ระบบการรู้จำใบหน้าโดยใช้กระบวนการถ่ายโอนการเรียนรู้

Facial recognition using transfer learning

นายธันยาธรณ์ บินตอเล็บ

Tanyatorn Bintoleb

นางสาวพนัชกร วรโชติชนาภัทร

Panatchakorn Worachotchanapat

นางสาวอาทิตยา จันทมี

Athitaya Chanthami

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2563

คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงงาน ระบบรู้จำใบหน้าโดยใช้กระบวนการถ่ายโอนการเรียนรู้
Facial recognition using transfer learning

นิสิต	นายธันยาธรณ์	บินตอเล็บ	60102010566
	นางสาวพนัชกร	วรโชติชนาภัทร	60102010571
	นางสาวอาทิตยา	จันทมี	60102010587

ปริญญา วิทยาศาสตร์บัณฑิต (วท.บ.)

ภาควิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงงาน อ.ดร. วีรยุทธ เจริญเรืองกิจ

ลงชื่อ.....

(อ.ดร. วีรยุทธ เจริญเรืองกิจ)

อาจารย์ที่ปรึกษาโครงงาน

บทคัดย่อ

งานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาระบบการรู้จำใบหน้าโดยจะทำการศึกษา Pre-trained Model สำหรับการทำระบบรู้จำใบหน้า และเปรียบเทียบค่าความถูกต้องของโมเดลสำหรับการทำการรู้จำใบหน้า ซึ่งใช้ในการระบุตัวตนของคนที่เข้ามาในระบบ โดยระบบมีการตรวจจับใบหน้า (Face Detection) เพื่อตรวจจับเฉพาะใบหน้า โดยการทำงานของระบบคือ เมื่อมีภาพหรือบุคคลเข้ามาในระบบ ระบบจะทำการตรวจจับหาใบหน้าบุคคล หลังจากนั้นจะทำการตีกรอบเฉพาะใบหน้า และทำการรู้จำใบหน้า (Face Recognition) ตามลำดับ ค่าความถูกต้อง ของระบบขึ้นอยู่กับโมเดลที่ใช้ซึ่งควรผ่านการเทรนกับชุดข้อมูลขนาดใหญ่เพื่อให้ได้ประสิทธิภาพที่ดีสำหรับการนำใช้สำหรับศึกษาต่อกับชุดข้อมูลและระบบรู้จำใบหน้าของงานวิจัยนี้ เพื่อลดระยะเวลาในการทำงานของระบบ ซึ่งโมเดลที่ใช้ในการทำวิจัยนี้เป็น Pre-trained Model 3 ตัว ได้แก่ VGG16, MobileNet, ResNet50 เนื่องจากตัวโมเดลเหล่านี้สามารถให้ประสิทธิภาพ และค่าความถูกต้องที่ค่อนข้างสูงถึงสูงมาก จึงเลือกใช้โมเดลเหล่านี้มาใช้ในการเปรียบเทียบเพื่อเลือกโมเดลที่เหมาะสมกับชุดข้อมูลและระบบรู้จำใบหน้า งานวิจัยนี้ใช้ภาพทั้งหมด 2,923 ภาพ 106 บุคคล โดยใช้ชุดข้อมูลของ Labelled Faces in the Wild (LFW) จำนวน 2,292 ภาพทั้งหมด 57 บุคคล และชุดข้อมูลของนิสิตวิทยาการคอมพิวเตอร์ จำนวน 631 ภาพ 49 บุคคลโดยแบ่งเป็นจำนวนภาพในการ train ทั้งหมด 2,045 ภาพ และจำนวนภาพในการ test ทั้งหมด 878 ภาพ สามารถรู้จำใบหน้าได้ค่าความถูกต้อง 95 %

Abstract

This research aims to develop a face recognition system from pre-trained models and compare the accuracy of models in the face recognition application. The system is used to identify people entering the system. The system operates on an image or person entering the system. The system detects the face of a person and creates frames around the face before sending it to face recognition. The accuracy of the system depends on the model, which should be trained with large datasets to achieve good performance for our dataset and face recognition system of this research. The models used in this research are three pre-training models: VGG16, MobileNet, ResNet50. The results from the evaluation are compared to select the best model that is suitable for the dataset and face recognition system. This research uses 2,923 images of 106 people from two datasets: 2,292 images of 57 people from the Labelled Faces in the Wild (LFW) dataset and 631 images of Computer Science 49 students. The dataset is divided as a training set of 2,045 images and as a test set of 878 images. The best recognition result obtained from the experiment achieves with 95% in accuracy.

กิตติกรรมประกาศ

โครงงาน ระบบการรู้จำใบหน้าโดยใช้กระบวนการถ่ายโอนการเรียนรู้ (Facial recognition using transfer learning) สามารถสำเร็จลุล่วงได้ด้วยความกรุณาจาก อ.ดร. วีรยุทธ เจริญเรืองกิจ อาจารย์ที่ปรึกษาโครงงาน ที่กรุณาเสียสละเวลาเพื่อคอยช่วยเหลือ ให้คำปรึกษา คำแนะนำแนวทางในการแก้ปัญหาและแก้ไขข้อบกพร่องที่เกิดขึ้นระหว่างการทำงาน คณะผู้จัดทำขอกราบพระคุณเป็นอย่างสูง

ขอขอบพระคุณอาจารย์ทุก ๆ ท่านในภาควิชาวิทยาการคอมพิวเตอร์ที่กรุณาให้ความรู้ และสามารถนำความรู้ที่ได้เรียนมาประยุกต์ใช้ในการดำเนินโครงงาน รวมทั้งให้คำปรึกษาและคำแนะนำที่เป็นประโยชน์ ทำให้คณะผู้จัดทำสามารถแก้ไขโครงงานได้สมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณมหาวิทยาลัยศรีนครินทรวิโรฒ คณะวิทยาศาสตร์ และภาควิชาวิทยาการคอมพิวเตอร์ ที่เป็นแหล่งศึกษาหาความรู้ ให้การสนับสนุนสถานที่และอุปกรณ์ต่าง ๆ เพื่อนำไปประยุกต์ใช้ในการทำงานและการดำเนินชีวิตต่อไป

ขอขอบพระคุณบิดา มารดา รวมถึงครอบครัวที่ได้ให้การช่วยเหลือและสนับสนุน ตลอดจนถึงความห่วงใยและยังเป็นกำลังใจสำคัญในการทำโครงงานเสมอมา

ขอขอบคุณเพื่อน ๆ ที่ช่วยสละเวลาให้ความร่วมมือในการเก็บรวบรวมชุดข้อมูล ตลอดจนถึงให้คำแนะนำคำปรึกษา และคอยช่วยเหลือ ทำให้โครงงานมีความสมบูรณ์มากยิ่งขึ้น

สุดท้ายขอขอบคุณผู้มีส่วนเกี่ยวข้องทุก ๆ ท่านที่ทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี และหวังว่าโครงงาน ระบบการรู้จำใบหน้าโดยใช้กระบวนการถ่ายโอนการเรียนรู้ (Facial recognition using transfer learning) จะเป็นประโยชน์ต่อผู้ที่สนใจศึกษา และสามารถเป็นส่วนหนึ่งสำหรับผู้ที่สนใจจะนำไปต่อยอดในบริบทต่าง ๆ ต่อไป

คณะผู้จัดทำ

สารบัญ

หัวข้อ **หน้า**

สารบัญ (ต่อ)

หัวข้อ **หน้า**

สารบัญ (ต่อ)

หัวข้อ **หน้า**

สารบัญรูปภาพ

หน้า

สารบัญรูปภาพ (ต่อ)

หน้า

สารบัญตาราง

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบันมีการใช้ระบบตรวจจับใบหน้ากันอย่างแพร่หลายในสถานที่ที่ต้องการความปลอดภัย ไม่ว่าจะเป็น บริษัท ร้านค้า ร้านอาหาร บ้าน ถนน หรือตามสถานีรถไฟและรถไฟฟ้าใต้ดิน ทางคณะผู้จัดทำจึงเล็งเห็นถึงความสำคัญของประสิทธิภาพและค่าความถูกต้องของโมเดลที่นำมาใช้งาน สำหรับการรู้จำใบหน้า โดยส่วนใหญ่มีเป้าหมายเพื่อปรับปรุงคุณภาพและประสิทธิภาพของการรักษาความปลอดภัยไม่ว่าจะเป็นแบบดิจิตอลหรือเทคโนโลยีสารสนเทศ

โดยทางคณะผู้ทำได้ทำการศึกษาโมเดลต่าง ๆ ที่เกี่ยวข้องสำหรับการนำมาใช้ในการระบบรู้จำใบหน้า โดยระบบตรวจจับใบหน้ามีข้อจำกัดมากมาย มีสิ่งรบกวนซึ่งเป็นสิ่งที่ท้าทายในการวิเคราะห์ระบบให้ได้ความถูกต้องแม่นยำ ข้อมูลใบหน้าส่วนใหญ่อาจจะสูญหายไปเนื่องจากสิ่งรบกวน เช่น แสงรอบข้างต่ำหรือความละเอียดของภาพต่ำเกินไป ซึ่งเป็นสิ่งที่ทำให้ยากต่อการตรวจจับใบหน้าของวัตถุที่อยู่ไกลจากกล้องออกไป

โดยทางคณะผู้จัดทำได้นำเสนอขอบเขตการทำงานสำหรับการเปรียบเทียบประสิทธิภาพของโมเดลสำหรับระบบรู้จำใบหน้า ซึ่งประกอบไปด้วย ขั้นตอนวิธีการตรวจจับใบหน้า (Face Detection) และการรู้จำใบหน้า (Face Recognition) โดยขั้นตอนวิธีการรู้จำใบหน้าที่นำเสนออยู่ภายใต้ การเปรียบเทียบโมเดล 3 โมเดลด้วยกัน คือ MobileNet, VGG16, ResNet50 ซึ่งทางคณะผู้จัดทำได้ทำการศึกษา และปรับปรุงค่าความถูกต้องรวมถึงพัฒนาประสิทธิภาพของตัวโมเดลนั้น ๆ โดยใช้วิธีการถ่ายโอนการเรียนรู้ (Transfer Learning)

1.2 วัตถุประสงค์ของโครงงาน

ในการวิจัยครั้งนี้ผู้วิจัยได้ตั้งวัตถุประสงค์ในการทำงานวิจัยไว้ดังนี้

1.2.1 เพื่อทำการศึกษา Pre-trained Model สำหรับ Face Recognition

1.2.2 เปรียบเทียบค่าความถูกต้องของโมเดลสำหรับ Face Recognition

1.2.3 สามารถระบุตัวบุคคลโดยการทำ Face recognition จากโมเดลที่ได้นำไปปรับปรุงค่าประสิทธิภาพ

1.3 ขอบเขตของโครงงาน

ในการวิจัยครั้งนี้มีขอบเขตในการทำงานวิจัยดังนี้

1.3.1 สามารถเพิ่มประสิทธิภาพของระบบรู้จำใบหน้าได้ จากการศึกษาโมเดลต่าง ๆ

1.3.2 สามารถระบุตัวตนของบุคคลที่เข้ามาใช้งานในระบบได้ จากระบบรู้จำใบหน้าที่ได้ใช้ โมเดลที่พัฒนามาแล้ว

1.3.3 พัฒนาระบบรู้จำใบหน้าเพื่อทำการตรวจจับและวิเคราะห์ใบหน้าบุคคลที่เข้ามาในระบบ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถทำการรู้จำใบหน้าได้อย่างถูกต้อง

1.4.2 เข้าใจการเลือกใช้โมเดลสำหรับ Face Recognition Application

1.4.3 ได้เพิ่มประสิทธิภาพของโมเดลสำหรับ Face Recognition

บทที่ 2
องค์ความรู้ที่เกี่ยวข้อง

2.1 องค์ความรู้เกี่ยวกับ Face Recognition

ระบบการรู้จำใบหน้าหรือ ระบบการจดจำใบหน้า (Facial Recognition System) คือระบบการตรวจหาใบหน้าของมนุษย์และการปรับภาพใบหน้าอัตโนมัติ กรอบจะปรากฏขึ้นบนใบหน้าที่ถูกตรวจจับและโฟกัส สี และค่าการวัดแสงจะถูกปรับโดยอัตโนมัติ นอกจากนั้นเมื่อบันทึกด้วยคุณภาพแบบ HD เทคโนโลยีการบีบอัดจะจัดสรรความจุของข้อมูลให้ลดลง แต่ได้ข้อมูลที่เป็นประโยชน์มากขึ้นเพื่อปรับคุณภาพของภาพ ข้อมูลที่ได้จะถูกนำไปเปรียบเทียบกับข้อมูลตัวอย่างที่บันทึกไว้ อาจจะทั้งใบหน้าหรือเพียงบางส่วน ขึ้นอยู่กับชนิดของวิธีแยกเอกลักษณ์ใบหน้า ระบบการรู้จำใบหน้าเป็นส่วนหนึ่งของเทคโนโลยีปัญญาประดิษฐ์ในส่วนเนื้อหาของเรื่อง การรับรู้ของเครื่อง (Machine perception)

2.2 องค์ความรู้เกี่ยวกับ Face Detection

ในระบบการจดจำภาพอินพุตที่เข้ามาหลายใบหน้าและพื้นหลังที่ซับซ้อนมีความยากและซับซ้อนจึงต้องมีการตรวจจับใบหน้าที่เข้ามา ซึ่งจะขึ้นอยู่ที่ใบหน้าที่เข้ามาและการคำนวณพื้นที่ที่สนใจ โดยการตรวจจับใบหน้าเป็นเป็นขั้นตอนแรกสำหรับการวิเคราะห์ใบหน้าทั้งหมดรวมถึงการจัดตำแหน่งใบหน้าในการที่จะรู้จำใบหน้า จึงต้องมีการภาพใบหน้าทั่วไปของมนุษย์เช่น ตา จมูก ปาก เป็นต้น จากนั้นกำหนดจำนวนใบหน้า ตำแหน่งและขนาดของใบหน้าทั้งหมด

