ระบบแจ้งเตือนการช่วยเหลือผู้ป่วยบนพื้นฐานการติดตามท่ามือและขั้นตอนวิธีเหมือนฮาร์ที่ถูกดัดแปร

นายธนภัทร รัชธร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

PATIENT AID NOTIFICATION SYSTEM BASED ON HAND GESTURE TRACKING AND MODIFIED HAAR-LIKE ALGORITHM

Mr. Tanapat Ratchatorn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering
Department of Electrical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2019

หัวข้อวิทยานิพนธ์	ระบบแจ้งเตือนการช่วยเหลือผู้ป่วยบนพื้นฐานการติดตามท่ามือและขั้นตอนวิธีเหมือนฮาร์ที่ถูกดัดแปร
โดย	นายธนภัทร รัชธร
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วันเฉลิม โปรา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.สัญญา มิตรเอม)

ธนภัทร รัชธร : ระบบแจ้งเตือนการช่วยเหลือผู้ป่วยบนพื้นฐานการติดตามท่ามือและขั้นตอนวิธีเหมือนฮาร์ที่ถูกดัดแปร. (PATIENT AID NOTIFICATION SYSTEM BASED ON HAND GESTURE TRACKING AND MODIFIED HAAR-LIKE ALGORITHM) อ.ที่ปรึกษาหลัก : ผศ. ดร.สุรีย์ พุ่มรินทร์

ในปัจจุบันการปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ (Human-Computer Interaction: HCI) ได้เข้ามามีบทบาทสำคัญอย่างมากในชีวิตประจำวันของเรา โดยหนึ่งในการประยุกต์ใช้ที่สำคัญคือการใช้งานด้านการแพทย์ งานวิจัยชิ้นนี้มุ่งเน้นไปที่การออกแบบระบบที่ช่วยให้ผู้ป่วยที่มีปัญหาด้วยการสื่อสารด้วยเสียงสามารถติดต่อกับผู้ดูแลได้โดยง่ายโดยใช้การแสดงท่าทางมือ ซึ่งได้ใช้อุปกรณ์เป็น Raspberry Pi ซึ่งเป็นอุปกรณ์ขนาดเล็กและราคาถูกเพื่อที่จะสามารถเข้าถึงผู้ใช้งานได้โดยง่าย สำหรับการทำงานของระบบนั้นได้ถูกแบ่งเป็น 2 ส่วนหลัก ในส่วนแรกระบบจะตรวจหาท่ากำมือเพื่อใช้เป็นสัญญาณในการเริ่มระบบแจ้งเตือน ซึ่งได้ใช้กระบวนวิธีแบบฮาร์สำหรับการตรวจจับ ผลการทดลองพบว่ามีความแม่นยำในการตรวจจับสูงโดยมีค่า F1-Score อยู่ที่ 0.991 ส่วนที่สองระบบจะรับภาพที่ได้จากการตรวจจับในตอนแรกเพื่อกำหนดพื้นที่และทำการแบ่งส่วนพื้นผิวมนุษย์ (Human Skin Segmentation) เพื่อลดความซับซ้อนของข้อมูลภาพ ก่อนที่จะส่งภาพไปยังโครงข่ายประสาทแบบคอนโวลูชั่นเพื่อจำแนกท่าทางนิ้วมือ 1 - 5 นิ้ว โดยในงานวิจัยนี้ได้ใช้สถาปัตยกรรมแบบ MobileNetV2 ที่มีความเร็วในการประมวลผลสูงสำหรับการจำแนก และได้ใช้ภาพมือขาว/ดำจำนวน 19,000 ภาพสำหรับการฝึกสอนโมเดล โดยผลลัพธ์จากการจำแนกท่ามือมีความถูกต้องมากกว่า 96% ผลลัพธ์ที่ได้จากการจำแนกท่ามือจะถูกแปลเป็นข้อความตามที่ได้กำหนดไว้และส่งไปยังแอพพลิเคชั่น LINE ของผู้ดูแลต่อไป วิธีการนี้ช่วยให้ผู้ป่วยสามารถสื่อสารกับผู้ดูแลได้โดยง่ายและยังช่วยลดความตึงเครียดของผู้ดูเลเนื่องจากไม่จำเป็นต้องอยู่กับผู้ป่วยตลอดเวลา

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต .....

ปีการศึกษา 2562 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

5970181621 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: Patient Care, Computer Vision, Haar-Like Feature, convolutional neural network (CNN), MobileNetV2, Monitoring System

Tanapat Ratchatorn : PATIENT AID NOTIFICATION SYSTEM BASED ON HAND GESTURE TRACKING AND MODIFIED HAAR-LIKE ALGORITHM. Advisor: Asst. Prof. SUREE PUMRIN, Ph.D.

Human-Computer Interaction (HCI) has an important role in our everyday lives. One of the important applications is using in the medical field. This research aims to create a monitoring system that help patients with speaking problem to communicate with their caregiver with an ease using hand gesture. Raspberry Pi is used as a hardware due to its small size and low price which make it easier to use for the patient. The algorithm contains 2 main phases. The first phase, system detects patient's fist as a signal to turn on a notification system. Haar-like features are applied as a detection process. The result shows high accuracy with F1-Score = 0.991. For the second phase, the image from phase 1 is used to limit region of interest and segment human skin to reduce complexity of the image. The segmented image is used as an input for Convolutional Neural Network to classify hand gesture from 1 to 5 fingers. MobileNetV2 architecture has been chosen for this work due to its low latency. The model is trained using 19,000 binary images of hand gesture as dataset. The experimental results show more than 96% accuracy. The labels from classification are then translated to messages and send to caregiver's LINE application. This method allows patients to communicate with their caregiver easier and reduces the stress of caregiver since it is not necessary for them to stay with the patients all the time.

Field of Study: Electrical Engineering Student's Signature .....

Academic Year: 2019 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีด้วยด้วยความช่วยเหลือของ ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์ ที่คอยให้คำแนะนำ ปรึกษา สนับสนุนในการทำวิจัย ตลอดจนการสอนให้ความรู้ในการด้านต่าง ๆ ตลอดการเรียนในระดับปริญญาโท

ขอขอบคุณ รองศาสตราจารย์ ดร.วันเฉลิม โปรา ทั้งในฐานประธานกรรมการสอบวิทยานิพนธ์และในฐานะอาจารย์ที่เป็นผู้สอน สำหรับความรู้และมุมมองต่าง ๆ ในทางวิศกรรม

ขอขอบคุณ รองศาสตราจารย์ ดร.สัญญา มิตรเอม ผู้ทรงคุณวุฒิภายนอกมหาวิทยาลัย สำหรับทัศนะและคำแนะนำในการปรับปรุงวิทยานิพนธ์

ขอขอบคุณเพื่อน ๆ ที่คอยช่วยเหลือ เป็นกำลังใจและให้คำปรึกษาในด้านต่าง ๆ พร้อมอยู่เคียงข้างกันในวันเวลาที่ประสบปัญหา

ขอขอบคุณจุฬาลงกรณ์มหาวิทยาลัย สถาบันการศึกษาอันเป็นที่รักและผูกพันธ์ยิ่ง

สุดท้ายนี้ขอขอบคุณบิดา มารดา และครอบครัว สำหรับความรัก ความอบอุ่น และการสนับสนุนในทุก ๆ เรื่องมาโดยตลอด

ธนภัทร รัชธร

สารบัญ

หน้า

สารบัญรูปภาพ

หน้า

สารบัญตาราง

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมา

ในยุคสมัยปัจจุบันนี้ เทคโนโลยีได้กลายมาเป็นส่วนหนึ่งในชีวิตประจำวันของมนุษย์ หนึ่งในตัวอย่างที่สำคัญก็คือ การปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ (Human-Computer Interaction: *HCI*) ซึ่งเป็นเทคโนโลยีที่ได้รับความสนใจนักวิจัยมาหลายทศวรรษ การปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์นั้น เป็นการผนวกความองค์ความรู้หลายแขนงไว้ด้วยกัน เช่น วิทยาการคอมพิวเตอร์ สรีระวิทยา พฤติกรรมศาสตร์ จิตวิทยา และการออกแบบ เป็นต้น โดยในปัจจุบันการปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ได้ขยายวงกว้างอย่างแพร่หลายและถูกใช้งานในหลาย ๆ ด้าน อาทิ อุปกรณ์ให้ความบันเทิง อุปกรณ์อำนวยความสะดวกในครัวเรือน ระบบจัดการภัยพิบัติ ระบบช่วยเหลือ และระบบสุขภาพและการแพทย์ [1-3]

ในด้านเทคโนโลยีทางการแพทย์อย่างเช่นระบบดูแลสุขภาพและดูแลผู้ป่วย การปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ได้เข้ามามีส่วนสำคัญในการลดจำนวนอุบัติเหตุที่เกิดกับผู้ป่วย/ผู้สูงอายุ ด้วยการประยุกต์ใช้ระบบเฝ้าระวังที่ช่วยติดตามการเคลื่อนไหวและอารมณ์ของผู้ป่วย/ผู้สูงอายุ [4] ซึ่งระบบเฝ้าระวังผู้ป่วยนั้นช่วยให้ผู้ดูแลสามารถดูแลผู้ป่วยได้โดยไม่มีความจำเป็นต้องอยู่กับผู้ป่วยตลอดเวลา ซึ่งเป็นการลดภาระและลดความเครียดของผู้ดูแลได้ นอกจากนี้ยังสามารถช่วยอำนวยความสะดวกให้ผู้ป่วยสามารถสื่อสารกับผู้ดูแลได้ง่ายขึ้น

วัตถุประสงค์หลักของวิทยานิพนธ์ฉบับนี้มุ่งเน้นไปที่การช่วยเหลือผู้ป่วย/ผู้สูงอายุที่มีปัญหาในการสื่อสารด้วยการพูดให้สามารถติดต่อสื่อสารกับผู้ดูแลได้ง่ายขึ้นโดยการอาศัยท่าทางมือ หลักการทำงานของงานวิจัยนี้แบ่งเป็น 2 ส่วนหลักได้แก่ การตรวจหาภาพมือด้วยเทคนิคการเรียนรู้ของเครื่องจักร (Machine Learning) ด้วยหลักการทำงานของ Viola-Jones และการจำแนกท่าทางมือด้วยเทคนิคโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN) ซึ่งท่าทางมือแต่ละท่าจะถูกตั้งค่าข้อความตามที่

กำหนดไว้เพื่อสื่อสารให้ผู้ดูแลทราบได้โดยง่าย และข้อความดังกล่าวจะถูกส่งไปยังโทรศัพท์เคลื่อนที่ของผู้ดูแล

1.2 วัตถุประสงค์ของงานวิจัย

1. สร้างระบบแจ้งเตือนช่วยเหลือผู้ป่วยบนพื้นฐานวิสัยทัศน์แบบทันที (Real Time) ด้วยขั้นตอนวิธีเหมือนฮาร์ ร่วมกับเทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน
2. พัฒนาระบบให้สามารถใช้งานได้ในภาพที่ฉากหลังมีความซับซ้อนหรือมีการเคลื่อนไหว

1.3 ขอบเขตของงานวิจัย

1. ระบบทำงานบนอุปกรณ์ Raspberry Pi และรับภาพจากโมดูลกล้อง Raspberry Pi Camera
2. ระบบจำแนกและแปรผลท่าทางมือตามข้อความที่กำหนดไว้และส่งข้อความแจ้งเตือนไปยังแอพพลิเคชั่น LINE ของผู้ดูแล
3. ระบบสามารถทำงานกับภาพที่ฉากหลังซับซ้อนหรือเคลื่อนไหวได้ แต่ฉากหลังไม่ควรมีรูปร่าง/สีสันคล้ายคลึงกับมือมนุษย์มากเกินไป

1.4 ประโยชน์จากวิจัย

1. ช่วยให้ผู้ป่วย/ผู้สูงอายุที่มีปัญหาด้านการพูดสามารถสื่อสารกับผู้ดูแลได้โดยง่าย
2. ช่วยลดภาระและความเครียดของผู้แล

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวของ

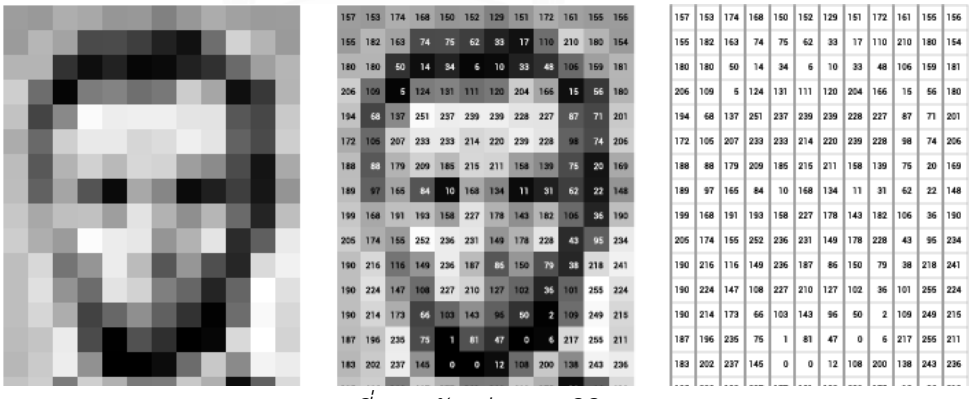
งานวิจัยชิ้นนี้อาศัยหลักการและทฤษฎีที่เกี่ยวข้องหลายส่วนซึ่งสามารถแบ่งออกได้เป็น 3 หัวข้อหลักดังนี้

1. การประมวลผลภาพ (Image Processing)
2. การตรวจจับภาพมือ (Hand Detection)
3. โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN)

2.1 การประมวลผลภาพ (Image Processing)

2.1.1 พื้นฐานรูปภาพดิจิตอล (Digital Image Basics)

รูปภาพดิจิตอล (Digital Image) ประกอบด้วยกลุ่มของพิกเซลรวมตัวกัน โดยพิกเซล (Pixel) คือองค์ประกอบพื้นฐานที่เล็กที่สุดในภาพดิจิตอลและไม่สามารถแบ่งย่อยได้อีก หากแบ่งภาพออกเป็นตารางย่อย ตารางย่อยที่เล็กที่สุดนั้นจะนับเป็น 1 พิกเซล โดยค่าของพิกเซลแต่ละพิกเซลคือค่าที่แทนสีหรือความเข้มของแสงที่ปรากฏในตารางย่อยนั้น [5]



รูปที่ 2.1 ตัวอย่างภาพดิจิตอล

ที่มา: https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

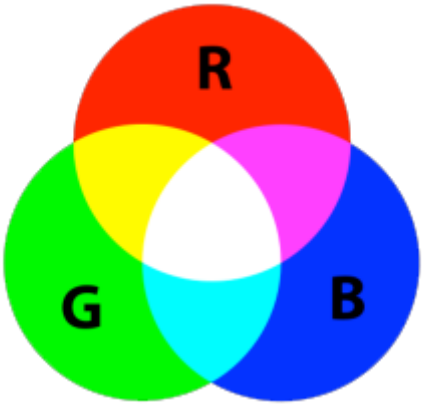
พิจารณารูปที่ 2.1 เป็นตัวอย่างของภาพที่มีความละเอียด 12 พิกเซลในแนวนอน และ 15 พิกเซลในแนวตั้ง ภาพดังกล่าวจึงมีจำนวนพิกเซลเท่ากับ 12 x 15 ซึ่งได้เท่ากับ 180 พิกเซล โดยในรูปตัวอย่างนี้เป็นภาพระดับเทา (Grayscale) ซึ่งมีค่าความสว่างของแต่ละพิกเซลอยู่ในช่วง 0 – 255 โดยค่า 0 หมายถึงความสว่างต่ำสุด และค่า 255 หมายถึงค่าความสว่างสูงสุด

ทั้งนี้ในการจัดเก็บและแสดงผลภาพดิจิตอลนั้นมีหลายประเภทขึ้นอยู่กับหมวดสีของภาพ ซึ่งในงานวิจัยชิ้นนี้มีการดำเนินการกับรูปภาพทั้งหมด 4 หมวดสีได้แก่ หมวดสี RGB หมวดสี YCbCr หมวดสีระดับเทา และหมวดสีขาว/ดำ (Binary)

2.1.2 หมวดสี RGB (RGB Color Mode)

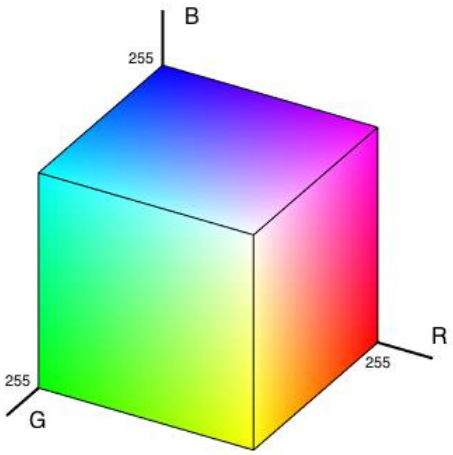
RGB เป็นหนึ่งในหมวดสีที่ได้รับความนิยมในการใช้งาน พิกเซลในหมวดสี RGB จะประกอบด้วยตัวเลขชุด 3 ตัว โดยตัวเลขแต่ละตำแหน่งค่าความเข้มของแสงในช่องแสงสีแดง (Red: R) สีเขียว (Green: G) และสีน้ำเงิน (Blue: B) ตามลำดับ โดยแต่ละช่องสีจะมีค่าได้ตั้งแต่ 0 – 255

การแสดงผลในหมวดสี RGB เป็นการคำนวนแบบรวม (Additive) เปรียบเหมือนการนำแสงในแต่ละช่องสีมาซ้อนกัน โดยค่าสีที่มีค่ามากจะยิ่งมีความสว่างมากและเข้าใกล้สีขาวมากขึ้น รูปที่ 2.2 แสดงให้เห็นการผสมของสีในช่องสี RGB จะเห็นว่าเมื่อรวมแสงสีแดงเข้ากับสีเขียวจะได้สีเหลือง เมื่อรวมแสงสีแดงเข้ากับสีน้ำเงินจะได้สีชมพู เมื่อรวมแสงสีน้ำเงินเข้ากับสีเขียวจะได้สีฟ้า และเมื่อรวมแสงทั้ง 3 สีเข้าด้วยกันจะได้เป็นแสงสีขาว



รูปที่ 2.2 ภาพการรวมกันขององค์ประกอบสี

หมวดสี RGB ยังสามารถแสงให้เห็นในรูปแบบลูกบาศก์ โดยแต่ละแกนของลูกบาศก์จะแทนค่าความเข้มของแสงสีแดง เขียว และน้ำเงิน ซึ่งเมื่อรวมกันทั้ง 3 ช่องสี จะสามารถให้สีที่มีความแตกต่างกันได้ถึง 256 x 256 x 256 เท่ากับ 16,777,216 สี ดังแสดงในรูปที่ 2.3 [6]



รูปที่ 2.3 ลูกบาศก์สี RGB

ที่มา: https://www.c4dcafe.com/ipb/forums/topic/92173-how-does-one-make-a-rgb-gradient-that-acts-like-uv-map/

2.1.3 หมวดสี YCbCr (YCbCr Color Mode)

หมวดสี YCbCr เป็นหมวดสีในระบบภาพดิจิตอลวีดีโอที่นิยมใช้อย่างแพร่หลายในงานวีดีโอดิจิตอล เช่น การบีบอัดไฟล์ชนิด MPEG ในระบบดีวีดี ดิจิตอลทีวี การส่งสัญญาณผ่านพอร์ตดิจิตอลของกล้องถ่ายภาพ เป็นต้น

คำว่า YCbCr ตัวอักษร Y แทนค่าความสว่าง (Luminance) ตัวอักษร Cb คือค่าสีน้ำเงินที่ลบ Luminance ออกไป (B-Y) และตัวอักษร Cr คือค่าสีแดงที่ตัด Luminance ออกไป (R-Y)

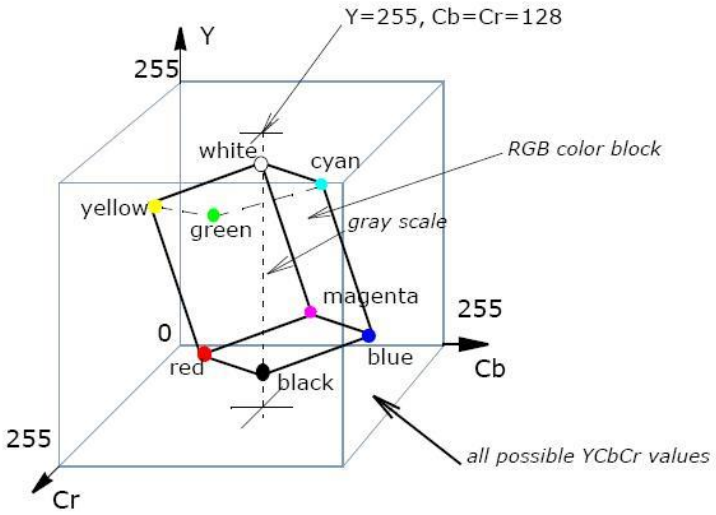
ค่าสี YCbCr สามารถแปลงไปเป็นหมวดสี RGB ได้ดังสมการที่ 2.1 – 2.3

$$Y' = 0.257*R' + 0.504*G' + 0.098*B' + 16 \quad (2.1)$$

$$Cb' = -0.148*R' - 0.291*G' + 0.439*B' + 128 \quad (2.2)$$

$$Cr' = 0.439*R' - 0.368*G' - 0.071*B' + 128 \quad (2.3)$$

สำหรับงานวิจัยนี้ ได้ใช้หมวดสี YCbCr ในขั้นตอนการแบ่งส่วนพื้นผิวมนุษย์ (Human Skin Segmentation) เนื่องจากมีผลการทดลองจากงานวิจัยและบทความหลายชิ้นได้พบว่าหมวดสี YCbCr เป็นหมวดสีที่มีประสิทธิภาพสำหรับการแบ่งส่วนพื้นผิวมนุษย์ออกจากส่วนอื่นของภาพ [7-9]



รูปที่ 2.4 ลูกบาศก์แสดงหมวดสี YCbCr เทียบกับ RGB

ที่มา: https://software.intel.com/en-us/node/503873

2.1.4 หมวดสีระดับเทา (Grayscale Color Mode)

ภาพระดับเทา เป็นภาพที่มีค่าช่องสีเพียงค่าเดียว โดยจะอยู่ระหว่าง 0 – 255 ซึ่งค่าสี 0 หมายถึงความสว่างต่ำสุด (สีดำ) และค่าความสว่างจะค่อย ๆ เพิ่มขึ้นตามลำดับ จนถึงค่า 255 หมายถึงค่าความสว่างสูงสุด (สีขาว)

ภาพระดับเทาสามารถแสดงข้อมูลได้น้อยกว่าภาพที่มี 3 ช่องสีอย่าง RGB หรือ YCbCr แต่มีข้อดีที่มีความซับซ้อนน้อยกว่า ทำให้สามารถประมวลผลได้รวดเร็วกว่า ซึ่งเป็นประโยชน์อย่างมากสำหรับการใช้งานที่เน้นความสำคัญของความรวดเร็วในการประมวลผล เนื่องจากภาพระดับเทาจะคำนวณเฉพาะค่าความสว่างเท่านั้น

รูปที่ 2.5 การไล่เฉดความเข้มของภาพระดับเทาจากระดับความสว่างต่ำสุด (0) ไปจนถึงระดับความสว่างสูงสุด (255)

2.1.5 หมวดสีขาว/ดำ (Binary Color Mode)

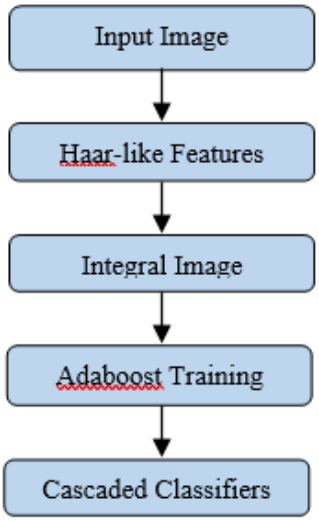
หมวดสีขาว/ดำหรือไบนารี เป็นหมวดสีที่มีค่าช่องสีเพียงช่องเดียวเหมือนกันภาพระดับเทา แต่จะมีค่าแสงได้เพียงแค่ 2 ค่าคือ 0 (สีดำ) และ 1 (สีขาว) เท่านั้น จึงทำให้หมวดสีขาว/ดำมีความซับซ้อนน้อยที่สุดและใช้เนื้อที่จัดเก็บน้อยที่สุดเพียง 1 บิตต่อพิกเซล

