



การจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์
Real-time Workplace Sign Language Recognition

พลกฤต ช่างสลัก รหัส 64172110222-9

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์และเทคโนโลยี
มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน
ปีการศึกษา 2566

การจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์
Real-time Workplace Sign Language Recognition

พลกฤต ช่างสลัก รหัส 64172110222-9

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรม สตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์และเทคโนโลยี
มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน
ปีการศึกษา 2566



ใบรับรองปริญญานิพนธ์

คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน

ปริญญา	วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)
ชื่อปริญญานิพนธ์	การจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์
นักศึกษา	นายพลกฤต ช่างสลัก รหัส 64172110222-9
ที่ปรึกษาปริญญานิพนธ์	อาจารย์กัระชาติ สุขสุทธิ
สาขา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2566

สาขาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

(อาจารย์วิทยา ศีลกุล)
หัวหน้าสาขา

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
(อาจารย์ดิพร จันทร์กลิ่น)

.....กรรมการ
(อาจารย์กัระชาติ สุขสุทธิ)

.....กรรมการ
(อาจารย์เกตุกาญจน์ โพธิจิตติกานต์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์พรภัสสร อ่อนเกิด)

ชื่อปริญญานิพนธ์	การจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์
นักศึกษา	นายพลกฤต ช่างสลัก รหัส 64172110222-9
ที่ปรึกษาปริญญานิพนธ์	อาจารย์กัระชาติ สุขสุทธิ
สาขา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2566

บทคัดย่อ

ปริญญานิพนธ์นี้เน้นการพัฒนาโปรแกรมการจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์ เนื่องจากผู้พิการทางการได้ยินมีความยากลำบากในการสื่อสารกับผู้อื่นที่ไม่เข้าใจภาษามือ โปรแกรมนี้มุ่งเน้นการใช้เทคโนโลยีการประมวลผลภาพและการรู้จำภาพจากวัตถุ เพื่อแปลความหมายของท่าทางภาษามือไทยในสถานการณ์เรียลไทม์ วัตถุประสงค์หลักคือการเพิ่มประสิทธิภาพในการสื่อสารของผู้พิการทางการได้ยินในสถานที่ทำงาน เป้าหมายคือการพัฒนาโปรแกรมที่แม่นยำและทันสมัยในการจำแนกภาษามือในสถานที่ทำงาน ปริญญานิพนธ์นี้มีลักษณะเป็นพื้นฐานและใช้กรอบทฤษฎีที่เกี่ยวข้องกับการประมวลผลภาพและการรู้จำภาพ ผลลัพธ์ที่คาดว่าจะได้รับคือการช่วยเพิ่มโอกาสในการมีส่วนร่วมในชีวิตประจำวันและการทำงานของผู้พิการทางการได้ยินในสังคมและสถานที่ทำงาน การพัฒนาโปรแกรมนี้นี้จะมีผลการใช้งานที่สามารถนำไปใช้ในการสื่อสารและการทำงานของผู้พิการทางการได้ยินในสถานที่ทำงานและชีวิตประจำวันอย่างมีประสิทธิภาพและสะดวกสบายมากขึ้น

คำสำคัญ: ช่วยเหลือผู้พิการทางการได้ยิน, แปลภาษามือไทย, การรู้จำภาพจากวัตถุ

Thesis Title Real-time Workplace Sign Language Recognition
Learner Ms.Ponkit Changsalak 64172110222-9
Adviser: Mr.Keerachart Suksut
Department: Computer Engineering
Academic Year: 2023

ABSTRACT

This thesis focuses on the development of real-time sign language recognition software for workplace environments. Due to the communication challenges faced by individuals with hearing impairments in interacting with those who do not understand sign language, this program emphasizes the utilization of image processing technology and object recognition to interpret Thai sign language gestures in real-time scenarios. The primary objective is to enhance the communication efficiency of individuals with hearing impairments in the workplace. The goal is to create a precise and contemporary program for sign language recognition in work settings. This thesis is foundational in nature and is grounded in theoretical frameworks related to image processing and object recognition. The anticipated outcomes include improving opportunities for participation in daily life and work for individuals with hearing impairments in both societal and workplace settings. The development of this software is expected to have practical applications, contributing to more effective and convenient communication and work interactions for individuals with hearing impairments in both professional and daily life contexts.

Keywords: Hearing Impairment Assistance, Thai Sign Language Interpretation, Object Recognition, Image Processing

กิตติกรรมประกาศ

รายงานวิจัยฉบับนี้สำเร็จลุล่วงด้วยดีเนื่องจากความกรุณาและความช่วยเหลืออย่างดียิ่งจาก อาจารย์กีระชาติ สุขสุทธิ ท่านได้เสียสละเวลาอันมีค่ายิ่ง ในการให้คำปรึกษาการดำเนินงานวิจัย ตลอดจนได้ตรวจสอบและแก้ไขข้อบกพร่องต่าง ๆ อันเป็นประโยชน์ในการจัดทำปริญญานิพนธ์ตั้งแต่เริ่มดำเนินการจนกระทั่งดำเนินการเสร็จสมบูรณ์

ขอขอบพระคุณ สมาคมคนหูหนวกแห่งประเทศไทย ที่ได้จัดทำระบบฐานข้อมูลภาษามือไทย ในรูปแบบดิจิทัลแพลตฟอร์ม มีส่วนช่วยในการรวบรวมชุดข้อมูลที่ถูกต้องและน่าเชื่อถือ จึงสามารถจัดทำโปรแกรมแปลภาษามือได้อย่างถูกต้อง

คุณค่าและประโยชน์ของการศึกษาค้นคว้านี้ ผู้จัดทำขอมอบเป็นเครื่องบูชาบุพการี บุรพจารย์ และผู้มีอุปการะคุณทุกท่านที่มีส่วนส่งเสริมให้ผู้ศึกษาประสบความสำเร็จในการดำเนินชีวิตและก้าวหน้าในหน้าที่การงาน ผู้ศึกษาขอกราบขอบพระคุณเป็น อย่างสูงไว้ ณ โอกาสนี้

พลกฤต ช่างสลัก

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
ABSTRACT	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญ (ต่อ)	จ
สารบัญตาราง	ฉ
สารบัญรูป	ช
สารบัญรูป (ต่อ)	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 ประโยชน์ที่ได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	3
1.6 แผนการดำเนินงานของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ความรู้ทั่วไปเกี่ยวกับภาษามือ	
2.2 การเรียนรู้ของเครื่อง	
2.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks : CNN)	
2.4 การตรวจจับวัตถุ (Object Detection)	
2.5 ภาษาไพธอน (Python Programming Language)	
2.6 โยโล (You Only Look Once : Yolo)	
2.7 โรโบฟลว์ (Roboflow)	
2.8 การวัดประสิทธิภาพโมเดล	
บทที่ 3 ขั้นตอนการดำเนินงาน	22
3.1 หลักการทำงานของระบบการจำแนกภาษามือและการแปลความหมาย	22
3.2 รวบรวมชุดข้อมูลคำศัพท์และประโยคภาษามือที่ใช้ในชีวิตประจำวัน	23
3.3 การเตรียมชุดข้อมูลก่อนนำเข้าโมเดล YOLOv8	25
3.3.1 จัดทำรูปภาพของท่าทางตามคำศัพท์	25

สารบัญ (ต่อ)

	หน้า
3.3.2 นำเข้าเฟรมเวิร์คโรโบโพล์เพื่อกำหนดวัตถุให้กับรูปภาพ	28
3.3.3 ออกแบบชุดข้อมูลการเทรน	29
บทที่ 4 การทดลองและผลดำเนินงาน	-
บทที่ 5 บทสรุปการดำเนินงานและข้อเสนอแนะ	-
เอกสารอ้างอิง	35
เอกสารอ้างอิง (ต่อ)	36
ภาคผนวก	-
ภาคผนวก ก	-

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงานโครงการวิศวกรรมคอมพิวเตอร์ 1	3
1.2 แผนการดำเนินงานโครงการวิศวกรรมคอมพิวเตอร์ 1 (ต่อ)	4
3.1 บทการสนทนาทั่วไป	23
3.2 การเสนอความช่วยเหลือ	23
3.3 การเสนอความคิดเห็น	24
3.4 การตอบสนองสื่อสาร	24