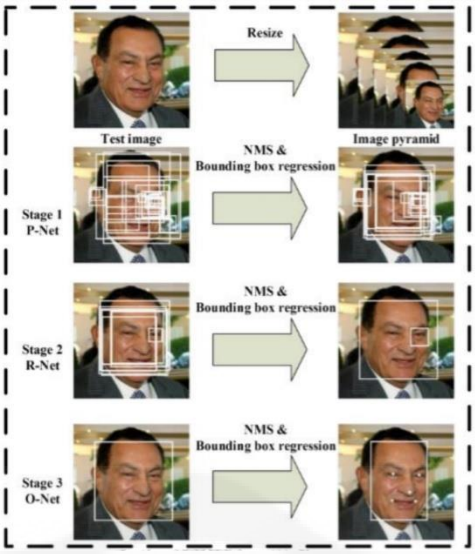
อัลกอริทึมตรวจจับใบหน้าที่เสนอโดย Viola และ Jones [1] อัลกอริทึมจะสแกนหน้าต่างย่อยที่สามารถตรวจจับใบหน้าจากภาพอินพุตที่เข้ามา การตรวจจับใบหน้าแบบ Viola-Jones มีสามแนวคิดหลักที่สามารถใช้แบบเรียลไทม์คือ การ Integral Image, classifier learning โดย AdaBoost และcascade structure. วิธีการดังกล่าวจะช่วยลดขนาดภาพให้มีขนาดแตกต่างกันจากนั้นจะตรวจจับขนาดผ่านภาพเหล่านี้อย่างคงที่ อย่างไรก็ตามอาจใช้เวลาการคำนวณนานเนื่องจากภาพมีขนาดที่แตกต่างกันออกไป

Histogram of Oriented Gradients (HOG) ใช้เพื่อตรวจจับวัตถุ computer vision และ image processing โดยมีแนวคิดพื้นฐานคือข้อมูลรูปร่างท้องถิ่นที่อธิบายโดยการกระจายของการไล่ระดับสีเข้มหรือทิศทางขอบ ถึงแม้ไม่มีข้อมูลที่แม่นยำเกี่ยวกับตำแหน่งของขอบตัวเอง อัลกอริทึมจะแบ่งภาพออกเป็นภาพย่อยขนาดเล็กเรียกว่า "เซลล์" จากนั้นรวบรวมฮิสโตแกรมของการวางขอบภายในเซลล์นั้น สุดท้ายคือรวม histogram และใช้ feature vector อธิบายวัตถุนั้น อัลกอริทึมยังนับการวางแนวไล่ระดับสีของภาพ นอกจากนี้ตัวอธิบาย HOG จะใช้ Support Vector Machine (SVM) เป็นตัวจำแนก ผลลัพธ์ของอัลกอริทึมที่ได้คือการตรวจจับใบหน้า

บางงานวิจัยสามารถพิสูจน์ได้ว่าเครือข่ายประสาทเทียมสามารถปรับปรุงประสิทธิภาพของการตรวจจับใบหน้าได้ แต่จะเพิ่มเวลาประมวลผล [2] Farfade และคณะ เสนอวิธีการที่เรียกว่า Deep Dense Face Detector (DDFD) ซึ่งสามารถตรวจจับใบหน้าในหลากหลายทิศทางโดยใช้แบบจำลองเดียวบนพื้นฐานของโครงข่ายประสาทเทียมเชิงลึก โดยวิธีการไม่จำเป็นต้องมีการใช้ segmentation, bounding-box regression หรือ SVM classifiers จากงานวิจัยพบว่าวิธีนี้สามารถตรวจจับใบหน้าจากมุมที่แตกต่างกันได้และมีความสัมพันธ์ระหว่างการกระจายตัวของ positive example (ภาพที่มีใบหน้า) ใน training set และคะแนนของเครื่องตรวจจับใบหน้าที่นำเสนอ

เนื่องจากคุณสมบัติของอัลกอริทึม Viola และ Jones มีความสามารถในการแสดงที่จำกัด ซึ่งมีขนาด feature pool size ขนาดใหญ่ จึงมีการใช้อัลกอริทึมโดยใช้คุณสมบัติของ channel features ที่เรียกว่า aggregate channel features อัลกอริทึมแยกคุณสมบัติโดยตรงเป็นค่าพิกเซลโดยไม่ต้องคำนวณผลรวมรูปสี่เหลี่ยมผืนผ้าที่ตำแหน่งและสเกลต่าง ๆ โดยใช้ integral images ซึ่งมีเพียง serval thousands ของ features pool size สิ่งนี้นำไปสู่การคำนวณคุณสมบัติที่เร็วขึ้นสำหรับการส่งเสริมการเรียนรู้ การทดลองพิสูจน์ว่า LUV channel และ gradient magnitude และ 6-bin histograms คำนวณบนพื้นที่สี RGB เป็นตัวเลือกที่ดีที่สุดสำหรับการตรวจจับใบหน้า ขนาดหน้าต่างการตรวจจับที่ใหญ่ขึ้นนั้นจะให้ประสิทธิภาพที่ดี แต่อาจทำให้หลาดใบหน้าเล็ก ๆ และนำไปสู่การตรวจจับที่ไม่มีประสิทธิภาพ ดังนั้น 80 x 80 และปัจจัยย่อยตัวอย่างของ 4 จึงเป็นค่าที่เหมาะสมที่สุดและสมเหตุสมผลที่สุดตามการทดลอง ดังนั้นวิธี face method โดยใช้ aggregate channel features จะแสดงให้เห็นถึงประสิทธิภาพการแข่งขันกับงานก่อนหน้าในขณะที่การคำนวณเร็วขึ้นบน VGA images

Unified Cascaded Convolutional Neural Networks (CNN) ถูกใช้เพื่อตรวจจับใบหน้าและการจัดแนว โดยการเรียนรู้แบบ multi-task learning ซึ่งเรียกว่า MTCNN.Framework ประกอบด้วยสามขั้นตอน: ขั้นแรกเพื่อให้ได้การถดถอยกล่องขอบเขตพวกซึ่งใช้เครือข่าย convolutional ที่เรียกว่าเครือข่ายข้อเสนอ (P-Net) หลังจากได้รับ estimated bounding box regression candidates โดยประมาณแล้วใช้ non-maximum suppression (NMS) เพื่อละเว้นกล่องขอบที่ทับซ้อนกันอย่างมาก candidates ทั้งหมดจะถูกป้อนเข้าสู่เครือข่ายการปรับแต่ง (R-Net), R-Net มีเป้าหมายที่จะปฏิเสธผู้สมัครปลอมจำนวนมากทำการปรับเทียบด้วยการ bounding box regression และ NMS candidate merge ในที่สุด Output Network (O-Net) จะอธิบายรายละเอียดของใบหน้าและตำแหน่งห้าสถานที่สำคัญบนใบหน้า ใช้เวลา 16fps สำหรับการตรวจจับใบหน้าและการจัดแนวบน CPU 2.60 GHz และ 99fps บน Nvidia Titan Black GPU วิธีการดังกล่าวบรรลุความแม่นยำโดยเฉลี่ย 95% ในแต่ละขั้นตอนในหลาย ๆ ประเด็นที่ท้าทายการผลิตในขณะที่ยังคงประสิทธิภาพตามเวลาจริง



รูปที่ 1 : Pipeline ของ frameworkเพื่อตรวจจับและจัดแนวใบหน้าด้วยสามขั้นตอน

(ที่มา : http://cuir.car.chula.ac.th/bitstream/123456789/60143/1/5970375021.pdf, 2020)

2.3 องค์ความรู้เกี่ยวกับ Face Alignment [3]

การจัดตำแหน่งใบหน้าเป็นโมดูลที่สำคัญใน pipeline หลังจากการตรวจจับใบหน้าและก่อนที่จะแยกคุณสมบัติและการจำแนกประเภทของขั้นตอนวิธีการวิเคราะห์ใบหน้าส่วนใหญ่ นอกจากนี้รูปภาพใบหน้าอาจจำเป็นต้องผ่านการปรับจัดตำแหน่งใบหน้าก่อนประมวลผล เนื่องจากการแสดงผลแบบจะไม่ระนาบ และเป็นเรื่องปกติที่ใบหน้าจะไม่อยู่กึ่งกลางภาพ เพื่อจัดการกับปัญหานี้เราใช้การแปลงภาพเพื่อจัดกึ่งกลางใบหน้าตามตำแหน่งของจุดสังเกต

อัลกอริทึมที่เสนอเพื่อประเมินตำแหน่งจุดสังเกตของใบหน้าอย่างแม่นยำโดยใช้ regression tree วิธีการนี้สามารถประมาณค่าได้โดยตรงจากชุดความเข้มของพิกเซลแบบเบาบางในเวลาจริง พร้อมการคาดการณ์คุณภาพสูง และเสนอการปรับค่าและหาค่าเฉลี่ยการทำนายโดย multiple regression trees เป็นวิธีการทำให้เป็นมาตรฐานในการ shape regression , เพื่อลดและสร้างการทำให้เป็นมาตรฐานดีขึ้น วิธีนี้ผ่านการ train ด้วย iBUG 300-W และ 17 HELEN ฐานข้อมูลที่มีความหลากหลายในรูปแบบการส่องสว่าง , พื้นหลัง ,การบิดเบี้ยว และคุณภาพของภาพ เป้าหมายของวิธีนี้คือการกำหนดจุดแลนด์มาร์ค ดังแสดงในรูปที่9 จุดสังเกตสำคัญเหล่านี้ประกอบด้วยคางขอบตาคิ้วปากและจมูก ในที่สุดใช้ 68 จุดเหล่านี้ในการหมุนปรับขนาดและแปลใบหน้าให้สอดคล้องกับภาพใบหน้าด้านหน้า

รูปที่ 2 : การแสดงพิกัดจุดที่สำคัญทางใบหน้า 68 ภาพจาก iBUG 300-W ชุด

(ที่มา : http://cuir.car.chula.ac.th/bitstream/123456789/60143/1/5970375021.pdf, 2020)

รูปที่ 3 : การตรวจจับ 68 จุดแลนด์มาร์ค

(ที่มา : http://cuir.car.chula.ac.th/bitstream/123456789/60143/1/5970375021.pdf, 2020)

วิธีประเมินสถานที่สำคัญบนใบหน้าสามารถประเมินได้โดยใช้ cascade ของ regressor โดยทั่วไปจุดใบหน้าจะแสดงเป็นเวกเตอร์คุณลักษณะ S = (x1y1, x2y2, … , xpyp) ∈ R2p ซึ่งแต่ละคู่ (xi,yi) เป็นพิกัดของจุดที่สำคัญทางใบหน้าที่ i ในรูปใบหน้า I และ p คือจำนวนของจุดสังเกตในตัวอย่างนี้คือ 68 ดังแสดงในรูป10 ให้ $\hat{S}$ (t) แทนการประมาณค่าปัจจุบันของเวกเตอร์ใบหน้า S แต่ละอัน regressor rt (.,.), ใน cascade ทำนายการปรับปรุงเวกเตอร์จากภาพและ $\hat{S}$(t) ซึ่งจะถูกเพิ่มเข้าไปในประมาณการรูปร่างปัจจุบัน $\hat{S}$ (t)

$$S^{(t+1)} = S^{(t)} + r_t(I, S^{(t)})$$

regressor rt ทำให้การคาดการณ์ของมันขึ้นอยู่กับคุณสมบัติเช่นค่าความเข้มของพิกเซลที่คำนวณจากภาพ I และการจัดทำดัชนีสัมพันธ์กับการประมาณรูปร่างปัจจุบัน $\hat{S}$ (t)

ความท้าทายของการจัดตำแหน่งใบหน้าในแอปพลิเคชันโลกแห่งความจริงมาจากการเปลี่ยนแปลงขนาดความสว่าง เสนอกรอบการทำงานเพื่อแก้ไขปัญหาเหล่านี้โดยใช้ CNNs แบบรวมกลุ่มแบบต่อเรียงโดยการเรียนรู้แบบหลายงาน ข้อเสนอประกอบด้วยสามขั้นตอน ขั้นแรกให้สร้างหน้าต่างตัวเลือกผ่านอย่างรวดเร็ว CNN ประการที่สองปฏิเสธจำนวนที่ไม่ใช่ใบหน้าในหน้าต่าง สุดท้ายใช้ CNN เพื่อปรับแต่งผลลัพธ์และส่งออกตำแหน่งจุดสังเกตใบหน้า

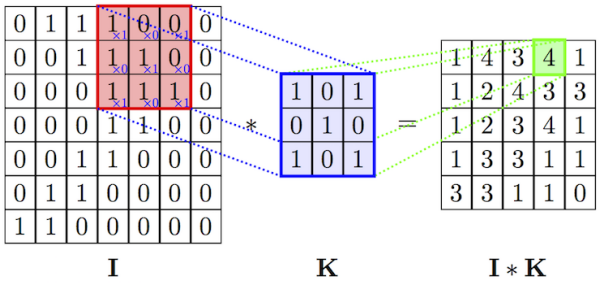
2.4 องค์ความรู้เกี่ยวกับ Convolutional Neural Network (CNN) [4]

Convolutional Neural Networks (CNN) หรือ โครงข่ายประสาทแบบคอนโวลูชัน เป็นโครงข่ายประสาทเทียมอย่างหนึ่งในกลุ่ม bio - inspired โดยที่ CNN จะจำลองการมองเห็นของมนุษย์ที่มองเป็นพื้นที่ย่อย ๆ และนำกลุ่มของพื้นที่ย่อย ๆ มาผสมกัน เพื่อดูว่าสิ่งที่มองเห็นอยู่นั้นเป็นอะไร ซึ่ง CNN นั้นจะคล้ายกันกับโครงข่ายประสาทปกติ แต่เซลล์ประสาทของ CNN จะส่งสัญญาณเข้าด้วยกัน ซึ่งจะประกอบด้วย ค่าน้ำหนัก (Weight) และ Bias ที่จะสามารถที่ในการศึกษาต่อได้ แต่ละเซลล์ประสาทจะได้รับอินพุต (Input) ในการดำเนินการ ในการจำแนกอินพุต (Input) ที่เป็นรูปภาพที่สามารถมองเห็นได้จากโครงสร้างทางสถาปัตยกรรมของรูปภาพ จากนั้นจะใช้ฟังก์ชันในการส่งต่อเพื่อลดจำนวนพารามิเตอร์ในโครงข่าย โดย CNN จะมีระดับชั้นของเซลล์ประสาทเป็นมิติ ได้แก่ ความกว้าง ความสูง และช่องสี (ซึ่งเรียกว่าความลึกในโครงข่ายประสาทเทียม)