การแปลงภาพจากหมวดสีอื่นมาเป็นภาพขาว/ดำนั้น สามารถทำได้โดยง่ายโดยการตั้งค่าขอบเขต (Threshold) ซึ่งหาพิกเซลใดมีค่ามากกว่าขอบเขตที่กำหนดเมื่อแปลงเป็นภาพขาว/ดำแล้ว พิกเซลนั้นจะมีค่าเป็น 1 (สีขาว) แต่หากพิกเซลใดมีค่าต่ำกว่าค่าขอบเขตเมื่อแปลงเป็นภาพขาว/ดำแล้ว พิกเซลนั้นจะมีค่าเป็น 0 (สีดำ) [10]

รูปที่ 2.6 ตัวอย่างผลลัพธ์การแปลงภาพจากภาพระดับเทา (ขวา) เป็นภาพแบบขาว/ดำ (ซ้าย)

2.2 การตรวจจับภาพมือ (Hand Detection)

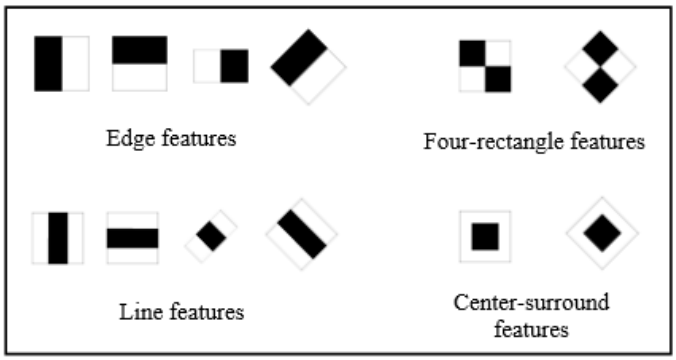
ในงานวิจัยนี้ ได้อาศัยการตรวจจับภาพมือโดยใช้ขั้นตอนวิธีของ Paul Viola และ Michael Jones ที่ได้เสนอไว้ในปี 2001 [11] ซึ่งประกอบด้วย 4 ส่วนหลักได้แก่ คุณลักษณะแบบฮาร์ ภาพอินทิกรัล ขั้นตอนวิธีแบบเอดาบูสท์ และการจำแนกแบบลำดับขั้น



รูปที่ 2.7 แผนภูมิแสดงการทำงานตามขั้นตอนวิธีของ Viola-Jones

2.2.1 คุณลักษณะแบบฮาร์ (Haar-Like Features)

เทคนิคการตรวจจับวัตถุในภาพบนพื้นฐานคุณลักษณะแบบฮาร์ ถูกเสนอขึ้นเป็นครั้งแรกในปี 2001 โดย Paul Viola และ Michael Jones โดยคุณลักษณะแบบฮาร์นั้น คือการหาผลต่างระหว่างผลรวมของค่าความสว่างในส่วนที่เป็นสีขาวและสีดำภายในกรอบที่สนใจ



รูปที่ 2.8 ตัวอย่างคุณลักษณะแบบฮาร์

ซึ่งค่าผลต่างนี้จะถูกคำนวณและเปรียบเทียบกับค่าขอบเขต (Threshold) ที่กำหนดเพื่อใช้ในการจำแนกวัตถุ [11-13]

อย่างไรก็ตาม ในการคำนวณผลต่างระหว่างผลรวมของค่าความสว่างในส่วนที่เป็นสีขาวและสีดำภายในกรอบที่สนใจนั้น เป็นการคำนวณที่กินทรัพยากรประมวลผลและใช้เวลามากเนื่องจากจำนวนคุณลักษณะแบบฮาร์นั้นมีจำนวนมากและต้องคำนวณทุกจุดพิกเซลในกรอบที่สนใจไล่ไปจนครบทุกส่วนของภาพ ดังนั้นจึงจำเป็นต้องอาศัยเทคนิคการคำนวณภาพอินทิกรัลเข้ามาช่วยเพื่อลดเวลาในการประมวลผล

2.2.2 ภาพอินทิกรัล (Integral Image)

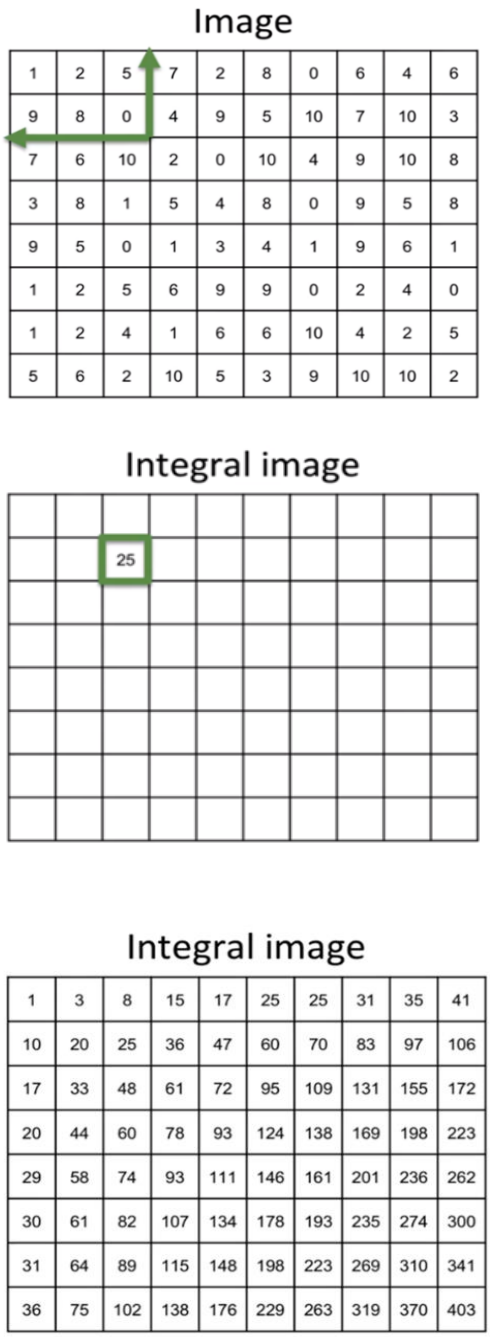
ภาพอินทิกรัล หรือที่รู้จักในอีกชื่อหนึ่งว่า ตารางผลรวมภายในพื้นที่ เป็นวิธีที่มีประสิทธิภาพและรวดเร็วในการคำนวณผลรวมของค่าความสว่างของพิกเซลภายในพื้นที่สี่เหลี่ยมที่กำหนด

ยกตัวอย่างจากรูปที่ 2.9 หากต้องการคำนวณผลรวมภายในพื้นที่สี่เหลี่ยมที่สนใจซึ่งมี 12 ค่า จะต้องทำการบวกตัวเลขทั้งหมด 11 ครั้ง ซึ่งจำนวนครั้งของการบวกนี้จะยิ่งมากขึ้นถ้าหากภาพมีขนาดใหญ่ ซึ่งในการใช้งานจริงภาพส่วนมากมักมีขนาดหลักแสน – หลักล้านพิกเซล และการคำนวณผลรวมในพื้นที่นี้ก็ต้องคำนวณหลายครั้งเนื่องจากต้องคำนวณหาคุณลักษณะแบบฮาร์จำนวนมากและคำนวณในหลายพื้นที่ตลอดทั้งภาพ ส่งผลให้การคำนวณใช้เวลามาก

1	2	5	7	2	8	0	6	4	6
9	8	0	4	9	5	10	7	10	3
7	6	10	2	0	10	4	9	10	8
3	8	1	5	4	8	0	9	5	8
9	5	0	1	3	4	1	9	6	1
1	2	5	6	9	9	0	2	4	0
1	2	4	1	6	6	10	4	2	5
5	6	2	10	5	3	9	10	10	2

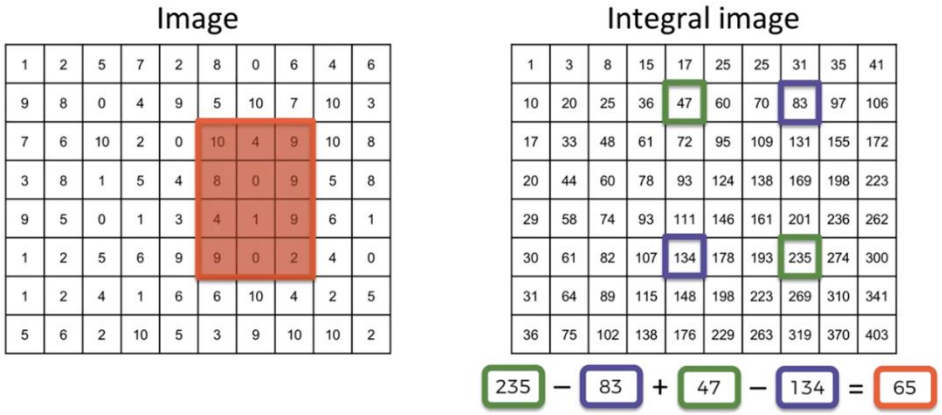
รูปที่ 2.9 การหาผลรวมในพื้นที่สี่เหลี่ยมที่สนใจ

การใช้ภาพอินทิกรัลเข้ามาช่วยกัน เริ่มต้นโดยการสร้างภาพอินทิกรัลที่เกิดจากการคำนวณผลรวมของค่าในพิกเซลที่อยู่ด้านบนและด้านซ้ายของจุดที่สนใจ โดยคำนวณไปทีละพิกเซลจนครบทั้งภาพดังแสดงในรูปที่ 2.10



รูปที่ 2.10 การคำนวณสร้างภาพอินทิกรัล

ในรูปที่ 2.11 ด้วยหลักการของภาพอินทิกรัล หากสังเกตดูจะพบว่าค่าผลรวมในพื้นที่พื้นที่สีแดงนั้น จริง ๆ แล้วก็คือค่าอินทิกรัลที่พิกเซลมุมขวาล่างของสี่เหลี่ยมหักลบด้วยค่าอินทิกรัลที่พิกเซลมุมขวาบนและซ้ายล่าง แล้วจึงบวกด้วยค่าอินทิกรัลที่พิกเซลซ้ายบน จะเห็นว่าด้วยวิธีการใช้ภาพอินทิกรัลเข้ามาช่วยนั้นทำให้สามารถคำนวณผลรวมของความสว่างของพิกเซลในพื้นที่ได้ด้วยการใช้การบวกแค่ 3 ครั้งเท่านั้น [14]



รูปที่ 2.11 การคำนวณผลรวมในพื้นที่สี่เหลี่ยมด้วยการใช้ภาพอินทิกรัล

การคำนวณผลรวมในพื้นที่สี่เหลี่ยมด้วยการใช้ภาพอินทิกรัล สามารถเขียนสรุปเป็นสมการได้ดังนี้

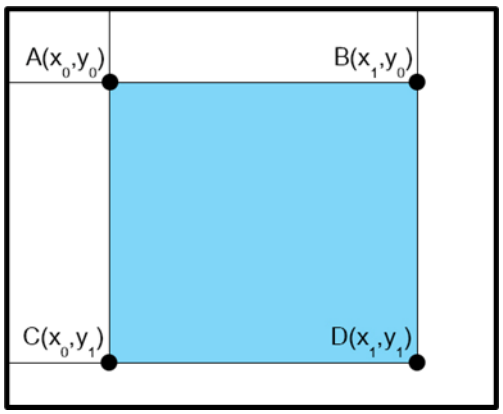
$$I(x,y) = \sum_{x' \le x,\ y' \le y} i(x',y') \tag{2.4}$$

โดย *I(x,y)* คือค่าการคำนวณภาพอินอิกรัล ณ ตำแหน่งพิกัด (x, y)

i(x,y) คือค่าของพิกเซล ณ ตำแหน่งพิกัด (x, y)

เมื่อคำนวณได้ภาพอินทิกรัลแล้ว เราสามารถใช้ผลลัพธ์ดังกล่าวมาคำนวณผลรวมในพื้นที่สี่เหลี่ยมใด ๆ ที่สนใจได้ดังนี้

$$\sum_{x_0 < x \le x_1,\ y_0 < y \le y_1} i(x,y) = I(D) + I(A) - I(B) - I(C) \tag{2.5}$$



รูปที่ 2.12 ภาพประกอบแสดงการคำนวณผลรวมในพื้นที่สี่เหลี่ยมโดยใช้ภาพอินทิกรัล

โดยจุด A, B, C และ D คือตำแหน่งพิกเซลซ้ายบน ขวาบน ซ้ายล่างและขวาล่างของสี่เหลี่ยมตามลำดับ (ดูรูปที่ 2.12 ประกอบ)

2.2.3 ขั้นตอนวิธีแบบเอดาบูสท์ (Adaboost Algorithm)

ขั้นตอนวิธีแบบเอดาบูสท์ (Adaboost หรือ Adaptive Boosting) เป็นเทคนิคการเรียนรู้ของเครื่องจักรที่ถูกเสนอโดย Freund และ Schapire ในปี 1997 [15] ซึ่งมีหลักการทำงานดังนี้

- นำเข้าภาพตัวอย่าง $(x_1, y_1),\ldots, (x_m, y_m)$ โดยที่ $x_i \in X$ และ y_i มีค่าเป็น -1 หรือ +1 หมายความว่า หากภาพอยู่ใน Positive Sample จะมีค่า y เป็น 1 และหากภาพอยู่ใน Negative Sample จะมีค่า y เป็น -1
- กำหนดค่าน้ำหนักเริ่มต้น $D_1(i) = 1/m$ โดยที่ $i = 1,\ldots, m$ โดยค่าน้ำหนักเริ่มต้นจะเท่ากันทั้งหมดสำหรับทุกภาพตัวอย่าง
- ในรอบการเรียนรู้ที่ $t = 1,\ldots, T$
  - สร้างตัวจำแนกแบบอ่อน (Weak Classifier) โดยสำหรับทุกภาพตัวอย่างโดยใช้ค่าน้ำหนัก D_t
  - แต่ละรอบการเรียนรู้ จะมีการเพิ่มน้ำหนักของภาพที่ทายผิด เพื่อที่จะบังคับให้ตัวจำแนกแบบอ่อนเรียนรู้ภาพที่ทายผิด
  - ตัวจำแนกแบบอ่อน ถูกสร้างด้วย Decision Stump $h_t : X \longrightarrow \{-1,+1\}$

- เลือกตัวจำแนกแบบอ่อน h_t ที่มีค่าความผิดพลาด ε_t ต่ำที่สุด โดย

$$\varepsilon_t = Pr_{i\sim D_t}[h_t(x_i) \neq y_i] \quad (2.6)$$

โดยที่ $h_t(x_i) \neq y_i$ หมายความว่าเราสนใจเฉพาะภาพที่ค่าจากตัวจำแนกไม่ตรงกับค่าจริง (ภาพที่ทายผิด)

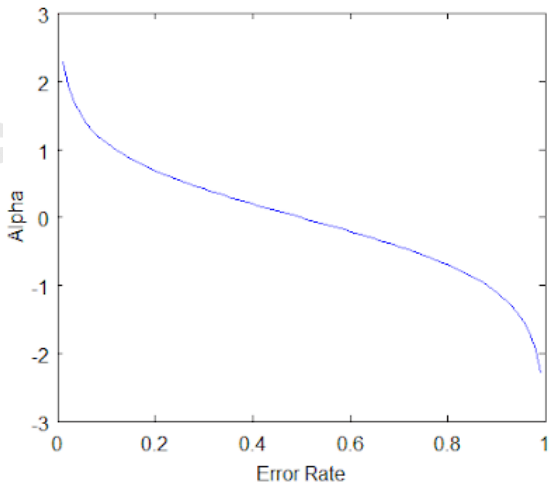
- คำนวณค่า Voting

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (2.7)$$

ของตัวจำแนกแบบอ่อนในรอบนั้น ๆ ซึ่งจากสมการ 2.7 เราสามารถสรุปความสัมพันธ์ระหว่างค่า Voting (α_t) และค่าความผิดพลาด (ε_t) ได้คร่าว ๆ ดังนี้

- ค่าความผิดพลาดเข้าใกล้ 0 ⟶ ค่า Voting จะเพิ่มขึ้นแบบ exponential
- ค่าความผิดพลาดเท่ากับ 0.5 ⟶ ค่า Voting จะเท่ากับ 0
- ค่าความผิดพลาดเข้าใกล้ 1 ⟶ ค่า Voting จะเพิ่มขึ้นแบบ exponential ในทิศเป็นลบ

ซึ่งสามารถเขียนออกมาเป็นกราฟความสัมพันธ์ได้ดังรูปที่ 2.13



รูปที่ 2.13 กราฟแสดงความสัมพันธ์ระหว่าง Voting (α_t) และค่าความผิดพลาด (ε_t)

- ปรับค่าน้ำหนักสำหรับแต่ละภาพตัวอย่าง

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad (2.8)$$

ซึ่งจากสมการ 2.8 จะเห็นว่าหากค่าแบ่งประเภทจริง (y_i) และค่าจากตัวจำแนกแบบอ่อน ($h_t(x_i)$) ตรงกัน จะส่งผลให้ค่าน้ำหนักลดลง (ค่า exponential น้อยกว่า 1) แต่หากค่าแบ่งประเภทจริง (y_i) และค่าจากตัวจำแนกแบบอ่อน ($h_t(x_i)$) ไม่ตรงกัน จะส่งผลให้ค่าน้ำหนักลดลง (ค่า exponential เป็นค่าเศษส่วนน้อยกว่า 1)

- จากสมการ 2.8 ค่า Z_t คือค่า Normalization Factor เพื่อทำให้ผลรวมของค่าน้ำหนักทั้งหมดในรอบถัดไปเป็น 1

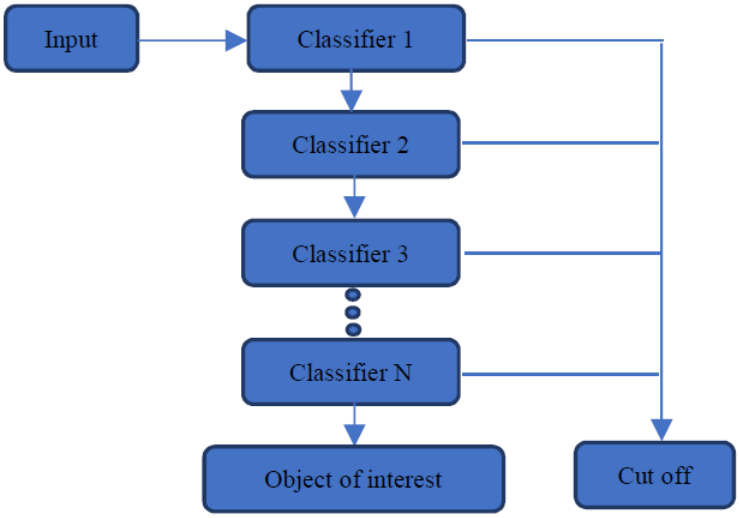
- ทำซ้ำกระบวนการในข้อที่ผ่านมาไปเรื่อย ๆ จนถึงจำนวนรอบ T ที่กำหนด
- ในขั้นตอนสุดท้าย เราจะได้ตัวจำแนกแบบแข็งแกร่ง (Strong Classifier) ที่เกิดจากผลลัพธ์การบวกของผลคูณระหว่างค่า Voting และค่าตัวจำแนกแบบอ่อนในแต่ละรอบดังสมการที่ 2.9 โดยเลือกสนใจเฉพาะค่าเครื่องหมายบวก/ลบของผลลัพธ์เท่านั้น

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \quad (2.9)$$

2.2.4 การจำแนกแบบลำดับขั้น (Cascaded Classifier)

ในการจำแนกแบบลำดับขั้น เริ่มต้นโดยการกำหนดพื้นที่หน้าต่างย่อยที่สนใจใน ภาพแล้วตรวจหาคุณลักษณะที่สนใจตามลำดับ หากในหน้าต่างพื้นย่อยที่ดังกล่าวตรวจไม่ พ บ คุณลักษณะที่สนใจก็จะทำการตัดหน้าต่างพื้นที่นั้นทิ้งไป และไปพิจารณาพื้นที่อื่นต่อไป แ ต่หากในพื้นที่ดังกล่าวตรวจเจอคุณลักษณะที่สนใจ ก็จะทำการตรวจหาคุณลักษณอื่นด้วย ตั ว จำแนกลำดับถัดไป โดยดำเนินการตามขั้นตอนนี้ซ้ำไปเรื่อย ๆ จนถึงตัวจำแนกลำดับสุดท้าย หากหน้าต่างพื้นที่ดังกล่าวสามารถผ่านตัวจำแนกลำดับสุดท้ายได้ก็จะถือว่าตรวจพบวัตถุที่สนใจในพื้นที่ดังกล่าว

การจำแนกแบบลำดับขั้นนี้ ช่วยลดเวลาที่ใช้ในการตรวจหาวัตถุได้เนื่องจากสามารถตัดพื้นที่หน้าต่างย่อยออกไปได้โดยไม่จำเป็นต้องผ่านตัวจำแนกทุกตัว



รูปที่ 2.14 การทำงานของตัวจำแนกแบบลำดับขั้น

2.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network)

2.3.1 การเรียนรู้เชิงลึก (Deep Learning)

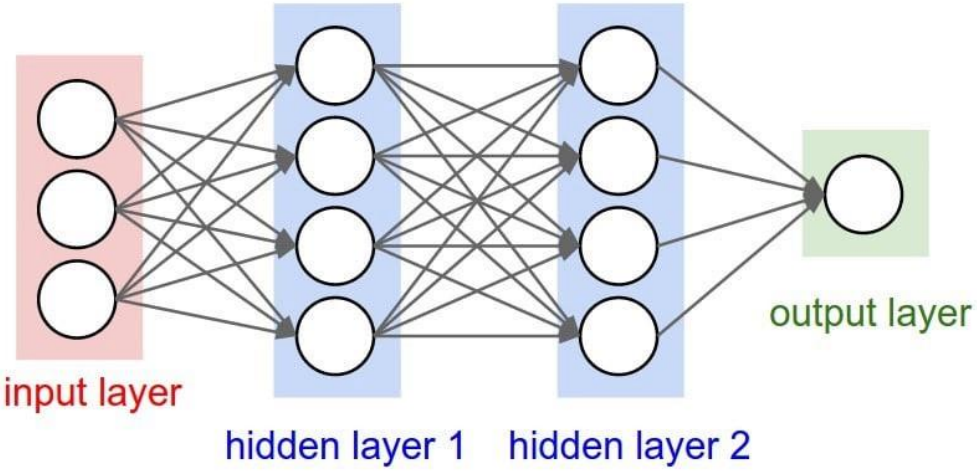
การเรียนรู้เชิงลึกหรือ Deep Learning เป็นสาขาวิชาหนึ่งในกลุ่มเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence: AI) ซึ่งกำลังเป็นที่นิยมและได้รับความสนใจมากในยุคปัจจุบัน โดยการเรียนรู้เชิงลึกถือเป็นแขนงหนึ่งในเทคโนโลยีการเรียนรู้ของเครื่องจักร (Machine Learning) ที่มีการใช้โครงข่ายประสาทเทียม (Artificial Neural Network: ANN) เข้ามาช่วยในการทำงาน

2.3.2 โครงข่ายประสาทเทียม (Artificial Neural Network)

โครงข่ายประสาทเทียม (Artificial Neural Network: ANN) เป็นรูปแบบหนึ่งของการเรียนรู้ของเครื่องจักรที่เรียนรู้จากข้อมูลและรูปแบบการรู้จำ โครงข่ายประสาทเทียมมีแนวความคิดมาจากการทำงานของระบบประสาทของมนุษย์ โดยลอกเลียนแบบการติดต่อ

สื่อสารระหว่างเซลล์ประสาทแต่ละเซลล์กับเซลล์อื่น แนวความคิดนี้ได้รับการเสนอครั้งแรกตั้งแต่ปี 1943 โดย McCulloch และ Pitts [16-19] และผ่านการต่อยอดมาจนถึงปัจจุบัน

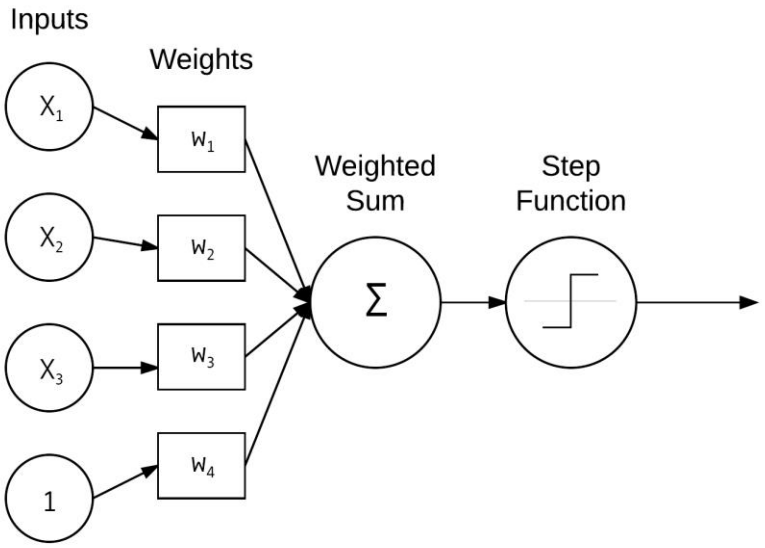
ในระบบโครงข่ายประสาทเทียมประกอบด้วย Node จำนวนมากที่ทำหน้าที่เปรียบเสมือนเซลล์ประสาท โดยมีการแบ่งเป็นชั้น ๆ และแต่ละ Node จะมีการส่งผ่านข้อมูลไปยังชั้นถัดไปเรื่อย ๆ โดยมีค่าน้ำหนักกำกับไว้