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 การเปรียบเทียบคำ “พ่อ” ระหว่างภาษามือไทยและภาษามือออสเตรเลีย	7
รูปที่ 2.2 ภาษามือไวยากรณ์ภาษาไทย ฉัน ไป โรงเรียน “ฉันไปโรงเรียน”	7
รูปที่ 2.3 ภาษามือไวยากรณ์ภาษาไทย คุณ กิน ข้าว แล้ว หรือ ยัง “คุณกินข้าวแล้วหรือยัง”	8
รูปที่ 2.4 ระบบโครงข่ายประสาทของมนุษย์และโครงข่ายประสาทเทียม	9
รูปที่ 2.5 อัลกอริทึม Deep Neural Network สำหรับการเรียนรู้จดจำใบหน้า	10
รูปที่ 2.6 ตัวอย่างการจำแนกลักษณะภาพด้วยโครงข่ายประสาทแบบคอนโวลูชัน	10
รูปที่ 2.7 การกำหนดค่าใน ตัวกรอง หรือ เคอร์เนล	11
รูปที่ 2.8 การกำหนดค่าใน ตัวกรอง หรือ เคอร์เนล	11
รูปที่ 2.9 รวบรวมผลลัพธ์ของการทำคอนโวลูชันทั้งหมดไว้ใน ผังคุณลักษณะ	12
รูปที่ 2.10 การทำ Padding ให้กับ Input Data	12
รูปที่ 2.11 การกำหนดค่า Stride มากขึ้นจะทำให้เราได้ผังคุณลักษณะที่มีขนาดเล็กลง	13
รูปที่ 2.12 การลดขนาดของภาพด้วยการ Pooling	13
รูปที่ 2.13 ตัวอย่างทำงานของการตรวจจับวัตถุต่าง ๆ	14
รูปที่ 2.14 รูปแบบโครงสร้างของภาษาไพทอนในการแสดง Hello, World!	16
รูปที่ 2.15 YOLOv8 Architecture, Visualisation Made	19
รูปที่ 2.16 ความสามารถของ YOLOv8 ที่ถูกพัฒนาขึ้นอย่างต่อเนื่อง	19
รูปที่ 2.17 Roboflow มีส่วนช่วยในการทำ Dataset	20
รูปที่ 2.18 ตาราง Confusion Matrix	21
รูปที่ 3.1 ผังงานโดยรวมของระบบจำแนกภาษามือและแปลความหมาย	22
รูปที่ 3.2 ผังงานโดยรวมการทำงานของระบบจัดทำรูปภาพ	25
รูปที่ 3.3 จากโปรแกรมจะได้ภาพที่มีชื่อ และถูกเก็บลงในโฟลเดอร์ที่กำหนด	27
รูปที่ 3.4 การอัปโหลดรูปภาพไปยัง Roboflow	28
รูปที่ 3.5 ขั้นตอนการกำหนด Bounding Box ให้กับภาพ	28
รูปที่ 3.6 การ Augmentation ที่มีใน Roboflow	29
รูปที่ 3.7 การแบ่งอัตราส่วนชุดข้อมูลก่อนการเทรน	30
รูปที่ 3.8 เมื่อดาวโหลดสำเร็จจะได้ไฟล์โมเดลสำหรับการฝึกสอนและชุดข้อมูล	31
รูปที่ 3.9 ตัววัดที่ใช้เพื่อประเมินคุณภาพและประสิทธิภาพของโมเดล	32

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3.10 กราฟแสดงค่าการสูญเสียและประสิทธิภาพของโมเดลในแต่ละรอบการฝึก	33
รูปที่ 3.11 ตาราง Confusion Matrix แสดงผลลัพธ์การทำนายของคลาสต่าง ๆ	33
รูปที่ 3.12 ภาพตัวอย่างการทำนายคลาสของโมเดลที่ผ่านการเทรน	34

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปริญญานิพนธ์

การสื่อสารเป็นสิ่งสำคัญในมนุษย์ ภาษาช่วยให้เราสามารถแบ่งปันประสบการณ์และความรู้ทางวัฒนธรรม และสร้างความเข้าใจร่วมกันในกลุ่มสังคม โดยทั่วไปแล้วภาษามีส่วนสำคัญในการแสดงความคิดเห็น การสื่อสารข้อมูล และการสร้างความเข้าใจร่วมกันในสังคมของมนุษย์ สามารถที่จะถ่ายทอดความรู้ ประสบการณ์ และความเข้าใจทางวัฒนธรรมอื่น ๆ ในรูปแบบของคำพูด การเขียน และการอ่าน จากการสำรวจขององค์การอนามัยโลก (WHO) พบว่า ในปี 2564 มีผู้พิการทางการได้ยินทั่วโลกประมาณ 466 ล้านคน หรือคิดเป็นร้อยละ 5.3 ของประชากรทั้งหมด โดยส่วนใหญ่อาศัยอยู่ในประเทศกำลังพัฒนา ในประเทศไทย มีผู้พิการทางการได้ยินประมาณ 1.2 ล้านคน หรือคิดเป็นร้อยละ 1.5 ของประชากรทั้งหมด ปัญหาการสื่อสารระหว่างผู้พิการทางหูและการใช้ภาษามือกับบุคคลทั่วไปนั้น เกิดจากความแตกต่างระหว่างภาษามือและภาษาพูด โดยภาษามือเป็นภาษาที่ประกอบด้วยท่าทางมือ ท่าทางใบหน้า และการแสดงออกทางสีหน้า ซึ่งมีลักษณะเฉพาะตัวและแตกต่างจากภาษาพูดอย่างสิ้นเชิง ทำให้บุคคลทั่วไปไม่สามารถเข้าใจภาษามือ นอกจากนี้ ท่าทางภาษามือยังสามารถใช้แทนคำศัพท์ได้หลายคำ ขึ้นอยู่กับบริบทของการใช้งาน ทำให้ระบบการแปลภาษามือในปัจจุบัน ไม่สามารถแปลความหมายของท่าทางภาษามือได้อย่างถูกต้อง หากท่าทางภาษามือนั้นคลุมเครือหรือสามารถแทนคำศัพท์ได้หลายคำ ปัญหานี้ทำให้ผู้พิการทางการได้ยินไม่สามารถมีส่วนร่วมในสังคมได้อย่างเท่าเทียมกับผู้อื่น โดยภาษามือเป็นรูปแบบการสื่อสารที่มีลักษณะเฉพาะตัว แตกต่างจากภาษาพูดที่ใช้เสียงเป็นหลัก ภาษามือมีโครงสร้างและไวยากรณ์ที่แตกต่างกันไปในแต่ละประเทศและภูมิภาค การสื่อสารด้วยภาษามือจึงมักจำกัดอยู่เฉพาะในกลุ่มคนหูหนวกหรือผู้พิการทางการได้ยิน การได้ยินมักประสบปัญหาในการสื่อสารกับผู้อื่น เนื่องจากคนส่วนใหญ่ไม่สามารถเข้าใจภาษามือได้ โปรแกรมการจำแนกภาษามือจึงเป็นเทคโนโลยีที่มีศักยภาพในการแก้ไขปัญหาการสื่อสารของคนหูหนวกหรือผู้พิการทางการได้ยิน โดยช่วยในเรื่องของการสื่อสารกับกลุ่มผู้พิการการได้ยิน ทำให้คนปกติสามารถที่จะสื่อสารกับผู้พิการทางการได้ยินได้อย่างมีประสิทธิภาพมากขึ้น การจำแนกภาษามือแบบเรียลไทม์ จึงมีความสำคัญต่อการพัฒนาเทคโนโลยีสำหรับผู้พิการทางการได้ยิน

ดังนั้น ปริญญานิพนธ์เรื่อง "การจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์" จึงเป็นที่มาของการพัฒนาโปรแกรมที่ใช้ในการวิเคราะห์แยกแยะภาษามือให้ออกมาเป็นคำศัพท์ต่าง ๆ ในรูปของประโยคแบบเรียลไทม์ โดยอาศัยเทคโนโลยีการประมวลผลภาพและการรู้จำภาพจากวัตถุ ซึ่งจะ

ช่วยสามารถแปลความหมายของท่าทางภาษามือของภาษาไทยที่ใช้ในสถานที่ทำงานได้ การพัฒนาเทคโนโลยีนี้จะเป็นประโยชน์อย่างยิ่งในการลดช่องว่างทางการสื่อสารระหว่างผู้พิการทางการได้ยินและบุคคลทั่วไปในบริบทของการทำงาน อันจะนำไปสู่การสร้างโอกาสและความเท่าเทียมในการประกอบอาชีพสำหรับผู้พิการทางการได้ยินต่อไป

1.2 วัตถุประสงค์ของปฏิญานิพนธ์

1.2.1 เพื่อศึกษาการจำแนกภาษามือแบบเรียลไทม์ ที่ช่วยให้บุคคลทั่วไปสามารถเข้าใจภาษามือที่ใช้ในสถานที่ทำงานของผู้พิการทางการได้ยินได้อย่างถูกต้อง

1.2.2 เพื่อพัฒนาโปรแกรมการจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์ โดยสามารถแปลความหมายของท่าทางภาษามือ และรูปประโยคของภาษาไทยให้เข้าใจง่าย

1.2.3 เพื่อสร้างโปรแกรมการจำแนกภาษามือที่ใช้ในสถานที่ทำงานแบบเรียลไทม์ ที่สามารถวิเคราะห์แยกแยะภาษามือให้ออกมาเป็นคำต่าง ๆ และในรูปประโยคได้อย่างเข้าใจง่าย

1.3 ขอบเขตของปฏิญานิพนธ์

1.3.1 พัฒนาโปรแกรมที่ใช้งานได้ในระบบปฏิบัติการ Windows ซึ่งใช้ภาษาไพธอน ในการพัฒนาโปรแกรม

