  0.19514172  0.03188299 -0.11572585  0.1025712
  0.01862897  0.2160418   0.28378522  0.08486986  0.00403396 -0.09580179
  0.10772225 -0.2273424   0.0705988   0.14081752  0.07180372  0.03235561
  0.09678373 -0.19317414 -0.00088086  0.09939131 -0.13803799  0.01948881
  0.00653491 -0.06652246 -0.04319426  0.04047258  0.1997477   0.10664738
 -0.12269793 -0.05064882  0.12488936 -0.02238242  0.03014256  0.02169436
 -0.19150327 -0.20767984 -0.22891052  0.08110818  0.37884849  0.18467768
 -0.21384551  0.01372666 -0.18862456  0.035554    0.08151729  0.01256399
 -0.07114346 -0.13153212 -0.04020823  0.04839402  0.08507413  0.03427088
 -0.04834593  0.22429292 -0.02123625  0.06345314  0.01217165  0.05531278
 -0.1475258  -0.02315399 -0.17892462 -0.07756063  0.01705166 -0.03153926
  0.02891058  0.12965578 -0.22984637  0.06226313  0.01275493 -0.03555039
  0.02730239  0.07433654 -0.02594047 -0.03417763  0.06341785 -0.2379019
  0.24267898  0.2552098   0.0329073   0.17940402  0.07169842  0.03061812
 -0.0125663  -0.03109422 -0.16603501 -0.05279541  0.05403206  0.06724691
  0.0795987   0.00267015]
 ,...,
[-0.14411928  0.09458829  0.13313006 -0.1392405  -0.09781284 -0.00103198
 -0.05000006 -0.09765496  0.24595229 -0.12307285  0.23012051  0.00111175
 -0.20965301 -0.02456653 -0.07008316  0.18956485 -0.19806431 -0.07467343
 -0.04133435  0.00813394  0.13566573  0.06126674  0.09221797  0.09713555
 -0.11071274 -0.31878656 -0.05114327 -0.10671535 -0.09687568 -0.18335156
  0.05998817  0.06451394 -0.15463494 -0.00631319 -0.07899575  0.08842622
 -0.07298511 -0.15771748  0.14134026  0.05146674 -0.23750761  0.01146818
  0.00070768  0.22127227  0.22343794  0.00175962  0.06348499 -0.1177499
  0.12133771 -0.36427152 -0.07287825  0.10433811 -0.06383633  0.04363741
  0.04165027 -0.20685937  0.10612108  0.04156399 -0.27018544  0.06045081
  0.10124242 -0.15201347  0.03975698  0.02482021  0.22097833  0.1842965
 -0.10080785 -0.18980883  0.10810453 -0.16462916 -0.04534761  0.15249154
 -0.13989796 -0.23980023 -0.22340946 -0.02166521  0.34440309  0.13149667
 -0.0572816   0.05452351 -0.1750547   0.04552171 -0.03271984  0.1271624
  0.00860763  0.02258067 -0.13043576  0.02647449  0.18058133 -0.0366239
 -0.00610749  0.36915681  0.02487309 -0.05129103  0.01946596  0.0607686
 -0.13251895  0.08742316 -0.08063795 -0.00191762 -0.00815073 -0.07969596
  0.05485324  0.04562411 -0.23052792  0.1777593  -0.12722747 -0.01953356
 -0.0422897  -0.10495269 -0.0900012   0.05131699  0.16358545 -0.27341849
  0.19630294  0.20804438  0.13041268  0.18598427  0.07378151  0.10058322
 -0.02738298 -0.02864192 -0.11622647  0.01855359  0.00800369 -0.0080822
  0.0395123   0.03963761] ]