รูปที่ 2.15 ตัวอย่างโครงสร้างของโครงข่ายประสาทเทียม

ที่มา: https://mc.ai/a-to-z-about-artificial-neural-networks-ann-theory-n-hands-on

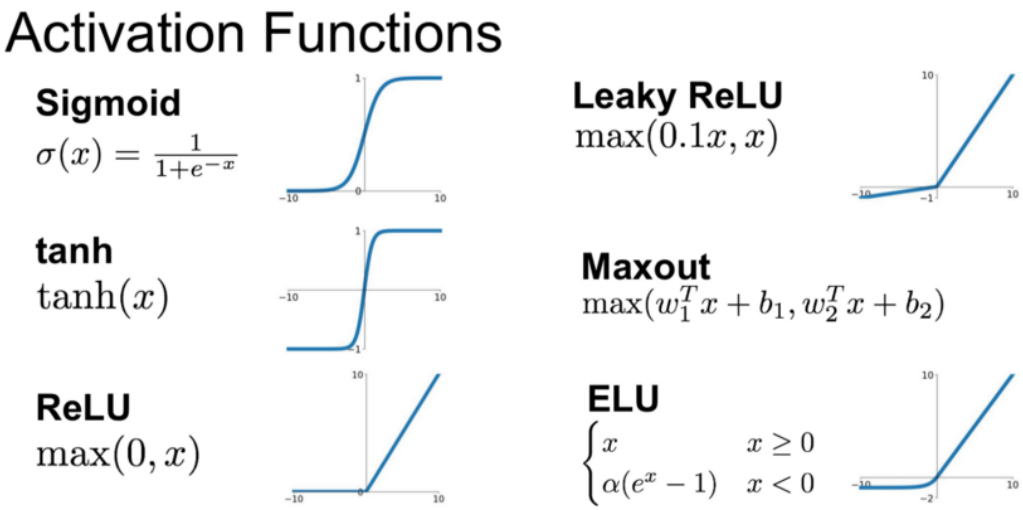
รูปที่ 2.15 เป็นการแสดงตัวอย่างโครงสร้างของระบบโครงข่ายประสาทเทียม ซึ่งเริ่มจากการรับข้อมูลเข้ามาในชั้นขาเข้า (Input Layer) ก่อนที่ข้อมูลจะถูกส่งต่อไปประมวลผลในชั้นซ่อนตัว (Hidden Layer) ซึ่งเป็นชั้นที่อยู่ระหว่างชั้นขาเข้าและขั้นขาออก (Output Layer) โดยชั้นซ่อนตัวอาจมีได้ตั้งแต่ 1 ชั้นไปจนถึงหลายชั้น และในการส่งต่อข้อมูลแต่ละครั้งจะมีการให้ค่าน้ำหนักกำกับไว้ หากค่าน้ำหนักมีค่ามากก็หมายถึงข้อมูลจาก Node นั้นมีอิทธิพลต่อผลลัพธ์สูง แต่หาก Node ไหนที่มีค่าน้ำหนักต่ำหมายถึงข้อมูลนั้นมีอิทธิพลต่อผลลัพธ์น้อย



รูปที่ 2.16 การทำงานของเซลล์ประสาทในโครงข่ายประสาทเทียม

ที่มา: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

จากรูปที่ 2.16 จะเห็นว่าข้อมูลขาเข้า x แต่ละตัว จะมีการเชื่อมต่อกับเซลล์ประสาทในชั้นต่อไปโดยผ่านการคูณค่าน้ำหนัก w หลังจากนั้นจึงรวมผลลัพธ์ทั้งหมดเข้าด้วยกันเป็นผลรวมถ่วงน้ำหนัก (Weighted Sum) ซึ่งจะถูกดำเนินการด้วยฟังก์ชั่นกระตุ้น (Activation Function) ต่อไปก่อนที่ข้อมูลจะถูกส่งไปยัง Node ถัดไป โดยฟังก์ชั่นกระตุ้นคือฟังก์ชั่นที่เป็นตัวกำหนดผลลัพธ์ของ Node นั้น ๆ ซึ่งในแต่ละชั้นอาจมีฟังก์ชั่นกระตุ้นที่แตกต่างกันได้ โดยฟังก์ชั่นกระตุ้นที่นิยมใช้มีหลายประเภทดังรูปที่ 2.17 แต่ในปัจจุบันฟังก์ชั่นที่ได้รับความนิยมสูงที่สุดคือ ReLu [20]

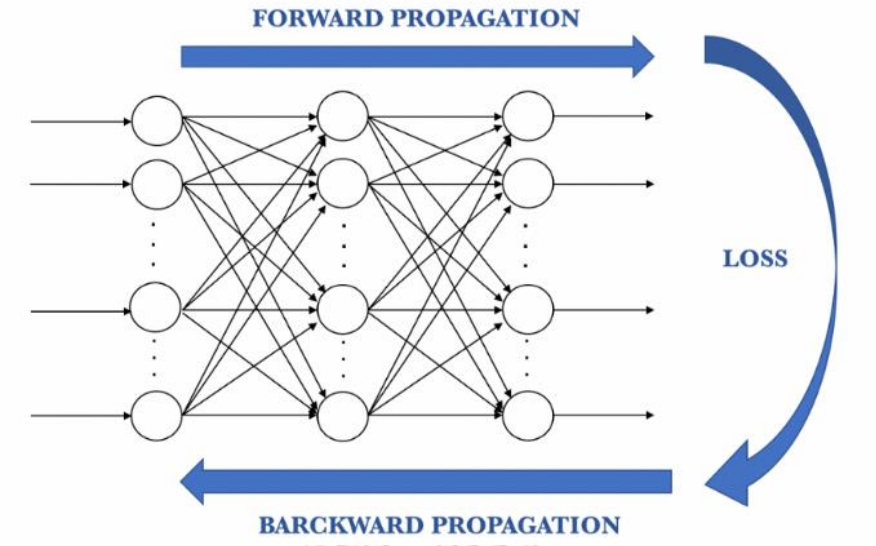


รูปที่ 2.17 ตัวอย่างฟังก์ชั่นกระตุ้นแบบต่าง ๆ

ที่มา: https://medium.com/@krishnakalyan3/introduction-to-exponential-linear-unit-d3e2904b366c

ในขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียม เมื่อข้อมูลขาเข้าถูกส่งผ่านและคำนวณผ่าน Node ในโครงข่ายประสาทเทียมแล้ว สุดท้ายข้อมูลทั้งหมดจะไปรวมกันที่ชั้นขาออกซึ่งจะประกอบด้วย Node เพียง 1 Node ที่ทำหน้าที่รวบรวมข้อมูลและสรุปออกมาเป็นผลลัพธ์ ขั้นตอนเหล่านี้รวมเรียกว่า “การส่งผ่านไปข้างหน้า (Forward Pass)” ซึ่งผลลัพธ์ที่ได้มานั้นปกติแล้วจะยังไม่ได้ตรงกับค่าจริงทั้งหมดจึงจำเป็นต้องอาศัยกระบวนการ “ส่งผ่านย้อนหลัง (Backward Pass)” เข้ามาช่วย

กระบวนการส่งผ่านย้อนหลัง (Backward Pass) จะคำนวณค่า Cost Function ซึ่งเป็นค่าบ่งบอกถึงความแตกต่างระหว่างค่าผลลัพธ์ที่ทำนายได้และค่าจริง (ซึ่งอาจมีได้หลายรูปแบบ) จากนั้นจึงส่งผ่านค่าย้อนกลับไปยังโครงข่ายประสาทเทียมและทำการปรับค่าน้ำหนักของแต่ละ Node โดยมีเป้าหมายคือลดค่า Cost Function ลงให้ต่ำที่สุด โดยจะปรับเฉพาะค่าน้ำหนักเท่านั้น (ไม่มีการปรับค่าข้อมูลขาเข้า) กระบวนนี้จะทำทำสลับกับกระบวนการส่งผ่านไปข้างหน้าและทำซ้ำทั้ง 2 กระบวนการเพื่อปรับค่าน้ำหนักในโครงข่ายประสาทเทียมไปเรื่อย ๆ จนกว่าจนครบจำนวนรอบที่กำหนดหรือจนกว่าจะได้ค่า Cost Function ที่ต่ำจนเป็นที่พอใจ [21]

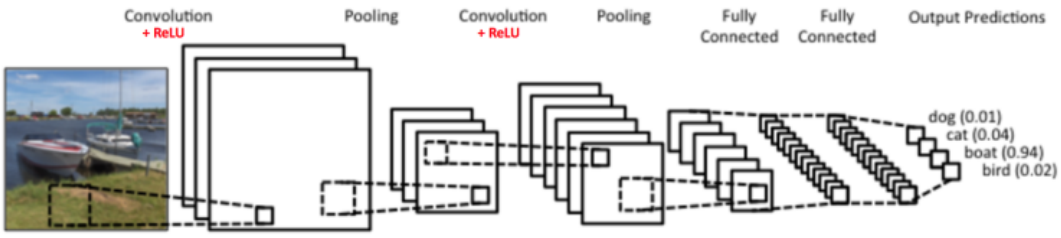


รูปที่ 2.18 การส่งผ่านไปข้างหน้าและการส่งผ่านย้อนหลัง

ที่มา: https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9

2.3.3 โครงข่ายประสาทแบบคอนโวลูชั่น (Convolutional Neural Network)

โครงข่ายประสาทเทียมแบบคอนโวลูชั่น (Convolutional Neural Network: CNN) เป็นการต่อยอดแนวความคิดจากโครงข่ายประสาทเทียมโดยมีจุดประสงค์เพื่อนำมาใช้งานในการทำนายข้อมูลที่เชิงวิสัยทัศน์ โครงข่ายประสาทเทียมแบบคอนโวลูชั่นจะรับข้อมูลข้าเข้าเป็นรูปภาพและให้ผลลัพธ์ออกมาเป็นการทำนายจำแนกประเภท โดยสามารถแบ่งขั้นตอนการทำงานตามลำดับได้คร่าว ๆ ได้แก่ ชั้นคอนโวลูชั่น (Convolution Layer) ชั้นพูลลิ่ง (Pooling Layer) ชั้นการเชื่อมต่ออย่างสมบูรณ์ (Fully Connected Layer) และชั้นการทำให้แบนราบ (Flattening Layer) [22]

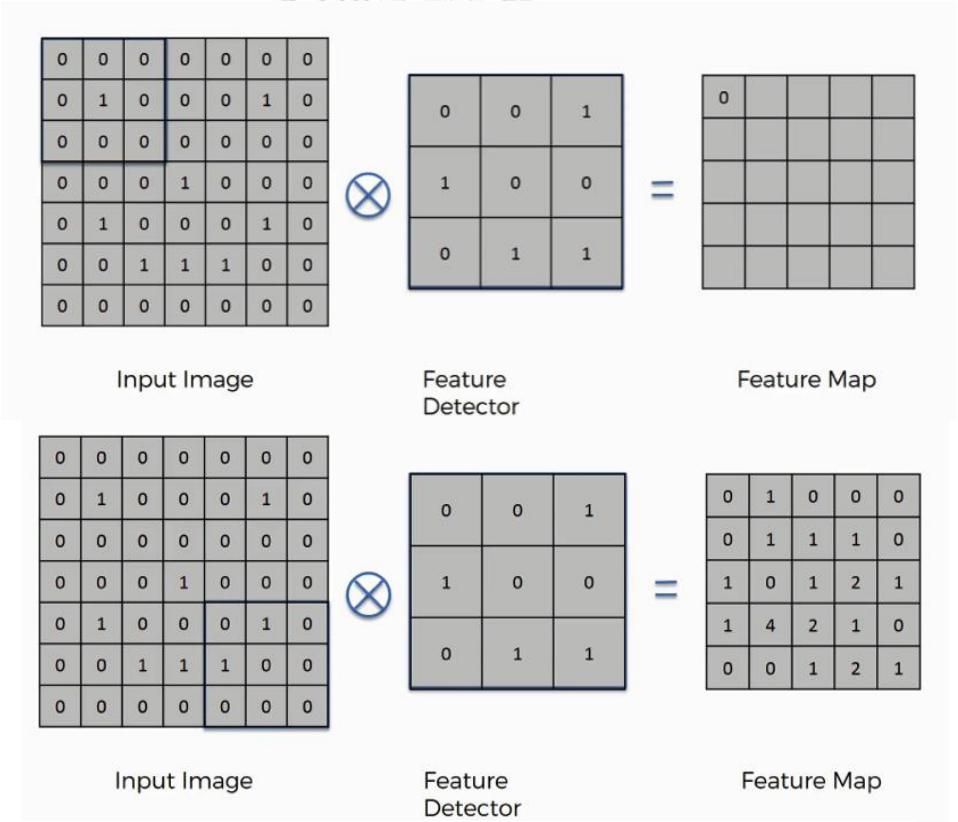


รูปที่ 2.19 ตัวอย่างโครงข่ายประสาทเทียมแบบคอนโวลูชั่น

ที่มา: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets

2.3.3.1 ชั้นคอนโวลูชั่น (Convolutional Layer)

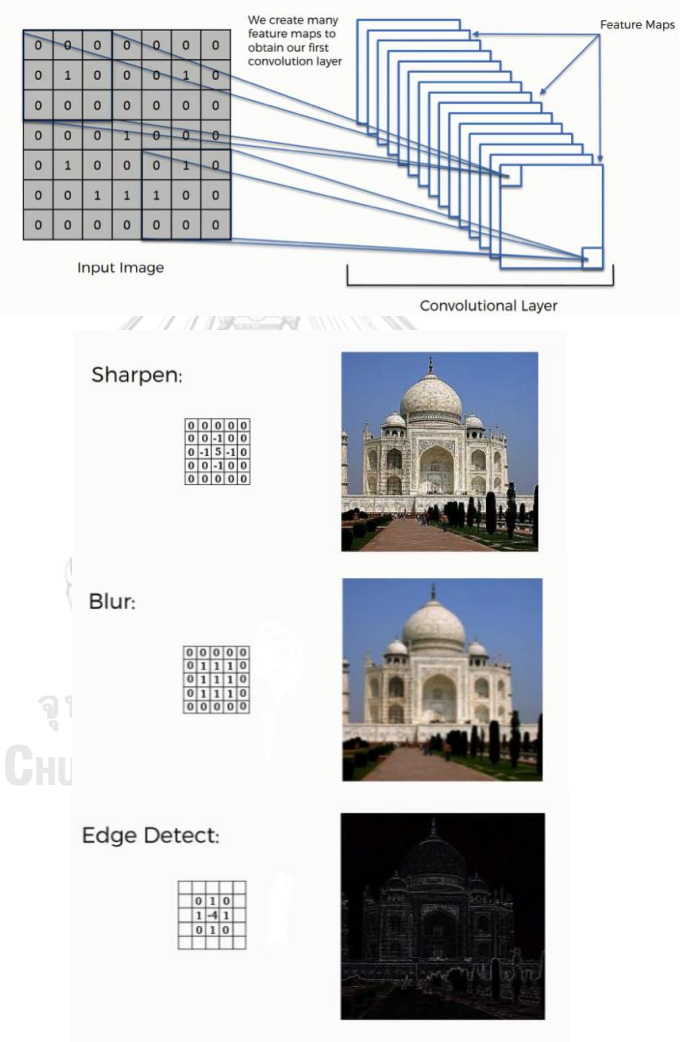
ชั้นคอนโวลูชั่นเป็นชั้นที่สกัดเอาคุณลักษณะของภาพขาเข้าออกมา โดยใช้ตัวตรวจจับคุณลักษณะ (Feature Detector) ซึ่งปกติแล้วจะมีขนาดเล็กกว่าภาพขาเข้า ทาบไปบนภาพ แล้วนำพิกเซลที่อยู่ตำแหน่งเดียวกันมาคูณกัน จากนั้นจึงนำผลจากการคูณมารวมกัน ผลลัพธ์ที่ได้คือค่าที่อยู่ตำแหน่งพิกเซลนั้น ๆ บนแผนที่คุณลักษณะ (Feature Map) โดยดำเนินการเช่นนี้ไปจนครบพื้นที่ของภาพขาเข้าดังแสดงในรูป 2.20 โดยในการเลื่อนตัวจับคุณลักษณะไปยังพื้นที่ต่อไปนั้น อาจเลื่อนครั้งละ 1 พิกเซลหรือมากกว่าก็ได้ โดยหากยิ่งเลื่อนตัวตรวจจับคุณลักษณะไปไกลขึ้น แผนที่คุณลักษณะที่ได้ออกมาก็จะยิ่งเล็กลง ทั้งนี้ในบางครั้งอาจเรียกตัวตรวจจับคุณลักษณะว่าเคอร์เนล (Kernel) หรือตัวกรอง (Filter) ก็ได้เช่นกัน



รูปที่ 2.20 แสงการกระบวนการการคำนวณหาแผนที่คุณลักษณะ

ที่มา: https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

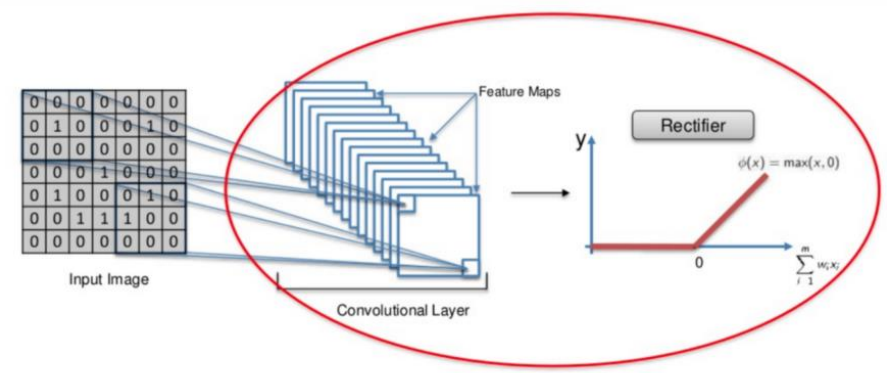
การนำภาพขาเข้ามาผ่านกระบวนการด้วยตัวตรวจจับคุณลักษณะและได้ออกมาเป็นแผนที่คุณลักษณะนั้น ทำให้ได้ภาพใหม่ที่ปรากฏคุณลักษณะบางอย่างชัดเจนขึ้น โดยขึ้นอยู่กับรูปแบบของตัวตรวจจับคุณลักษณะที่นำมาใช้ซึ่งสามารถมีได้หลายรูปแบบและโดย ปกติในชั้นคอนโวลูชั่นนี้ ภาพขาเข้า 1 ภาพ มักจะผ่านการดำเนินการตรวจจับคุณลักษณะด้วยตัวกรองหลายรูปแบบ ทำให้ได้ผลลัพธ์ออกมาเป็นแผนที่คุณลักษณะจำนวนมากดังแสดงในรูปที่ 2.21



รูปที่ 2.21 การคำนวณหาแผนที่คุณลักษณะในชั้นคอนโวลูชั่นและตัวอย่างจากการใช้ตัวตรวจจับคุณลักษณะในรูปแบบต่าง ๆ

ที่มา: https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

แผนที่คุณลักษณะที่ได้มานั้น บ่อยครั้งที่จะพบความเป็นเชิงเส้น (Linearity) ปรากฏอยู่มาก ซึ่งผิดกับธรรมชาติความเป็นจริงของรูปภาพ ดังนั้นภาพแผนที่คุณลักษณะจะต้องถูกนำมาผ่าน Rectified Linear Unit หรือ ReLu ก่อน เพื่อลดความเป็นเชิงเส้นของข้อมูลลง [23]



รูปที่ 2.22 การใช้ฟังก์ชั่น ReLu กับแผนที่คุณลักษณะ

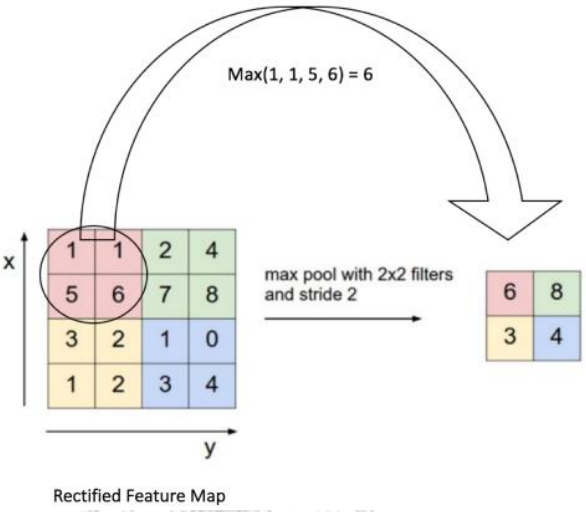
ที่มา: https://medium.com/@pradyasin/what-is-convolution-neural-network-bf2e525089f5

2.3.3.2 ชั้นพูลลิ่ง (Pooling Layer)

ชั้นพูลลิ่งเป็นชั้นที่อยู่ถัดมาจากชั้นคอนโวลูชั่น จุดประสงค์หลักของชั้นพูลลิ่งคือการลดขนาดของภาพจากชั้นก่อนหน้าลงเพื่อลดความซับซ้อนของข้อมูล เพื่อเพิ่มความเร็วในการประมวลผลแต่จะพยายามรักษาข้อมูลสำคัญไว้ให้ได้มากที่สุด พูลลิ่งสามารถมีได้หลายรูปแบบ โดยแบบที่นิยมใช้งานได้แก่ พูลลิ่งสูงสุด (Max Pooling) พูลลิ่งค่าเฉลี่ย (Average Pooling) และพูลลิ่งผลรวม (Sum Pooling) เป็นต้น

รูปที่ 2.23 เป็นตัวอย่างการพูลลิ่งสูงสุด โดยใช้ตัวกรองขนาด 2 x 2 พิกเซล ทาบไปบนภาพที่ผ่านชั้นคอนโวลูชั่นและ ReLu มาแล้ว แล้วจึงเลือกเฉพาะค่าที่สูงที่สุดจากนั้นจึงเลื่อนไปดำเนินการต่อในพื้นที่ถัดไป ทั้งนี้การเลื่อนตำแหน่งของตัวกรองในชั้นพูลลิ่งนั้น พื้นที่ที่เคยผ่านการดำเนินการไปแล้วจะไม่มีการพิจารณาซ้ำอีก (แตกต่างกับการดำเนินการในชั้นคอนโวลูชั่น)

การพูลลิ่งนั้น นอกจากจะเป็นการลดขนาดของข้อมูลแล้วยังเป็นการลดจำนวนพารามิเตอร์ ซึ่งทำให้สามารถป้องกันการ overfitting ได้อีกทางหนึ่งด้วย นอกจากนี้ยังทำ ให้ภาพทนต่อการเปลี่ยนแปลง เนื่องจากการเปลี่ยนแปลงภาพเล็กน้อย (เช่น การบิดภาพ การหมุนภาพ การยืด-หดภาพ) จะไม่เปลี่ยนแปลงผลลัพธ์ของการพูลลิ่ง [24]

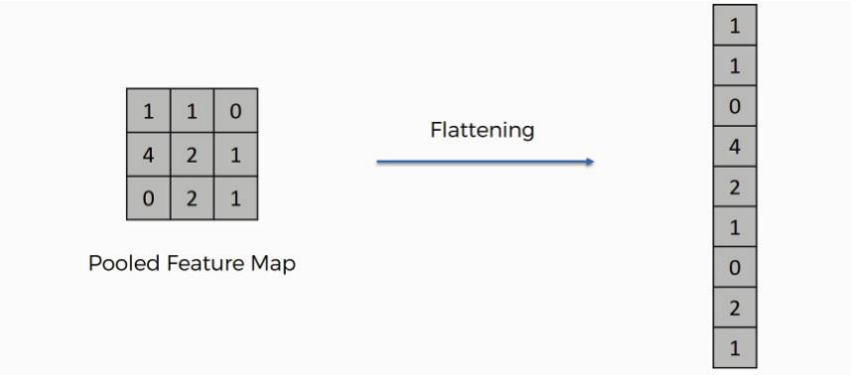


รูปที่ 2.23 ตัวอย่างของการพูลลิ่งสูงสุด

ที่มา: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

2.3.3.3 ชั้นการทำให้แบนราบ (Flattening Layer)