1.3.2 แปลความหมายท่าทางของภาษามือที่ใช้สื่อสารในสถานที่ทำงานของภาษาไทย ออกมาในรูปแบบของข้อความ

1.3.3 สามารถนำความหมายของท่าทางที่ถูกแปลคำศัพท์เรียงต่อเป็นประโยคได้

1.3.4 การแปลความหมายของท่าทางในภาษามือแบบมือเปล่า ซึ่งเป็นภาษามือที่ใช้เพียงมือ ในการสื่อความหมายเท่านั้น

1.3.5 ชุดข้อมูลที่ถูกนำมาแปลความหมายนั้น ได้อ้างอิงมาจากแหล่งข้อมูลต่าง ๆ ที่น่าเชื่อถือ

1.3.6 พัฒนาโปรแกรมที่ช่วยให้บุคคลทั่วไปสามารถเข้าใจการสื่อสารภาษามือ ของคนหูหนวกได้อย่างเข้าใจมากยิ่งขึ้น

1.3.7 ความสามารถในการแปลความหมายท่าทางของภาษามือแบบมือเปล่าที่ใช้ในการสื่อสารในสถานที่ทำงาน โดยแบ่งออกเป็นจำนวนประโยค 33 ประโยค และคำศัพท์ทั้งหมด 50 คำ

1. บทการสนทนาทั่วไป จำนวน 12 ประโยค
2. การเสนอความช่วยเหลือ จำนวน 5 ประโยค
3. การเสนอความคิดเห็น จำนวน 7 ประโยค
4. การตอบสนองสื่อสาร จำนวน 9 ประโยค

บทที่ 2

ทฤษฎีและงานที่วิจัยเกี่ยวข้อง

ในบทนี้กล่าวถึงทฤษฎีที่เกี่ยวกับภาษาศาสตร์ของภาษามือและ ทฤษฎีการเรียนรู้ของเครื่องคอมพิวเตอร์ในรูปแบบการเรียนรู้แบบเชิงลึก เน้นที่การใช้เครือข่ายประสาทเทียมเพื่อทำให้เครื่องคอมพิวเตอร์สามารถเรียนรู้และทำนายข้อมูลได้อย่างมีประสิทธิภาพ ซึ่งเป็นกระบวนการที่เครื่องคอมพิวเตอร์สามารถเรียนรู้จากข้อมูลที่ถูกนำเข้ามาในรูปแบบของภาพหรือสัญลักษณ์ภาษามือ ทำให้เครื่องคอมพิวเตอร์สามารถทำนายและเข้าใจความหมายของภาพต่าง ๆ ได้ เครือข่ายประสาทเทียมประกอบด้วยชั้นของเซลล์ประสาทซึ่งทำหน้าที่เรียนรู้ความสัมพันธ์ระหว่างข้อมูลที่นำเข้าและข้อมูลที่เอาต์พุต โดยแต่ละเซลล์ประสาทรุ่นมีความสามารถในการปรับแต่งน้ำหนักของการเชื่อมต่อ เครือข่ายประสาทเทียมสามารถเรียนรู้ลักษณะและลำดับของข้อมูลที่ซับซ้อนได้ ดังรายละเอียดต่อไปนี้

- 2.1 ความรู้ทั่วไปเกี่ยวกับภาษามือ
- 2.2 การเรียนรู้ของเครื่อง
- 2.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks : CNN)
- 2.4 การตรวจจับวัตถุ (Object Detection)
- 2.5 ภาษาไพทอน (Python Programming Language)
- 2.6 โยโล (You Only Look Once : Yolo)
- 2.7 โรโบฟโฟลว์ (Roboflow)
- 2.8 การวัดประสิทธิภาพโมเดล

2.1 ความรู้ทั่วไปเกี่ยวกับภาษามือ

ในประเทศไทย [1] นอกจากมีภาษาไทยมาตรฐานอันเป็นภาษาราชการและภาษาประจำชาติแล้ว ยังมีภาษาไทยอีกลักษณะหนึ่งรวมอยู่ด้วยคือภาษามือไทย ซึ่งเป็นภาษาที่ใช้กันในกลุ่มคนหูหนวกชาวไทยจำนวนนับแสนคนและใช้กันอยู่ในเฉพาะประเทศไทยเท่านั้น ภาษามือ คือภาษาแรกของคนหูหนวกที่ใช้ในการติดต่อสื่อสาร คนทั่วไปมักเข้าใจผิดว่าภาษามือเป็น "ภาษาสากล" และภาษามือคือการทำท่ามือที่เปลี่ยนท่ามือไปตามภาษาพูดหรือภาษาเขียน การทำท่ามือออกมาให้ได้ถูกต้องก็เหมือนกับการออกเสียงให้ถูกต้องในภาษาพูด มีความแตกต่างเพียงเล็กน้อยในรูปแบบท่ามือ การหันมือหรือพลิกฝ่ามือ, ตำแหน่งของมือ, การเคลื่อนไหวของลำตัว จังหวะการทำท่า มือ และการแสดงออกทางสีหน้า ซึ่งสิ่งเหล่านี้เป็นองค์ประกอบของการใช้ภาษามือ การใช้ภาษามือที่จะก่อให้เกิดประสิทธิภาพสูงที่สุดนั้น ควรต้องใช้องค์ประกอบต่าง ๆ เหล่านี้ทั้งหมดมาผสมผสานกัน และท่ามือที่เกิดจากองค์ประกอบเหล่านี้ สามารถเปลี่ยนความหมายของคำได้อย่างสิ้นเชิง

ภาษามือเป็นระบบการสื่อสารด้วยมือ ร่างกาย และใบหน้า แทนการใช้เสียงพูด ภาษามือเป็นภาษาที่สมบูรณ์แบบ เช่นเดียวกับภาษาพูด สามารถใช้สื่อความหมายได้หลากหลาย เช่น นามธรรม จินตนาการ อารมณ์ และความรู้สึก ภาษามือมีต้นกำเนิดมาอย่างยาวนาน ต่อมาภาษามือได้พัฒนาไปตามกาลเวลา ในแต่ละภูมิภาคของโลกมีภาษามือที่แตกต่างกันออกไป

2.1.1 ลักษณะของภาษามือ

ภาษามือในแต่ละประเทศไม่เหมือนกัน [2] ภาษามือเป็นภาษาที่มีวัฒนธรรมเหมือนกับภาษาพูด ภาษามือไทยเป็นวัฒนธรรมของคนหูหนวกไทย ซึ่งต่างจากภาษาพูดไทยที่เป็นวัฒนธรรมของคนที่มีการได้ยิน คนหูหนวกใช้สายตาในการมองภาพต่าง ๆ มากกว่าคนที่มีการได้ยิน ส่วนใหญ่คนทั่วไปคิดว่า ภาษามือของคนหูหนวกในแต่ละประเทศเหมือนกัน เพราะการใช้ภาษามือดูแล้วเหมือน ๆ กับภาษาใบ้ แต่ที่จริงแล้ว ภาษามือในแต่ละประเทศมีความแตกต่างกันมาก เช่น คำศัพท์ในภาษามือไทย “พ่อ” ไม่เหมือนกับคำศัพท์ในภาษามือออสเตรเลียน “พ่อ” (ดังรูปที่ 2.1) เช่นเดียวกับ ภาษาไทย ภาษาจีนและภาษาพูดอิสราเอลที่ใช้คำศัพท์ต่างกันในคำที่มีความหมายเหมือนกัน



รูปที่ 2.1 การเปรียบเทียบคำ “พ่อ” ระหว่างภาษามือไทยและภาษามือออสเตรเลียน [2]

2.1.2 ประเภทของภาษามือ

ภาษามือสามารถแบ่งออกเป็นประเภทใหญ่ ๆ ได้ 2 ประเภท คือ

- ภาษามือสากลเป็นภาษามือที่ใช้กันทั่วโลก ใช้ในการสื่อสารระหว่างผู้พิการทางการได้ยินจากประเทศต่าง ๆ
- ภาษามือเฉพาะประเทศ เป็นภาษามือที่ใช้กันเฉพาะในประเทศหรือภูมิภาคหนึ่ง ๆ เช่น ภาษามือไทย ภาษามืออังกฤษ ภาษามืออเมริกัน

2.1.3 ภาษามือตามไวยากรณ์ภาษาไทย

ภาษามือตามไวยากรณ์ภาษาไทย เป็นการใชภาษามือที่คัดลอกคำต่อคำจากประโยคภาษาไทยสำหรับการสื่อสารระหว่างครูที่มีการได้ยินและนักเรียนหูหนวก (Branson, Miller & Sri-on, 2005) ครูที่มีการได้ยินเป็นผู้คิดทำภาษามือตามประโยคภาษาไทยเพื่อสอนให้เด็กหูหนวกฝึก