โดยปกติ ลำดับชั้นที่ใช้ในการสร้าง CNN จะประกอบด้วยสถาปัตยกรรม 3 ประเภท ได้แก่ Convolution layer, Pooling layer และ Fully-connected layer

2.4.1 องค์ความรู้เกี่ยวกับ Convolutional Layer

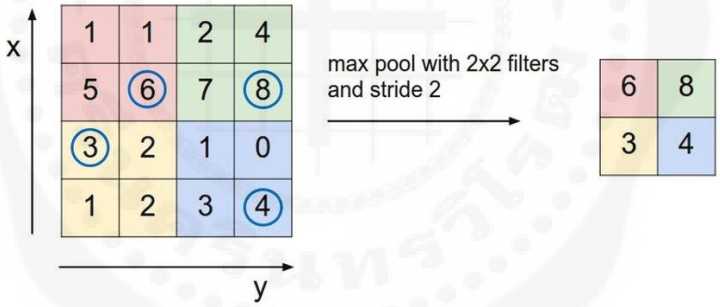
Convolutional Layer (ชั้นคอนโวลูชัน) เป็นลักษณะเด่นของโครงข่ายแบบ CNN ก็คือการทำงานของ Convolutional Layer ที่คำนวณเพื่อหาชั้นของผลลัพธ์ซึ่งเรียกว่า Feature Map ด้วยการนำพื้นที่ส่วนย่อยรูปภาพ (Sub-region) ไปคำนวณแบบ dot product กับเคอร์นอล (Kernel) โดย Kernel ที่นำมาคำนวณจะต้องมีขนาดเล็กกว่ารูปภาพ โดยจุดประสงค์ของชั้น Convolutional คือการ Training คุณสมบัติของ Object ในรูปภาพ พารามิเตอร์ของชั้น Convolutional ประกอบด้วยชุดของตัวกรองที่ Training มาได้จากฟิลเตอร์ที่แปลงไปสู่ค่าอินพุต (Input) นั้น โดยฟิลเตอร์นั้นอาจจะเป็น ขอบ, เส้น, ความคมชัด เป็นต้น โดยจะได้ Output คือ Kernel ในการประมวลผลรูปภาพ ให้ค่าภาพ I คือภาพสองมิติ และ Kernel คือ K โดยจะคำนวณแนวโน้มของรูปภาพจาก I*K โดยการซ้อนทับตัวกรอง (Kernel) ที่ด้านบนของภาพ จากด้านซ้ายไปขวา[5] ตามลำดับ



รูปที่ 4 : รูปแสดงตัวอย่าง Convolution ในการรับอินพุต (Input) ของรูปภาพ และ ฟิลเตอร์ *(ที่มา : https://anhvnn.wordpress.com/2018/02/01/deep-learning-computer-vision-and-convolutional-neural-networks/, 2020)*

2.4.2 องค์ความรู้เกี่ยวกับ Pooling Layer

Pooling Layer เป็นชั้นที่เชื่อมจาก Convolution Layer โดยมีเป้าหมายคือทำให้ขนาดของ Feature Map ลดลง ในการคำนวณสามารถใช้ค่าต่ำสุด (Min Pooling) ค่าสูงสุด (Max Pooling) ผลรวม (Sum Pooling) และค่าเฉลี่ย (Average Pooling) ในการคำนวณ Feature Map จะถูกแบ่งออกเป็นบล็อคขนาดต่าง ๆ ซึ่งหากใช้วิธี Max Pooling ในการคำนวณ ค่าที่ได้ก็คือค่าสูงสุด (Max Value) ของแต่ละบล็อค



รูปที่ 5 : รูป A pooling of the input image with a 2x2 filters and stride of 2
(ที่มา : https://anhvnn.wordpress.com/2018/02/01/deep-learning-computer-vision-and-convolutional-neural-networks/, 2020)

2.4.3 องค์ความรู้เกี่ยวกับ Fully-connected Layer

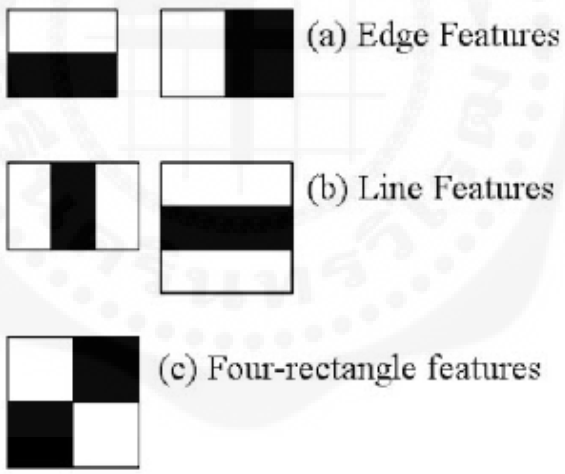
Fully-Connected Layer คือ Hidden Layer และ Output Layer ของโครงข่ายประสาทเทียม ดังนั้น Fully-Connected Layer จึงทำหน้าที่ในการเรียนรู้ (Training) และการจำแนกประเภทของวัตถุ โดยผลลัพธ์ ที่ได้ก็คือจำนวนของ Class ที่ต้องการจำแนก

2.4.4 องค์ความรู้เกี่ยวกับ Hyperameter

Hyperameter คือ ตัวแปรเริ่มต้นก่อนที่จะทำการฝึกอบรมโมเดล โครงข่ายประสาทเทียมสามารถมีไฮเปอร์พารามิเตอร์ได้หลายตัว ซึ่งประกอบด้วย optimizer's hyperparameters เช่น learning rate, decay rates, step size และ batch size รวมถึง model's hyperparameters เช่น จำนวนของเลเยอร์, จำนวนของ unit แต่ละชั้น, dropout rate และ activation function (ReLu, Sigmoid, Tanh)

2.5 Haar cascade classifiers Detection [6]

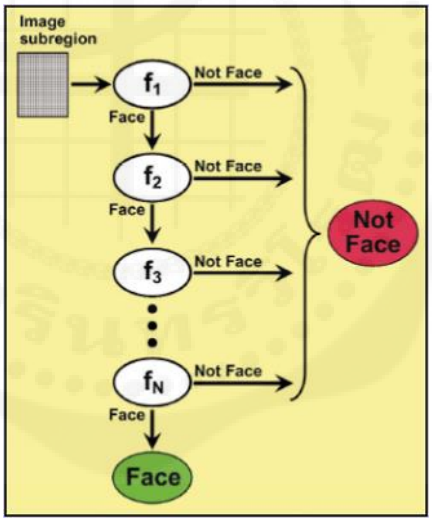
Haar cascade classifiers เป็นวิธีการตรวจจับวัตถุที่มีประสิทธิภาพที่เสนอโดย Paul Viola และ Michael Jones ในงานวิจัยของเขา โดยสามารถนำมาใช้งานร่วมกับการตรวจจับใบหน้าได้ ในขั้นต้น อัลกอริทึมต้องการรูปภาพจำนวนมากในการฝึกจำแนก classifier ทั้ง positive images (ภาพใบหน้า) และ negative images (ภาพที่ไม่มีใบหน้า) ซึ่งจะต้องดึงคุณสมบัติของมันออกมาใช้ โดยมันจะเหมือนกับ convolutional kernel แต่ละ Features คือ single value ที่ได้จากการลบ (subtracting) ผลรวมของพิกเซลภายใต้สีเหลี่ยมสีขาวจากจากผลรวมของพิกเซลภายใต้สี่เหลี่ยมสีดำ ดังรูป



รูปที่ 6 : รูปตัวอย่างคุณสมบัติของ Haar cascade classifier

(ที่มา : https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html, 2020)

โดยในการคำนวนแต่ละคุณสมบัติเราจำเป็นต้องค้นหาผลรวมของพิกเซลภายใต้สี่เหลี่ยมสีขาวและสีดำ ซึ่งในการแก้ปัญหาโดยทั่วไปมักจะเรียกว่า Viola-Jones method ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นมีการนำเสนอวิธีการแทนรูปภาพที่ เรียกว่า"Integral Image" ซึ่งช่วยให้การคำนวณ feature ทำได้รวดเร็วขึ้นและได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้นฐานจาก AdaBoost ซึ่งเลือกเอาเฉพาะ critical features ที่ให้ classifiers ที่มีประสิทธิภาพสูงสุด นอกจากนี้ยังได้อธิบายถึงการรวม classifiers แบบ cascade ซึ่งช่วยให้ส่วนพื้น หลังของภาพถูกปฏิเสธได้เร็วและเน้นการคำนวณไปที่บริเวณที่มีลักษณะคล้ายวัตถุที่สนใจมากขึ้น หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการสแกน sub-window เพื่อตรวจหาใบหน้าจากรูปภาพอินพุต การประมวลผลภาพแบบทั่วไปจะใช้การปรับขนาดภาพขาเข้าแตกต่างกันหลายๆขนาด และใช้ตัวตรวจจับ (Detector) ที่มีขนาดคงที่ค้นหาวัตถุ และได้มีการสร้างตัวจำแนกประเภทแบบ cascaded (Cascaded classifier) ซึ่งเมื่อ sub-window ถูกจัดประเภทเป็น ไม่ใช่ใบหน้า (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูกจำแนกเป็น มีโอกาสเป็นใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายิ่งมีจำนวนชั้น ของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าจะยิ่งมีมากขึ้น

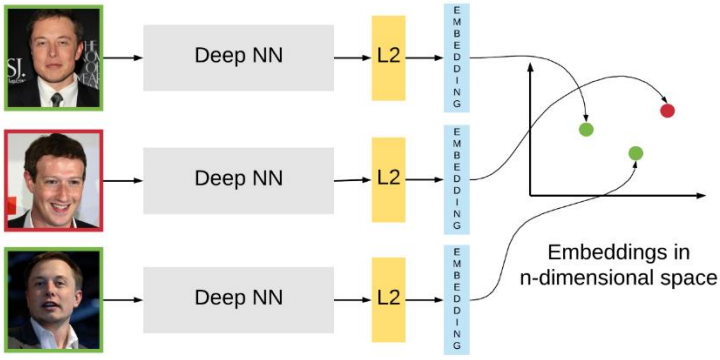


รูปที่ 7 : รูปตัวอย่างแสดงการทำงาน Classifier cascade

(ที่มา : http://www.mns-smartpro.com/blog/, 2020)

2.6 องค์ความรู้เกี่ยวกับ FaceNet

FaceNet เป็นหนึ่งใน Embedding Learning Framework ที่ใช้ในการทำ Face Recognition โดยมีจุดเด่นคือการสร้าง Embedding Features จากภาพใบหน้า ให้อยู่ในรูป Euclidean space และสามารถใช้ Distance สื่อความหมายของใบหน้าได้โดยตรง[7]



รูปที่ 8 : รูปตัวอย่างการทำ Face embedding

(ที่มา : https://medium.com/datawiz-th/facenet-triplet-loss, 2020)

2.6.1 องค์ความรู้เกี่ยวกับ Euclidean space

โดยทั่วไปแล้วโมเดลอื่น ๆ ที่ทำ Face recognition มักจะเทรน Classification layer จากใบหน้าบุคคล และใช้ Bottleneck layer รองสุดท้าย (ก่อน Output layer) มาเป็น Embedding features ข้อเสียของวิธีนี้มี 2 ข้อหลัก ๆ คือ

- Indirectness : จากการใช้ Bottleneck layer ที่หวังว่ามันจะ Generalize เพียงพอและนำไปใช้ในการสร้าง Face embeddings กับหน้าใหม่ ๆ ได้
- Inefficiency : จากการใช้ Bottleneck layer ที่ปกติแล้วจะได้ Embeddings ที่มีขนาดใหญ่ (กว่าพัน Dimensions) ทำให้ใช้เวลามากกว่าในการคำนวณ

2.6.2 องค์ความรู้เกี่ยวกับ Triplet loss

Triplet หมายถึง การเกี่ยวกับกลุ่ม 3 คนหรือแฝด 3 ซึ้งก็บ่งบอกถึงการเทรนโมเดลของ FaceNet ที่จะใช้การเปรียบเทียบ Embeddings ของ 3 ตัวอย่างพร้อมกันในการคำนวณหา loss

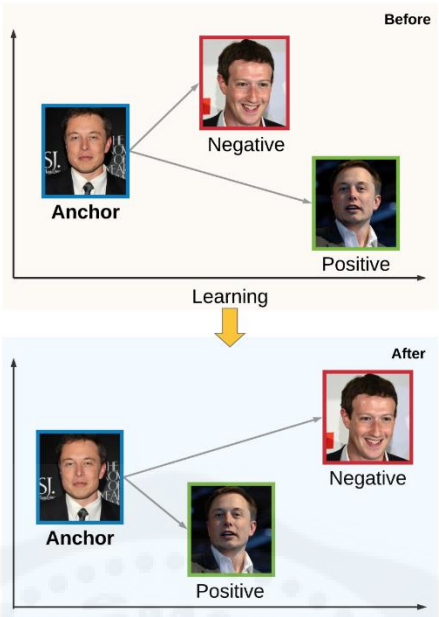
Objective ของการหา Triplet loss คือ Embeddings ของใบหน้าคนเดียวกัน อยู่ใกล้กันมากกว่าใบหน้าของคนอื่น