ชั้นการทำให้แบนราบ คือชั้นที่รับข้อมูลภาพที่ผ่านการพูลลิ่งแล้วมาแปลงให้กลายเป็นข้อมูล 1 คอลัมน์เพื่อที่สามารถส่งต่อเป็นข้อมูลขาเข้าของชั้นการเชื่อมต่ออย่างสมบูรณ์ได้ โดยการทำให้แบนราบนั้นจะทำเพียงครั้งเดียวก่อนที่จะส่งข้อมูลไปยังชั้นการเชื่อมต่ออย่างสมบูรณ์เท่านั้น แตกต่างกันชั้นคอนโวลูชั่นและชั้นพูลลิ่งที่โดยปกติมักจะมีการ ทำซ้ำหลายครั้ง

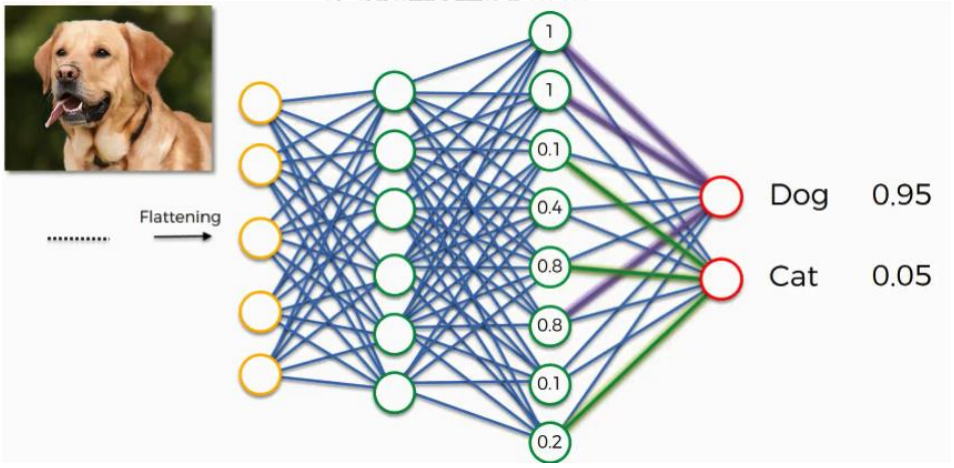


รูปที่ 2.24 ตัวอย่างการทำให้แบนราบ

ที่มา: https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

2.3.3.4 ชั้นการเชื่อมต่ออย่างสมบูรณ์ (Fully Connected Layer)

ชั้นการเชื่อมต่ออย่างสมบูรณ์ เป็นการนำผลลัพธ์จากชั้นก่อนหน้าส่งต่อเข้าไปยังโครงข่ายประสาทเทียมและผ่านการประมวลผลในชั้นซ่อนตัวของโครงข่าย



รูปที่ 2.25 ชั้นการเชื่อมต่ออย่างสมบูรณ์

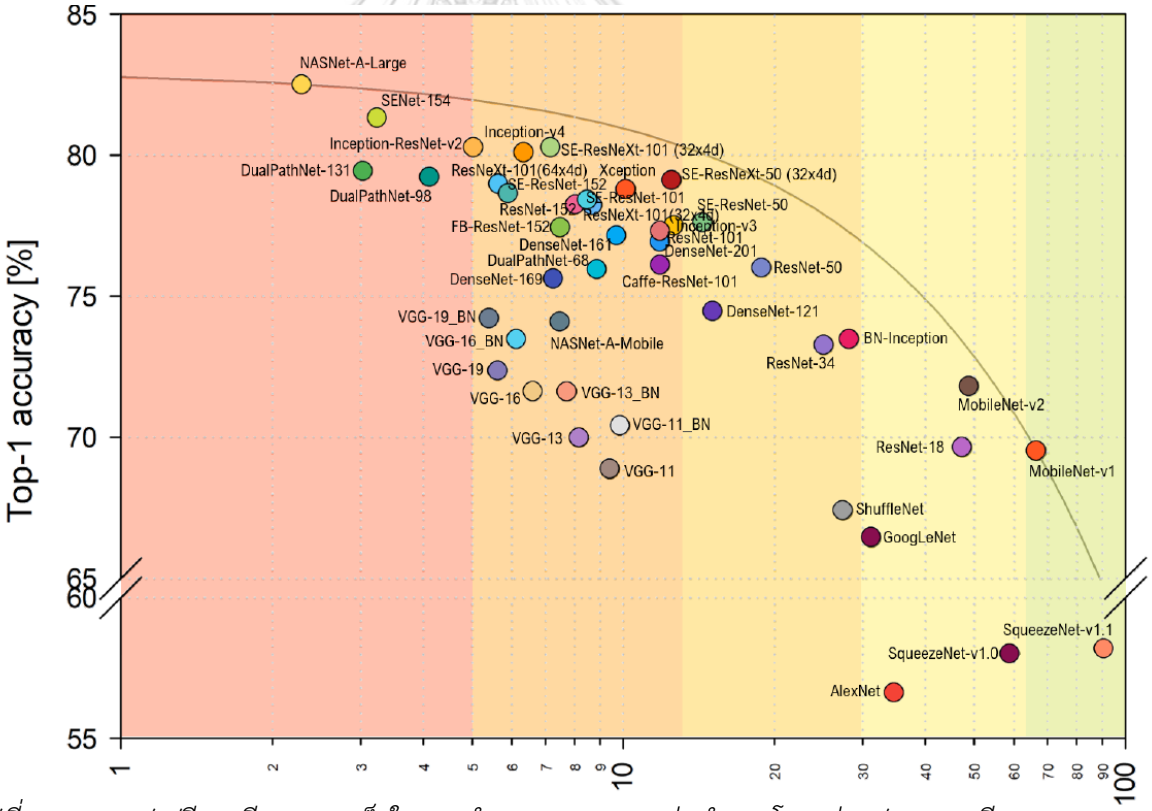
ประสาทเทียมก่อนที่จะได้ผลลัพธ์ออกมาเป็นการทำนายจำแนกประเภท

ที่มา: https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

2.3.4 โครงสร้างแบบ MobileNetV2

โครงข่ายประสาทแบบคอนโวลูชั่นนั้นมีหลากหลายรูปแบบขึ้นอยู่กับการจัดเรียงลำดับชั้นและตัวแปรปรับค่าได้ต่าง ๆ ซึ่งโครงสร้างแต่ละแบบนั้นจะมีความแม่นยำและความรวดเร็วในการฝึก/ทำนายแตกต่างกันไป

สำหรับงานวิจัยชิ้นนี้ ผู้วิจัยได้ทำการสืบค้นข้อมูลจากงานวิจัยอื่น ๆ เพื่อหาโครงสร้างที่เหมาะสมสำหรับการใช้งานกับระบบในงานวิจัยนี้ ทั้งนี้เนื่องจากอุปกรณ์ที่ใช้คือบอร์ด Raspberry Pi ซึ่งมีกำลังประมวลผลต่ำ ดังนั้นโครงสร้างของโครงข่ายประสาทแบบคอนโวลูชั่นที่เหมาะสมกับการใช้งานจึงจำเป็นต้องมีความเร็วในการประมวลผลที่สูงและใช้งานทรัพยากรระบบต่ำ ในการทดลองนี้จึงเลือกใช้โครงสร้างสถาปัตยกรรมแบบ MobileNetV2 ซึ่งตรงกับความต้องการและยังมีความแม่นยำอยู่ในระดับสูงดังแสดงในรูปที่ 2.26 [25, 26]

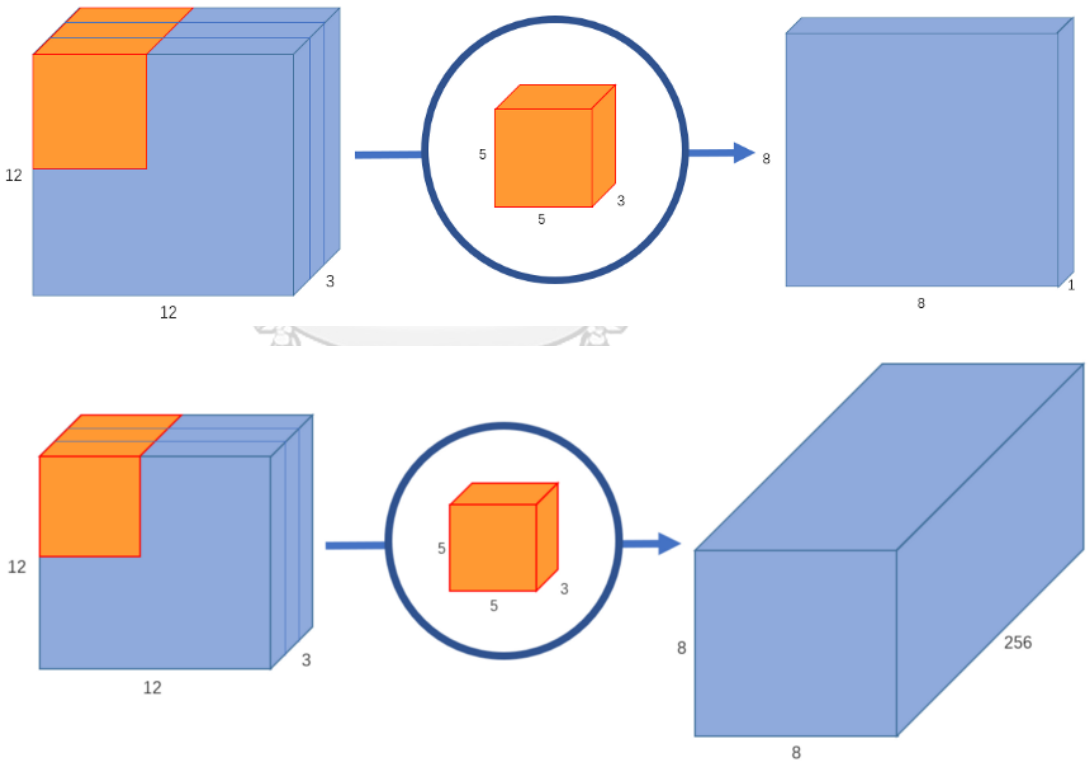


รูปที่ 2.26 กราฟเปรียบเทียบความเร็วในการทำงานและความแม่นยำของโครงข่ายประสาทเทียมคอนโวลูชั่นรูปแบบต่าง ๆ [26]

โครงสร้างแบบ MobileNetV2 เป็นสถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชั่นที่พัฒนาโดย Google Inc. ซึ่งมุ่งเน้นไปที่การใช้งานบนอุปกรณ์หรือ

ระบบที่มีกำลังประมวลผลต่ำเช่นอุปกรณ์ IoT หรือโทรศัพท์มือถือ โดยแทนที่คอนโวลูชั่นแบบดั้งเดิมด้วยการใช้คอนโวลูชั่นเชิงลึกแบบแบ่งแยกได้ (Depthwise Separable Convolution) ซึ่งเป็นการใช้คอนโวลูชั่นแบบเชิงลึก (Depthwise Convolution) มาคำนวณร่วมกับคอนโวลูชั่นเชิงจุด (Pointwise Convolution) เพื่อลดเวลาในการคำนวณลงจากคอนโวลูชั่นแบบดั้งเดิม

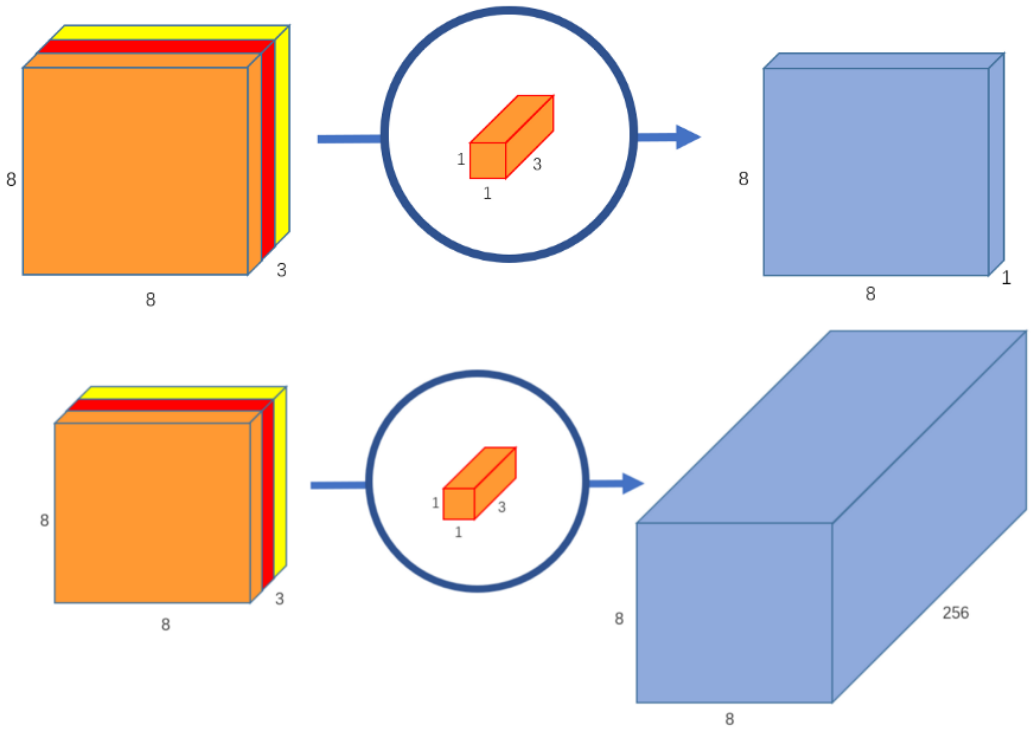
รูปที่ 2.27 แสดงการทำคอนโวลูชั่นแบบดั้งเดิมบนภาพขาเข้าขนาด 12 x 12 x 3 พิกเซล ด้วยการใช้ฟิลเตอร์ขนาด 5 x 5 x 3 พิกเซล ซึ่งจะต้องใช้ฟิลเตอร์ที่มีจำนวนชั้นหรือ Channel หนาเท่ากับจำนวนชั้นของภาพขาเข้าและให้ผลลัพธ์ออกมาเป็นแผนที่คุณลักษณะขนาด 8 x 8 x 1 พิกเซล ซึ่งหากต้องการให้ได้แผนที่คุณลักษณะจำนวน 256 ชั้น จำเป็นต้องทำการคำนวณทั้งสิ้น 256 x 3 x 5 x 5 x 8 x 8 = 1,228,800 ครั้ง



รูปที่ 2.27 การทำคอนโวลูชั่นแบบดั้งเดิม

ที่มา: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

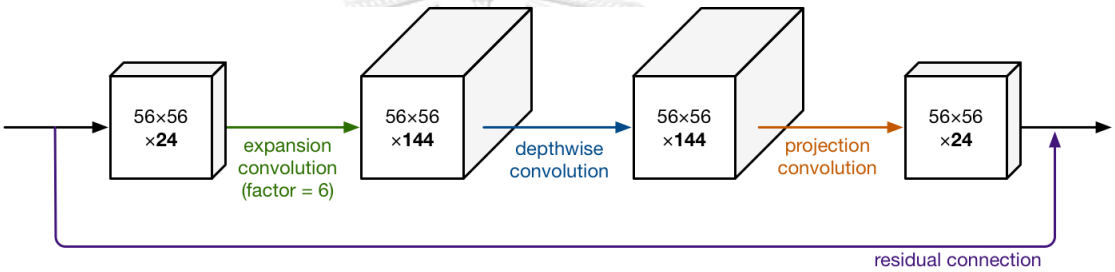
รูปที่ 2.28 เป็นการแสดงการทำคอนโวลูชั่นเชิงลึกแบบแบ่งแยกได้บนภาพขาเข้าขนาด 12 x 12 x 3 พิกเซล ด้วยการใช้คอนโวลูชั่นเชิงลึกซึ่งเป็นการใช้ฟิลเตอร์ขนาด 5 x 5 x 1 พิกเซลมาทำคอนโวลูชั่น โดยต้องทำทั้งหมด 3 ครั้ง จะเห็นว่าการทำคอนโวลูชั่นเชิงลึกนั้นเป็นการใช้ฟิลเตอร์ที่มีความหนาเพียงชั้นเดียวแต่เปลี่ยนฟิลเตอร์ไปทีละชั้นของภาพขาเข้าได้เป็นแผนที่คุณลักษณะขนาด 8 x 8 x 3 พิกเซล และตามด้วยการใช้คอนโวลูชั่นเชิงจุด ซึ่งเป็นการคอนโซลูชั่นโดยใช้ฟิลเตอร์ขนาด 1 x 1 x 3 พิกเซล หรือฟิลเตอร์พิกเซลเดียวที่หนาเท่าภาพตั้งต้นและจำได้ผลลัพธ์เป็นแผนที่คุณลักษณะขนาด 8 x 8 x 1 โดยหากต้องการให้ได้แผนที่คุณลักษณะจำนวน 256 ชั้น จำเป็นต้องทำการคำนวณทั้งสิ้น (3 x 5 x 5 x 8 x 8) + (256 x 1 x 1 x 3 x 8 x 8) = 53,952 ครั้งเท่านั้น ซึ่งจะเห็นว่าใช้การคำนวณน้อยกว่าการทำคอนโวลูชั่นแบบดั้งเดิมมาก ด้วยวิธีการนี้จึงทำให้สถาปัตยกรรมแบบ MobileNetV2 ที่ใช้การคอนโวลูชั่นเชิงลึกแบบแบ่งแยกได้นั้นมีความเร็วในการคำนวณสูงมากโดยยังมีความแม่นยำอยู่ในระดับสูง



รูปที่ 2.28 การทำคอนโวลูชั่นเชิงลึกแบบแบ่งแยกได้

ที่มา: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

นอกจากนี้ ในโครงสร้าง MobileNetV2 ยังมีการใช้คอนโวลูชั่นแบบขยาย (Expansion Convolution) ซึ่งเป็นการใช้คอนโวลูชั่นเชิงจุดในการเพิ่มจำนวนชั้นของภาพขาเข้าเพื่อเพิ่มความละเอียดของข้อมูล ก่อนที่จะส่งไปยังคอนโวลูชั่นเชิงลึกแบบแบ่งแยกได้ต่อไป รวมทั้งมีการใช้ Residual Connection ซึ่งเป็นการเพิ่มเส้นการเชื่อมต่อเพื่อสร้างทางลัดของข้อมูลจากข้อมูลขาเข้าไปยังข้อมูลขาออกโดยไม่ผ่านชั้นคอนโวลูชั่น ซึ่งข้อมูลขาเข้าดังกล่าวจะถูกนำไปรวมกันข้อมูลที่ผ่านการคอนโวลูชั่นแล้วอีกทีหนึ่งดังแสดงในรูปที่ 2.27 การเพิ่มชั้นคอนโวลูชั่นแบบขยายและการเพิ่มเส้นทางลัดดังกล่าว ช่วยให้โครงสร้าง MobileNetV2 มีความแม่นยำเพิ่มมากขึ้น โดยกลุ่มชั้นที่กล่าวข้างต้นรวมกันเรียกว่าชั้นคอขวด หรือ Bottleneck Layer



รูปที่ 2.29 กลุ่มชั้นคอขวด

ที่มา: https://machinethink.net/blog/mobilenet-v2/

โดยในโครงสร้างแบบ MobileNetV2 นั้น ประกอบด้วยกลุ่มชั้นคอขวดและชั้นอื่น ๆ โดยมีค่าตัวแปรต่าง ๆ ดังแสดงในรูปที่ 2.29 โดยที่ t แทนจำนวนเท่าของการขยายในคอนโวลูชั่นแบบขยาย c แทนค่าจำนวนชั้นของภาพขาออก n แทนจำนวนครั้งที่ดำเนินการ และ s แค่ค่า stride หรือการขยับของฟิลเตอร์ในแต่ละครั้งที่ทำการคอนโวลูชั่น

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

รูปที่ 2.30 ชั้นต่าง ๆ ในโครงสร้าง MobileNetV2 [25]

บทที่ 3
วิธีดำเนินการทดลอง

ในงานวิจัยชิ้นนี้มีวัตถุประสงค์ในการพัฒนาระบบที่ช่วยให้ผู้ป่วย/ผู้สูงอายุสามารถสื่อสารกับผู้ดูแลได้โดยง่าย ทั้งนี้ยังต้องการให้ระบบสามารถติดตั้งได้ง่าย ราคาย่อมเยาว์ และสามารถผู้ใช้สามารถเข้าถึงได้โดยไม่ลำบาก

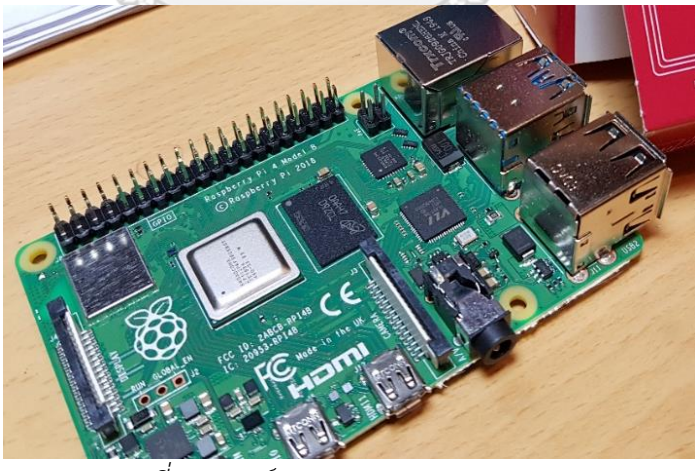
3.1 อุปกรณ์และเครื่องมือ

3.1.1 ฮาร์ดแวร์ (Hardware)

เพื่อให้สอดคล้องต่อวัตถุประสงค์ของงานวิจัย ในการทดลองนี้จึงใช้ฮาร์ดแวร์หลักคือ Raspberry Pi และกล้อง Raspberry Pi Camera Module เนื่องจากเป็นอุปกรณ์ที่มีขนาดเล็ก ติดตั้งง่าย และราคาถูก

3.1.1.1 Raspberry Pi

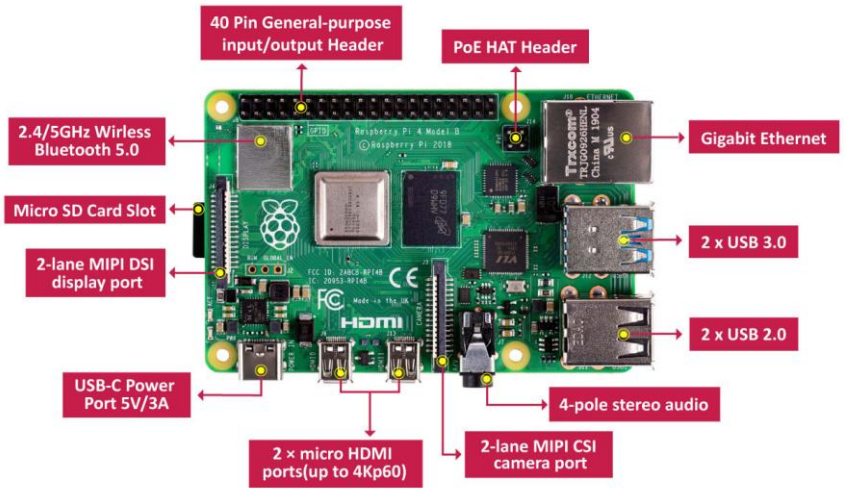
Raspberry Pi คือคอมพิวเตอร์ส่วนบุคคลรุ่นหนึ่งที่ถูกสร้างโดย Raspberry Pi Foundation ซึ่งเป็นองค์กรการกุศลที่มีจุดมุ่งหมายในการช่วยให้ผู้คนสามารถเข้าถึงคอมพิวเตอร์ได้



รูปที่ 3.1 บอร์ด Raspberry Pi 4 Model B

Raspberry Pi ถูกเปิดตัวครั้งแรกตั้งแต่ปี 2012 และมีการพัฒนารุ่นใหม่ ๆ มาจนถึงปัจจุบันที่เป็นรุ่นที่ 4 โดยเป็นคอมพิวเตอร์ที่เหมาะสำหรับการเรียนรู้การเขียนโปรแกรม การสร้างโครงงาน การทำระบบบ้านอัจฉริยะ การทำอุปกรณ์ Internet of Things (IoT) ตลอดจนการประยุกต์ใช้ในงานระดับอุตสาหกรรม