เขียนภาษาไทยได้ถูกต้องตามหลักไวยากรณ์ เช่น ฉัน ไป โรงเรียน “ฉันไปโรงเรียน” และ คุณ กิน ข้าว แล้ว หรือ ยัง “คุณกินข้าวแล้วหรือยัง” ตัวอย่างดังรูปที่ 2.2, 2.3



รูปที่ 2.2 ภาษามือไวยากรณ์ภาษาไทย ฉัน ไป โรงเรียน “ฉันไปโรงเรียน” [2]



รูปที่ 2.3 ภาษามือไวยากรณ์ภาษาไทย คุณ กิน ข้าว แล้ว หรือ ยัง “คุณกินข้าวแล้วหรือยัง” [2]

2.1.4 องค์ประกอบเชิงโครงสร้างของภาษามือ

องค์ประกอบเชิงโครงสร้างของภาษามือเป็นองค์ประกอบที่ใช้ในการประกอบท่าทางการแสดงสีหน้า และการเคลื่อนไหวของร่างกาย เพื่อสื่อความหมาย องค์ประกอบเชิงโครงสร้างของภาษามือสามารถแบ่งออกได้เป็นประเภทต่าง ๆ ดังนี้

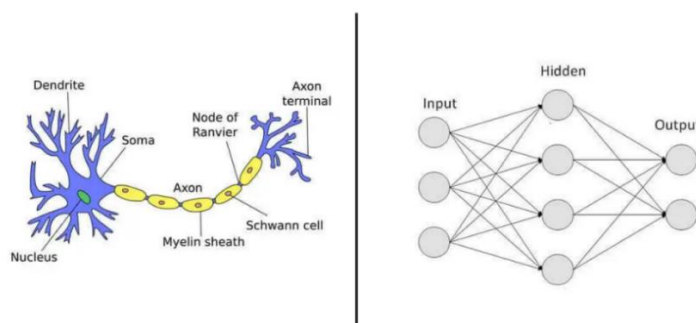
- รูปร่างของมือ เป็นองค์ประกอบที่ใช้กำหนดรูปร่างมือ เช่น มือแบน มืองอ มือกำ
- ตำแหน่งของมือ เป็นองค์ประกอบที่ใช้กำหนดตำแหน่งของมือ เช่น มืออยู่ด้านหน้า มืออยู่ด้านหลัง มืออยู่ด้านข้าง
- ทิศทางของมือ เป็นองค์ประกอบที่ใช้กำหนดทิศทางของมือ เช่น มือชี้ไปข้างหน้า มือชี้ไปข้างหลัง มือชี้ไปด้านบน
- การเคลื่อนไหวของมือ เป็นองค์ประกอบที่ใช้กำหนดการเคลื่อนไหวของมือ เช่น มือเคลื่อนไหวไปข้างหน้า มือเคลื่อนไหวไปข้างหลัง มือเคลื่อนไหวไปด้านบน
- การแสดงสีหน้า เป็นองค์ประกอบที่ใช้แสดงอารมณ์ เช่น ยิ้ม หัวเราะ เศร้า
- การเคลื่อนไหวของร่างกาย เป็นองค์ประกอบที่ใช้แสดงท่าทางและอารมณ์ เช่น พยักหน้า กะพริบตา

2.2 การเรียนรู้ของเครื่อง (Machine Learning)

การเรียนรู้ของเครื่อง [3] เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่ช่วยให้คอมพิวเตอร์สามารถเรียนรู้จากข้อมูลทำให้คอมพิวเตอร์สามารถทำงานบางอย่างได้โดยไม่ต้องเข้ารหัสทุกบรรทัดของอัลกอริทึมที่ใช้ทำงานนั้น ๆ เป็นเรื่องที่ทำนายและน่าทึ่งในโลกที่เรามีวันนี้ หนึ่งในวิธีที่ถูกนำเสนออย่างกว้างขวางคือการใช้เทคนิคที่เรียกว่า โครงข่ายประสาทเทียม

2.2.1 โครงข่ายประสาทแบบตื้น (Shallow Neural Networks)

โครงข่ายประสาทเทียม [3] ทำให้คอมพิวเตอร์สามารถเรียนรู้และทำนายผลลัพธ์ของข้อมูลโดยไม่ต้องเข้ารหัสทุกบรรทัดของอัลกอริทึม (Algorithm) โดยนำเอาหลักการทำงานของสมองมนุษย์มาใช้เป็นแนวทางในการพัฒนาโมเดล (Model) การเรียนรู้ของเครื่องมีหลากหลายประเภท การจำลองการทำงานของสมองมนุษย์ โครงข่ายประสาทเทียมเป็นโมเดลทางคณิตศาสตร์ที่สร้างขึ้นเพื่อจำลองการทำงานของสมองมนุษย์ ประกอบด้วยชั้นหลาย ๆ ชั้นที่มีหน้าที่แตกต่างกัน



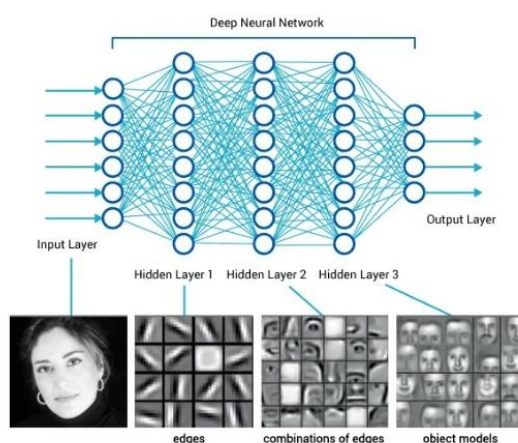
รูปที่ 2.4 ระบบโครงข่ายประสาทของมนุษย์และโครงข่ายประสาทเทียม [4]

- ชั้นอินพุต (Input Layer) ชั้นนี้รับข้อมูลเข้าสู่เครือข่าย เปรียบเหมือน "ตา" ของระบบที่มองเห็นข้อมูลเข้ามา
- ชั้นซ่อน (Hidden Layer) ชั้นนี้เป็นที่ทำหน้าที่ประมวลผลข้อมูลจากชั้นอินพุตไหนในชั้นซ่อนใช้ฟังก์ชันทางคณิตศาสตร์เพื่อจัดการกับข้อมูลที่มาจากชั้นอินพุต
- ชั้นเอาต์พุต (Output Layer) ชั้นนี้ส่งผลลัพธ์ออกมา มักจะเป็น "ตา" ที่มองเห็นผลลัพธ์หลังจากการประมวลผลผลลัพธ์

2.2.2 โครงข่ายประสาทแบบลึก (Deep Neural Network : DNN)

เครือข่ายประสาทเทียมเชิงลึก เป็นประเภทหนึ่งของเครือข่ายประสาทเทียม (Neural Network) ที่มีเลเยอร์ (Layer) มากกว่า 1 เลเยอร์ โดยเลเยอร์เหล่านี้จะเชื่อมต่อกันด้วยการเชื่อมต่อแต่ละการเชื่อมต่อจะมีค่าน้ำหนัก (Weight) ที่กำหนดความสำคัญของข้อมูลที่ได้รับมา DNN ได้รับความพัฒนามาจากเครือข่ายประสาทเทียมแบบดั้งเดิม ซึ่งมักมีเพียง 1-2 เลเยอร์เท่านั้น เครือข่าย

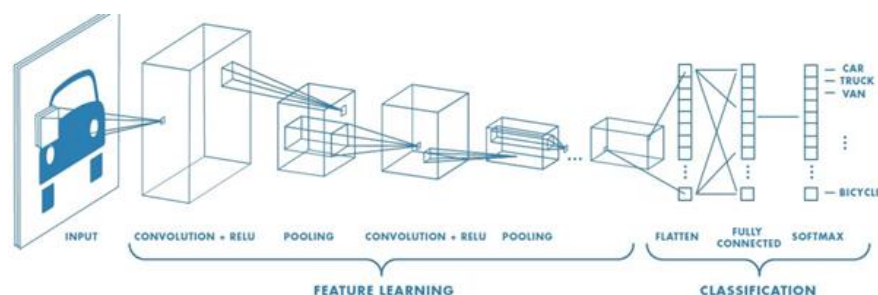
ประสาทเทียมแบบดั้งเดิมสามารถแก้ปัญหาทางคณิตศาสตร์ที่ง่าย และปัญหาคอมพิวเตอร์ รวมถึงโครงสร้างพื้นฐานพร้อมตารางความจริง แต่เป็นเรื่องยากสำหรับเครือข่ายเหล่านี้ในการแก้ปัญหาการประมวลผลภาพ การมองเห็นด้วยคอมพิวเตอร์ และการประมวลผลภาษาธรรมชาติที่ซับซ้อน สำหรับปัญหาเหล่านี้ เราใช้เครือข่ายประสาทเทียมเชิงลึก ซึ่งมักมีโครงสร้างเลเยอร์ซ่อนที่ซับซ้อนพร้อมกับเลเยอร์ที่หลากหลายมากมาย ช่วยให้โมเดลเข้าใจปัญหาได้ดีขึ้นและให้โซลูชันที่เหมาะสมที่สุดสำหรับโครงการที่ซับซ้อน