[ FORD , FORD ,..., WIN ]
```

ภาพที่ 3.4 ตัวอย่างการเก็บข้อมูลเข้ารหัสใบหน้า

บทที่ 4

ผลการดำเนินงานวิจัย

จากการศึกษาและทดสอบประสิทธิภาพหาค่าความแม่นยำของทั้ง 3 โมเดลคือโมเดล SVM,KNN และ MLP ในการจดจำใบหน้า จะกล่าวถึงผลการทดลองและบันทึกผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของโมเดล โดยใช้การทดสอบแบบ 5-fold Cross Validation มาใช้ในการฝึกโมเดลและทดสอบโมเดลและเปรียบเทียบประสิทธิภาพ เพื่อหาโมเดลที่ดีที่สุดในการใช้งาน ผลการทดลองจะแบ่งเป็นสามส่วนตามแต่ละโมเดล

4.1 ผลการทดลองของแต่ละโมเดล

4.1.1 ผลการทดลองของโมเดล SVM

การทดสอบโมเดล SVM ได้ผลการทดสอบดังต่อไปนี้ กลุ่มที่ 1 ได้ค่าความแม่นยำ 98.08%, กลุ่มที่ 2 ได้ค่าความแม่นยำ100.00%, กลุ่มที่ 3 ได้ค่าความแม่นยำ 98.08%, กลุ่มที่ 4 ได้ค่าความแม่นยำ 96.15% และ กลุ่มที่ 5 ได้ค่าความแม่นยำ 96.15% ดังภาพที่ 4.1

4.1.2 ผลการทดลองของโมเดล KNN

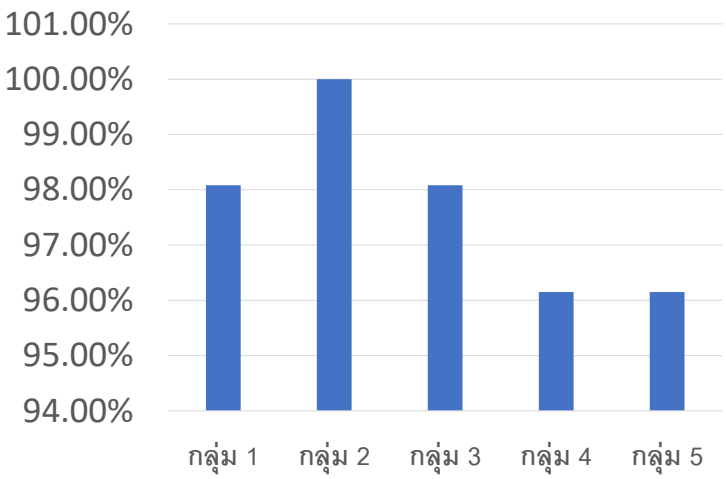
การทดสอบโมเดล KNN ได้ผลการทดสอบดังต่อไปนี้ กลุ่มที่ 1 ได้ค่าความแม่นยำ 98.08%, กลุ่มที่ 2 ได้ค่าความแม่นยำ 100.00%, กลุ่มที่ 3 ได้ค่าความแม่นยำ 100.00%, กลุ่มที่ 4 ได้ค่าความแม่นยำ 100.00% และ กลุ่มที่ 5 ได้ค่าความแม่นยำ 98.08% ดังภาพที่ 4.2

4.1.3 ผลการทดลองของโมเดล MLP

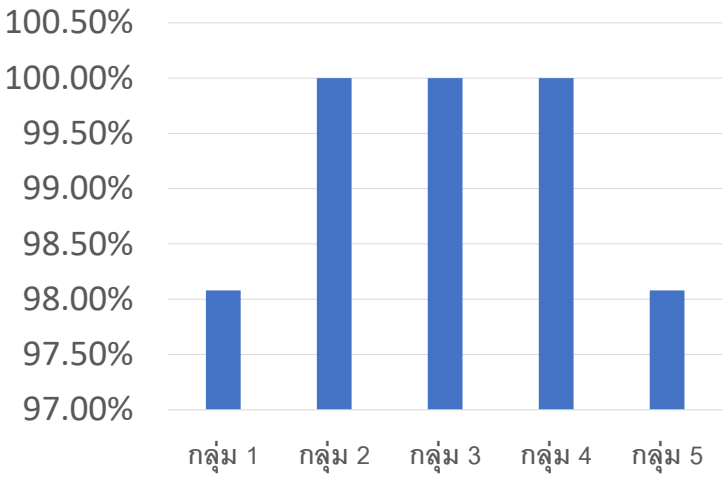
การทดสอบโมเดล MLP ได้ผลการทดสอบดังต่อไปนี้ กลุ่มที่ 1 ได้ค่าความแม่นยำ 100.00%, กลุ่มที่ 2 ได้ค่าความแม่นยำ 100.00%, กลุ่มที่ 3 ได้ค่าความแม่นยำ 100.00%, กลุ่มที่ 4 ได้ค่าความแม่นยำ 100.00% และ กลุ่มที่ 5 ได้ค่าความแม่นยำ 98.08% ดังภาพที่ 4.3

4.2 สรุปผลการทดสอบโมเดล

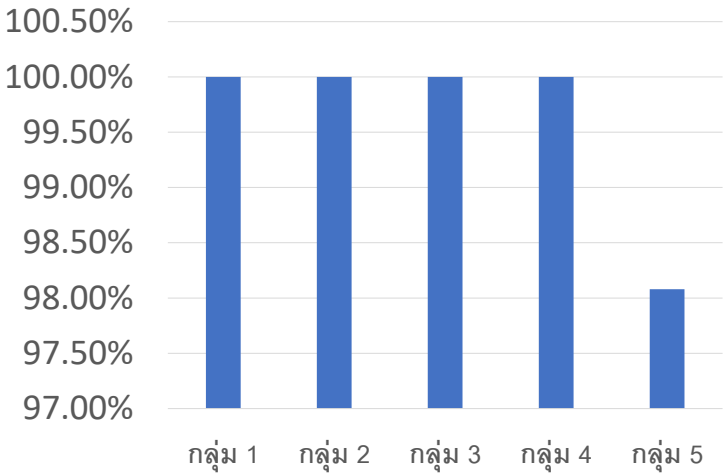
จากผลการทดสอบโมเดลที่สร้างจากอัลกอริทึม SVM,KNNและMLP ได้ค่าเฉลี่ยของแต่ละโมเดลอยู่ที่ 97.69%, 99.23%และ 99.62% ตามลำดับดังภาพที่ 4.4



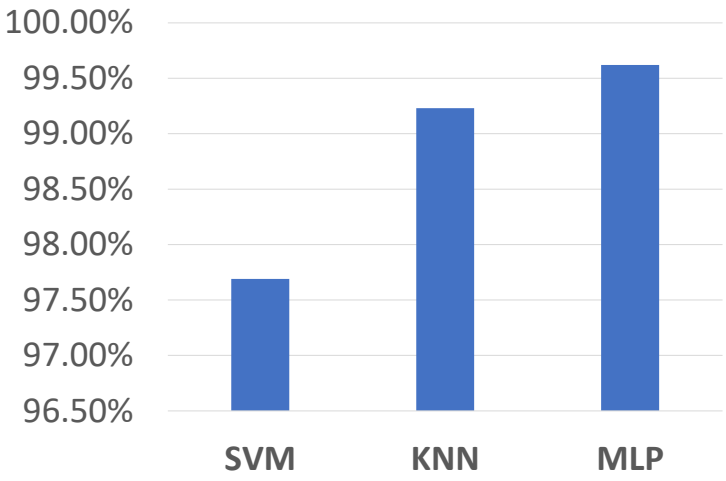
ภาพที่ 4.1 ผลการทดสอบโมเดล SVM



ภาพที่ 4.2 ผลการทดสอบโมเดล KNN



ภาพที่ 4.3 ผลการทดสอบโมเดล MLP



ภาพที่ 4.4 ผลการทดสอบเฉลี่ยของแต่ละโมเดล

บทที่ 5

สรุปและอภิปรายผล

5.1 สรุปผลและอภิปรายผล