ซึ่งจะประกอบไปด้วย 3 ส่วนดังนี้

Anchor : ตัวอย่างตั้งต้นที่ใช้ในการเปรียบเทียบ

Positive : ตัวอย่างของใบหน้าเดียวกับ Anchor

Negative : ตัวอย่างขงใบหน้าที่ต่างกัน Anchor



รูปที่ 9 : รูปตัวอย่างขั้นตอนการ Learning ด้วย Triplet loss

(ที่มา : https://medium.com/datawiz-th/facenet -triplet-loss, 2020)

โดย Objective คือ

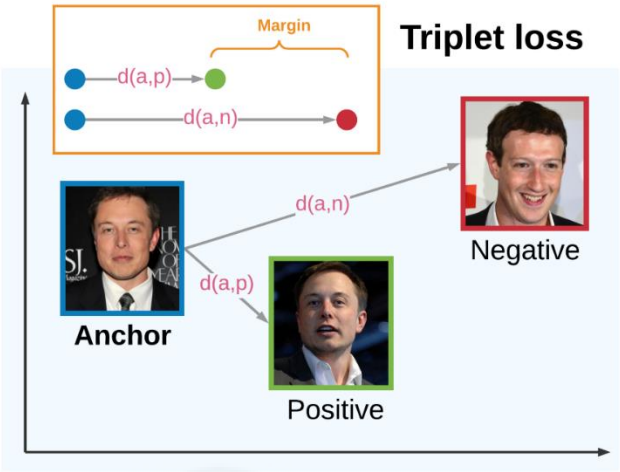
- ลด Distance ระหว่าง Anchor และ Positive (เป็นบุคคลเดียวกัน)
- เพิ่ม Distance ระหว่าง Anchor และ Negative (เป็นคนละคนกัน)

โดยระยะทางบน Embeddings space ก็คือ Loss ของ Triplet และสามารถเขียนออกมาเป็นสูตรได้คือ

$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

มี A เป็น Anchor input, P เป็น Positive input, N เป็น Negative input, และค่า Margin ระหว่าง d(a, p) และ d(a, n) อธิบายง่าย ๆ คือ Optimize จนได้ d(a, n) - d(a, p) มีค่ามากกว่าหรือเท่ากับ Margin (loss เป็นศูนย์) ลองย้ายสมการ

- d(a, p) - d(a, n) + margin =< 0
- d(a, n) >= d(a, p) + margin
- d(a, n) - d(a, p) >= margin

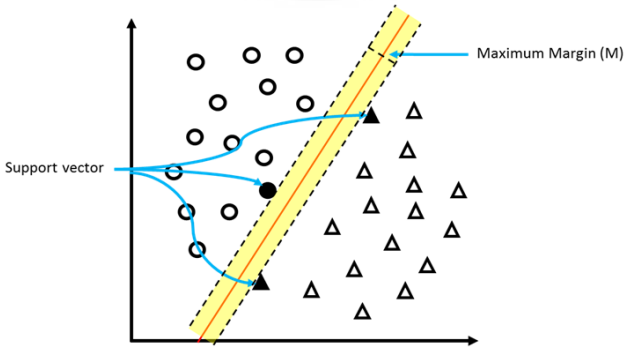


รูปที่ 10 : รูปตัวอย่างแสดงรูปภาพการทำงานของ Triplet loss

(ที่มา : https://medium.com/datawiz-th/facenettriplet-los, 2020)

2.7 องค์ความรู้เกี่ยวกับ Support Vector Machine (SVM)

ซัพพอร์ตเวกเตอร์แมชชิน (Support Vector Machine: SVM)[8] เป็นตัวจำแนกเชิงเส้น (Linear Classifier) แบบ 2 คลาส ซึ่งเป็นที่ยอมรับถึงประสิทธิภาพของการจำแนกที่เหนือว่าวิธีการจำแนกอื่น ๆ ข้อได้เปรียบของ SVM คือมีประสิทธิภาพในการจำแนกข้อมูลที่มีมิติจำนวนมากได้ นอกจากนี้การใช้ฟังก์ชันเคอร์เนล (Kernel Function) เพื่อแปลงข้อมูลไปยังมิติที่สูงขึ้นในปริภูมิคุณลักษณะ (Feature Space) สามารถจำแนกข้อมูลที่มีความคลุมเครือได้อย่างมีประสิทธิภาพ หลักการของ SVM คือการหาเส้นตรงที่มีมาร์จินที่โตที่สุด (Maximum Margin) ที่สามารถแบ่งข้อมูลออกเป็น 2 คลาส ดังตัวอย่างในภาพที่ 2 เป็นข้อมูลขนาด 2 มิติ โดนถูกจำแนกออกเป็น 2 คลาส ได้แก่ + (O) และคลาส – (Δ) โดยเส้นตรงที่ใช้แบ่งข้อมูลมีมาร์จินเท่ากับ $M=2w$ ซึ่ง เป็นความกว้างระหว่างเส้นตรงกับซัพพอร์เวคเตอร์[9] ของข้อมูลทั้ง 2 คลาส (● และ ▲)

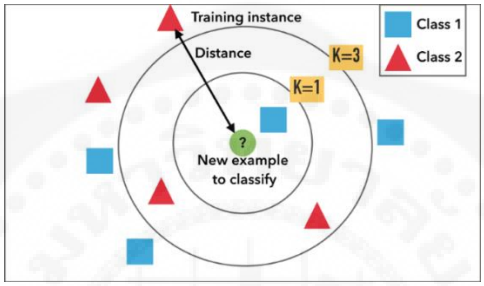


รูปที่ 11 : รูปตัวอย่างของตัวแบบจำแนก SVM บนข้อมูลขนาด 2 มิติ

(ที่มา : https://knowledge.snru.ac.th/, 2020)

2.8 องค์ความรู้เกี่ยวกับ K-Nearest Neighbors (KNN)

K-Nearest Neighbors เป็นอัลกอลิทึม ที่ใช้สำหรับการจัดกลุ่มข้อมูล (Classification) ซึ่งเป็นอัลกอลิทึมที่อยู่ในกลุ่มของ Supervised learning โดยแนวคิดและหลักการพื้นฐานของ KNN ก็คือ กำหนดขนาดของค่า K แล้วคำนวณหาระยะห่าง (Distance) ของข้อมูลที่ต้องการพิจารณากับกลุ่มข้อมูลตัวอย่าง หลังจากนั้นก็จะจัดเรียงลำดับของระยะห่างและเลือกพิจารณาชุดข้อมูลที่ใกล้จุดที่ต้องการพิจารณาตามจำนวน k ที่ได้กำหนดไว้[10] และพิจารณาข้อมูลจำนวน k ชุด และสังเกตว่ากลุ่ม (Class) ไหนที่ใกล้จุดที่พิจารณาเป็นจำนวนมากที่สุด ต่อด้วยกำหนด Class ให้กับจุดที่พิจารณา



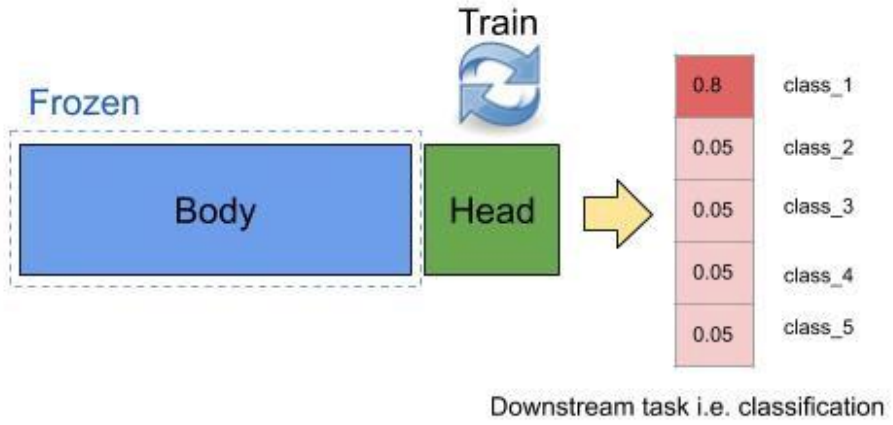
รูปที่ 12 : รูปตัวอย่างของตัวแบบจำแนกClass ของ KNN

(ที่มา :https://medium.com/@kongruksiamza/, 2020)

2.9 องค์ความรู้เกี่ยวกับ Transfer Learning

การทำ Transfer learning คือ การถ่ายทอดความรู้จาก model ที่ผ่านเทรน Deep Learning มาแล้วเรียบร้อยมาใช้งาน ซึ่งส่วนใหญ่จะผ่านการเทรนจากชุดข้อมูลที่มีขนาดใหญ่ เช่น imageNet ซึ่งนำมาใช้เป็นส่วนหนึ่งของ model ใหม่[11] โดยการนำ weight ของ model ที่เทรนมาแล้ว มาใช้ initialize ค่า weight ของ model ที่เราจะทำการเทรนใหม่ ทำให้ประหยัดเวลาในการเทรน model converge และสามารถเรียนรู้จากข้อมูลที่ไม่มากได้ เมื่อเทียบกับการที่ต้องเทรนตั้งแต่ต้น

ทำได้โดยการ freeze ส่วน body ของ model เพื่อไม่ให้มีการอัพเดทค่า weight และเทรนเฉพาะส่วน head เพื่อปรับ weight ส่วนนี้ก่อน เนื่องจากเป็นส่วนที่เรา random initialize ขึ้นมาใหม่ หลังจากนั้นทำการ unfreeze ทั้ง model เพื่ออัพเดท weight พร้อมกันทั้ง model โดยกำหนดให้ learning rate ต่ำ ๆ สำหรับ DCNN layer ตื้น ๆ เนื่องจาก pattern พื้นฐาน เป็นอะไรที่ไม่ต้องปรับมาก และกำหนด learning rate สูง ๆ สำหรับ DCNN layer ลึก ๆ และส่วน head ของ model



รูปที่ 13 : รูปภาพตัวอย่างการทำ Transfer learning

(ที่มา : https://medium.com/datawiz-th-google-image,2021)

2.10 องค์ความรู้เกี่ยวกับ Pre-train Model

คือโมเดลที่นักวิจัยคนอื่น ๆ ได้เทรนโมเดลด้วยรูปภาพจำนวนมากมาแล้ว เพื่อให้ได้โมเดลที่ได้เรียนรู้ฟีเจอร์ของรูปภาพและเก็บ Weight และ Bias มาแล้ว มาให้ได้โหลดมาใช้งานได้ง่าย ๆ ไม่ต้องมาเทรนเองทั้งหมด ซึ่งเรียกว่า Transfer Learning

2.10.1 องค์ความรู้เกี่ยวกับ VGG16

VGG เป็นโครงข่ายใยประสาทแบบชั้น 16 โดยไม่นับเลเยอร์ maxpool และ softmax ในตอนท้าย มันยังเรียกว่า VGG16 สถาปัตยกรรมเป็นสิ่งที่เราทำงานด้วยด้านบน สแต็คแบบซ้อนซ้อน รวมชั้นตามด้วย ANN ที่เชื่อมต่ออย่างสมบูรณ์

2.10.2 องค์ความรู้เกี่ยวกับ MobileNet

MobileNet คือ โมเดลขนาดเล็ก ที่ทำงานได้เร็ว Latency ต่ำ ใช้พลังงานในการประมวลผลไม่มาก ถูกออกแบบมาสำหรับงานที่มีทรัพยากรจำกัด MobileNet สามารถใช้งานได้ทั้ง Classification, Detection, Embedding และ Segmentation เหมือนกับโมเดลที่เป็นที่นิยมอื่น ๆ เช่น ResNet, Inception, U-Net

2.10.3 องค์ความรู้เกี่ยวกับ ResNet50

ResNet-50 เป็นโครงข่ายประสาทเทียมที่มีความลึก 50 ชั้น สามารถโหลด pretrained ของเครือข่ายการเทรนในมากกว่าหนึ่งล้านภาพจากฐานข้อมูล ImageNet เครือข่ายที่กำหนดไว้ล่วงหน้าสามารถจำแนกรูปภาพ

ออกเป็นหมวดหมู่วัตถุได้ 1,000 ประเภทเช่นแป้นพิมพ์, เมาส์, ดินสอและสัตว์หลายชนิด เป็นผลให้เครือข่ายได้เรียนรู้การแสดงคุณลักษณะที่หลากหลายสำหรับภาพที่หลากหลาย เครือข่ายมีขนาดอินพุตภาพ 224 x 224 สำหรับเครือข่าย pretrained อื่น ๆ ใน MATLAB สามารถใช้ classify เพื่อจัดประเภทรูปภาพใหม่โดยใช้โมเดล ResNet-50 ทำตามขั้นตอนของการจัดประเภทรูปภาพโดยใช้ GoogLeNet และแทนที่ GoogLeNet ด้วย ResNet-50 หากต้องการเทรน Neural Network อีกครั้ง ให้ทำตามขั้นตอนของ Train Deep Learning Network เพื่อจัดประเภทรูปภาพใหม่และโหลด ResNet-50 แทน GoogLeNet

2.11 โครงงานที่เกี่ยวข้อง

2.11.1 งานวิจัยเรื่อง “Deep Learning Transfer Learning ” by Rattanachot Phwanwilai”[12]

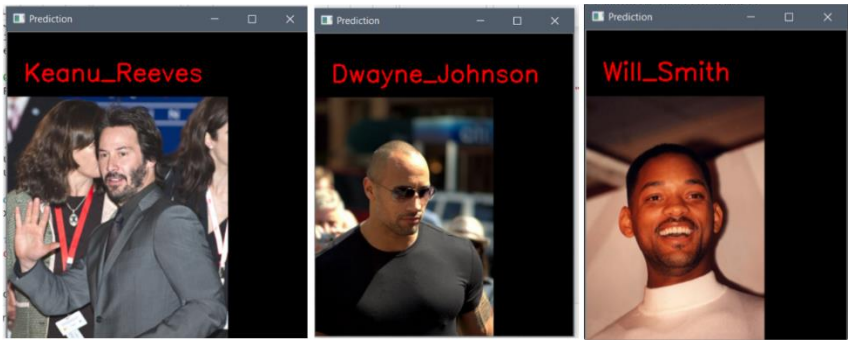
บทความนี้เป็นการใช้เทคนิค Transfer Learning โดยใช้ Pre-trained models ของโมเดล Vgg16 และ Vgg19 โดยใช้โครงสร้างของ Convolution Neaural Network ที่ผ่านการเทรนมาแล้วมาใช้งาน แล้วทำการปรับแต่งในขั้นตอนสุดท้ายในชั้น Fully Connected ซึ่งจะทำให้ประหยัดเวลาในการเทรนโมเดลอย่างมาก เพื่อจะทำการจำแนกข้อมูลภาพตามจำนวนชั้นของข้อมูลตามที่ได้กำหนด ซึ่งผลลัพธ์ที่ได้จะเป็น Class