รูปที่ 2.5 อัลกอริทึมโครงข่ายประสาทแบบลึกสำหรับการเรียนรู้จัดจำใบหน้า [6]

2.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks : CNN)

โครงข่ายประสาทแบบคอนโวลูชัน เป็น [7] โครงข่ายประสาทเทียมหนึ่งในกลุ่ม Bio-Inspired โดยที่ CNN จะจำลองการมองเห็นของมนุษย์ที่มองพื้นที่เป็นที่ย่อย ๆ และนำกลุ่มของพื้นที่ที่ย่อย ๆ มาผสมกัน เพื่อดูว่าสิ่งที่เห็นอยู่เป็นอะไรกันแน่ การมองพื้นที่ย่อยของมนุษย์จะมีการแยกคุณลักษณะ (Feature) ของพื้นที่ย่อยนั้น เช่น ลายเส้น และการตัดกันของสี ซึ่งการที่มนุษย์รู้ว่าพื้นที่ตรงนี้เป็นเส้นตรงหรือสีตัดกัน เพราะมนุษย์ดูทั้งจุดที่สนใจและบริเวณรอบ ๆ ประกอบกัน



รูปที่ 2.6 ตัวอย่างการจำแนกลักษณะภาพด้วยโครงข่ายประสาทแบบคอนโวลูชัน [7]

2.3.1 ลักษณะของฟิลเตอร์หรือเคอร์เนล (Filter or Kernel)

ในการทำงานด้านการประมวลผลภาพ [7] การคำนวณนี้จะเริ่มจากการกำหนดค่าในฟิลเตอร์ หรือเคอร์เนลที่ช่วยดึงคุณลักษณะที่ใช้ในการรู้จำวัตถุออก โดยปกติตัวกรอง หรือเคอร์เนลอันหนึ่งจะดึงคุณลักษณะที่สนใจออกมาได้หนึ่งอย่าง เราจึงจำเป็นต้องตัวกรองหลายตัวกรองด้วย เพื่อหาคุณลักษณะทางพื้นที่หลายอย่างประกอบกัน

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

รูปที่ 2.7 การกำหนดค่าในตัวกรอง หรือเคอร์เนล [7]

กระบวนการทำคอนโวลูชัน [8] จะทำการเลื่อนตัวกรองนี้ไปที่ข้อมูลนำเข้า ณ ทุก ๆ ตำแหน่ง และทำการคูณค่าของแต่ละพิกเซล (Pixel) แล้วรวมผลลัพธ์ทั้งหมด เพื่อให้ได้ค่าผลรวมนั้นไปไว้ในผังคุณลักษณะ (Feature Map) ที่เกิดขึ้น พื้นที่สี่เหลี่ยมที่คอนโวลูชันถูกทำนี้เรียกว่า พื้นที่ที่รับสัญญาณ (Receptive Field) โดยเนื่องจากขนาดของตัวกรองทำให้พื้นที่รับรู้ (Receptive Field) นั้นเป็นขนาด 3x3

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

Input x Filter

4		

Feature Map

รูปที่ 2.8 การกำหนดค่าในตัวกรอง หรือเคอร์เนล [8]

ที่นี้ตัวกรองอยู่ที่ด้านบนซ้าย, ผลลัพธ์จากการทำคอนโวลูชันคือ "4" และถูกแสดงใน ผังคุณลักษณะ (Feature Map) ที่เกิดขึ้น จากนั้นเราจะเลื่อนตัวกรองไปทางขวาและทำการดำเนินการเดียวกัน โดยการบวกผลลัพธ์เข้าไปในผังคุณลักษณะด้วย

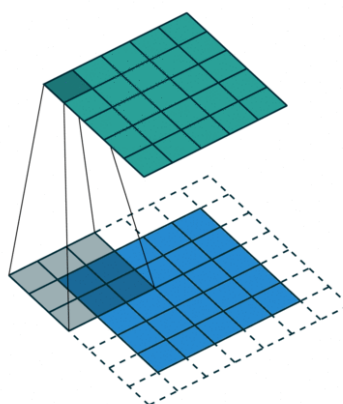
1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

รูปที่ 2.9 รวบรวมผลลัพธ์ของการทำคอนโวลูชันทั้งหมดไว้ใน ผังคุณลักษณะ [8]

2.3.2 การเสริมขอบ (Padding)

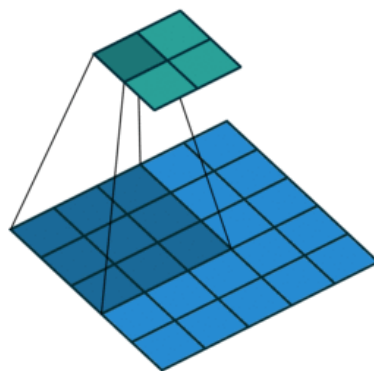
ในกระบวนการคอนโวลูชัน (convolution) เคอร์เนลจะเลื่อนไปที่ละช่องบนข้อมูลอินพุต (input data) เพื่อหาบัพที่ละส่วน โดยจะเคลื่อนที่ไปจนครอบคลุมทั้งหมดหากไม่ทำการเสริมขอบให้กับข้อมูลอินพุต พิกเซลที่อยู่ตามขอบจะไม่ถูกหาบัพโดยเคอร์เนล เนื่องจากเคอร์เนลไม่สามารถเคลื่อนที่ออกนอกขอบได้ การเสริมขอบจึงเป็นวิธีที่ช่วยให้พิกเซลตามขอบสามารถถูกหาบัพได้ นอกจากนี้ ยังช่วยปรับปรุงคุณสมบัติของผังคุณลักษณะ เพราะจะมีข้อมูลจากบริเวณขอบรวมอยู่ด้วย ซึ่งอาจมีข้อมูลสำคัญที่ช่วยในการระบุคุณสมบัติของวัตถุในภาพได้ โดยทั่วไป วิธีเสริมขอบที่นิยมใช้คือการเสริมขอบด้วยศูนย์ (Zero Padding) ซึ่งเป็นการเติมค่า 0 ให้กับขอบของข้อมูลอินพุต



รูปที่ 2.10 การทำการเสริมขอบให้กับพิกเซลรูปภาพ [9]

2.3.3 สไลด์ริง (Striding)

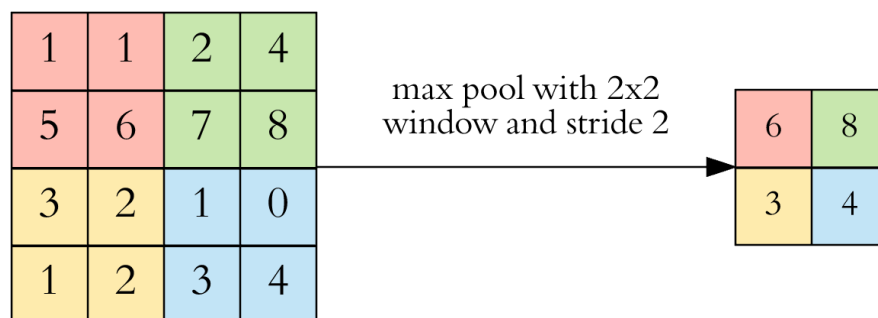
เป็นกระบวนการ [9] ในการทำคอนโวลูชันโดยการเลื่อนแผ่นเคอร์เนลไปบนข้อมูลภาพ (Image Data) ซึ่งโดยพื้นฐานของคอนโวลูชันแล้วค่าสไลด์ริง (Stride) จะมีค่าเท่ากับ 1 คือจะมีการเลื่อน เคอร์เนลไปบนข้อมูลภาพ ครึ่งละ 1 พิกเซล สามารถลดขนาดของภาพที่แต่ละพิกเซลมีความสัมพันธ์กันในเชิงพื้นที่ได้โดยการเพิ่มค่าสไลด์ริง ซึ่งเมื่อมีการกำหนดค่าสไลด์ริงมากขึ้น จะทำให้การเลื่อนกันของเคอร์เนลตอนที่มีการทับกับข้อมูลภาพ และขนาดของฟังก์ชันลักษณะลดลงเมื่อกำหนดค่าสไลด์ริงเท่ากับ 2 แล้วเคอร์เนลขนาด 3×3 จะถูกค่าสไลด์ริงข้ามพิกเซลของข้อมูลภาพ ขนาด 5×5 ที่ละ 2 พิกเซล ทำให้ได้ฟังก์ชันลักษณะขนาด 2×2 (ดังรูปที่ 2.12)



รูปที่ 2.11 การกำหนดค่าสไลด์ริงมากขึ้นจะทำให้เราได้ฟังก์ชันลักษณะที่มีขนาดเล็กลง [9]

2.3.4 การรวมข้อมูล (Pooling)