จากผลการทดลองเปรียบเทียบอัลกอริทึมโดยอาศัยข้อมูลการเข้ารหัสใบหน้า ผู้วิจัยได้เตรียมภาพไว้จำนวน 260 ภาพ เพื่อทำการเข้ารหัสใบหน้า และหลังจากเข้ารหัสใบหน้าแล้วจะได้รับข้อมูลการเข้ารหัสใบหน้า 260 ชุด x 128 มิติ สำหรับการฝึกและทดสอบโมเดล ในกระบวนการฝึกและทดสอบโมเดลนี้ ผู้วิจัยเลือกใช้อัลกอริทึมสำหรับการฝึกและทดสอบโมเดลได้แก่ SVM, KNN และ MLP สำหรับการวัดประสิทธิภาพความแม่นยำ วัดโดยการใช้เทคนิค 5-fold cross-validation ซึ่งผลเป็นดังนี้ SVM ได้ค่าความแม่นยำ 97.69% KNN ค่าความแม่นยำ 99.23% และ MLP ค่าความแม่นยำ 99.62%

จากผลการทดลองนี้ สรุปได้ว่าข้อมูลที่ได้จากการเข้ารหัสใบหน้า ด้วยฟังก์ชัน การเข้ารหัสใบหน้า ของไลบรารี การจดจำใบหน้า สามารถนำมาใช้ในการเรียนรู้ของโมเดลได้อย่างมีประสิทธิภาพ และในงานวิจัยนี้โมเดลที่มีประสิทธิภาพมากที่สุดในการจดจำใบหน้า จากข้อมูลการเข้ารหัสใบหน้าคือ โมเดลจากอัลกอริทึม MLP

5.2 ข้อจำกัด

1. ในงานวิจัยนี้ ผู้วิจัยได้ใช้ค่าเริ่มต้นของอัลกอริทึมเพียงเท่านั้นไม่ได้ปรับค่าพารามิเตอร์ในการสร้างโมเดล
2. ในงานวิจัยนี้ ใช้ข้อมูลภาพเพียงแค่ 260 จาก 20 คนเท่านั้น
3. ในงานวิจัยนี้ ใช้เพียงแค่ 3 อันกอริทึมในการเปรียบเทียบกันเท่านั้น

5.3 ข้อเสนอแนะและงานวิจัยในอนาคต

จากการทดลองการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า มีแนวทางในการพัฒนาให้สามารถนำไปใช้ประโยชน์ได้อย่างมีประสิทธิภาพมากขึ้น โดยมีข้อเสนอแนะและงานวิจัยในอนาคตดังนี้

1. เพิ่มข้อมูลรูปภาพในการทดลองให้มากขึ้น
2. เพิ่มอัลกอริทึมในการทดลอง

บรรณานุกรม

บรรณานุกรม

Dagher, I., & Nachar, R. (2006). Face recognition using ipca-ica algorithm. *IEEE transactions on pattern analysis and machine intelligence*, *28*(6), 996–1000.

Delbiaggio, N. (2017). A comparison of facial recognition's algorithms.

Lu, X. (2003). Image analysis for face recognition. *Personal notes*, 36.

Paul, S., & Acharya, S. K. (2020). A comparative study on facial recognition algorithms.

Salama AbdELminaam, D., Almansori, A. M., Taha, M., & Badr, E. (2020). A deep facial recognition system using computational intelligent algorithms. *Plos one*, *15*(12), e0242269.

ภาคผนวก

ภาคผนวก ก

โค้ดการทำงานของโปรแกรม

โค้ดการทำงานของโปรแกรมสำหรับการจัดประเภทใบหน้าด้วยข้อมูลการเข้ารหัสใบหน้า มีรายละเอียดดังต่อไปนี้

ก.1 โค้ดการนำเข้าไลบรารีและโมดูล