โดยข้อดีของการใช้ Transfer Learning using pre-trained models คือประหยัดเวลาในการเทรนโมเดล โดยไม่จำเป็นต้องใช้ชุดข้อมูลที่มีขนาดใหญ่เพื่อใช้ในการเทรนก็ได้ และยังสามารถประยุกต์ข้อมูลที่หลากหลายอาทิเช่น Image Classification, Face Recognition เป็นต้น

2.11.2 งานวิจัยเรื่อง “ Face Recognition using Transfer Learning on MobileNet ” by Akashdeep Gupta[13]

บทความนี้เป็นการใช้ Transfer Learning เพื่อสาธิตการทำงานของ Transfer Leaning โดยใช้ pre-trained model ที่ผ่านการเทรนมาแล้ว โดยในบทความนี้จะมีเรียกใช้โมเดล MobileNet ซึ่งมันจะทำให้ลดเวลาและทรัพยากรในการคำนวนเป็นอย่างมาก

โดยทำการสาธิตตั้งแต่การเตรียมชุดข้อมูลเพื่อที่จะนำมาใช้ในการเทรน ซึ่งชุดข้อมูลที่นำมาใช้ในการเทรนเป็นข้อมูลรูปภาพของบุคคล เพื่อที่จะใช้ในการทำการจดจำและทำนายใบหน้าในขั้นตอนถัดไป และความถูกต้องของแบบจำลองที่ผ่านการทำ Transfer Learning สูงถึง 91% โดยใช้เวลาและพลังงานในการคำนวณน้อยมาก



รูปที่ 14 : ภาพตัวอย่างการทำนายโดยใช้การ Transfer Learning using MobileNet model

(ที่มา : https://medium.com/analytics-vidhya/face-recognition-using-transfer-learning-on-mobilenet-cf632e25353e, 2021)

2.11.3 Top 4 Pre-Trained Models for Image Classification with Python Code by Purvra Huilgol**[14]**

งานวิจัยนี้เป็นการทดสอบ pre-trained model ที่ผ่านการเทรนมาแล้ว 4 อันดับแรกที่เป็นที่นิยมสำหรับการทำ Classification โดยจะทำการอธิบายและทดสอบการใช้งานของแต่และโมเดลที่นำมาเปรียบเทียบกันเพื่อดูประสิทธิภาพของแต่ละโมเดล โดย pre-trained model ที่นำมาเปรียบเทียบมีดังนี้ VGG16 , ResNet50, InceptionV3 และ EfficientNet โดยชุดข้อมูลที่นำมาใช้คือรูปสนัขและแมว แล้วทำการเปรียบเทียบประสิทธิภาพและความถูกต้องของ pre-trained model แต่ละตัว ดังแสดงในภาพตัวอย่างที่ 13

Model	Year	Number of Parameters	Top-1 Accuracy
VGG-16	2014	138 Million	74.5%
ResNet-50	2015	25 Million	77.15%
Inception V3	2015	24 Million	78.8%
EfficientNetB0	2019	5.3 Million	76.3%
EfficientNetB7	2019	66 Million	84.4%

ตารางที่ 1 : ตารางการเปรียบเทียบประสิทธิภาพของ Pre-trained models

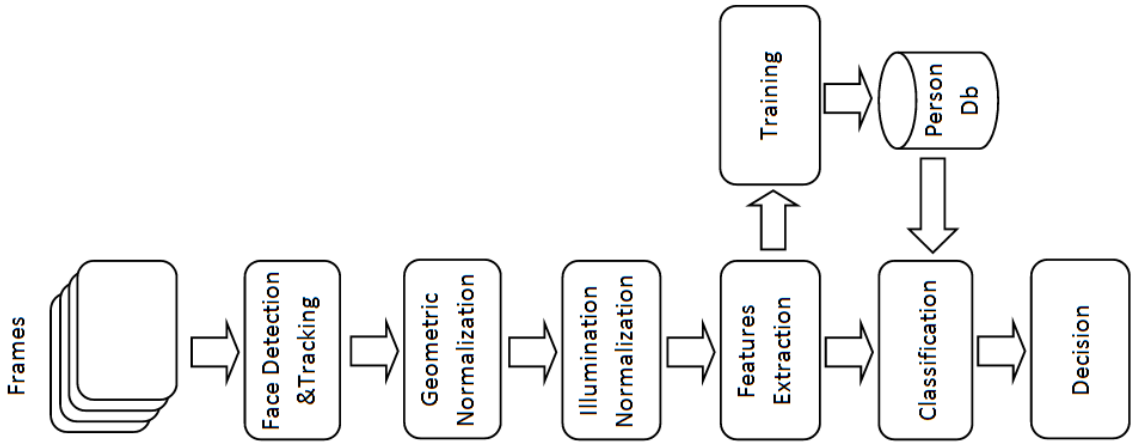
(ที่มา : https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/, 2021)

2.11.4 งานวิจัยเรื่อง “Evaluating OpenFace: an open-source automatic facial comparison algorithm for forensics [15]

งานวิจัยนี้เป็นการตรวจสอบการรู้จำใบหน้ากับการจัดกลุ่มของ OpenFace ของข้อมูลภาพนิ่งในชุดข้อมูลหลายชุด เพื่อหาประสิทธิภาพของชุดข้อมูล LFW ที่นำมาใช้งานกับ OpenFace ซึ่งการเทรนโมเดลนี้สามารถปรับปรุงประสิทธิภาพของ OpenFace บนชุดข้อมูลที่เกี่ยวข้องได้ โดยจะทำการเปรียบเทียบชุดข้อมูลจำนวน 4 ชุด ได้แก่ LFW-raw, LFW-deep funnelled, SCface และ ForenFace เพื่อหาค่าความต่างของค่า Energy Efficiency Ratio (EER), Area under the curve (AUC), Threshold ของแต่ละชุดข้อมูล และจากนั้นทำการเปรียบเทียบประสิทธิภาพของโมเดลจำนวน 4 โมเดล ได้แก่ nn4.v1, nn4.v2, nn4.small2.v1 และ nn4.small1.v1 เพื่อหาค่าความต่างของค่า Energy Efficiency Ratio (EER), Area under the curve (AUC), Threshold และ Runtime(min) หลังจากนั้นเลือกโมเดลที่มีประสิทธิภาพดีที่สุดใน 4 โมเดลนี้ แล้วนำชุดข้อมูลทั้ง 4 มาใช้กับโมเดล nn4.small.v2 เพื่อเปรียบเทียบประสิทธิภาพการทำงานระหว่างชุดข้อมูลกับโมเดล

2.11.5 งานวิจัยเรื่อง “Real-Time Video Face Recognition for Embedded Devices” By Gabriel Costache, Sathish Mangapuram, Alexandru Drimbarean, Petronel Bigioi and Peter Corcoran [16]

งานวิจัยนี้นำเสนอการจดจำใบหน้าแบบเรียลไทม์บนอุปกรณ์ฝังตัว (Embedded platform) โดยในส่วนแรกนำเสนอประเด็นหลักที่จำเป็นต้องได้รับการแก้ไขเมื่อมีการออกแบบระบบและเสนอวิธีแก้ปัญหาที่เป็นไปได้ของการจดจำใบหน้า (General architecture of a Video face recognition system) และส่วนที่สองคือการอธิบายถึงวิธีการแก้ปัญหาการทำงานโดยใช้คุณสมบัติ Local Binary Patterns (LBP) ที่สามารถใช้ระบุตัวตนซึ่งสามารถคำนวณได้อย่างรวดเร็ว มีความทนทานต่อการเปลี่ยนแปลงรูปแบบต่าง ๆ และสามารถดึงข้อมูลที่เป็นประโยชน์ออกมาจากส่วนใบหน้า เพื่อให้ได้ระบบการจดจำที่มีประสิทธิภาพ ซึ่งระบบจะดึงเฉพาะคุณสมบัติที่มีค่าเท่ากันในหลาย ๆ รูปแบบของบุคคลเดียวกันเพื่อเพิ่มความแม่นยำของระบบ ผลลัพธ์ที่ได้สำหรับระบบนี้คือได้รับการทดสอบและนำไปใช้งานบนแพลตฟอร์มแบบฝังตัว ซึ่งได้ความแม่นยำที่ดีในข้อมูลอินพุตและพารามิเตอร์ที่มีความหลากหลาย ซึ่งตรงกับเงื่อนไขสำหรับการประมวลผลการจดจำใบหน้าแบบเรียลไทม์



รูปที่ 15 : รูปแสดง Architecture ของระบบ VFR system *(ที่มา : https://www.intechopen.com/books/new-approaches-to-characterization-and-recognition-of-faces/real-time-video-face-recognition-for-embedded-devices, 2020)*

บทที่ 3
วิธีการดำเนินโครงงาน

3.1 วิธีการดำเนินงาน

3.1.1 ประชุมและวางแผนการดำเนินงาน

3.1.1.1 วางแผนจัดทำโครงการ

3.1.1.2 หาข้อมูลศึกษาเกี่ยวกับแนวคิดและทฤษฎี

3.1.1.3 เลือกเครื่องมือ

3.1.1.4 ศึกษางานที่เกี่ยวข้อง

3.1.2 การวิเคราะห์ระบบ

3.1.2.1 ศึกษาและค้นคว้าโค้ดที่เกี่ยวกับการทำงานของ Face Recognition ด้วยรูปภาพโดยใช้ Pre-trained model ต่าง ๆ (VGG16, MobileNet, ResNet50)

3.1.2.2 ปรับปรุงแก้ไขข้อผิดพลาดของระบบ

3.1.3 พัฒนาระบบ

3.1.3.1 รวบรวมชุดข้อมูล (Dataset) เพื่อนำไปการเทรนโมเดลโดยใช้ Pre-trained model

3.1.3.2 นำ Pre-trained model ที่ได้จากการเทรนโมเดลใหม่มาใช้กับระบบรู้จำใบหน้า

3.1.3.3 พัฒนาระบบให้ถูกต้องและมีประสิทธิภาพเพียงพอสำหรับใช้งานได้จริง

3.1.4 สรุปและจัดทำเล่มรายงานวิจัย

3.2 แผนการดำเนินงานตลอดโครงงาน

ขั้นตอนการดำเนินงาน	เดือน			
	1	2	3	4
ประชุมและวางแผนการทำงาน				
1. วางแผนการทำงานและศึกษาทฤษฎีและแนวคิดที่ใช้ในงานวิจัย				
เก็บรวบรวมข้อมูล				

2. เก็บรวบรวมข้อมูล				
3. วิเคราะห์ข้อมูล				
การพัฒนาระบบ				
4. ออกแบบการทำงานของระบบ				
5. พัฒนาการทำงานของระบบ				
6. ทดสอบและปรับปรุงประสิทธิภาพการทำงานของระบบ				
สรุปผล				
7. สรุปผลและรายงานผลงานวิจัย				

ตารางที่ 2 : แสดงผลระยะเวลาของการดำเนินโครงการวิจัย

3.3 อุปกรณ์และเครื่องมือที่ใช้

3.3.1 ฮาร์ดแวร์

3.3.1.1 PC Processor: Intel(R) Core (TM) i7-6700HQ CPU @ 2.60GHz 2.59GHz RAM: 8.00 GB

3.3.2 ซอร์ฟแวร์

3.3.2.1 Jupyter Notebook

3.3.2.2 Google Colab

3.3.3 ภาษาที่ใช้

3.3.3.1 Python

3.4 ชุดข้อมูล (Dataset Set) ที่ใช้งาน

ชุดข้อมูลที่ใช้ในการทำงานวิจัย มีดังนี้

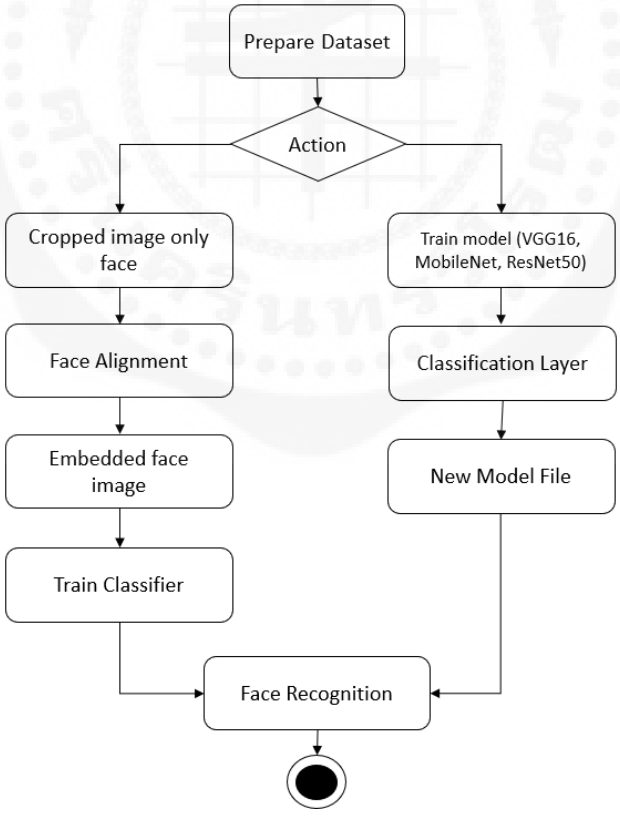
- LFW Dataset: มีจำนวนทั้งหมด 2,292 ภาพ 57 คน [17]

- Nisit ComSci Dataset: มีจำนวนทั้ง 631 ภาพ 49 คน
  รวมทั้งสิ้นจำนวนทั้งหมด 106 คน 2,923 ภาพ