บอร์ด Raspberry Pi สามารถใช้งานต่อจอแสดงผล ใช้งานร่วมกับอุปกรณ์ต่อพ่วงหรือใช้งานระบบเน็ตเวิร์กได้เช่นเดียวกับคอมพิวเตอร์ส่วนบุคคลทั่วไป เพียงแต่มีขนาดเล็กเท่าบัตรเครดิต ราคาถูก และมีพลังการประมวลผลน้อยกว่าคอมพิวเตอร์ปกติทั่วไป โดยในงานวิจัยชิ้นนี้ได้ใช้บอร์ด Raspberry Pi 4 Model B 8GB ซึ่งมีคุณสมบัติดังแสดงในรูปที่ 3.2 และตารางที่ 3.1



รูปที่ 3.2 ส่วนประกอบของบอร์ด Raspberry Pi 4 Model B

ที่มา: https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html

ตารางที่ 3.1 รายละเอียดคุณสมบัติของบอร์ด Raspberry Pi 4 Model B 8GB

ส่วนประกอบ	รายละเอียด
หน่วยประมวลผล	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
หน่วยความจำ	8 GB LPDDR4-3200 SDRAM
หน่วยประมวลผลกราฟฟิก	Broadcom VideoCore VI @ 500MHz
การเชื่อมต่อไร้สาย	2.4GHz and 5GHz 802.11b/g/n/ac, Bluetooth 5.0, BLE
การเชื่อมต่อเครือข่าย	Gigabit Ethernet
พอร์ต USB	2 USB 3.0 ports; 2 USB 2.0 ports
GPIO	40-pin GPIO header
HDMI	2 × micro HDMI ports (up to 4Kp60 supported)
พอร์ตแสดงผล	2-lane MIPI DSI

พอร์ตกล้อง	2-lane MIPI CSI
เสียง	3.5mm analogue audio-video jack
พื้นที่เก็บข้อมูล	Micro-SD card slot
แหล่งพลังงาน	5 V DC, minimum 3 A
ระบบปฏิบัติการ	Debian Linux 10 based

3.1.1.2 Raspberry Pi Camera Module V2

งานวิจัยนี้ได้ใช้โมดูลกล้องของ Raspberry Pi ในการรับภาพเคลื่อนไหวแบบทันที (Real Time) เพื่อส่งต่อข้อมูลไปประมวลผลที่บอร์ด Raspberry Pi ต่อไป โดยเชื่อมต่อผ่านพอร์ต Camera Serial Interface (CSI) ที่อยู่บนบอร์ด ซึ่งตัวกล้องมีคุณสมบัติดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 คุณสมบัติของ Raspberry Pi Camera Module V2

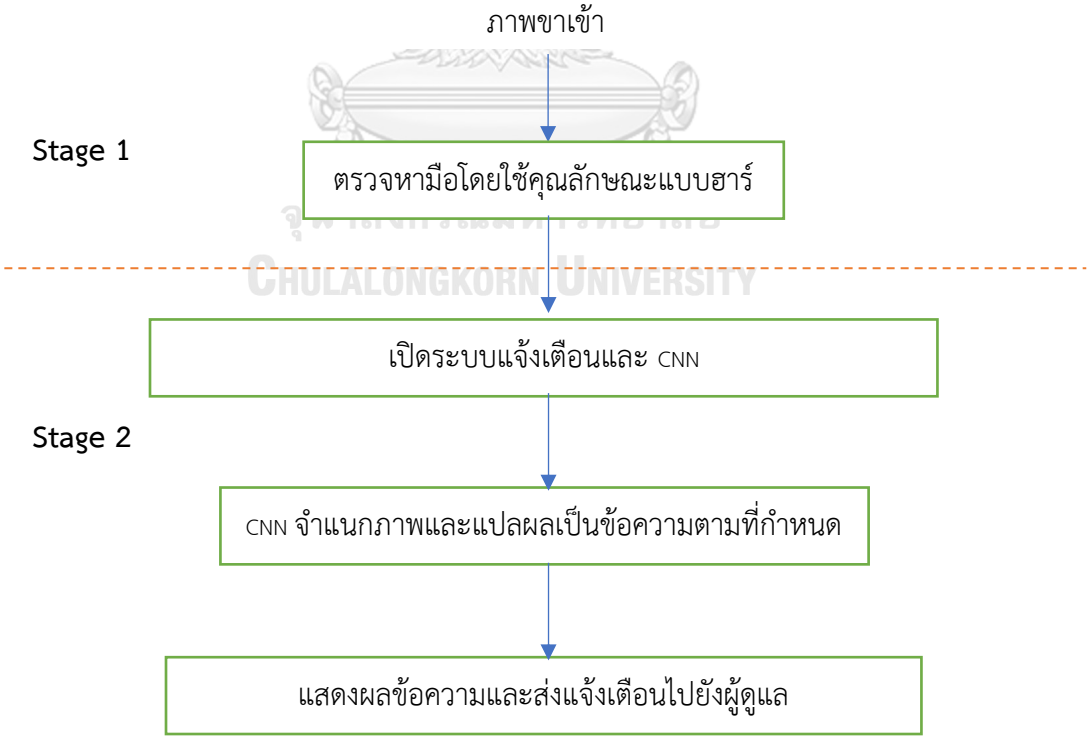
ส่วนประกอบ	**รายละเอียด**
เซ็นเซอร์	Sony IMX219
ความละเอียดเซ็นเซอร์	3280 × 2464 pixels
ความละเอียดภาพนิ่ง	8 Megapixels
การถ่ายภาพเคลื่อนไหว	1080p30, 720p60, 640×480p60/90
ฟอร์แมตภาพ	JPEG (accelerated), JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888
ฟอร์แมตวีดีโอ	raw h.264 (accelerated)

3.1.2 ซอฟท์แวร์ (Software)

- Python 3.6.10: ภาษาที่ใช้เขียนโปรแกรมในงานวิจัย
- Tensorflow 2.0.0: ชุดเครื่องมือสำหรับพัฒนาการเรียนรู้ของเครื่องจักร
- Keras 2.2.4: ชุดเครื่องมือพัฒนาการเรียนรู้ของเครื่องจักร ใช้ร่วมกับ Tensorflow
- OpenCV 3.4.1: ชุดเครื่องมือสำหรับประมวลผลภาพ
- Nvidia CUDA 10.0.130: ชุดเครื่องมือสำหรับการประมวลผลแบบคู่ขนานโดยใช้ GPU
- ชุดเครื่องมืออื่น ๆ เช่น NumPy, Time, request, urllib, sklearn และ matplotlib

3.2 กระบวนการทดลอง

งานวิจัยชิ้นนี้มุ่งเน้นไปที่การสร้างระบบแจ้งเตือนที่จำเป็นต้องเปิดใช้งานระบบตลอดเวลาเพื่อเฝ้าดูผู้ป่วยในยามที่ผู้ดูแลไม่ได้อยู่ด้วย ทั้งนี้การเปิดใช้งานระบบตลอดเวลานั้น ก่อให้เกิดปัญหาหลัก 2 อย่างคือ เป็นภาระของอุปกรณ์ที่ต้องประมวลผลหนักตลอดเวลาและอาจเกิดการแปลผล/แจ้งเตือนโดยไม่ตั้งใจเมื่อผู้ใช้มีการขยับมือโดยไม่ได้เจตนาจะส่งข้อความ ดังนั้นในงานวิจัยชิ้นนี้จึงได้แบ่งการทำงานของระบบออกเป็น 2 สเตจหลัก โดยช่วงแรกจะเป็นสเตจสแตนด์บายซึ่งในช่วงนี้จะปิดระบบแจ้งเตือนทั้งหมดไว้และเปิดการทำงานเฉพาะการตรวจหามือโดยใช้คุณลักษณะแบบฮาร์เท่านั้น โดยเป็นการให้สัญญาณเข้าสู่สเตจถัดไป หากมีการตรวจเจอมือแล้ว ระบบก็จะคำนวณหาพื้นที่ที่สนใจและส่งต่อไปยังสเตจที่ 2 โดยในสเตจนี้จะเปิดการใช้งานการจำแนกท่ามือด้วยโครงข่ายประสาทแบบคอนโวลูชั่นและเปิดระบบแจ้งเตือน ซึ่งในสเตจ 2 ได้มีการกำหนดหน้าต่างเวลาไว้ 3 วินาที หากภายใน 3 วินาทีดังกล่าวระบบตรวจไม่พบท่ามือใด ๆ ระบบก็จะย้อนกลับไปเป็นสเตจ 1 แผนภาพการทำงานแสดงในรูปที่ 3.3



รูปที่ 3.3 แผนภาพการทำงานของระบบ

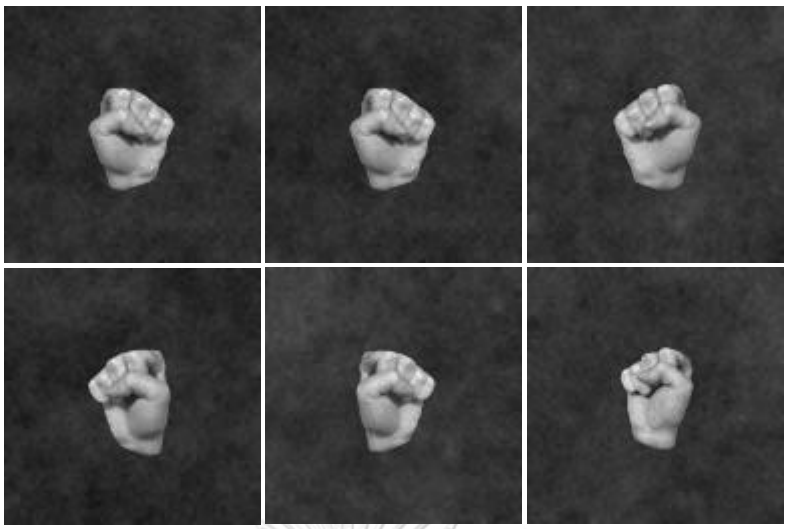
3.2.1 การตรวจหามือด้วยคุณลักษณะแบบฮาร์

ในการตรวจหามือด้วยคุณลักษณะแบบฮาร์ตามกระบวนขั้นตอนของ Viola-Jones นั้นจำเป็นต้องมีโมเดลเพื่อใช้สำหรับการตรวจจับหามือ โดยโมเดลดังกล่าวได้มาจากการฝึกสอนเครื่องจักรด้วยวิธีการเรียนรู้แบบเอดาบูสท์ ซึ่งจำเป็นต้องมีชุดข้อมูลภาพเพื่อใช้ในการเรียนรู้ ชุดข้อมูลภาพที่ใช้สำหรับการเรียนรู้นี้ประกอบด้วย 2 ชุดคือชุดข้อมูลของภาพที่สนใจ (Positive Samples) และชุดของมูลของภาพอื่น ๆ ที่ไม่มีส่วนของวัตถุที่สนใจประกอบอยู่ในภาพ (Negative Samples) ซึ่งชุดข้อมูลทั้ง 2 จะต้องมีจำนวนมากพอสมควรเพื่อให้ระบบสามารถเรียนรู้ได้

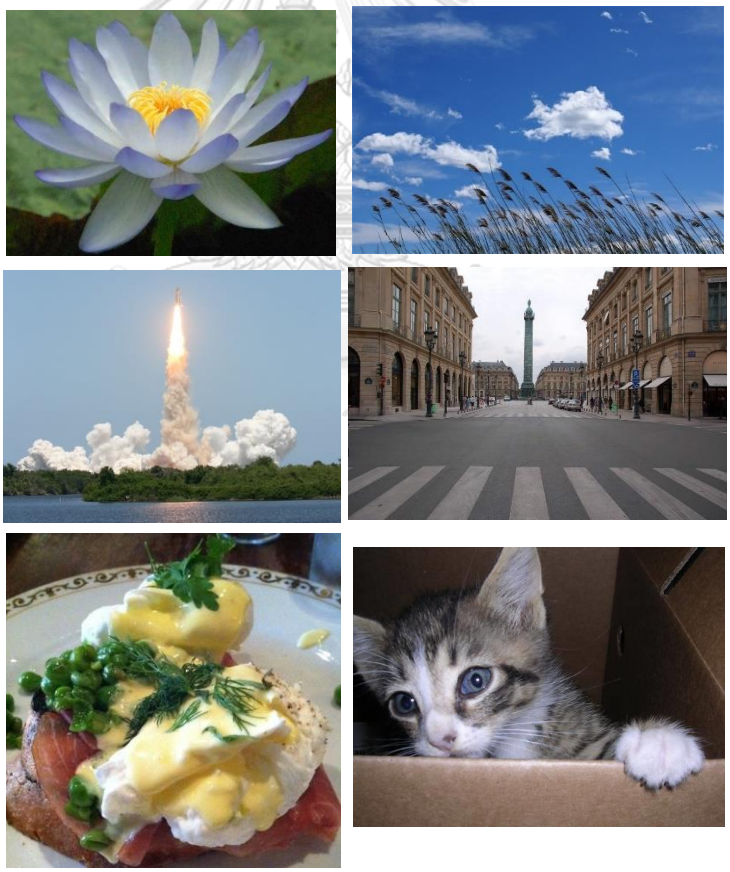
ชุดข้อมูลรูปภาพที่เตรียมไว้จะถูกนำไปใช้ในการเรียนรู้เพื่อหาคุณลักษณะแบบฮาร์ ทั้งนี้เนื่องจากคุณลักษณะแบบฮาร์นั้นสามารถมีได้จำนวนมหาศาล จึงต้องใช้การเรียนรู้แบบเอดาบูสท์เข้ามาช่วยในการคัดเลือกเฉพาะคุณลักษณะที่ให้ค่าความผิดพลาดในการทำนายน้อยที่สุดโดยคุณลักษณะแต่ละอันจะถือเป็นตัวจำแนกประเภทแบบอ่อน (Weak Classifier) ซึ่งเมื่อนำมารวมกันแล้วจะได้เป็นตัวจำแนกประเภทแบบแข็งแกร่ง (Strong Classifier) แล้วจึงนำไปใช้ในขั้นตอนการจำแนกแบบลำดับขั้น (Cascaded Classifier) เพื่อทำการตรวจหามือต่อไป

ในการทดลองนี้ ในช่วงสเตจที่ 1 ระบบจะปิดการแจ้งเตือนทั้งหมดไว้ โดยได้กำหนดให้ใช้ท่ากำมือเป็นท่าที่เราต้องการตรวจหา หากไม่เจอท่ากำมือในเฟรมภาพ ระบบก็จะอยู่ในสเตจที่ 1 ไปเรื่อย ๆ แต่หากเจอท่ากำมือในเฟรมภาพ ระบบก็จะส่งต่อพื้นที่ที่สนใจ (Region of Interest: ROI) ไปยังสเตจ 2 ซึ่งจะทำหน้าที่พิจารณาและจำแนกของท่ามือในพื้นที่ดังกล่าวต่อไป

เนื่องจากในขั้นตอนนี้ ระบบต้องการตรวจหาท่ากำมือ ดังนั้นชุดข้อมูลของภาพที่สนใจจึงเป็นชุดข้อมูลภาพกำมือ ในการทดลองนี้ได้ใช้ภาพทั้งหมด 1705 ภาพ ซึ่งส่วนใหญ่เป็นภาพที่ได้มาจากการรวบรวมชุดข้อมูลจากเว็บไซท์ Kaggle [27] ขณะที่อีกส่วนจะเป็นชุดข้อมูลของภาพทั่วไปที่ไม่มีส่วนประกอบของมืออยู่ในภาพ ซึ่งในการทดลองนี้ได้ใช้ภาพทั้งหมด 8730 ภาพ



รูปที่ 3.4 ตัวอย่างชุดข้อมูลภาพที่สนใจ [27]



รูปที่ 3.5 ตัวอย่างชุดข้อมูลภาพที่ไม่มีส่วนประกอบของวัตถุที่สนใจ

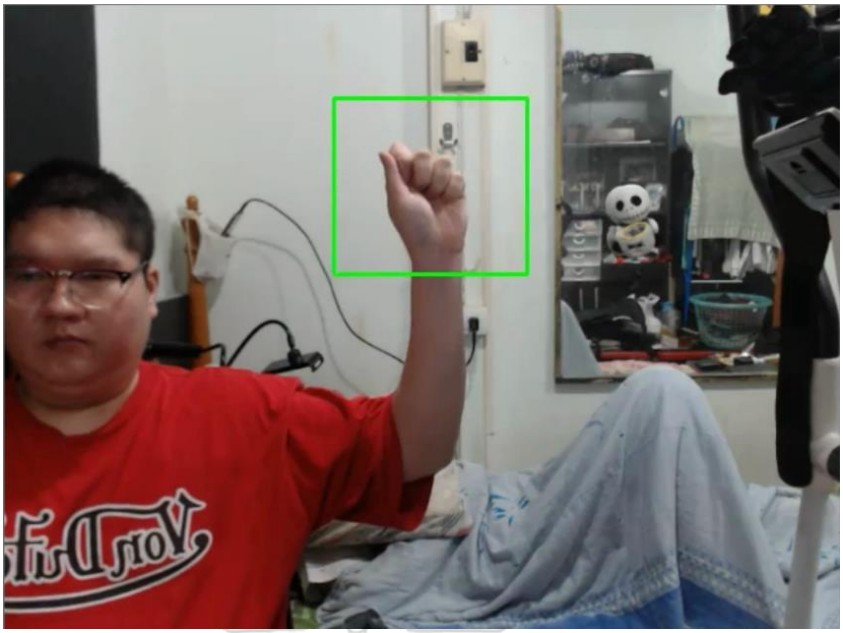
ในขั้นตอนการเรียนรู้แบบเอดาบูสท์ด้วยการใช้คุณลักษณแบบฮาร์ในงานวิจัยนี้ ได้ใช้ชุดอุปกรณ์สำหรับการฝึกสอนซึ่งใช้ภาษา Python และชุดอุปกรณ์ OpenCV เป็นหลัก โดยใช้คอมพิวเตอร์ส่วนบุคคลที่ใช้ ซีพียู intel core i5-8400 2.8GHz และหน่วยความจำขนาด 16 GB สำหรับการฝึกสอน โดยได้มีการกำหนดค่าตัวแปรต่าง ๆ ดังนี้

- ขนาดภาพขาเข้า: 24 พิกเซล x 24 พิกเซล
- อัตราการเจอขั้นต่ำ (Min Hit Rate): 0.995
- อัตราการผิดพลาดเชิงบวกสูงสุด (Max False Alarm Rate): 0.5
- จำนวนชั้นของการเรียนรู้: 20 ชั้น

หลังจากที่โมเดลได้รับการเรียนรู้แบบเอดาบูสท์ด้วยการใช้คุณลักษณะแบบฮาร์แล้ว จะได้ไฟล์โมเดลที่มีนามสกุลไฟล์ .xml ซึ่งโมเดลดังกล่าวสามารถนำไปใช้เพื่อทำการตรวจหาท่ากำมือในภาพต่อไป

การตรวจจับหาท่ากำมือนั้น เมื่อรับภาพขาเข้ามาจากกล้องแล้ว ภาพจะถูกทำสำเนาและภาพสำเนาจะถูกเปลี่ยนให้เป็นภาพระดับเทาเพื่อที่จะสอดคล้องกับคุณลักษณะแบบฮาร์ (โดยไม่เปลี่ยนแปลงภาพต้นฉบับ) จากนั้นจึงใช้ภาพระดับเทาดังกล่าวไปทำการตรวจหาท่ากำมือ โดยในงานวิจัยนี้ใช้ฟังก์ชั่น detectMultiScale ซึ่งเป็นส่วนหนึ่งของชุดเครื่องมือ OpenCV ซึ่งใช้สำหรับตรวจจับวัตถุที่ต้องการในเฟรมภาพ และหากตรวจพบวัตถุที่ต้องการแล้ว ฟังก์ชั่นดังกล่าวจะให้ค่าผลลัพธ์ออกมาเป็นตัวเลข 4 ค่าคือ พิกัดในแนวแกน X ของวัตถุ พิกัดในแนวแกน Y ของวัตถุ ความกว้างของภาพวัตถุ และความสูงของภาพวัตถุ

หลังจากที่ได้ผลลัพธ์จากฟังก์ชั่น detectMultiScale แล้ว ระบบจะใช้ค่าผลลัพธ์ดังกล่าวเพื่อทำการตีกรอบรอบวัตถุในภาพต้นฉบับไว้และกำหนดพื้นที่ดังกล่าวเป็นบริเวณที่สนใจหรือ ROI วิธีการนี้จะทำให้ได้ผลลัพธ์เป็นพื้นที่ที่สนใจโดยมีหมวดสีเป็น RGB เพื่อที่จะส่งต่อไปดำเนินการในกระบวนการถัดไปในสเตจที่ 2



รูปที่ 3.6 การตรวจจับท่ากำมือด้วยการใช้คุณลักษณะแบบฮาร์

3.2.2 การจัดเตรียมข้อมูลภาพ (Image Pre-processing)

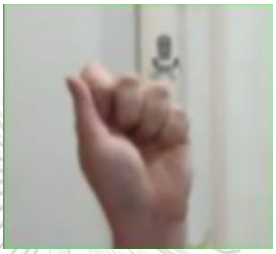
ผลลัพธ์จากขั้นตอนก่อนหน้าทำให้ระบบได้พื้นที่ที่สนใจและเปิดใช้งานการจำแนกท่ามือด้วยโครงข่ายประสาทแบบคอนโวลูชั่น ทั้งนี้ภาพของบริเวณที่สนใจดังกล่าวยังไม่สามารถนำไปใช้เป็นข้อมูลขาเข้าของโครงข่ายประสาทแบบคอนโวลูชั่นได้ในทันที แต่จำเป็นต้องผ่านกระบวนการจัดเตรียมข้อมูลภาพเพื่อปรับปรุงภาพให้เหมาะสมก่อนที่จะนำไปใช้งานในขั้นตอนถัดไป ซึ่งกระบวนการจัดเตรียมข้อมูลภาพที่ใช้ในงานวิจัยนี้มีหลายขั้นตอน ได้แก่

รูปที่ 3.7 ตัวอย่างภาพที่ตัดมาเฉพาะ ROI

- การปรับขนาดภาพ (Resize): เนื่องจาก ROI ที่ได้มานั้นมีขนาดไม่แน่นอน แต่โครงข่ายประสาทแบบคอนโวลูชั่นต้องการภาพขาเข้าที่มีขนาดคงที่ที่ได้กำหนดไว้แล้ว ดังนั้น ROI ที่ได้มาจึงต้องถูกปรับขนาดให้พอดีกับความ

ต้องการของโครงข่ายประสาทแบบคอนโวลูชั่น ซึ่งในการทดลองนี้ได้ใช้ขนาดภาพขาเข้าเท่ากับ 128 พิกเซล x 128 พิกเซล

- การทำให้ภาพราบเรียบ (Smoothing): เป็นการปรับภาพให้เบลอกว่าเดิมเล็กน้อยเพื่อพื้นผิวภาพเสมอกันมากขึ้นส่งผลให้ขอบภาพดูนุ่มขึ้นและเป็นการขจัดสัญญาณรบกวนหรือ Noise ที่มีในภาพ โดยในการทดลองนี้ได้ใช้ตัวกรองชนิด Gaussian ขนาด 5 พิกเซล x 5 พิกเซล



รูปที่ 3.8 ภาพที่ผ่านกระบวนการทำให้ภาพราบเรียบด้วยตัวกรอง Gaussian

- การเปลี่ยนหมวดสี: ภาพที่ได้จากกระบวนการผ่าน ๆ มานั้นมีหมวดสีเป็น RGB จึงจำเป็นต้องทำการเปลี่ยนหมวดสีให้เป็น YCbCr เนื่องจากเป็นหมวดสีที่ประมวลผลได้เร็วและมีค่าสีเหมาะแก่การแบ่งส่วนสีพื้นผิวมนุษย์ในขั้นถัดไป [7]

รูปที่ 3.9 ภาพที่ผ่านกระบวนการเปลี่ยนหมวดสีให้กลายเป็น YCbCr

- การแบ่งสีพื้นผิวมนุษย์ (Human Skin Segmentation): เป็นการกำหนดค่าขอบเขต (Threshold) ของค่าสีขึ้นมาแล้วจึงคัดเลือกเฉพาะพิกเซลที่มีค่าสีอยู่ในเกณฑ์ที่ตั้งไว้ โดยผลลัพธ์ของขั้นตอนนี้จะได้ออกมาเป็นภาพขาว/ดำหรือไบนารี่ โดยพิกเซลที่ผ่านเกณฑ์จะมีสีขาว และพิกเซลที่ไม่ผ่านเกณฑ์จะเป็นสีดำ ทั้งนี้เนื่องจากค่าขอบเขตที่เหมาะสมนั้นอาจ

แตกต่างกันได้ขึ้นกับสีผิวของแต่ละบุคคลและสภาพของแสงในพื้นที่ใช้งาน ทางผู้ทดลองจึงได้ออกแบบโปรแกรมให้ผู้ใช้สามารถปรับค่าขอบเขตที่ต้องการได้เองด้วยการกดปุ่มคีย์บอร์ดเพื่อเพิ่ม-ลดระดับค่าขอบเขต