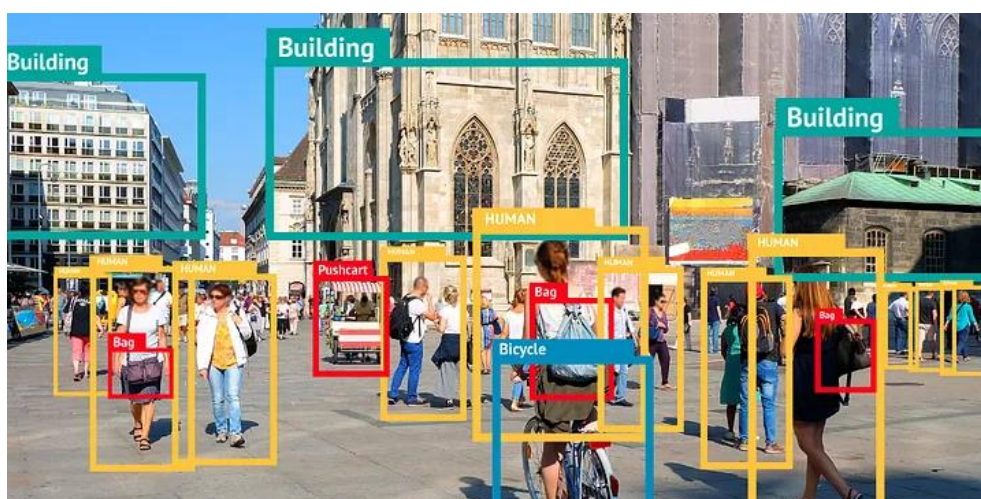
นอกจากการลดขนาดของภาพด้วยการ [9] เพิ่มค่าสไลด์ริงโดยการเลื่อนเคอร์เนลข้ามพิกเซลของข้อมูลภาพตามระยะทางที่กำหนดแล้ว ยังมีอีกวิธีหนึ่งในการลดขนาดของภาพ คือการทำลดค่าสูงสุดข้อมูล (Max Pooling) หรือการลดค่าเฉลี่ยข้อมูล (Average Pooling) โดยการรวมข้อมูลจะเป็นกระบวนการทำงานภายนอก CNN Layer



รูปที่ 2.12 การลดขนาดของภาพด้วยการรวมข้อมูล [9]

2.4 การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุ [10] คือ การสอนให้คอมพิวเตอร์ทำหน้าที่เสมือนดวงตา ที่สามารถรับรู้ได้ว่าในรูปภาพที่แสดงอยู่ มีวัตถุอะไรบ้างที่เราสนใจ และบอกตำแหน่งของวัตถุต่าง ๆ เหล่านั้น ว่าตั้งอยู่ตรงไหนของภาพ (จากรูปที่ 2.14) จะเห็นถึงวัตถุ และตำแหน่งต่าง ๆ ที่คอมพิวเตอร์ตอบกลับมา ว่าเห็นอะไรบ้างในภาพ โดยตีกรอบล้อมรอบเพื่อระบุตำแหน่งของวัตถุ โดยก่อนที่จะมาเป็นโมเดลตรวจจับวัตถุ ที่แสดงผลดังภาพข้างบนได้ เราต้องให้คอมพิวเตอร์เรียนรู้ภาพจำนวนหนึ่ง (Dataset) ของวัตถุที่เราต้องการจะตรวจจับ พร้อมทั้งบอกให้คอมพิวเตอร์รู้ว่าวัตถุที่เราให้มันเรียนรู้ยู่ตอนนี้คือวัตถุอะไร (Class) และ ตำแหน่งอยู่ตรงไหน ซึ่งจะต่างจากการทำการจำแนกภาพ (Image Classification) ทั่ว ๆ ไปที่ต้องการรู้แค่เพียงชนิดของรูปภาพเพียงอย่างเดียว



รูปที่ 2.13 ตัวอย่างทำงานของการตรวจจับวัตถุต่าง ๆ [10]

2.5 ภาษาไพธอน (Python Programming Language)

ประวัติความเป็นมาของไพธอน [11] เป็นภาษาการเขียนโปรแกรมระดับสูงที่พัฒนาโดย Guido van Rossum ชาวดัตช์ ในปี 1989 เดิมทีไพธอนเป็นเพียงโครงการงานอดิเรกของ van Rossum เพื่อพัฒนาภาษาการเขียนโปรแกรมที่อ่านง่ายและเขียนง่าย เนื่องจากเขารู้สึกเบื่อหน่ายกับภาษาการเขียนโปรแกรมอื่น ๆ ในขณะนั้นไพธอนเวอร์ชันแรก คือ 0.9.0 ได้รับการเผยแพร่ในปี 1991 โดยมีคุณสมบัติต่าง ๆ เช่น ประเภทข้อมูลและฟังก์ชันบางส่วนสำหรับการจัดการข้อผิดพลาด

- ไพธอน 1.0 ได้รับการเผยแพร่ในปี 1994 พร้อมฟังก์ชันใหม่เพื่อประมวลผลรายการข้อมูลได้อย่างง่ายดาย เช่น Map, Filter และ Reduce

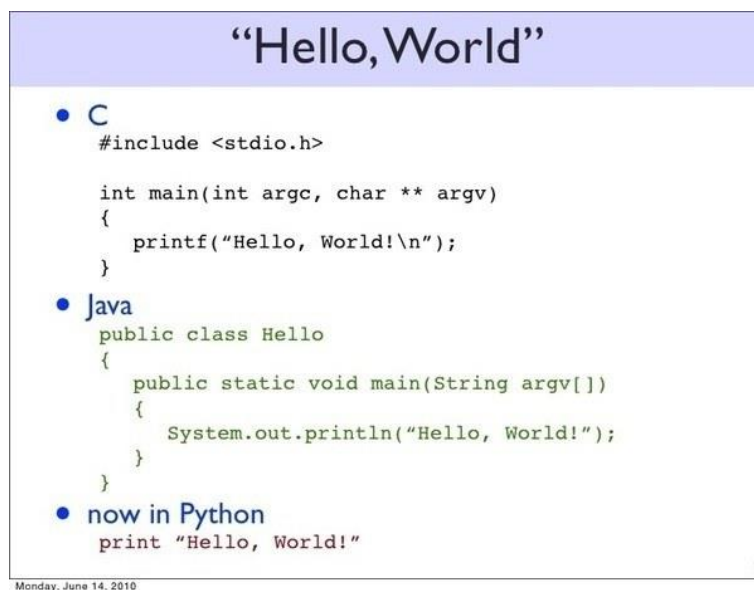
- ไพธอน 2.0 ได้รับการเผยแพร่เมื่อวันที่ 16 ตุลาคม 2000 พร้อมคุณสมบัติใหม่ที่เป็นประโยชน์สำหรับโปรแกรมเมอร์ เช่น การรองรับอักขระ (Unicode) และวิธีที่สั้นกว่าในการวนลูปรายการ
- ไพธอน 3.0 ได้รับการเผยแพร่เมื่อวันที่ 3 ธันวาคม 2008 ซึ่งมีคุณสมบัติต่าง ๆ เช่น ฟังก์ชันการพิมพ์และการสนับสนุนเพิ่มเติมสำหรับการแบ่งหมายเลขและการจัดการข้อผิดพลาด
- เวอร์ชันปัจจุบันของไพธอน คือ ไพธอน 3.12.0 ซึ่งได้รับการเผยแพร่เมื่อวันที่ 3 ตุลาคม 2566

ไพธอนเวอร์ชัน 3.0 นั้นเป็นเวอร์ชันที่สำคัญและมีการเปลี่ยนแปลงที่สำคัญในโค้ด (Code) และโครงสร้างของภาษา ซึ่งทำให้โค้ดที่เขียนสำหรับเวอร์ชัน 2.x ไม่สามารถทำงานกับเวอร์ชัน 3.x ได้โดยตรง ดังนั้น, นักพัฒนาต้องปรับปรุงโค้ดของพวกเขาเพื่อให้เข้ากับการเปลี่ยนแปลงนี้ บางครั้ง, การอัปเดตโค้ดจะต้องทำในลักษณะที่ซับซ้อนและต้องมีการแก้ไขบางส่วนของโค้ดที่ไม่สามารถทำงานกับไพธอน 3.x ได้ ต่อมา ไพธอน 3.x ได้รับการปรับปรุงและพัฒนาต่อเนื่องที่สำคัญทั้งในด้านประสิทธิภาพ, ประสิทธิภาพการทำงาน, และความปลอดภัย, การจัดการสตริง (String), การจัดการไบนารี (Binary) และการเพิ่มประสิทธิภาพในการทำงานของการแปล (Interpreter)