3.5 แนวคิดที่ใช้ในการทำงานวิจัย

ศึกษาและค้นคว้าการทำ Transfer learning ที่มีการเรียกใช้ Pre-trained model ต่าง ๆ เพื่อนำมาประยุกต์ใช้ให้เหมาะสมกับระบบรู้จำใบหน้าและชุดข้อมูลที่ได้รวบรวมไว้ โดยจะเริ่มจากการ ทำความเข้าใจโมเดลต่าง ๆ และนำโมเดลนั้น ๆ มาทำการ Classification เพื่อเปรียบเทียบประสิทธิภาพ และค่าความถูกต้อง ของโมเดลกับชุดข้อมูล เพื่อใช้ในการตัดสินใจที่จะนำไปใช้ต่อในการทำ ระบบรู้จำใบหน้า เพื่อให้ระบบรู้จำใบหน้าได้ประสิทธิภาพและผลลัพธ์ในการตรวจจับหน้าที่มีความแม่นยำมากขึ้น โดย Pre-trained model ที่ได้นำมาทำการ Transfer learning เป็นโมเดลที่ได้รับความนิยมและเหมาะสำหรับการนำมาใช้งานกับระบบรู้จำใบหน้า ซึ่ง Pre-trained model ที่ได้พิจารณาสำหรับการเลือกนำมาใช้มี 3 โมเดล ดังนี้ VGG16, MobileNet, ResNet50

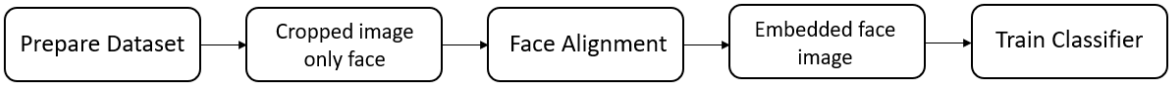
3.6 ขั้นตอนการทำงานของระบบ



รูปที่ 16: รูปตัวอย่างของขั้นตอนการทำงานของระบบ

ทางคณะผู้จัดทำได้แบ่งการทำงานของระบบหลัก ๆ ออกเป็น 2 ส่วนคือส่วนของการ Train Classifier และการทำ Pre-trained model โดยใช้วิธีการ Transfer learning

3.6.1 ในส่วนของการ Train Classifier



รูปที่ 17: รูปตัวอย่างของขั้นตอนการทำงานของ Train Classifier

3.6.1.1 Prepare dataset

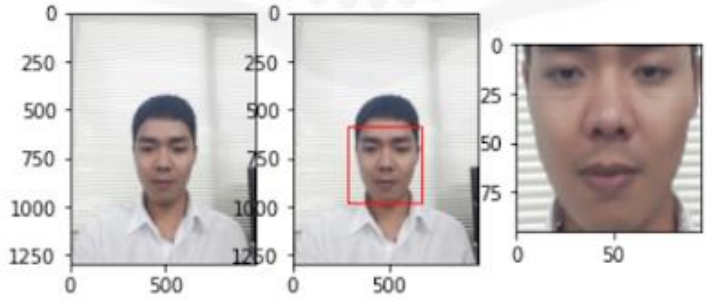
การจัดเตรียมชุดข้อมูล (Dataset) ทางกลุ่มได้ดาวน์โหลดชุดข้อมูลของ LFW แล้วนำมาใช้ในบางส่วน เนื่องจาก LFW มีจำนวนบุคคลในรูปภาพค่อนข้างมาก แต่บางบุคคลมีเพียง 1-2 รูป ทางกลุ่มนิสิตเลยคัดเลือกเพียงบางบุคคลที่นำมาใช้ในการTrain ครั้งนี้ รวมถึงรวบรวมชุดข้อมูลจากกลุ่มนิสิตคณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยศรีนครินทรวิโรฒ จำนวน 49 คน โดยใบหน้าที่เก็บข้อมูลจะเป็นใบหน้าที่ไม่มีสิ่งบดบังใบหน้า เช่น หน้ากากอนามัย หรือเครื่องประดับต่าง ๆ เป็นต้น

3.6.1.2 การ Crop รูปภาพเพื่อเอาใบหน้า

การ Crop รูปภาพเพื่อเอาเฉพาะใบหน้า เพื่อคัดเลือกเฉพาะส่วนของใบหน้าที่ Cropped ไว้ไปใช้ในการทำ Face Alignment

3.6.1.3 การทำ Face Alignment รูปภาพ

การทำ Face Alignment โดยการรับรูปภาพเข้ามาจากชุดข้อมูล (Dataset) โดย Face Alignment นี้จะโหลดไฟล์ landmark.dat เข้ามาเพื่อใช้สำหรับการหาจุดต่าง ๆ บนในส่วนของตาและจมูก เพื่อให้ทำการจัดแนวและจับใบหน้าได้



รูปที่ 18 : รูปตัวอย่างจากการหาจุด Landmark บนใบหน้าในการทำ Face Alignment

3.6.1.4 การ Embedded รูปภาพ

การ Embedded รูปภาพ ใช้ Pre-train model nn4.small2 ตัวโมเดลจะถูกเทรนมาด้วย LFW Dataset ซึ่งมีใบหน้ามากกว่า 1 แสนใบหน้า โดยตัว Model จะทำการแปลงรูปภาพใบหน้าที่ผ่านการ Croped

และ Alignment มาแล้วให้เป็น Vector ขนาด 128-dimensions โดยจะทำการ Embedded รูปภาพทั้งหมดในทุก ๆ Folder ของชุดข้อมูล แล้ว Embedded ของแต่ละบุคคลในแต่ละ Folder นั้น ๆ โดยจะเก็บค่าที่ได้ไว้เป็น Embedded array

activation_73 (Activation)	(None, 3, 3, 256)	0	inception_5b_1x1_bn[0][0]
concatenate_13 (Concatenate)	(None, 3, 3, 736)	0	activation_71[0][0] zero_padding2d_45[0][0] activation_73[0][0]
average_pooling2d_7 (AveragePoo	(None, 1, 1, 736)	0	concatenate_13[0][0]
flatten_1 (Flatten)	(None, 736)	0	average_pooling2d_7[0][0]
dense_layer (Dense)	(None, 128)	94336	flatten_1[0][0]
norm_layer (Lambda)	(None, 128)	0	dense_layer[0][0]

Total params: 3,743,280
Trainable params: 3,733,968
Non-trainable params: 9,312

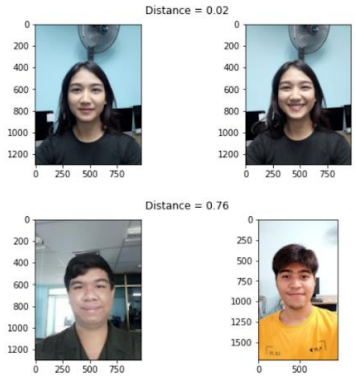
รูปที่ 19 : รูปตัวอย่างแสดงจำนวน Parameter ของ pre-trained model

```
print(embedded[i])
```

```
[ 0.09370779  0.23268628  0.02760166  0.09726511 -0.05658586  0.17924993
  0.09071765 -0.13949421 -0.03681837 -0.02455963  0.06572477 -0.04508472
  0.06159227 -0.07364009  0.04173109  0.02509906 -0.01985997  0.03657084
 -0.07230901 -0.07567628  0.0921608   0.01663779 -0.02783371  0.12809511
  0.07803781 -0.20700893 -0.14071229 -0.06643867  0.02954291  0.22289751
  0.0517538   0.052375   -0.09625575  0.09199942  0.04354984 -0.06472503
  0.05958683 -0.04607845  0.12316584  0.07551799 -0.04455547 -0.09579399
  0.09551688 -0.08538866  0.0313369  -0.07510354  0.09685358 -0.01149878
 -0.1053041   0.04511934  0.1599039   0.01044536  0.02941133  0.03807495
 -0.02655459  0.04787646  0.01131619  0.07251257  0.01627178 -0.01598059
 -0.07859755  0.05768196  0.02221816 -0.26679984  0.01947696  0.03384183
  0.0668691  -0.13526282 -0.25252974 -0.02658145 -0.01382246  0.06125072
 -0.08777971  0.03244327  0.03835078  0.00162256  0.0720126  -0.09542847
 -0.01074515  0.1631472   0.06240923 -0.07474183  0.02626797  0.01074197
 -0.04590464  0.1104816  -0.0846516  -0.23973855 -0.0902402   0.06190149
 -0.03801677 -0.1909555   0.02326789  0.02766829 -0.09368092 -0.00766715
 -0.04755592  0.05102276  0.04377303  0.08667275  0.04769826  0.0896568
 -0.04031251  0.20955978 -0.02627812  0.03036739  0.01614038  0.10780289
 -0.0690776   0.01435107 -0.03443241 -0.03477282 -0.02892669 -0.00035077
 -0.04532235  0.08123384 -0.02921794  0.16837855 -0.03242794  0.10482179
  0.02932264 -0.07926483 -0.01865693  0.01209966  0.04922369  0.06132053
 -0.01455353 -0.04321218]
```

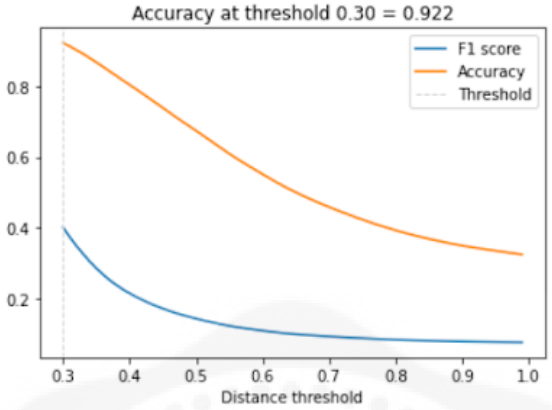
รูปที่ 20 : รูปตัวอย่างลักษณะข้อมูลรูปภาพที่ถูก Embedded ขนาด 128-dimensions

หลังจากนั้นจะทำการเปรียบเทียบรูปภาพ 2 รูป เพื่อหาค่า Distance ความต่างของแต่ละบุคคลนั้น ๆ ที่นำมาเทียบ โดยหาได้จากการนำค่า Embedded vector ของรูปภาพมาลบกันเพื่อหาผลลัพธ์ค่าความต่างระหว่าง 2 รูป



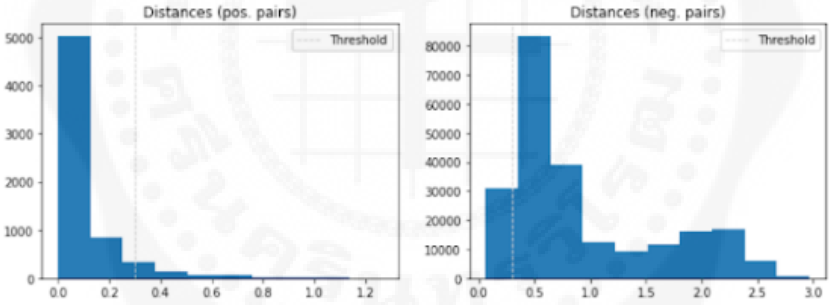
รูปที่ 21 : รูปตัวอย่างจากเปรียบเทียบการหาค่า Distance ความต่าง

เมื่อได้ค่า Distance ความต่างแล้ว จะนำค่า Embedded vector ของรูปภาพที่อยู่ในชุดข้อมูลมาคำนวณหาค่าความถูกต้อง (Accuracy) โดยจะหาจากค่า F1-Score ซึ่งคือค่าเฉลี่ยระหว่าง Precision และ Recall โดยจะปรับค่า Hyperparameter อยู่ที่ 0.3 และได้ค่าความถูกต้องอยู่ที่ 0.922

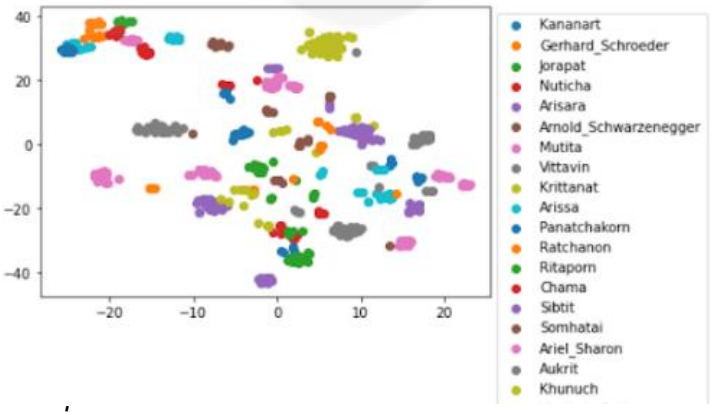


รูปที่ 22 : รูปตัวอย่างจากการเปรียบเทียบค่าความถูกต้อง กับ F1-Score

จากนั้นจะแบ่งเกณฑ์ความต่างของรูปภาพบุคคลจากค่า Distance โดยได้ผลสรุปออมาว่า ถ้าเป็นบุคคลคนเดียวกันส่วนมากค่าจะอยู่ในช่วงน้อยกว่า 0.3 และถ้าเป็นคนละบุคคลกันส่วนมากค่าจะอยู่ในช่วงมากกว่า 0.3