รูปที่ 3.10 ภาพที่ผ่านกระบวนการแบ่งสีพื้นผิวมนุษย์

- การกัดกร่อน (Erosion): การกัดกร่อนภาพเป็นการการแปลงสันฐาน (Morphological Transformations) รูปแบบหนึ่ง มีหลักการทำงานคือ เลื่อนเคอร์เนลไปทั่วทั้งภาพ หากพิกเซลที่พิจารณาอยู่นั้นถูกทาบด้วยเคอร์เนลแล้วพบว่าค่าที่อยู่ใต้เคอร์เนลทุกพิกเซลมีค่าเป็น 1 พิกเซลดังกล่าวก็จะมีค่าเป็น 1 แต่หากไม่เป็นเช่นนั้น พิกเซลดังกล่าวก็จะมีค่าเป็น 0 กระบวนการนี้ส่งผลให้พิกเซลที่อยู่บริเวณขอบวัตถุจะถูกทำให้หายไปและลดขนาดของวัตถุลง ซึ่งเหมาะสำหรับการขจัดสัญญาณรบกวนเล็ก ๆ ที่ไม่ต้องการออกจากภาพ

รูปที่ 3.11 ภาพที่ผ่านกระบวนการกัดกร่อน

- การพองตัว (Dilation): การพองตัวเป็นการแปลงสันฐานอีกรูปแบบหนึ่ง ซึ่งเป็นกระบวนการตรงข้ามของการกัดกร่อน โดยหากพื้นที่ที่แนบเคอร์เนลลงไปนั้น มีอย่างน้อย 1 พิกเซลที่มีค่าเป็น 1 ก็จะนับว่าผลลัพธ์พิกเซลนั้นเป็น 1 กระบวนการนี้เป็นการเพิ่มพื้นที่บริเวณขอบของวัตถุและทำให้วัตถุมีขนาดใหญ่ขึ้น ซึ่งเหมาะสำหรับการเติมเต็มพื้นที่ที่ขาดหายไปของวัตถุ

รูปที่ 3.12 ภาพที่ผ่านกระบวนการพองตัว

เมื่อผ่านกระบวนการจัดเตรียมข้อมูลข้างต้นเรียบร้อยแล้ว จะได้ผลลัพธ์เป็นภาพบริเวณที่สนใจที่มีขนาด 128 พิกเซล x 128 พิกเซล และมีหมวดสีเป็นภาพขาว/ดำ ซึ่งทั้งหมดนี้เป็นการกระทำเพื่อลดความซับซ้อนของภาพเพื่อเพิ่มความแม่นยำของการทำนายในขั้นตอนถัดไป

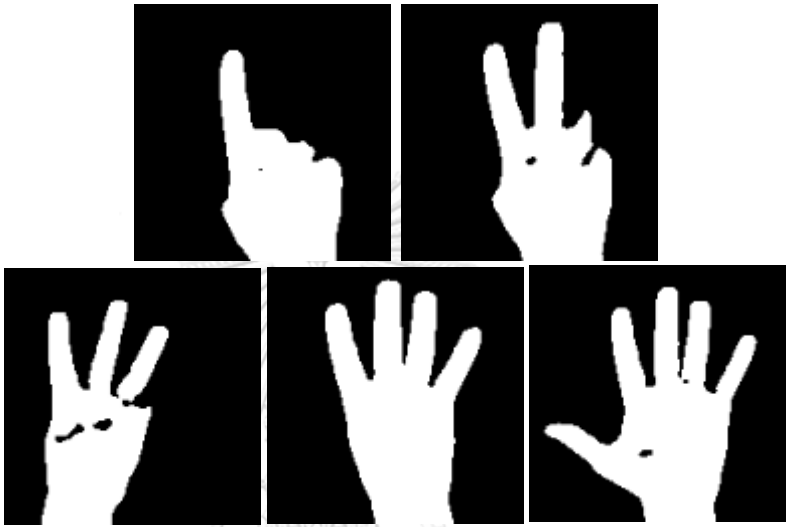
3.2.3 การจำแนกท่ามือด้วยโครงข่ายประสาทแบบคอนโวลูชั่น

ขั้นตอนนี้เป็นการรับเอาผลลัพธ์ที่ผ่านการจัดเตรียมข้อมูลจากขั้นตอนที่แล้วผ่านเข้าไปยังโครงข่ายประสาทแบบคอนโวลูชั่นเพื่อจำแนกท่าทาง โดยในงานวิจัยชิ้นนี้ได้ใช้ท่ามือเป็นการนับนิ้วมือ 1 นิ้วไปจนถึง 5 นิ้ว เพื่อใช้แทนการสื่อความหมายเป็นข้อความที่ผู้ป่วยต้องการสื่อสารไปถึงผู้ดูแล

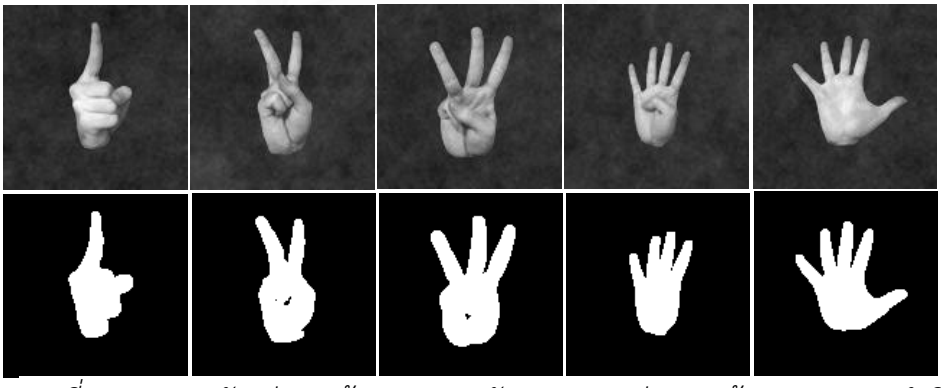
ก่อนที่จะใช้โครงข่ายประสาทแบบคอนโวลูชั่นเพื่อการทำนายจำแนกท่าทางนั้นจำเป็นต้องมีโมเดลสำหรับใช้ในการทำนายเสียก่อน ซึ่งในการเรียนรู้เพื่อที่จะสร้างโมเดลของโครงข่ายประสาทแบบคอนโวลูชั่นนั้นจำเป็นต้องใช้ชุดภาพข้อมูลจำนวนมาก ซึ่งในการทดลองนี้ต้องการทำนายภาพที่เป็นภาพขาว/ดำ ดังนั้นจึงได้ใช้ชุดข้อมูลที่รวบรวมมาจากเว็บไซท์ Kaggle โดย Ryan O'Shea [28] ซึ่งเป็นชุดข้อมูลภาพมือในหมวดสีขาว/ดำ นอกจากนี้เพื่อเพิ่มจำนวนชุดข้อมูลและเพิ่มความหลากหลายของภาพ จึงได้ใช้ภาพจาก Pavel Koryakin [27] ซึ่งเป็นชุดข้อมูลภาพมือในรูปแบบภาพระดับเทามาประกอบด้วย โดยได้ทำการแปลงจากภาพระดับเทาให้เป็นภาพขาว/ดำก่อนที่จะนำไปรวมกับชุดข้อมูลก่อนหน้า โดยรวมแล้วได้ออกมาเป็นชุดข้อมูลขนาด 19000 ภาพซึ่งได้ถูกแบ่งออกเป็น 3 ส่วนดังนี้

- ชุดฝึกสอน (Training Set): เป็นข้อมูลหลักในชุดข้อมูล ใช้สำหรับการฝึกสอนเครื่องจักร

- ชุดตรวจสอบ (Validation Set): เป็นชุดข้อมูลที่ ใช้สำหรับตรวจสอบโมเดลที่ได้จากการฝึกสอนเพื่อปรับค่าตัวแปรต่าง ๆ แล้วหาค่าตัวแปรที่ให้ผลลัพธ์ดีที่สุด
- ชุดทดสอบ (Test Set): เป็นชุดข้อมูลทดสอบ ใช้สำหรับวัดประสิทธิภาพของโมเดลที่ผ่านการฝึกสอนและปรับแก้ค่ามาแล้ว



รูปที่ 3.13 ตัวอย่างชุดข้อมูลภาพขาว/ดำ [28]



รูปที่ 3.14 (บน) ตัวอย่างชุดข้อมูลภาพระดับเทา [27] (ล่าง) ชุดข้อมูลภาพขาว/ดำที่ได้จากการแปลงภาพระดับเทา

โดยในการทดลองนี้ได้มีการแบ่งสัดส่วนของชุดรูปภาพตัวอย่างที่นำมาใช้ในการสร้างโมเดลได้ดังตารางที่ 3.3

ตารางที่ 3.3 การแบ่งสัดส่วนชุดข้อมูลที่ใช้ในการฝึกสอนโมเดล

ท่ามือ	ขนาดของชุดฝึกสอน	ขนาดของชุดตรวจสอบ	ขนาดของชุดทดสอบ
1 นิ้ว	2660 ภาพ	570 ภาพ	570 ภาพ
2 นิ้ว	2660 ภาพ	570 ภาพ	570 ภาพ
3 นิ้ว	2660 ภาพ	570 ภาพ	570 ภาพ
4 นิ้ว	2660 ภาพ	570 ภาพ	570 ภาพ
5 นิ้ว	2660 ภาพ	570 ภาพ	570 ภาพ

หลังจากได้ชุดข้อมูลที่ต้องการแล้ว ก่อนที่จะส่งข้อมูลเข้าไปยังโครงข่ายประสาทแบบคอนโวลูชั่น ภาพแต่ละภาพจะถูกผ่านกระบวนการ Data Augmentation ซึ่งเป็นกระบวนการเพื่อเพิ่มความหลากหลายของชุดข้อมูลด้วยการนำภาพไปปรับแต่ง ย่อ/ขยาย หมุนเอียง กลับภาพ ยืด/หด ฯลฯ เพื่อให้ได้ชุดข้อมูลที่หลากหลายและทนทานต่อการเปลี่ยนแปลง โดยสำหรับการทดลองนี้ได้ใช้ตัวแปรในกระบวนการ Data Augmentation คือ rotation_range=30 zoom_range=[0.9, 1.1] width_shift_range=0.2 height_shift_range=0.2 shear_range=15.0 และ horizontal_flip=True

ข้อมูลขาเข้าทั้งหมดที่ผ่านกระบวนการ Data Augmentation แล้ว จะถูกส่งต่อไปยังโครงข่ายประสาทแบบคอนโวลูชั่นเพื่อทำการเรียนรู้ โดยการในขั้นตอนการเรียนรู้นั้นสามารถออกแบบหรือเลือกใช้โครงสร้างสถาปัตยกรรมของโครงข่ายได้หลากหลายรูปแต่ แต่ในการทดลองนี้ได้ใช้โครงสร้างสถาปัตยกรรมแบบ MobileNetV2 ซึ่งเป็นสถาปัตยกรรมโครงข่ายประสาทคอนโวลูชั่นที่ออกแบบมาโดยเน้นให้มีความเร็วในการประมวลผลสูงและใช้ทรัพยากรระบบต่ำเหมาะสำหรับการใช้งานในอุปกรณ์พกพา อุปกรณ์ IoT หรืออุปกรณ์ที่มีทรัพยาการประมวลผลจำกัด โดยยังมีค่าความแม่นยำอยู่ในระดับค่อนข้างสูง [25, 26]

ในการฝึกสอนแบบโมเดลนั้น ผู้ทดลองได้ทำการทดสอบการฝึกสอนซ้ำหลายครั้ง แล้วจึงเลือกตัวแปรที่ได้ค่าความผิดพลาด (Loss) ของการทำนายต่ำที่สุด โดยได้ตั้งค่าจำนวนรอบการฝึกฝน (Epoch) อยู่ที่ 100 รอบ และมีขนาดการแบ่งชุดข้อมูล (Batch Size) อยู่ที่ 32 ตัวอย่าง โดยเลือกใช้ตัวปรับค่าน้ำหนัก (Optimizer) แบบ Stochastic Gradient Descent หรือ SGD และอัตราการเรียนรู้ (Learning Rate) ที่ 0.0001 โดยทำการฝึกฝนบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ ซีพียู intel core i5-8400 2.8GHz หน่วยความจำขนาด 16 GB และหน่วยประมวลผลกราฟฟิก Nvidia GTX1070 หน่วยความจำขนาด 8 GB โมเดลที่ผ่านการฝึกฝนสำเร็จแล้วจะถูกบันทึกไว้ในฟอร์แมต h5 นามสกุลไฟล์ .model

```
Epoch 80/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1495 - acc: 0.9514 - val_loss: 0.0513 - val_acc: 0.9867
Epoch 81/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1496 - acc: 0.9500 - val_loss: 0.0428 - val_acc: 0.9895
Epoch 82/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1440 - acc: 0.9546 - val_loss: 0.0610 - val_acc: 0.9842
Epoch 83/100
415/415 [==============================] - 83s 200ms/step - loss: 0.1482 - acc: 0.9523 - val_loss: 0.0791 - val_acc: 0.9719
Epoch 84/100
415/415 [==============================] - 85s 204ms/step - loss: 0.1405 - acc: 0.9560 - val_loss: 0.0371 - val_acc: 0.9891
Epoch 85/100
415/415 [==============================] - 83s 200ms/step - loss: 0.1416 - acc: 0.9529 - val_loss: 0.0311 - val_acc: 0.9905
Epoch 86/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1397 - acc: 0.9557 - val_loss: 0.0430 - val_acc: 0.9884
Epoch 87/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1347 - acc: 0.9570 - val_loss: 0.0299 - val_acc: 0.9916
Epoch 88/100
415/415 [==============================] - 84s 204ms/step - loss: 0.1302 - acc: 0.9594 - val_loss: 0.0313 - val_acc: 0.9923
Epoch 89/100
415/415 [==============================] - 85s 204ms/step - loss: 0.1346 - acc: 0.9570 - val_loss: 0.0354 - val_acc: 0.9891
Epoch 90/100
415/415 [==============================] - 84s 202ms/step - loss: 0.1254 - acc: 0.9598 - val_loss: 0.0303 - val_acc: 0.9926
Epoch 91/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1247 - acc: 0.9611 - val_loss: 0.0249 - val_acc: 0.9937
Epoch 92/100
415/415 [==============================] - 84s 202ms/step - loss: 0.1204 - acc: 0.9613 - val_loss: 0.0243 - val_acc: 0.9933
Epoch 93/100
415/415 [==============================] - 84s 202ms/step - loss: 0.1190 - acc: 0.9616 - val_loss: 0.0252 - val_acc: 0.9937
Epoch 94/100
415/415 [==============================] - 84s 202ms/step - loss: 0.1132 - acc: 0.9655 - val_loss: 0.0301 - val_acc: 0.9926
Epoch 95/100
415/415 [==============================] - 85s 204ms/step - loss: 0.1159 - acc: 0.9647 - val_loss: 0.0341 - val_acc: 0.9905
Epoch 96/100
415/415 [==============================] - 84s 202ms/step - loss: 0.1109 - acc: 0.9656 - val_loss: 0.0275 - val_acc: 0.9926
Epoch 97/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1100 - acc: 0.9674 - val_loss: 0.0254 - val_acc: 0.9937
Epoch 98/100
415/415 [==============================] - 84s 203ms/step - loss: 0.1140 - acc: 0.9662 - val_loss: 0.0256 - val_acc: 0.9923
Epoch 99/100
415/415 [==============================] - 85s 204ms/step - loss: 0.1066 - acc: 0.9670 - val_loss: 0.0223 - val_acc: 0.9951
Epoch 100/100
415/415 [==============================] - 87s 210ms/step - loss: 0.1023 - acc: 0.9688 - val_loss: 0.0233 - val_acc: 0.9940
```

รูปที่ 3.15 ตัวอย่างรายงานผลการเรียนรู้ของแบบจำลอง

หลังจากที่ได้โมเดลสำหรับการทำนายแล้ว โมเดลดังกล่าวจะถูกนำไปใช้ร่วมร่วมกับกระบวนการอื่น โดยข้อมูลภาพที่ได้จากการตัดพื้นที่ ROI ในขั้นตอนที่ 3.2.1 และผ่านกระบวนการการจัดเตรียมข้อมูลภาพในขั้นตอนที่ 3.2.2 จะถูกส่งต่อไปยังโครงข่ายประสาทแบบคอนโวลูชั่นที่ใช้โมเดลที่ฝึกฝนไว้และจำแนกออกมาเป็นประเภทท่ามือ 1 นิ้ว – 5 นิ้ว ซึ่งจะมีการแปลผลลัพธ์และแสดงข้อความพร้อมกับการแจ้งเตือนในขั้นตอนถัดไป

3.2.4 การแปลผลข้อความและการแจ้งเตือน

ผลลัพธ์จากโครงข่ายประสาทแบบคอนโวลูชั่นในขั้นตอนที่ 3.2.3 นั้นจะได้ออกมาเป็นการระบุท่าทางมือที่ตรวจจับได้ว่ามีความน่าจะเป็นใกล้เคียงกับท่าใดมากที่สุด ซึ่งท่ามือแต่ละท่าจะถูกกำหนดข้อความไว้ดังตารางที่ 3.4 และข้อความที่ได้จะแสดงไว้ที่มุมซ้ายบนของหน้าจอฝั่งผู้ใช้งาน โดยจะแสดงผลทั้งจำนวนนิ้ว ข้อความและเปอร์เซนต์ความมั่นใจของการทำหน่าย

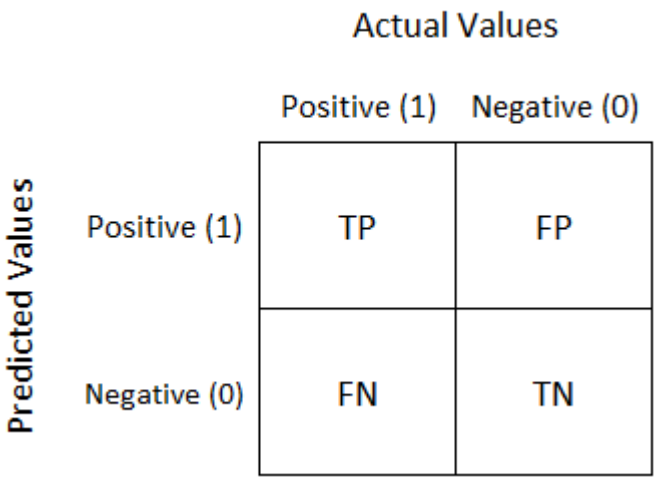
ตารางที่ 3.4 ท่ามือและข้อความที่กำหนด

ท่ามือ	ข้อความ
1 นิ้ว	Need Help
2 นิ้ว	Toilet
3 นิ้ว	Not Feeling Good
4 นิ้ว	Hungry/Thirsty
5 นิ้ว	Emergency

เมื่อได้ผลลัพธ์สุดท้ายแล้ว ข้อความที่ได้มาจะถูกส่งผ่านระบบอินเตอร์เน็ตเพื่อแจ้งเตือนข้อความไปยังสมาร์ทโฟนของผู้ดูแล โดยในงานวิจัยนี้ได้ใช้ LINE Notify API ซึ่งเป็นช่องทางการเชื่อมต่อข้อมูลระหว่างนักพัฒนากับเซิร์ฟเวอร์เพื่อส่งข้อมูลการแจ้งเตือนไปยังแอพพลิเคชั่น LINE ซึ่งเป็นหนึ่งในแพลทฟอร์มการรับ-ส่งข้อความแบบทันที (Instant Messenger) ที่ได้รับความนิยมสูงที่สุดในประเทศไทย [29]

บทที่ 4
ผลการทดลอง

ในการวัดประสิทธิภาพในการทำนายของระบบนั้น ในการทดลองนี้จะใช้การคำนวณจากตารางประเมินผลลัพธ์หรือ Confusion Mattrix ซึ่งเป็นวิธีการที่นิยมใช้สำหรับวัดประสิทธิภาพของการจำแนก



รูปที่ 4.1 Confusion Mattrix

ในการวัดผลด้วย Confusion Mattrix นั้น จะแบ่งผลการทำนายออกเป็น 4 กรณี ได้แก่

- True Positive (TP) คือกรณีที่ทำนายว่า “จริง” และมีค่าที่ถูกต้องเป็น “จริง”
- False Positive (FP) คือกรณีที่ทำนายว่า “จริง” แต่มีค่าที่ถูกต้องเป็น “ไม่จริง”
- True Negative (TN) คือกรณีที่ทำนายว่า “ไม่จริง” และมีค่าที่ถูกต้องเป็น “ไม่จริง”
- False Negative (FN) คือกรณีที่ทำนายว่า “ไม่จริง” แต่มีค่าที่ถูกต้องเป็น “จริง”

ซึ่งค่าทั้ง 4 สามารถนำมาใช้คำนวณประสิทธิภาพได้หลายวิธี โดยในงานวิจัยนี้จะเน้นการวัดผลไปที่ค่า Precision Recall และ F1-Score

- Precision หรือค่าความแม่นยำ เป็นค่าที่บ่งบอกถึงความถูกต้องของค่าที่ทำนายออกมา โดยคำนวณได้จาก

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall หรือค่าความไวของการทำนาย เป็นค่าบ่งบอกถึงความสามารถในการตรวจพบ โดยคำนวณได้จาก

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- F1-Score เป็นค่าเฉลี่ยฮาร์โมนิกระว่าง Precision และ Recall หาได้จาก

$$\text{F1} - \mathit{Score} = 2\left(\frac{\text{Precision} * \text{ Recall}}{\text{Precision} + \text{Recall}}\right)$$

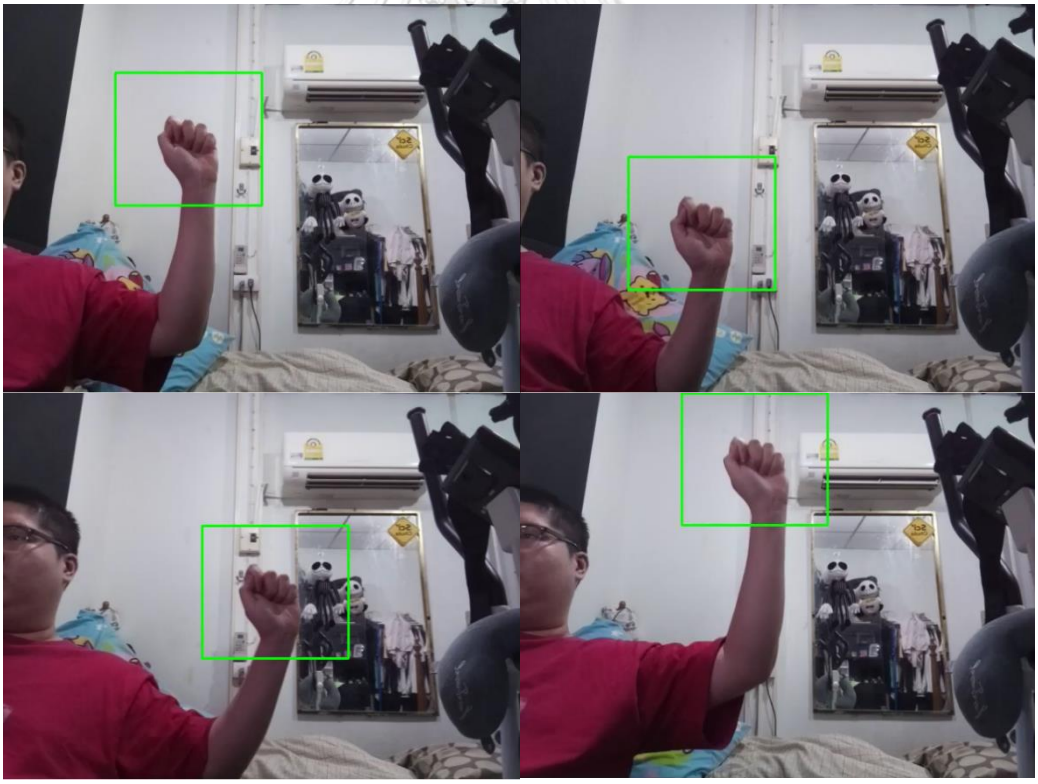
4.1 ผลการทดลองการตรวจหามือด้วยคุณลักษณะแบบฮาร์

ในการวัดผลการทดลองการตรวจหาท่ามือด้วยคุณลักษณะแบบฮาร์ ผู้ทดลองได้ทำการแสดงท่ากำมือโดยมีการเคลื่อนไหวและขยับมือเล็กน้อย โดยอยู่ห่างจากตัวกล้องประมาณ 50 – 70 เซนติเมตรในสภาพแสงแบบ Cool White และเปิดระบบเฉพาะ Stage ที่ 1 โดยทำการนับเฟรมที่มีการทำงานจำนวน 300 เฟรม เพื่อทำการตรวจจับท่ากำมือ และตีกรอบรอบบริเวณที่ตรวจจับวัตถุได้โดยได้ผลการทดลองดังตารางที่ 4.1