2.5.1 คุณสมบัติของไพธอน

มีจุดเด่นหลายประการที่ทำให้เป็นภาษาการเขียนโปรแกรม ที่ได้รับความนิยม ดังนี้

- อ่านง่ายและเขียนง่าย ไพธอนใช้คำที่เหมือนในภาษาอังกฤษ และไวยากรณ์ที่เรียบง่าย ทำให้ง่ายต่อการเรียนรู้และใช้งาน ตัวอย่างเช่น ตัวแปรในไพธอนประกาศโดยใช้คำกริยา เช่น $x = 1$ แทนที่จะใช้เครื่องหมายเท่ากับ เช่น $x = 1$
- ไพธอน สามารถใช้ในการเขียนโปรแกรมได้หลากหลายประเภท เช่น การพัฒนาเว็บ วิทยาศาสตร์ข้อมูล ปัญญาประดิษฐ์ และอื่น ๆ อีกมากมาย ตัวอย่างเช่น ไพธอนสามารถใช้เพื่อสร้างเว็บไซต์ วิเคราะห์ข้อมูล พัฒนาแอปพลิเคชัน AI และอื่น ๆ
- มีประสิทธิภาพ ไพธอน เป็นภาษาการเขียนโปรแกรมที่มีประสิทธิภาพสูง สามารถใช้สร้างแอปพลิเคชันที่มีประสิทธิภาพสูงได้ ตัวอย่างเช่น ไพธอน สามารถใช้สร้างเกม ซอฟต์แวร์กราฟิก และอื่นๆ
- ฟรีและโอเพนซอร์ส ไพธอน เป็นภาษาการเขียนโปรแกรมฟรีและโอเพนซอร์ส ซึ่งหมายความว่าทุกคนสามารถดาวน์โหลดและใช้งานได้ฟรี ตัวอย่างเช่น ไพธอนสามารถใช้เพื่อสร้างแอปพลิเคชันสำหรับใช้ในองค์กรหรือโรงเรียน



รูปที่ 2.14 รูปแบบโครงสร้างของภาษาไพทอนในการแสดง Hello, World! [11]

2.5.2 ไบบรารีและเฟรมเวิร์ค (library and framework)

ไลบรารีไพธอน [15] เป็นชุดของโค้ดที่ใช้อยู่ ซึ่งนักพัฒนาสามารถใช้ในโปรแกรมไพธอน เพื่อหลีกเลี่ยงการเขียนโค้ดขึ้นใหม่ทั้งหมด ไลบรารีไพธอน มักจัดเรียงตามประเภทของงาน เช่น วิทยาศาสตร์ข้อมูล **แมชชีนเลิร์นนิง** การพัฒนาเว็บ และอื่น ๆ ตัวอย่าง ไลบรารีไพธอนยอดนิยม ได้แก่

- NumPy เป็นไลบรารีสำหรับการจัดการข้อมูลตัวเลข ช่วยให้นักพัฒนาสามารถทำงานกับอาร์เรย์ตัวเลขได้อย่างมีประสิทธิภาพ
- Pandas เป็นไลบรารีสำหรับการวิเคราะห์ข้อมูล ช่วยให้นักพัฒนาสามารถวิเคราะห์ข้อมูลขนาดใหญ่ได้อย่างง่ายดาย
- Matplotlib เป็นไลบรารีสำหรับการสร้างกราฟและแผนภูมิ ช่วยให้นักพัฒนาสามารถแสดงผลข้อมูลได้อย่างมีประสิทธิภาพ
- SciPy เป็นไลบรารีสำหรับการคำนวณทางวิทยาศาสตร์ ช่วยให้นักพัฒนาสามารถดำเนินการคำนวณทางวิทยาศาสตร์ที่ซับซ้อนได้
- Scikit-learn เป็นไลบรารีสำหรับการเรียนรู้ของเครื่อง ช่วยให้นักพัฒนาสามารถสร้างโมเดลการเรียนรู้ของเครื่องได้
- TensorFlow เป็นไลบรารีสำหรับการเรียนรู้เชิงลึก ช่วยให้นักพัฒนาสามารถสร้างโมเดลการเรียนรู้เชิงลึกได้

เฟรมเวิร์คไพธอน เป็นชุดของโมดูลและแพ็คเกจที่ช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันประเภทต่าง ๆ ได้อย่างรวดเร็วและง่ายดาย ตัวอย่างเฟรมเวิร์คไพธอน ได้แก่

- Django เป็นเฟรมเวิร์คสำหรับการพัฒนาเว็บแบบเต็มสแต็ก ช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันเว็บที่ซับซ้อนได้อย่างง่ายดาย
- Flask เป็นเฟรมเวิร์คสำหรับการพัฒนาเว็บแบบไมโครสแต็ก ช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันเว็บขนาดเล็กได้อย่างรวดเร็วและง่ายดาย
- Apache MXNet เป็นเฟรมเวิร์คสำหรับการเรียนรู้เชิงลึก ช่วยให้นักพัฒนาสามารถสร้างโมเดลการเรียนรู้เชิงลึกได้อย่างรวดเร็วและง่ายดาย
- PyTorch เป็นเฟรมเวิร์คสำหรับการเรียนรู้เชิงลึก ช่วยให้นักพัฒนาสามารถสร้างโมเดลการเรียนรู้เชิงลึกได้อย่างรวดเร็วและง่ายดาย

ประโยชน์ของไลบรารีและเฟรมเวิร์คไพธอน ไลบรารีและเฟรมเวิร์คไพธอน มีประโยชน์มากมายสำหรับนักพัฒนาไพธอน ช่วยให้เขียนโปรแกรมได้รวดเร็วและง่ายขึ้น ไลบรารีและเฟรมเวิร์คไพธอน ช่วยให้นักพัฒนาสามารถเข้าถึงฟังก์ชันและความสามารถที่จำเป็นได้โดยไม่ต้องเขียนโค้ดขึ้นใหม่ทั้งหมด ช่วยให้เขียนโปรแกรมได้มีประสิทธิภาพมากขึ้น ไลบรารีและเฟรมเวิร์คไพธอน ได้รับการออกแบบมาเพื่อทำงานได้อย่างมีประสิทธิภาพสูงสุด ช่วยให้เขียนโปรแกรมได้ถูกต้องมากขึ้น ไลบรารีและเฟรมเวิร์คไพธอน มักได้รับการทดสอบอย่างเข้มงวด ซึ่งช่วยลดความเสี่ยงของข้อผิดพลาดในโปรแกรม

2.5.3 การใช้งานของไพธอน

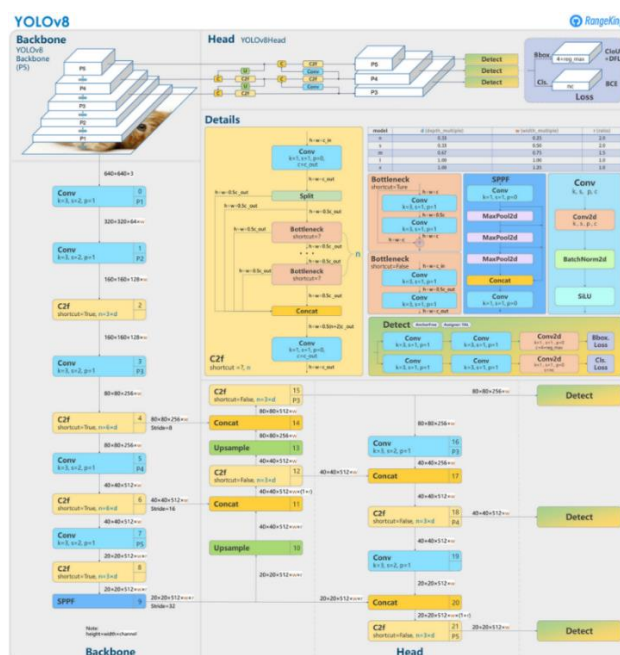
ไพธอนสามารถใช้ในหลากหลายประเภทของงาน การพัฒนาเว็บไพธอน เป็นที่นิยมอย่างมากสำหรับการพัฒนาเว็บ แอปพลิเคชันไพธอนมักใช้กับเฟรมเวิร์คเว็บ เช่น Django และ Flask เว็บไซต์ยอดนิยมหลายแห่ง เช่น Instagram, Spotify และ Reddit สร้างขึ้นโดยใช้ไพธอน

- วิทยาศาสตร์ข้อมูล ไพธอน มักใช้สำหรับการวิเคราะห์ข้อมูลวิทยาศาสตร์และการสร้างแบบจำลองทางสถิติ ตัวอย่างเช่น ไพธอน สามารถใช้เพื่อทำการวิเคราะห์ข้อมูลทางการเงิน วิเคราะห์ข้อมูลทางการแพทย์ และอื่น ๆ
- แมชชีนเลิร์นนิง ไพธอน เป็นที่นิยมอย่างมากสำหรับการพัฒนาแอปพลิเคชันดีปเลิร์นนิง ตัวอย่างเช่น ไพธอน สามารถใช้เพื่อสร้างโมเดลการจำแนกประเภท โมเดลการถดถอย และอื่น ๆ
- ปัญญาประดิษฐ์ ไพธอน มักใช้สำหรับการพัฒนาแอปพลิเคชันปัญญาประดิษฐ์ ตัวอย่างเช่น ไพธอน สามารถใช้เพื่อสร้างหุ่นยนต์ รถยนต์ไร้คนขับ และอื่น ๆ

2.6 โยโล (You Only Look Once : Yolo)