รูปที่ 23 : รูปตัวอย่างการแบ่งเกณฑ์ความเหมือนของบุคคลจากค่า Distance



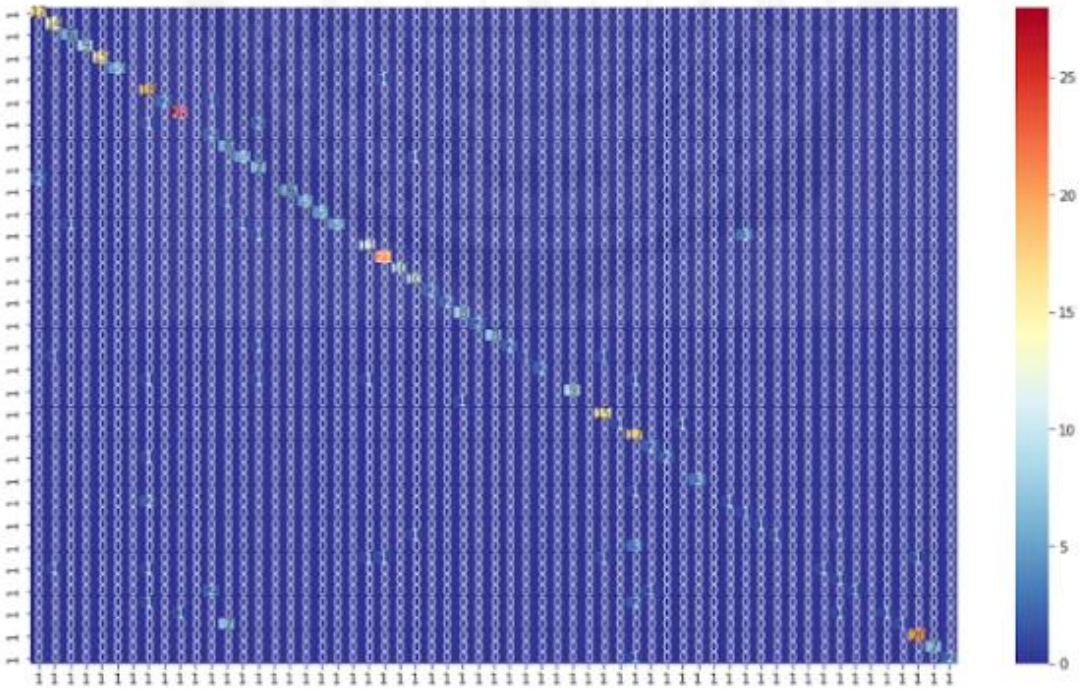
รูปที่ 24 : รูปตัวอย่างการ Plot การจับกลุ่มของแต่ละบุคคล

3.6.1.5 การ Train Classifier

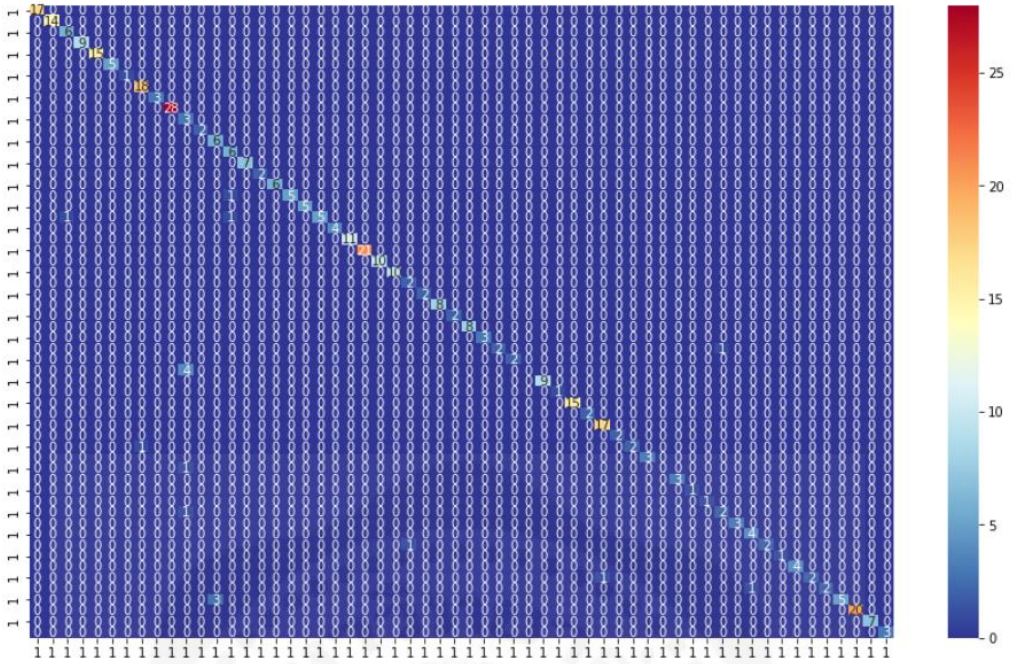
การ Train Classifier เนื่องจากต้องการที่จะแสดงประสิทธิภาพของโมเดล nn4.small2 จึงได้แบ่งชุดข้อมูลที่ใช้ในการเทรนแบบ Cross validation โดยกำหนด test_size = 0.55 และ กำหนด random_state = 42 และClassifier ที่ใช้ทดสอบโมเดล ก็คือ SVM (Support Vector Machine) และ KNN (K-Nearest Neighbor) โดยกลุ่มนิสิตต้องการใช้ Algorithm ที่มีค่า Accuracy ที่มากที่สุด จึงเลือกใช้ KNN (K-Nearest Neighbor) โดยการ Fit model ของ Embedded array กับชื่อบุคคล เพื่อให้ Classifier สามารถแบ่งกลุ่มของข้อมูลแต่ละบุคคลได้

Classifier	Accuracy	Precision	Recall	F1-Score	Support
SVM	0.89	0.73	0.69	0.67	376
KNN (n_neighbors = 1)	0.86	0.92	0.92	0.91	376
KNN (n_neighbors = 3)	0.90	0.92	0.90	0.91	376
KNN (n_neighbors = 5)	0.79	0.63	0.67	0.62	376

ตารางที่ 3 : แสดงผลการเปรียบเทียบ Classifier



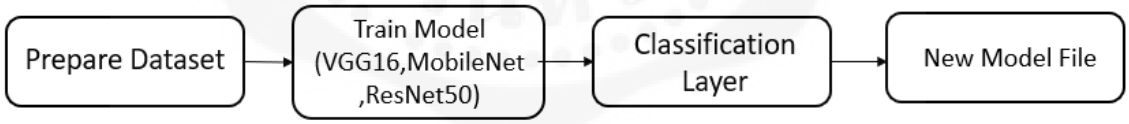
รูปที่ 25 : รูปตัวอย่าง Confusion matrix ของ Classifier SVM



รูปที่ 26 : รูปตัวอย่าง Confusion matrix ของ Classifier KNN; k=3

จากการทำ Confusion matrix เปรียบเทียบชุดข้อมูลระหว่างบุคคล กับบุคคล ซึ่งถ้าเป็นบุคคลเดียวกันจะได้ค่าในตาราง Grid ที่มากกว่าบุคคลที่ต่างกัน ซึ่งจะเห็นได้ว่าการกระจายตัวของค่าในตาราง Grid ของ SVM มีการกระจายตัวมากกว่า KNN เพราะค่าความถูกต้องของ KNN (k=3) มีค่ามากกว่า SVM

3.6.2 ในส่วนของการทำ Pre-trained model โดยใช้วิธีการ Transfer learning



รูปที่ 27: รูปตัวอย่างของขั้นตอนการทำงานของการทำ Classification layer

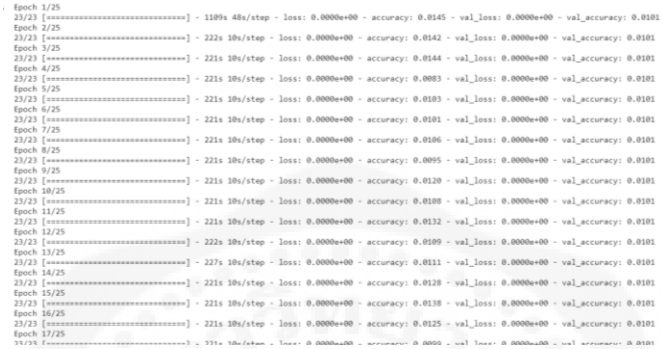
3.6.2.1 Prepare dataset

การจัดเตรียมชุดข้อมูล (Dataset) ทางกลุ่มได้ดาวน์โหลดชุดข้อมูลของ LFW แล้วนำมาใช้ในบางส่วน เนื่องจาก LFW มีจำนวนบุคคลในรูปภาพค่อนข้างมาก แต่บางบุคคลมีเพียง 1-2 รูป ทางกลุ่มนิสิตเลยคัดเลือกเพียงบางบุคคลที่นำมาใช้ในการTrain ครั้งนี้ รวมถึงรวบรวมชุดข้อมูลจากกลุ่มนิสิตคณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยศรีนครินทรวิโรฒ จำนวน 49 คน โดยใบหน้าที่เก็บข้อมูลจะเป็นใบหน้าที่ไม่มีสิ่งบดบังใบหน้า เช่น หน้ากากอนามัย หรือเครื่องประดับต่าง ๆ เป็นต้น หลังจากนั้นนำชุดข้อมูล (Dataset)

ทั้งหมดที่ได้ทำการรวบรวมมา แบ่งกลุ่มของชุดข้อมูล (Dataset) เป็น Test 30% และ Train 70% ของจำนวนรูปภาพทั้งหมด

3.6.2.2 การ Train Model

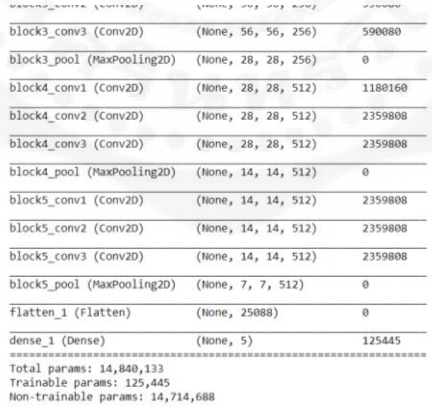
โดยทางคณะผู้จัดทำได้เลือก Pre-trained model มา 3 ตัว เพื่อเปรียบเทียบค่าความถูกต้องสำหรับการเลือกนำมาใช้กับระบบรู้จำใบหน้า



รูปที่ 28 : รูปตัวอย่างการ Train model

3.6.2.3 การทำ Classification Layer

เมื่อทำการเทรนโมเดลเป็นที่เรียบร้อยแล้ว ก็ทำการปรับแต่งโมเดลด้วยการทำ Classification layer โดยจะทำการ freeze ตัว Base model ของ Pre-trained model ไว้แล้วทำการเพิ่มเลเยอร์สุดท้ายเข้าไปเทรนใหม่ เพื่อให้ได้โมเดลตัวใหม่ออกมาจากการเพิ่มชุดข้อมูล (Dataset) ที่ได้รวบรวมไว้เข้าไปเทรนโมเดลใหม่



รูปที่ 29 : รูปตัวอย่างผลลัพธ์จากการทำ Classification layer

3.7 ปัญหาและอุปสรรค

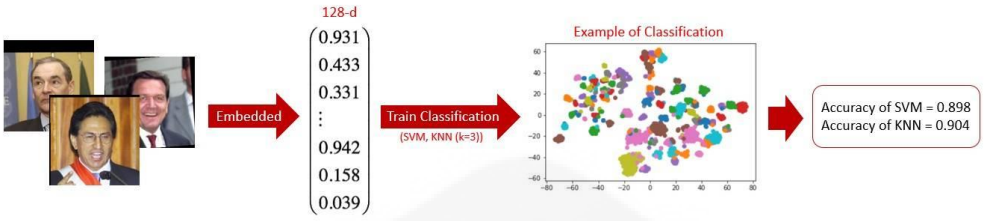
3.7.1 ใช้ระยะในการเทรนโมเดลค่อนข้างนานเนื่องจากมีการเพิ่มจำนวนชุดข้อมูลค่อนข้างมากเข้าไป

3.7.2 การรันไทม์บน Server มีปัญหาเกี่ยวกับ GPU บ่อยครั้งเมื่อรันหลายๆไฟล์พร้อมกัน

บทที่ 4

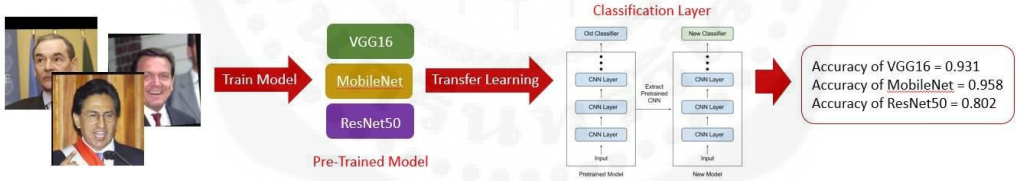
ผลการดำเนินโครงงาน

4.1 วิธีการทำงานของระบบ



รูปที่ 30 : รูปตัวอย่างการทำงานของระบบในส่วนของ Train Classifier

โดยอธิบายการทำงานของระบบในส่วนการ Train Classifier ได้ดังนี้ ทางคณะผู้จัดทำจะทำการเก็บรวบรวมชุดข้อมูล (Dataset) แล้วนำรูปภาพที่รวบรวมมาได้ทั้งหมดมาทำการ Embedding vector เมื่อได้แปลงรูปภาพเป็นข้อมูลตัวเลข 128-dimensions แล้ว ระบบจะทำการแบ่งกลุ่มข้อมูลรูปภาพโดยใช้อัลกอริทึม 2 ตัวที่นำมาทำการแบ่งกลุ่ม (Classification) คือ SVM และ KNN (k=3) และระบบจะทำการตรวจจับและระบุตัวตนใบหน้าออกมา จากชุดข้อมูลที่มีอยู่