```
import os
import face_recognition
from face_recognition.face_recognition_cli import image_files_in_folder
from sklearn.model_selection import cross_val_score
from sklearn import svm
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
```

ภาพที่ ก.1 โค้ดไลบรารีและโมดูล

จากภาพที่ ก.1 ผู้วิจัยได้นำเข้าไลบรารีและโมดูลที่จำเป็นสำหรับการทำงานมีดังต่อไปนี้ os ใช้ในการจัดการระบบไฟล์, face_recognition ใช้ในการตรวจหาใบหน้าและเข้ารหัสใบหน้า, face_recognition.face_recognition_cli ใช้ในการดึงรายชื่อไฟล์ภาพจากโฟลเดอร์, cross_val_score ใช้ในการทดสอบแบบจำลองด้วยวิธี cross-validation, svm ใช้ในสร้างแบบจำลอง Support Vector Machine (SVM), KNeighborsClassifier ใช้ในการสร้างแบบจำลอง K-Nearest Neighbors (KNN), MLPClassifier ใช้ในการสร้างแบบจำลอง Multi-Layer Perceptron (MLP)

ก.2 โค้ดกำหนดชนิดของไฟล์และโฟลเดอร์

```
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

train_dir = "faces"
```

ภาพที่ ก.2 โค้ดกำหนดชนิดของไฟล์และโฟลเดอร์

จากภาพที่ ก.2 ผู้วิจัยได้กำหนดชนิดของไฟล์ที่สามารถใช้ได้และกำหนดเส้นทางของโฟลเดอร์ที่ใช้เก็บข้อมูลรูปภาพ

ก.3 โค้ดการตรวจหาใบหน้าและเข้ารหัสใบหน้า

จากภาพที่ ก.3 เป็นโค้ดกระบวนการเตรียมข้อมูลที่จะถูกใช้ในการฝึกอบรมแบบจำลอง ฟังก์ชันเหล่านี้มีหน้าที่โหลดรูปภาพ,ตรวจหาใบหน้าจากภาพ และสกัดคุณลักษณะของใบหน้าจากภาพ