ตารางที่ 4.1 ผลการตรวจหาท่ากำมือด้วยคุณลักษณะฮาร์ในสภาพแสง Cool White

ท่ามือ	จำนวนเฟรม	จำนวนเฟรมที่ตรวจพบท่ามือ (TP)	จำนวนเฟรมที่ตรวจไม่พบท่ามือ (FN)	จำนวนเฟรมที่ตรวจพบอย่างอื่น (FP)
กำมือ	300	295	5	0

ผลการทดลองดังตารางที่ 4.1 แสดงให้เห็นว่าการตรวจจับท่ากำมือในสภาพแสง Cool White นั้น ให้ผลลัพธ์เป็นที่น่าพอใจ โดยมีค่า Precision Recall และ F1-Score เท่ากับ 1.00 0.983 และ 0.991 ตามลำดับ และจากการทดลองยังพบว่าระบบสามารถตรวจจับท่ากำมือได้แม้จะมีพื้นหลังซับซ้อน และยังสามารถทำงานได้แม้จะการเคลื่อนไหวและเปลี่ยนแปลงพื้นหลังรอบมือไปตามการเคลื่อนที่มือของผู้ทดลองดังแสดงในรูปที่ 4.2



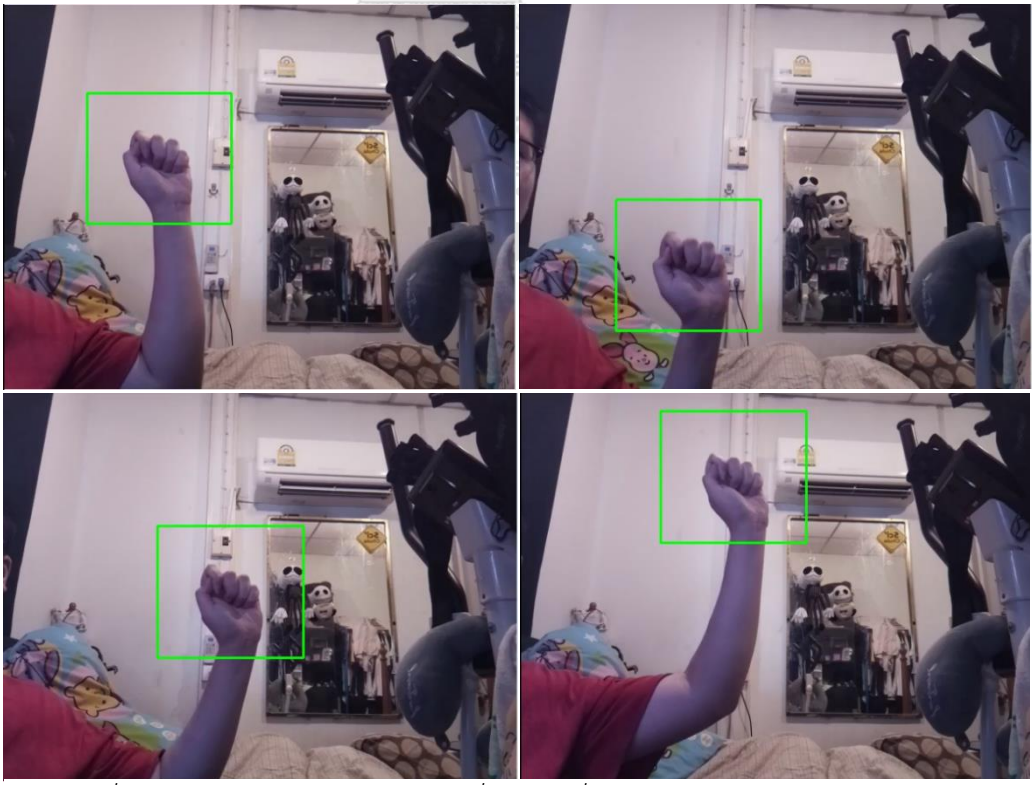
รูปที่ 4.2 การตรวจจับท่ากับมือขณะที่มีการเคลื่อนไหวในสภาพแสง Cool White

นอกจากนี้ ผู้ทดลองยังได้ทำการทดลองในสภาพแสงที่ต่างออกไป โดยได้ทำการแสดงท่ากำมือโดยมีการเคลื่อนไหวและขยับมือเล็กน้อย โดยอยู่ห่างจากตัวกล้องประมาณ 50 – 70 เซนติเมตรในสภาพแสงแบบ Warm White และเปิดระบบเฉพาะ Stage ที่ 1 โดยทำการนับเฟรมที่มีการทำงานจำนวน 300 เฟรม เพื่อทำการตรวจจับท่ากำมือ และตีกรอบรอบบริเวณที่ตรวจจับวัตถุได้โดยได้ผลการทดลองดังตารางที่ 4.2

ตารางที่ 4.2 ผลการตรวจหาท่ากำมือด้วยคุณลักษณะฮาร์ในสภาพแสง Warm White

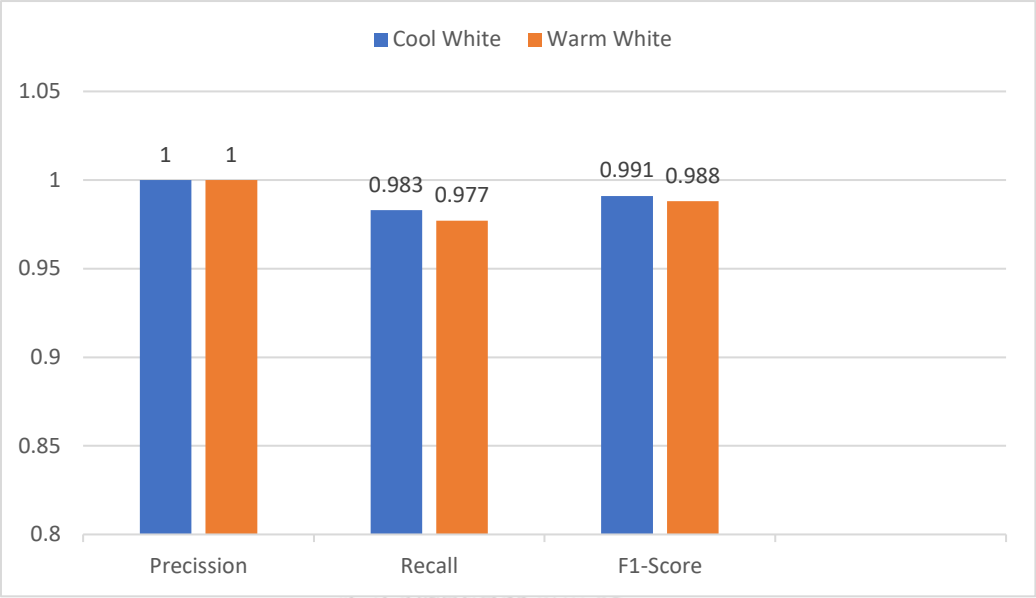
ท่ามือ	จำนวนเฟรม	จำนวนเฟรมที่ตรวจพบท่ามือ (TP)	จำนวนเฟรมที่ตรวจไม่พบท่ามือ (FN)	จำนวนเฟรมที่ตรวจพบอย่างอื่น (FP)
กำมือ	300	293	7	0

ผลการทดลองจากตารางที่ 4.2 แสดงให้เห็นว่าการตรวจจับท่ากำมือในสภาพแสง Warm White นั้น ให้ผลลัพธ์เป็นที่น่าพอใจ โดยมีค่า Precision Recall และ F1-Score เท่ากับ 1.000 0.977 และ 0.988 ตามลำดับ และจากการทดลองยังพบว่าระบบสามารถ



รูปที่ 4.3 การตรวจจับท่ากับมือขณะที่มีการเคลื่อนไหวในสภาพแสง Warm White

ตรวจจับท่ากำมือได้แม้จะมีพื้นหลังซับซ้อน และยังสามารถทำงานได้แม้มีการเคลื่อนไหวและเปลี่ยนแปลงพื้นหลังรอบมือไปตามการเคลื่อนที่มือของผู้ทดลองดังแสดงในรูปที่ 4.3



รูปที่ 4.4 กราฟเปรียบเทียบประสิทธิภาพของการตรวจจับท่ามือด้วยคุณลักษณะฮาร์ในสภาพแสงแบบ Cool White และ Warm White

เมื่อนำผลการทดลองจากสภาพแสงที่แตกต่างกันมาสร้างกราฟเปรียบเทียบประสิทธิภาพจะเห็นว่าการทดลองการตรวจหามือด้วยคุณลักษณะแบบฮาร์นั้นให้ผลลัพธ์ที่ใกล้เคียงกันและไม่ได้แตกต่างกันอย่างมีนัยยะสำคัญแม้ว่าจะมีการทดลองในสภาพแสงที่แตกต่างกันออกไปดังแสดงในรูปที่ 4.4 ทั้งนี้เนื่องจากคุณลักษณะแบบฮาร์นั้นจะสนใจเฉพาะระดับความเข้มของพิกเซลในหมวดสีระดับเทาเท่านั้น การเปลี่ยนโทนสีของแสงจึงไม่ส่งผลกระทบต่อผลลัพธ์มากนักตราบใดที่ยังมีแสงสว่างเพียงพอ

4.2 ผลการทดลองการจัดเตรียมข้อมูลภาพ

บริเวณในภาพที่ตรวจจับท่ามือได้จากขั้นตอนก่อนหน้าจะถูกตีกรอบและนำมาใช้เป็น ROI สำหรับจำแนกภาพต่อไป แต่ก่อนที่จะเข้าสู่ขั้นตอนจำแนกนั้น ภาพที่ถูกตัดมาเฉพาะบริเวณ ROI จะต้องผ่านการจัดเตรียมข้อมูลภาพเบื้องต้นก่อนที่จะส่งไปยังโครงข่ายประสาทแบบคอนโวลูชั่น

- การปรับขนาดภาพ (Resize): ภาพ ROI ที่ได้มาจากการตรวจหาท่ามือ ถูกปรับขนาดให้เป็น 128 พิกเซล x 128 พิกเซล

รูปที่ 4.5 ตัวอย่างภาพท่ามือที่ผ่านการปรับขนาดแล้ว

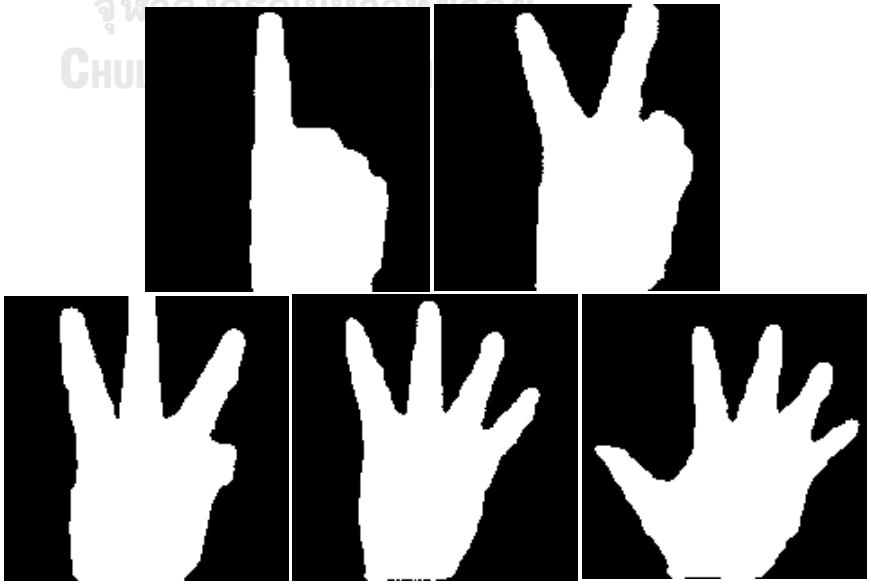
- การทำให้ภาพราบเรียบ (Smoothing): ภาพที่ถูกปรับขนาดแล้ว จะถูกทำให้เบลอด้วย Gaussian Filter ขนาด 5 พิกเซล x 5 พิกเซล จะสังเกตเห็นว่าภาพจะดูเรียบขึ้นและขอบวัตถุจะคมน้อยลง

รูปที่ 4.6 ตัวอย่างภาพท่ามือที่ผ่านการทำให้ภาพราบเรียบแล้ว

- การเปลี่ยนหมวดสี: ภาพที่ผ่าน Gaussian Filter แล้ว จะถูกเปลี่ยนหมวดสีเป็น YCbCr ดังแสดงในรูป 4.7

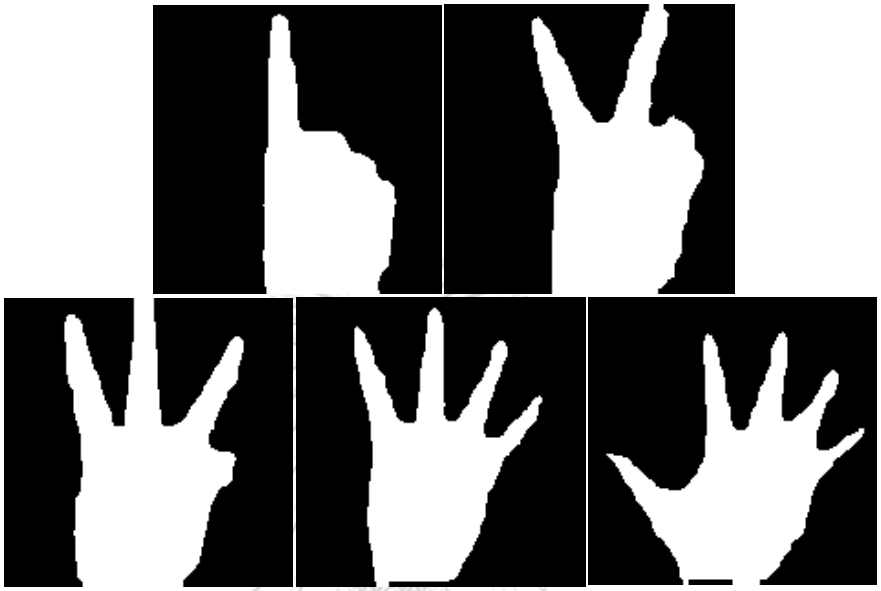
รูปที่ 4.7 ตัวอย่างภาพท่ามือที่ถูกเปลี่ยนเป็นโหมดสี YCbCr

- การแบ่งสีพื้นผิวมนุษย์ (Human Skin Segmentation): รูปที่ 4.8 แสดงให้เห็นภาพที่ได้จากการแบ่งสีพื้นผิวมนุษย์โดยการกำหนดขอบเขต โดยในสถานการณ์ตัวอย่างได้ใช้ขอบเขตอยู่ที่ $135 < Cr < 175$ และ $96 < Cb < 136$ ทั้งนี้ค่าขอบเขตดังกล่าวอาจมากหรือน้อยกว่านี้ได้ขึ้นกับสภาพแสงและสภาพแวดล้อมของผู้ใช้ โดยผู้ใช้สามารถปรับค่าได้เองแบบเรียลไทม์



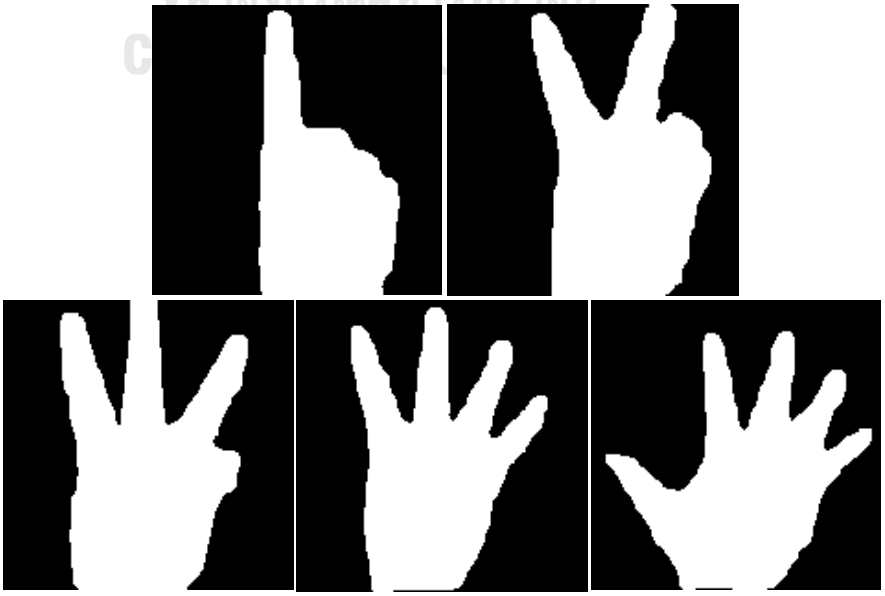
รูปที่ 4.8 ตัวอย่างภาพท่ามือที่ผ่านการแบ่งสีพื้นผิวมนุษย์

- การกัดกร่อน (Erosion): การกัดกร่อนเป็นการช่วยขจัด Noise หรือสัญญาณรบกวนขนาดเล็กที่อาจพบในภาพ อย่างไรก็ตามเนื่องจากสถานการณ์ตัวอย่างนั้น ภาพที่ได้หลังจากการแบ่งสีพื้นผิวมนุษย์ไม่มีสัญญาณรบกวนปรากฏอยู่ จึงสังเกตเห็นเพียงแค่ขอบวัตถุที่ถูกกัดกร่อนไป



รูปที่ 4.9 ตัวอย่างภาพท่ามือที่ผ่านการกัดกร่อน

- การพองตัว (Dilation): จากรูปที่ 4.10 จะสังเกตเห็นว่าพิกเซลที่แหว่งไปถูกเติมเต็มเข้ามา



รูปที่ 4.10 ตัวอย่างภาพท่ามือที่ผ่านการพองตัว

ในขั้นตอนนี้ ภาพพื้นที่ ROI ที่ได้จากการตรวจจับท่ามือด้วยคุณลักษณะฮาร์ได้ผ่านการบวนการจัดเตรียมข้อมูลภาพดังรูปที่ 4.5 – 4.10 โดยได้ผลลัพธ์เป็นภาพท่ามือในหมวดสีขาว/ดำซึ่งมีความซับซ้อนน้อยลงเมื่อเทียบกับภาพต้นฉบับ โดยภาพผลลัพธ์ถูกกำหนดให้มีขนาด 128 พิกเซล x 128 พิกเซล ซึ่งเป็นรูปแบบที่ตรงกับโครงข่ายประสาทแบบคอนโวลูชั่นที่ได้ทำการฝึกสอนไว้

4.3 ผลการทดลองการจำแนกท่ามือด้วยโครงข่ายประสาทแบบคอนโวลูชั่น

ในการฝึกสอนโครงข่ายประสาทแบบคอนโวลูชั่นด้วยขั้นตอนที่ระบุไว้ในหัวข้อ 3.2.3 นั้น ระหว่างการฝึกสอนตัวโปรแกรมจะมีการสร้างกราฟแสดงค่าความผิดพลาดและค่าความแม่นยำ ซึ่งได้ผลลัพธ์ดังรูปที่ 4.11

รูปที่ 4.11 เป็นกราฟแสดงค่าความผิดพลาดและความแม่นยำในระหว่างรอบการฝึกฝน โดยเส้นสีแดงและเส้นสีม่วงคือกราฟที่ได้จากการวัดผลของโมเดลในระหว่างการฝึกด้วยชุดข้อมูลฝึกสอน (Training Set) เส้นสีดำและเส้นสีน้ำเงินคือกราฟที่ได้จากวัดผลของโมเดลหลังจากการฝึกแต่ละรอบ (Epoch) ด้วยชุดข้อมูลตรวจสอบ (Validation Set) จากกราฟจะสังเกตุเห็นว่าค่า Training Accuracy จะค่อย ๆ เพิ่มขึ้นตามจำนวนรอบการฝึก ซึ่งสอดคล้องกับ Training Loss ที่ลดลง สำหรับ Validation Loss และ Validation Accuracy นั้น ในช่วงแรกของการฝึกจะให้ค่าที่แกว่งค่อนข้างสูงและยังไม่เสถียรนัก แต่

รูปที่ 4.11 กราฟแสดงค่าความผิดพลาดและค่าความแม่นยำของโมเดล

หลังจากการฝึกในรอบที่ 60 เป็นต้นไปจะสังเกตเห็นว่ากราฟมีแนวโน้มลู่เข้าไปในทิศทางเดียวกันกับค่าการฝึกที่ได้จากชุดข้อมูลฝึกสอน (Training Set) โดยหลังจากการฝึกครบ 100 รอบ โมเดลที่ได้มี ค่า Training Accuracy และ Validation Accuracy อยู่ที่ 96.98% และ 99.40% ตามลำดับ

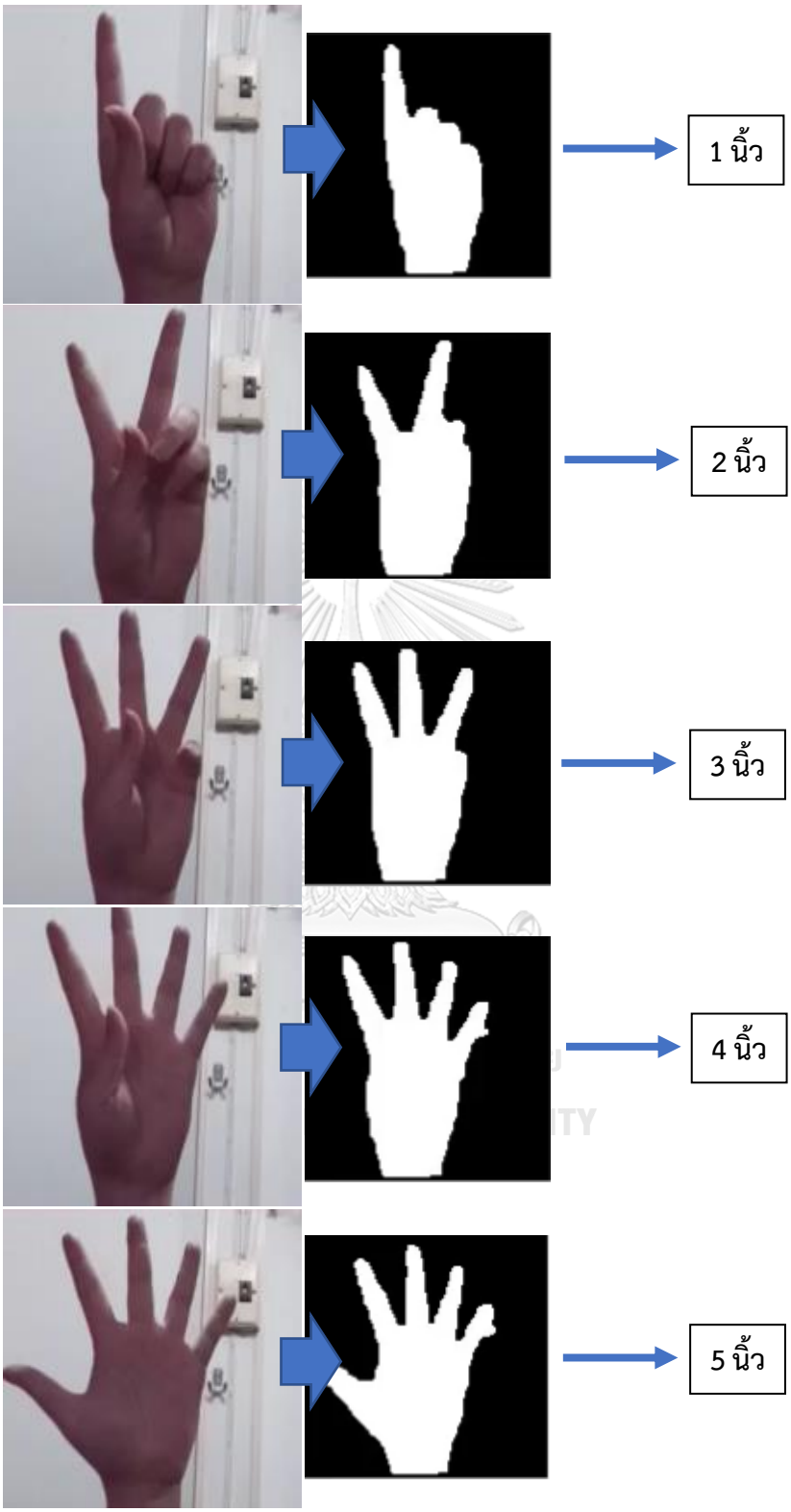
หลังจากการฝึกสอนโมเดลแล้วเสร็จ ระบบได้มีการทดสอบประสิทธิภาพของโมเดลโดยการจำแนกภาพท่ามือ 1 นิ้ว – 5 นิ้ว โดยทดสอบกับชุดข้อมูลทดสอบ (Testing Set) ซึ่งเป็นชุดข้อมูลภาพที่โมเดลไม่เคยเห็นมาก่อนจำนวน 570 ภาพต่อ 1 ท่ามือ ซึ่งได้ผลลัพธ์ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดสอบประสิทธิภาพของโมเดลด้วยชุดข้อมูลทดสอบ