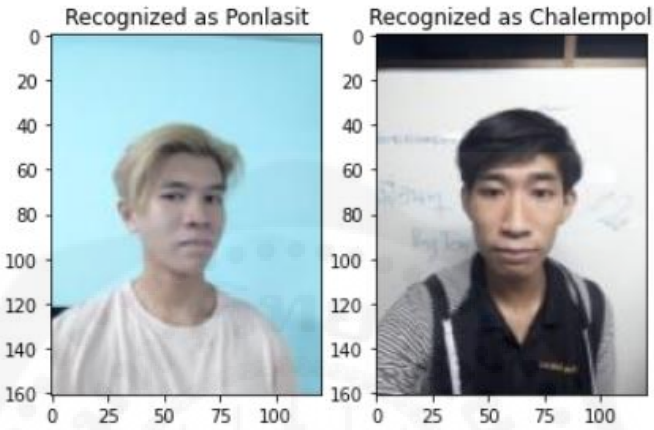


รูปที่ 31 : รูปตัวอย่างการทำงานของระบบจากการทำ Pre-trained model

โดยอธิบายการทำงานของระบบในส่วนการทำ Pre-trained model ได้ดังนี้ ทางคณะผู้จัดทำจะทำการเก็บรวบรวมชุดข้อมูล (Dataset) แล้วนำรูปภาพที่รวบรวมมาได้ทั้งหมดมาทำการเทรนโมเดลใหม่จากโมเดลที่มีอยู่แล้ว ซึ่งทางคณะผู้จัดทำได้ทำการเลือกโมแดลมา 3 โมเดล ได้แก่ VGG16, MobileNet, ResNet50 มาใช้ในการเทรนโมเดลใหม่กับชุดข้อมูล (Dataset) ที่ได้รวบรวมมา เพื่อทำการเปรียบเทียบประสิทธิภาพและค่าความถูกต้องของทั้ง 3 โมเดลและทำให้เลือกโมเดลที่ได้ค่าเหมาะสมที่สุดกับชุดข้อมูลที่ได้รวบรวมมา เพื่อที่จะให้ระบบสามารถทำการรู้จำใบหน้าต่อไป

4.2 ผลการดำเนินโครงการ

จากการดำเนินโครงงานวิจัย โดยการ Train Classifier โดยการแบ่งกลุ่มของบุคคลจากชุดข้อมูล (Dataset) ที่ได้รวบรวมไว้ สามารถระบุตัวตนบุคคลได้อย่างถูกต้อง และมีค่าถูกต้องที่ค่อนข้างสูงจากข้อมูลข้างต้นในข้อ 3.6.1.5 ซึ่งได้ผลลัพธ์ออกมาตามรูปตัวอย่างด้านล่างนี้



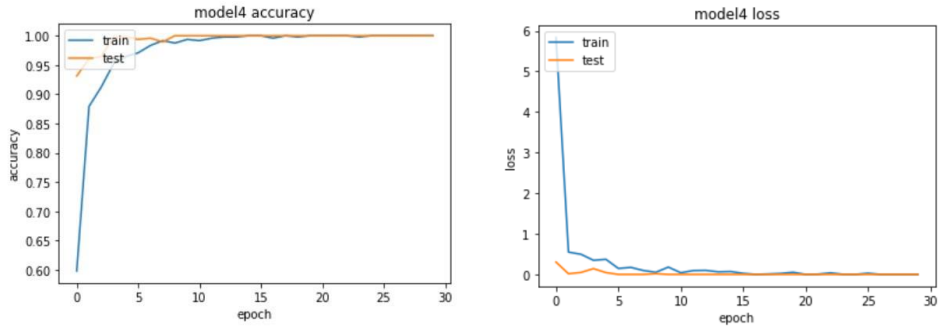
รูปที่ 32 : รูปตัวอย่างผลลัพธ์สำหรับ Face recognition จากการ Train Classifier

จากการดำเนินโครงงานวิจัย โดยการเปรียบเทียบ Pre-trained model จากการทำ Classification layer โดยการปรับค่าราเมเตอร์ และ ค่า Weight ของ Pre-trained model ที่นำมาใช้ เพื่อให้ได้ Class ของชุดข้อมูลออกมา และได้ค่าความถูกต้อง (Accuracy) ออกมาดังตารางด้านล่างนี้

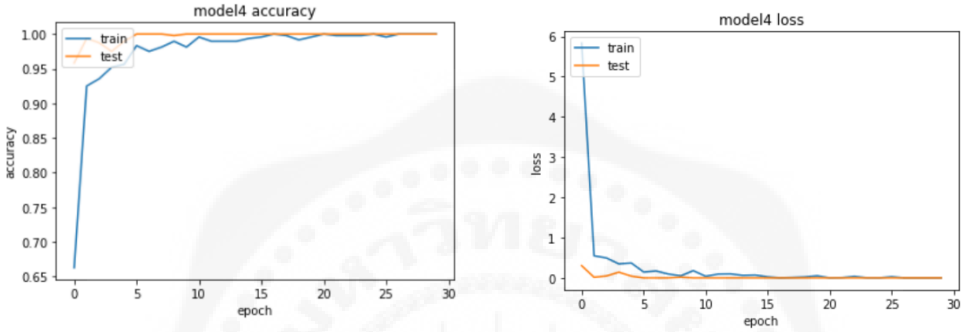
Model	Accuracy	Parameters
VGG16	0.9312	14,714,688
MobileNet	0.9583	3,228,864
ResNet50	0.8021	23,587,712

ตารางที่ 4 :การเปรียบเทียบ Pre-trained model จากการทำ Classification layer

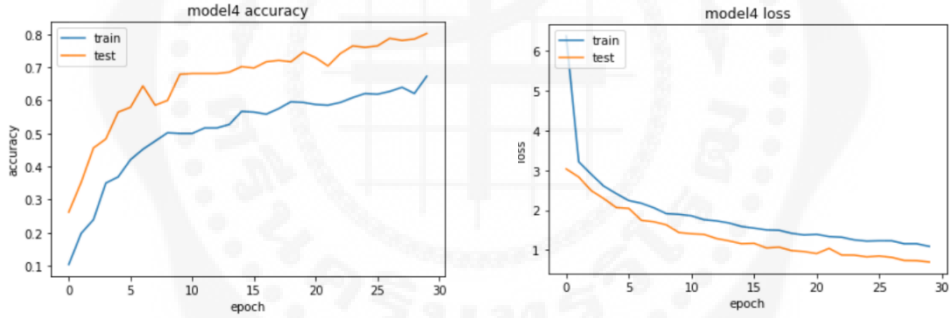
ตัวอย่างกราฟ Model accuracy และ Model loss สำหรับการเปรียบเทียบเบื้องต้นว่าโมเดลที่ Train แล้วมีประสิทธิภาพเป็นอย่างไร



รูปที่ 33 : รูปภาพตัวอย่าง Pre-trained model (VGG16)



รูปที่ 34 : รูปภาพตัวอย่าง Pre-trained model (MobileNet)



รูปที่ 35 : รูปภาพตัวอย่าง Pre-trained model (ResNet50)

จากการนำโมเดลที่สร้างมาใหม่ โดยใช้วิธีการของการทำ Transfer learning โดยทางคณะผู้จัดทำได้เลือกใช้ MobileNet สำหรับ Pre-trained model ที่นำเทรนมาเทรนใหม่กับชุดข้อมูล (Dataset) ที่ได้รวบรวมมา เนื่องจาก MobileNet เป็น Pre-trained model ที่ได้ค่าความถูกต้องมากถึง 95% จากการทำการ Classification layer และนำไปทำระบบรู้จำใบหน้าโดยได้ผลลัพธ์ ดังนี้

รูปที่ 36 : รูปตัวอย่างผลลัพธ์สำหรับ Face recognition จากการทำ Transfer learning

โดยทางคณะผู้จัดทำได้ทำการดาวน์โหลดวีดีโอตัวอย่างมาจากทาง Youtube จากนั้นระบบจะทำการ Face Recognition โดยจะตรวจจับใบหน้าในวีดีโอที่หน้าจอแสดงผลขึ้น เมื่อตรวจจับพบใบหน้าของบุคคลที่ไม่ได้อยู่ในชุดข้อมูล (Dataset) จะขึ้นผลลัพธ์มุมซ้ายของหน้าจอแสดงว่า “Unknow” และสำหรับบุคคลที่มีข้อมูลใบหน้าอยู่ในชุดข้อมูล (Dataset) ของเรา เมื่อตรวจจับพบใบหน้าของบุคคลระบบจะแสดงผลลัพธ์ด้านมุมซ้ายของหน้าจอแสดงผล เป็นชื่อของบุคคลนั้นๆ ออกมา

บทที่ 5
สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการศึกษาค้นคว้า

จากการศึกษาและเปรียบเทียบค่าความถูกต้องและประสิทธิภาพของโมเดลต่าง ๆ โดยใช้ Pre-trained model 3 โมเดล ดังนี้ VGG16, MobileNet, ResNet50 เพื่อศึกษาวิเคราะห์และหาค่าความถูกต้องเพื่อเปรียบเทียบ และทำความเข้าใจ เพื่อสามารถนำผลลัพธ์ที่ได้ไปปรับใช้ให้เหมาะสมกับชุดข้อมูล และวิธีการรู้จำใบหน้า เพื่อให้สามารถระบุตัวตนของบุคคลนั้นๆได้อย่างมีประสิทธิภาพและมีค่าความถูกต้องที่สูง

5.2 อภิปรายผล

จากการดำเนินการทำโครงงานเพื่อทำการศึกษาและเปรียบเทียบค่าความถูกต้องและประสิทธิภาพของโมเดลต่าง ๆ ที่เลือกนำมาใช้สำหรับระบบรู้จำใบหน้า Pre-trained model ที่เลือกมาใช้ ได้ค่าความถูกต้องดังนี้ VGG16 ได้ค่าความถูกต้องอยู่ที่ 0.9312, MobileNet ได้ค่าความถูกต้องอยู่ที่ 0.9583, ResNet50 ได้ค่าความถูกต้องอยู่ที่ 0.8021 ซึ่งสรุปผลได้ว่า MobileNet มีค่าความถูกต้องสูงที่สุดจาก 3 โมเดลที่เลือกนำมาใช้ และจากทั้ง 3 โมเดลที่เลือกนำมาศึกษาให้ค่าความถูกต้องค่อนข้างที่ดีสำหรับชุดข้อมูลที่นำใช้ระบบรู้จำใบหน้านี้

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

สำหรับชุดข้อมูล (Dataset) ควรเพื่อจำนวนรูปภาพในการ Train เพื่อจะทำให้มีประสิทธิภาพในการ Train ที่ดีมากขึ้นสำหรับการทำระบบรู้จำใบหน้า (Face recognition)

บรรณานุกรม

[1] S. Lee, “Understanding Face Detection with the Viola-Jones Object Detection Framework,” 2020. https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14 (accessed May 22, 2020).

[2] S. S. Farfade, M. Saberian, and L. J. Li, “Multi-view face detection using Deep convolutional neural networks,” *ICMR 2015 - Proc. 2015 ACM Int. Conf. Multimed. Retr.*, pp. 643–650, 2015, doi: 10.1145/2671188.2749408.

[3] Miw Nuntida, “Project การทำระบบ Automated Video Tagging โดยใช้ Face Recognition,” 2018. https://medium.com/@nuntida.s/thomson-reuters-intern-ai-project-ทำระบบ-automated-video-tagging-โดยใช้-face-recognition-955fb1f9fe6b (accessed May 22, 2020).

[4] “Deep Learning แบบฉบับคนสามัญชน EP 1 : Neural Network History,” 2019. https://medium.com/mmp-li/deep-learning-แบบฉบับคนสามัญชน-ep-1-neural-network-history-f7789236a9a3 (accessed May 15, 2020).

[5] “Applied Deep Learning - Part 4: Convolutional Neural Networks,” 2017. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 (accessed May 22, 2020).

[6] B. M. Nair, J. Foytik, R. Tompkins, Y. Diskin, T. Aspiras, and V. Asari, “Multi-pose face recognition and tracking system,” *Procedia Comput. Sci.*, vol. 6, pp. 381–386, 2011, doi: 10.1016/j.procs.2011.08.070.

[7] P. Wettayakorn, “FaceNet และ Triplet Loss ในการจดจำใบหน้า,” 2019. https://medium.com/datawiz-th/facenet-และ-triplet-loss-ในการจดจำใบหน้า-a523b71be9e1 (accessed May 22, 2020).

[8] Chaiyanan, “ซัพพอร์ตเวกเตอร์แมชชิน (Support Vector Machine: SVM),” 2018. https://knowledge.snru.ac.th/ซัพพอร์ตเวกเตอร์แมชชิน/ (accessed Aug. 27, 2020).

[9] Jaruwit Pratancheewin, “เรียนรู้และทำความเข้าใจเรื่อง Support Vector Machine (SVM) คืออะไร ,” 2019. https://www.glurgeek.com/education/support-vector-machine/ (accessed May 22, 2020).

[10] Nati Thaiyathum, “KNN หรือ K-Nearest Neighbors,” 2019. https://www.glurgeek.com/education/knn/ (accessed Oct. 25, 2020).

[11] W. Daroontham, “ทำ Image Classification/Transferred learning ด้วย Pytorch,” 2020. https://medium.com/datawiz-th/ทำ-image-classification-transferred-learning-จาก-google-image-บน-pytorch-f679f3a62f1e (accessed Jun. 20, 2020).

[12] Rattanachot Phwanwilai, “การทำ Deep Learning Transfer Learning,” 2019. https://medium.com/@605162020010/19-07-2562-หลังจากที่ผมหายไปนานติดอยู่หลายเรื่องทั้งเรื่องเรียน-ป-โท-498452e3fe84 (accessed Feb. 03, 2021).

[13] A. Gupta, “Face Recognition using Transfer Learning on MobileNet,” 2020. https://medium.com/analytics-vidhya/face-recognition-using-transfer-learning-on-mobilenet-cf632e25353e (accessed Feb. 24, 2021).

[14] PURVA HUILGOL, “Top 4 Pre-Trained Models for Image Classification with Python Code,” 2020. https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/ (accessed Feb. 28, 2021).

[15] Angeliki Fydanaki & Zeno Geradts, “Evaluating OpenFace: an open-source automatic facial comparison algorithm for forensics,” 2018. https://www.tandfonline.com/doi/full/10.1080/20961790.2018.1523703 (accessed Nov. 03, 2020).

[16] P. B. and P. C. Gabriel Costache, Sathish Mangapuram, Alexandru Drimbarean, “Real-Time Video Face Recognition for Embedded Devices,” 2011. https://www.intechopen.com/books/new-approaches-to-characterization-and-recognition-of-faces/real-time-video-face-recognition-for-embedded-devices (accessed Nov. 03, 2020).

[17] Jessica Li, “LFW-Dataset,” 2018. https://archive.org/details/lfw-dataset (accessed Jun. 20, 2020).