```python
def get_image_from_path(img_path):
    return face_recognition.load_image_file(img_path)

def get_1face_encodings(image):
    face_location = face_recognition.face_locations(image)
    if len(face_location) != 1:
        print("Image not suitable")
        return None, None
    else:
        return face_location, face_recognition.face_encodings(image, known_face_locations=face_location)[0]

def get_all_x_y(data_dir):
    X = []
    y = []

    for class_dir in os.listdir(data_dir):
        if not os.path.isdir(os.path.join(data_dir, class_dir)):
            continue

        for img_path in image_files_in_folder(os.path.join(data_dir, class_dir)):
            image = get_image_from_path(img_path)
            face_lo, face_en = get_1face_encodings(image)

            if face_lo:
                X.append(face_en)
                y.append(class_dir)

    return X, y

X, y = get_all_x_y(train_dir)
```

ภาพที่ ก.3 โค้ดการตรวจหาใบหน้าและเข้ารหัสใบหน้า

ก.4 โค้ดการสร้างฝึกและทดสอบของโมเดล SVM

จากภาพที่ ก.4 เป็นโค้ดกระบวนการสร้างโมเดล SVM และทำการฝึกและทดสอบแบบ5-fold cross-validation

ก.5 โค้ดการสร้างฝึกและทดสอบของโมเดล KNN

จากภาพที่ ก.5 เป็นโค้ดกระบวนการสร้างโมเดล KNN และทำการฝึกและทดสอบแบบ5-fold cross-validation

```python
def train_and_evaluate_svm(X, y):
    #สร้างโมเดล svm
    svm_classifier = svm.SVC(kernel='linear', C=1.0)

    #ฝึกและประเมินแบบจำลอง
    scores_svm = cross_val_score(svm_classifier, X, y, cv=5)
    return scores_svm

scores_svm = train_and_evaluate_svm(X, y)
for i, score in enumerate(scores_svm):
    print(f'Fold {i+1}: Accuracy = {score * 100:.2f}%')

average_accuracy_svm = scores_svm.mean()
print(f'Average Accuracy: {average_accuracy_svm * 100:.2f}%')
```

ภาพที่ ก.4 โค้ดการสร้างฝึกและทดสอบของโมเดล SVM

```python
def train_and_evaluate_knn(X, y):
    #สร้างโมเดล knn
    knn_model = KNeighborsClassifier(n_neighbors=13)

    #ฝึกและประเมินแบบจำลอง
    scores_knn = cross_val_score(knn_model, X, y, cv=5)
    return scores_knn

scores_knn = train_and_evaluate_knn(X, y)
for i, score in enumerate(scores_knn):
    print(f'Fold {i+1}: Accuracy = {score * 100:.2f}%')

average_accuracy_knn = scores_knn.mean()
print(f'Average Accuracy: {average_accuracy_knn * 100:.2f}%')
```

ภาพที่ ก.5 โค้ดการสร้างฝึกและทดสอบของโมเดล KNN

ก.6 โค้ดการสร้างฝึกและทดสอบของโมเดล MLP

จากภาพที่ ก.6 เป็นโค้ดกระบวนการสร้างโมเดล MLP และทำการฝึกและทดสอบแบบ5-fold cross-validation

```python
def train_and_evaluate_mlp(X, y):
    #สร้างโมเดล mlp
    mlp = MLPClassifier(hidden_layer_sizes=(200, 100), max_iter=1000, random_state=42)

    #ฝึกและประเมินแบบจำลอง
    scores_mlp = cross_val_score(mlp, X, y, cv=5)
    return scores_mlp

scores_mlp = train_and_evaluate_mlp(X, y)
for i, score in enumerate(scores_mlp):
    print(f'Fold {i+1}: Accuracy = {score * 100:.2f}%')

average_accuracy_mlp = scores_mlp.mean()
print(f'Average Accuracy: {average_accuracy_mlp * 100:.2f}%')
```

ภาพที่ ก.6 โค้ดการสร้างฝึกและทดสอบของโมเดล MLP

ประวัติย่อของผู้วิจัย

ประวัติย่อของผู้วิจัย

ชื่อ สกุล	นาย พฤทธิ์ สาริวงค์
วัน เดือน ปีเกิด	23 มกราคม 2543
สถานที่เกิด	22 หมู่ 17 ตำบลไม้ยา อำเภอพญาเม็งราย จังหวัดเชียงราย
เบอร์โทร	094-7467-403
ประวัติการศึกษา	
2561	ระดับ มัธยมศึกษา จบ การ ศึกษา จาก โรงเรียน เด็ก ดี พิทยาคม อำเภอเทิง จังหวัดเชียงราย
2555	ระดับ ประถม ศึกษา จบ การ ศึกษา จาก โรงเรียน เด็ก ดี พิทยาคม อำเภอเทิง จังหวัดเชียงราย