ท่ามือ	จำนวนภาพทดสอบ	Precision	Recall	F1-Score
1 นิ้ว	570	1.00	1.00	1.00
2 นิ้ว	570	1.00	1.00	1.00
3 นิ้ว	570	1.00	0.99	1.00
4 นิ้ว	570	0.99	1.00	1.00
5 นิ้ว	570	1.00	1.00	1.00

จากผลการทดสอบในตารางที่ 4.3 สังเกตเห็นได้ว่าประสิทธิภาพของโมเดลมีค่าสูงมากอยู่ในระหว่าง 0.99 – 1.00 ในทุกประเภทการทดสอบ อย่างไรก็ตามเนื่องจากภาพในชุดข้อมูลทดสอบเป็นภาพที่ค่อนข้างสมบูรณ์ ไม่มีสัญญาณรบกวนและมีคุณภาพใกล้เคียงชุดข้อมูลฝึกสอนเพราะมาจากแหล่งที่มาเดียวกัน จึงทำให้โมเดลสามารถทำนายท่ามือได้ในระดับความแม่นยำสูงมาก

โมเดลที่ผ่านการฝึกสอนแล้วได้ถูกนำไปใช้จำแนกภาพท่ามือที่เป็นผลลัพธ์จากการจัดเตรียมข้อมูลภาพในขั้นตอนก่อนหน้า โดยในการทดลองนี้เป็นการจำลองสถานการณ์ด้วยการใช้งานระบบจริงและจับเฟรมภาพโดยนับเฉพาะเฟรมที่มีการจำแนกเกิดขึ้นจำนวน 600 เฟรมภาพต่อ 1 ท่ามือ แล้วจึงบันทึกผล



รูปที่ 4.12 ภาพที่ได้จากการหาพื้นที่ ROI และการจัดเตรียมข้อมูลภาพเพื่อเป็นภาพขาเข้าสำหรับโครงข่ายประสาทแบบคอนโวลูชั่นและผลการจำแนกท่ามือ

ตารางที่ 4.4 ผลลัพธ์จากการจำแนกท่ามือด้วยโครงข่ายประสาทแบบคอนโวลูชั่น

ท่ามือ	จำนวนเฟรม	จำนวนเฟรมที่จำแนกได้ถูกต้อง	จำนวนเฟรมที่จำแนกผิด	เปอร์เซ็นต์ความถูกต้อง
1 นิ้ว	600	600	0	100.00%
2 นิ้ว	600	599	3	99.50%
3 นิ้ว	600	599	1	99.83%
4 นิ้ว	600	578	22	96.33%
5 นิ้ว	600	600	0	100.00%

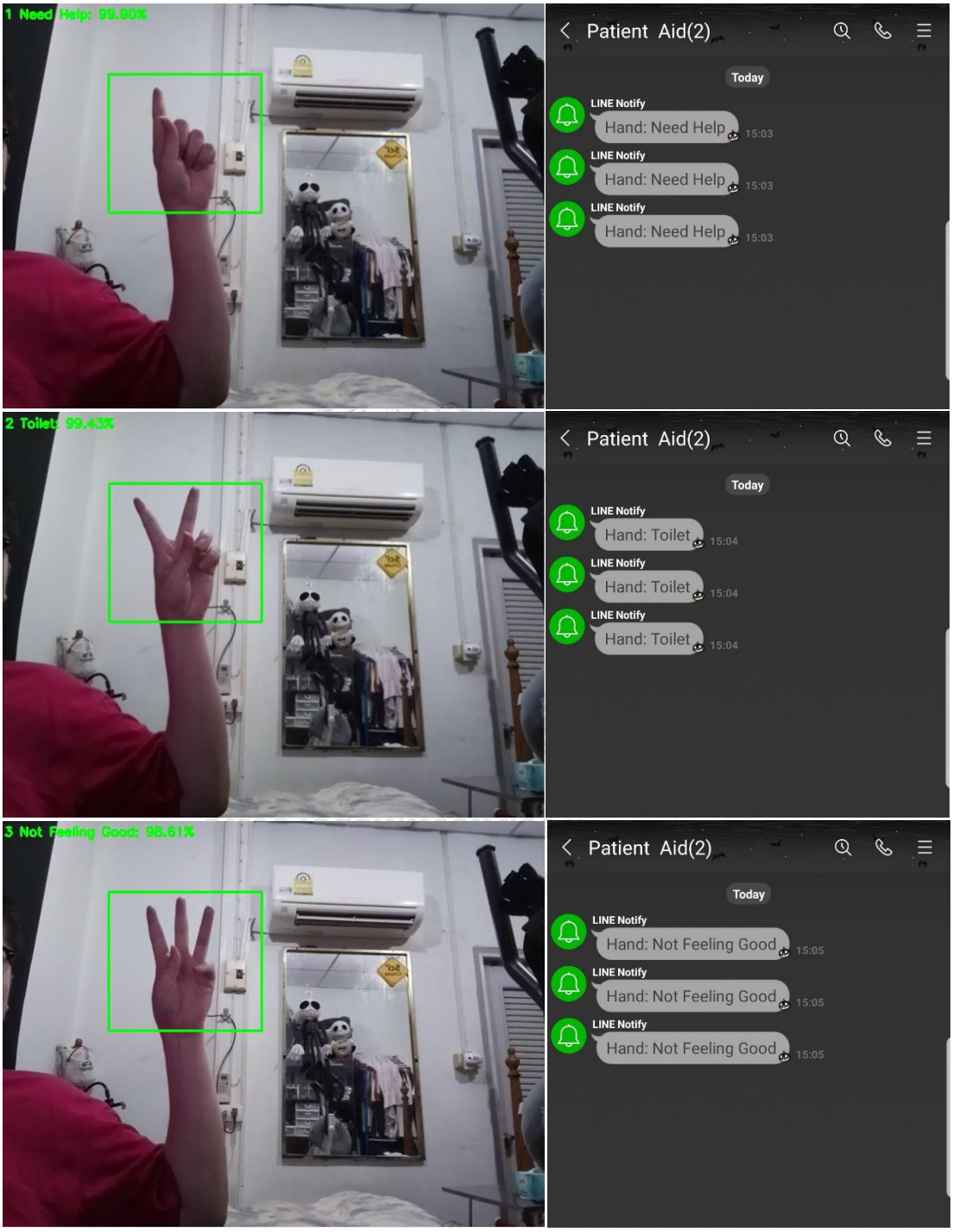
ตารางที่ 4.4 แสดงผลลัพธ์จากการทดลองจำแนกท่ามือด้วยโคงข่ายประสาทแบบคอนโวลูชั่นจำนวน 600 เฟรมต่อ 1 ท่ามือ โดยผลลัพธ์จากการทำนายที่ได้มีความแม่นยำอยู่ในระดับสูง มีเปอร์เซนต์ความถูกต้องมากกว่า 95% ในทุกท่ามือที่ทดลอง โดยท่ามือ 1 นิ้ว และ 5 นิ้วเป็นท่ามือที่มีเปอร์เซ็นต์ความแม่นยำสูงที่สุด โดยมีค่า 100% และท่ามือ 4 นิ้ว เป็นท่ามือที่มีเปอร์เซ็นต์ความแม่นยำต่ำที่สุด โดยมีค่า 96.33%

4.4 ผลการทดลองการแปลผลข้อความและการแจ้งเตือน

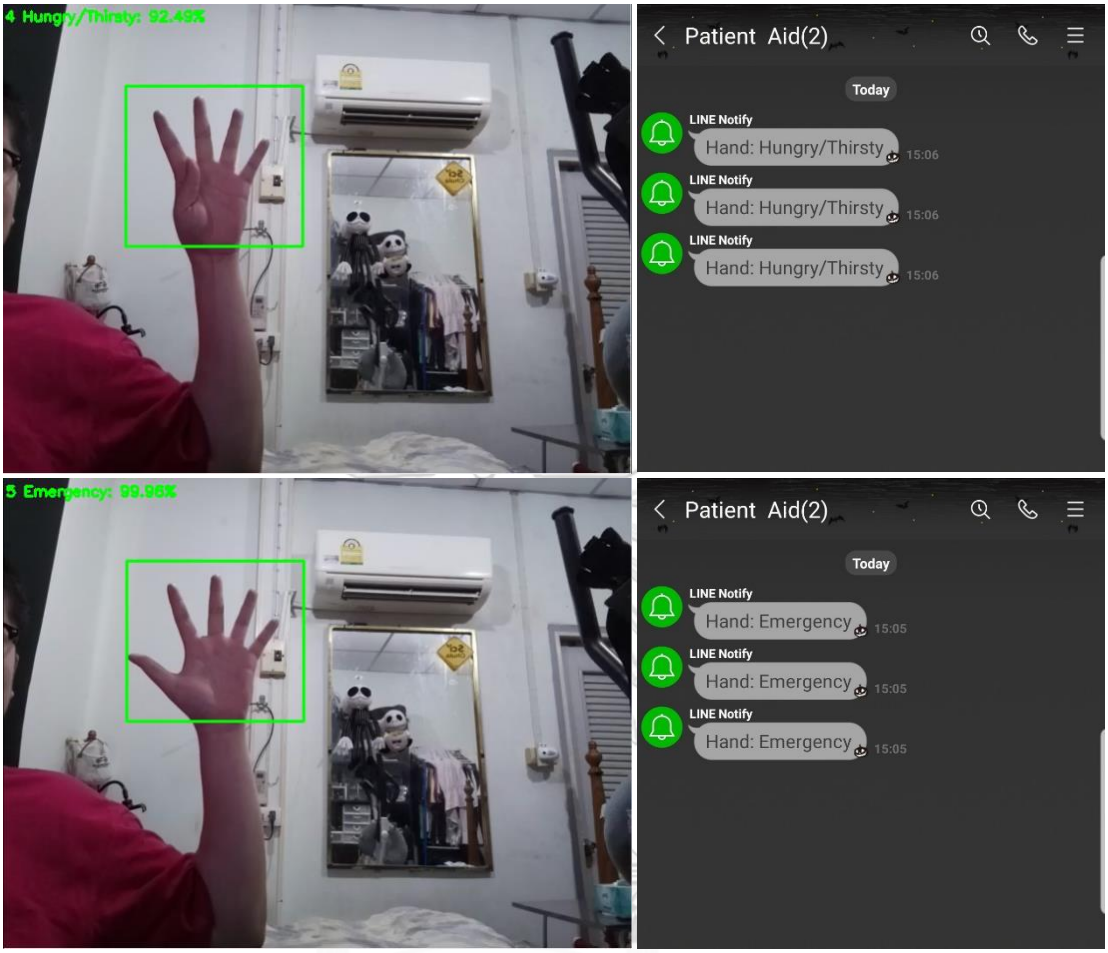
ผลการจำแนกท่ามือในขั้นตอนก่อนหน้า ได้ถูกกำหนดคู่กับข้อความตามที่ได้กำหนดไว้เพื่อให้ผู้ป่วย/ผู้สูงอายุสามารถสื่อสารไปยังผู้ดูแลได้ โดยเมื่อระบบสามารถจำแนกท่ามือได้ก็จะแสดงผล ตัวเลขจำนวนนิ้วมือ ข้อความแจ้งเตือนที่กำหนด และเปอร์เซ็นต์ความมั่นใจในการทำนาย โดยแสดงเป็นตัวอักษรสีเขียวไว้บริเวณมุมซ้ายบนของหน้าจอฝั่งผู้ใช้งาน พร้อมกันนี้ ข้อความแจ้งเตือนที่กำหนดก็ได้ถูกส่งผ่านระบบอินเตอร์เน็ตไปยังแอพพลิเคชั่น LINE บนโทรศัพท์พกพาของผู้ดูแล

เมื่อเปิดใช้งานระบบทุกอย่าง จากการทดลองใช้งาน พบว่าระบบมีความแม่นยำในระดับสูง แต่เนื่องจากอุปกรณ์ Raspberry Pi นั้นเป็นอุปกรณ์ที่สร้างขึ้นเพื่อให้มีราคาถูกและมีทรัพยากรในการประมวลผลที่จำกัด จึงทำให้ระบบมีดีเลย์ในการแสดงผลและแจ้งเตือน

เล็กน้อยประมาณ 0.5 – 1 วินาที ซึ่งยังถือว่าอยู่ในระดับที่สามารถใช้งานเป็นระบบแจ้งเตือนด้วยข้อความแบบเรียลไทม์ได้



รูปที่ 4.13 ภาพแสดงข้อความบนหน้าจอฝั่งผู้ใช้งานและข้อความแจ้งเตือนในแอพพิเคชั่น LINE ของผู้ดูแล



รูปที่ 4.14 ภาพแสดงข้อความบนหน้าจอฝั่งผู้ใช้งานและข้อความแจ้งเตือนในแอพพิเคชั่น LINE ของผู้ดูแล (ต่อ)

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยชิ้นนี้มีจุดประสงค์เพื่อการออกแบบระบบแจ้งเตือนบนอุปกรณ์ Raspberry Pi เพื่อช่วยให้ผู้ป่วย/ผู้สูงอายุที่มีความลำบากในการสื่อสารด้วยเสียงสามารถสื่อสารกับผู้ดูแลได้โดยง่ายด้วยการใช้ท่าทาง มือสื่อสารแทนคำพูดที่กำหนดและแจ้งเตือนไปยังโทรศัพท์พกพาของผู้ดูแล

ในการออกแบบระบบ ผู้ดูแลได้ทำการแบ่งการทำงานของระบบออกเป็น 2 ส่วนหลัก ๆ โดยส่วนแรกเป็นสแตนด์บายสเตจที่จะไม่มีการแจ้งเตือนใด ๆ และทำหน้าที่เพียงตรวจจับท่ากำมือซึ่งเป็นท่าที่ได้กำหนดไว้เพื่อเป็นสัญญาณในการเข้าสู่สเตจถัดไป โดยในการตรวจจับท่ากำมือ ในงานวิจัยชิ้นนี้ได้ใช้เทคนิคการตรวจจับวัตถุด้วยคุณลักษณะฮาร์ ซึ่งในการฝึกสอนโมเดลสำหรับตรวจจับ ผู้วิจัยได้ใช้ภาคท่ากำมือสำหรับฝึกสอนจำนวน 1705 ภาพ และภาพอ้างอิงที่ไม่มีส่วนประกอบของท่ากำมืออยู่ในภาพอีก 8730 ภาพ โดยกำหนดขนาดภาพขาเข้าที่ 24 พิกเซล x 24 พิกเซล และจำนวนชั้นของการเรียนรู้ที่ 20 ชั้น โดยใช้อัตราการเจอขั้นต่ำเท่ากับ 0.995 และอัตราการผิดพลาดเชิงบวกสูงสุดเท่ากับ 0.5 ซึ่งจากผลการทดลอง โมเดลดังกล่าวให้ผลลัพธ์ Precision Recall และ F1-Score เท่ากับ 1.00 0.983 และ 0.991 ตามลำดับในสภาพแสง Cool White และให้ผลลัพธ์ Precision Recall และ F1-Score เท่ากับ 1.000 0.977 และ 0.988 ตามลำดับในสภาพแสง Warm White

ตำแหน่งของมือที่ตรวจจับได้ในสเตจที่ 1 ถูกใช้เพื่อกำหนด ROI สำหรับสเตจที่ 2 ซึ่งเป็นส่วนของการจัดเตรียมข้อมูลภาพและการจำแนกท่ามือในขั้นตอนต่อไป โดยภาพในบริเวณ ROI จะถูกจัดเตรียมเพื่อให้มีความเหมาะสมผ่านการบวนการต่าง ๆ เช่น การปรับขนาด การปรับภาพให้เรียบ การเปลี่ยนหมวดสี การแบ่งส่วนพื้นผิวมนุษย์ การกัดกร่อนและการพองตัว ซึ่งทำให้ได้ผลลัพธ์เป็นภาพขาว/ดำ ขนาด 128 พิกเซล x 128 พิกเซล ซึ่งเป็นรูปแบบที่จำเป็นสำหรับการจำแนกท่ามือในขั้นตอนต่อไป

ในขั้นตอนการจำแนกท่ามือสำหรับงานวิจัยชิ้นนี้ ได้ใช้โครงข่ายประสาทแบบคอนโวลูชั่นในการจำแนก ซึ่งได้มีการฝึกสอนโมเดลด้วยชุดข้อมูลตัวอย่างที่เป็นภาพท่ามือ 1 นิ้ว – 5 นิ้วในหมวดสีภาพขาว/ดำจำนวน 19000 ภาพ โดยแบ่งเป็นชุดข้อมูลฝึกสอน 13300 ภาพ ชุดข้อมูลตรวจสอบจำนนวน 2850 ภาพ และชุดข้อมูลทดสอบจำนวน 2850 ภาพ รวมทั้งได้มีการใช้กระบวนการ Data Augmentation เพื่อเพิ่มความหลากหลายของข้อมูล และใช้สถาปัตยกรรมแบบ MobileNetV2 ในการฝึกสอน โดยมีจำนวนรอบการฝึกสอนที่ 100 รอบ และมีขนาดการแบ่งชุดข้อมูลที่ 32 ตัวอย่าง โดยเลือกใช้ตัวปรับค่าน้ำหนักแบบ Stochastic Gradient Descent และอัตราการเรียนรู้ที่ 0.0001 ซึ่งการฝึกสอนโมเดลให้ค่า Training Accuracy และ Validation Accuracy ของโมเดลอยู่ที่ 96.98% และ 99.40% ตามลำดับ โดยการทดสอบกับชุดข้อมูลทดสอบพบว่าทุกท่ามือมีค่า Precision Recall และ F1-Score อยู่ระหว่าง 0.99 – 1.00 และจากการทดสอบในสถานการณ์จำลองพบว่ามีเปอร์เซนต์ความถูกต้องมากกว่า 95% ในทุกท่ามือที่ทดลอง โดยท่ามือ 1 นิ้วและ 5 นิ้ว เป็นท่ามือที่มีเปอร์เซ็นต์ความแม่นยำสูงที่สุดโดยมีค่า 100% และท่ามือ 4 นิ้วเป็นท่ามือที่มีเปอร์เซ็นต์ความแม่นยำต่ำที่สุดโดยมีค่า 96.33%

ผลการจำแนกท่ามือได้ถูกกำหนดข้อความสำหรับการสื่อสารไว้ โดยเมื่อระบบจำแนกท่ามือใด ๆ ก็จะแสดงผลตัวเลขจำนวนนิ้วมือ ข้อความแจ้งเตือนที่กำหนดและเปอร์เซ็นต์ความมั่นใจในการทำนาย โดยแสดงเป็นตัวอักษรสีเขียวไว้บริเวณมุมซ้ายบนของหน้าจอฝั่งผู้ใช้งาน พร้อมกับส่งข้อความแจ้งเตือนที่กำหนดไว้ผ่านระบบอินเตอร์เน็ตไปยังแอพพลิเคชั่น LINE บนโทรศัพท์พกพาของผู้ดูแล ซึ่งให้ผลลัพธ์ในการใช้งานที่ดี แต่เนื่องจากทรัพยากรในการประมวลของของอุปกรณ์มีจำกัด ทำให้มีดีเลย์ในการแสดงผลได้เล็กน้อย

5.2 ข้อเสนอแนะ

งานวิจัยชิ้นนี้สามารถนำไปต่อยอดพัฒนาเพิ่มเติมเพื่อใช้งานได้จริงได้ โดยอาจปรับปรุงระบบให้ใช้งานง่ายขึ้น เพิ่ม GUI เพื่อให้สะดวกต่อผู้ใช้ ออกแบบแพ็คเก็จเพื่อให้เหมาะสมกับการติดตั้งในสถานการณ์จริง หรือหากใช้งานบนอุปกรณ์ที่มีทรัพยาการประมวลผลมากขึ้นก็อาจใช้เทคนิคอื่นร่วมด้วยเช่น SSD หรือ YOLO เป็นต้น

บรรณานุกรม

1. Dunn, L.J. *HCI factors affecting quality of information in crisis management systems*. in *Sixth Australian Conference on Computer-Human Interaction*. 1996. Hamilton.
2. Khan, M.N.H., et al., *Speech based text correction tool for the visually impaired*, in *18th International Conference on Computer and Information Technology (ICCIT)*. 2015: Dhaka. p. 150-155.
3. Shuang, L., et al., *A study on human-machine interaction computer games robot*, in *29th Chinese Control And Decision Conference (CCDC)*. 2017: Chongqing. p. 7643-7648.
4. Archarya, C., H. Thimbleby, and P. Oladimeji, *Human computer interaction and medical devices*, in *24th BCS Interaction Specialist Group Conference*. 2010. p. 168-176.
5. Sonka, M., V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*. 1993: Chapman & Hall.
6. Forsyth, D.A. and J. Ponce, *Computer Vision: A Modern Approach*. 2002: Pearson.
7. Kumar, A. and S. Malhotra. *Real-time Human Skin Color Detection Algorithm using Skin Color Map*. in *2015 2nd International Conference on "Computing for Sustainable Global Development*. 2015. New Delhi.
8. Shaik, K.B., et al., *Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space* in *3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)*. 2015.
9. Chhibber, N. *Skin Detection Using OpenCV Python*. 2018; Available from: https://nalinc.github.io/blog/2018/skin-detection-python-opencv/.
10. Bovik, A., *The Essential Guide to Image Processing*. Second Edition ed. 2009: Academic Press.
11. Viola, P. and M. Jones. *Rapid object detection using a boosted cascade of simple features*. in *the 2001 IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition. 2001. CVPR.

12. Lienhart, R. and J. Maydt, *An extended set of Haar-like features for rapid object detection*, in *Proceedings. International Conference on Image Processing*. 2002. p. I-900-I-903.
13. Mita, T., T. Kaneko, and O. Hori, *Joint Haarlike Features for Face Detection*, in *Proc. Int'l Conf. Computer Vision*. 2005. p. 1619-1626.
14. Crow, F., *Summed-area tables for texture mapping*, in *the 11th annual conference on Computer graphics and interactive techniques*. 1984. p. 207–212.
15. Freund, Y. and R.E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of Computer and System Sciences, 1997. **55**: p. 119-139.
16. McCulloch, W.S. and W. Pitts, *Neurocomputing: Foundations of Research*. 1988: MIT Press.
17. Rosenblatt, F., *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. 1962: Spartan.
18. J.Werbos, P., *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. 1974, Harvard University.
19. al, Y.L.e. *Efficient BackProp*. in *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*. 1998. London: Springer-Verlag.
20. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* Nature 521.7553, 2015: p. 436–444.
21. Nielsen, M., *NeuralNetworksAndDeepLearning.com*. 2017. 139, 143.
22. al., Y.L.e. *Gradient-Based Learning Applied to Document Recognition*. in *Proceedings of the IEEE*. 1998.
23. Kuo, C.-C.J. *Understanding Convolutional Neural Networks with A Mathematical Model*. 2016.
24. ujjwalkarn. *An Intuitive Explanation of Convolutional Neural Networks*. 2016; Available from: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.
25. Chen, M.S.a.A.H.a.M.Z.a.A.Z.a.L.-C., *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2018, arXiv.

26. BIANCO, S., et al., *Benchmark Analysis of Representative Deep Neural Network Architectures*. 2018, arXiv.

27. Koryakin, P. *Fingers*. 2019; Available from: https://www.kaggle.com/koryakinp/fingers.

28. O'Shea, R. *Finger Digits 0-5*. 2019; Available from: https://www.kaggle.com/roshea6/finger-digits-05.

29. BrandInside. *LINE* เผยพฤติกรรมผู้ใช้ชาวไทย ใช้บริการ *LINE Call 49* ล้านครั้งต่อวัน สูงที่สุดในโลก. 2020; Available from: https://brandinside.asia/line-thailand-2019-summary.

ประวัติผู้เขียน

ชื่อ-สกุล	ธนภัทร รัชธร
วัน เดือน ปี เกิด	30 พฤศจิกายน 2532
สถานที่เกิด	จังหวัดนราธิวาส
วุฒิการศึกษา	วิทยาศาสตรบัณฑิต (วท. บ.), จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	กรุงเทพมหานคร
ผลงานตีพิมพ์	T. Ratchatorn and S. Pumrin, “Patient aid message notification system based on hand movement tracking and haar-like features,” in ECTI-CON 2018 - 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2019, pp. 624–627.