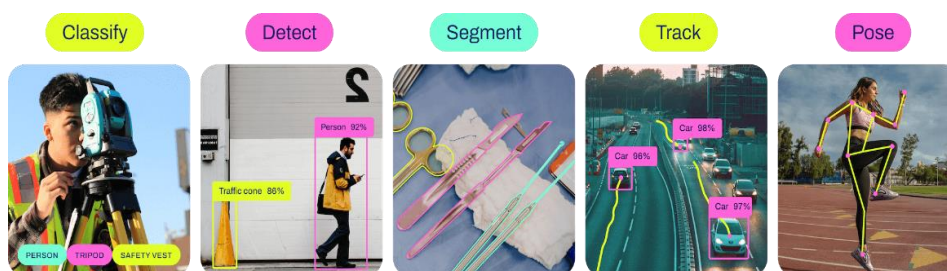
ในปัจจุบัน การใช้การเรียนรู้เชิงลึกเริ่มพบาทมากขึ้น ระบบประสาทเทียมแบบคอนโวลูชันสามารถทำงานกับข้อมูลรูปภาพได้ประสิทธิภาพที่ดีกว่า และเริ่มเข้ามาแทนที่วิธีการแบบเดิม ๆ อัลกอริทึมหนึ่งที่เป็นที่นิยมกันอย่างมากในช่วงเวลานี้คือ You Only Look Once หรือ โยโล ถูกพัฒนาขึ้นในปี ค.ศ.2016 โดย Joseph Redmon และคนอื่น ๆ ให้มีรูปแบบการทำงานเป็น Single Neural Network ที่สามารถทำนายกล่องขอบเขตของวัตถุ (Bounding Box) และทำนายค่าความน่าจะเป็นของประเภทของวัตถุ (Class Probabilities) จากข้อมูลรูปภาพ ได้อย่างรวดเร็ว สถาปัตยกรรมของ โยโล เวอร์ชันแรก ประกอบไปด้วย ชั้นคอนโวลูชันจำนวน 24 ชั้น และชั้นเชื่อมต่อทั้งหมด (Fully Connected layers) จำนวน 2 ชั้น หลักการของอัลกอริทึมโยโล ถูกนำไปพัฒนาอย่างต่อเนื่องเป็นเวอร์ชัน 2, 3 และ ให้มีประสิทธิภาพในการทำงานที่แม่นยำและรวดเร็วมากขึ้น

หลังจากนั้นไม่นาน Glenn Jocher ได้เปลี่ยนการทำงานของโยโล เวอร์ชัน 3 ไปใช้กับเฟรมเวิร์คไพธอน (Pytorch Deep Learning Framework) แทนการใช้เทนเซอร์ฟลว์ (Tensor Flow) เป็นที่รู้จักกันในชื่อโยโล เวอร์ชัน 4 ซึ่งไม่ได้เป็นการนำเสนอองค์ความรู้ใหม่ใดๆ จึงทำให้ยังไม่มีการตีพิมพ์บทความของโยโล เวอร์ชัน 5 และพบว่าในบางงานวิจัยโยโล เวอร์ชัน 5 มีประสิทธิภาพในการทำงานที่ดีกว่าโยโล เวอร์ชัน 4 และ โยโล เวอร์ชัน 3 แต่มีความเร็วในการทำงานใกล้เคียงกับ โยโล เวอร์ชัน 4 อย่างไรก็ตามก็เคยมีผู้ทำการศึกษางานวิจัยที่มีการแสดงผลการเปรียบเทียบผลการจำแนกประเภทของยานพาหนะโดยใช้หลักการโยโล ได้ผลว่าโยโล เวอร์ชัน 5 ทำงานได้ค่อนข้างดีกว่าตัวอื่น



รูปที่ 2.15 YOLOv8 Architecture, Visualisation Made [12]

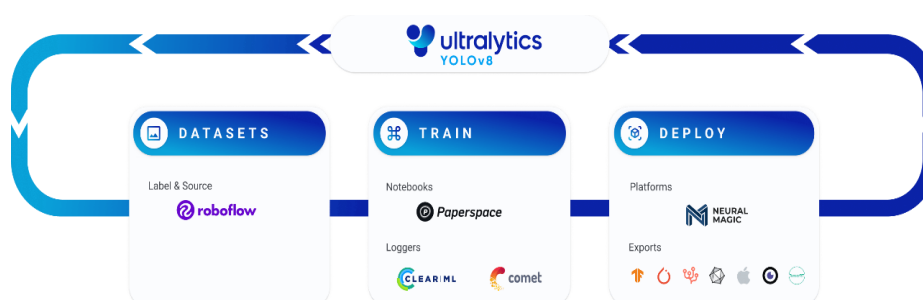
ในปีปัจจุบัน พ.ศ. 2023 Glenn Jocher และผู้พัฒนาอื่น ๆ ที่ใช้ชื่อว่า Ultralytics ได้พัฒนา โยโล เวอร์ชัน 8 ออกมา ซึ่งทำงานในการตรวจจับและจำแนกวัตถุได้ดีกว่า โยโล เวอร์ชัน 5 มีความแม่นยำและรวดเร็วกว่า อีกทั้งยังถูกพัฒนาให้สามารถติดตามวัตถุ รวมทั้งทำนายวัตถุแบบแบ่งส่วนรูปภาพหรือการแบ่งกลุ่ม (Segmentation) ด้วยตัวมันเองได้อีกด้วย



รูปที่ 2.16 ความสามารถของโยโล (เวอร์ชัน 8) ที่ถูกพัฒนาขึ้นอย่างต่อเนื่อง [12]

2.7 โรโบโฟลว์ (Roboflow)

โรโบโฟลว์เป็นแพลตฟอร์มที่ช่วยในการจัดการข้อมูลภาพและสนับสนุนกระบวนการมองเห็นของคอมพิวเตอร์ (Computer Vision) โดยทำให้ง่ายขึ้นในการเตรียมข้อมูลภาพสำหรับการฝึกอัลกอริทึมและโมเดลทางด้าน AI ที่มุ่งเน้นในการทำนายหรือจำแนกภาพ โรโบโฟลว์มีความสามารถในการทำงานกับข้อมูลที่มีขนาดใหญ่และหลายรูปแบบ รวมถึงการดำเนินการเตรียมข้อมูล เช่น การปรับขนาดภาพ, การตัดต่อ, การปรับแต่งความสว่าง, การปรับแต่งสี เป็นต้น ทั้งนี้เพื่อให้ข้อมูลพร้อมใช้งานในกระบวนการฝึกอัลกอริทึมของ AI นอกจากนี้โรโบโฟลว์ ยังมีคุณสมบัติการฝึกอัลกอริทึม (Training) ที่สะดวกและง่ายต่อผู้ใช้ และมีการเชื่อมต่อกับหลาย ๆ โมเดล Computer Vision ที่ได้รับความนิยม เพื่อให้ผู้ใช้สามารถฝึกโมเดลได้ด้วยข้อมูลที่ถูกรวบรวมไว้ในโรโบโฟลว์แล้ว



รูปที่ 2.17 Roboflow มีส่วนช่วยในการทำ Dataset [12]

โรโบโฟลว์มีภารกิจที่ชัดเจนที่จะทำให้กระบวนการทำ Computer Vision เป็นเรื่องที่เข้าใจและสะดวกสบายมากขึ้นสำหรับนักพัฒนาและวิจัยทางด้าน AI โดยทั่วไปโรโบโฟลว์ มีเป้าหมายที่จะลดเส้นทางที่ซับซ้อนในขั้นตอนการเตรียมข้อมูล มีจุดมุ่งหมายที่จะทำให้การใช้ข้อมูลและการฝึกโมเดล AI ด้วยข้อมูลภาพเป็นเรื่องง่ายขึ้น

2.8 การวัดประสิทธิภาพโมเดล (Model Evaluation)

การวัดประสิทธิภาพของโมเดลในเรียนรู้ของเครื่องเป็นขั้นตอนสำคัญที่ช่วยให้เราทราบว่าโมเดลของเราทำงานได้ดีเพียงใด และสามารถตอบโจทย์ปัญหาที่เราต้องการแก้ไขได้หรือไม่ วิธีการและเครื่องมือที่ใช้ในการวัดประสิทธิภาพมีหลายรูปแบบ ขึ้นอยู่กับประเภทของปัญหาและโมเดลที่ใช้

2.8.1 การแบ่งชุดข้อมูล (Training Set, Validation Set, Test Set)

- 1) Training Set ชุดข้อมูลที่ใช้สำหรับการฝึกโมเดล
- 2) Validation Set ชุดข้อมูลที่ใช้ปรับแต่งพารามิเตอร์ของโมเดลและป้องกันการเกิด overfitting
- 3) Test Set ชุดข้อมูลที่ใช้ทดสอบประสิทธิภาพของโมเดลที่ไม่เคยเห็นข้อมูลในขั้นตอนการฝึกหรือการปรับแต่ง

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

รูปที่ 2.18 ตาราง Confusion Matrix [13]

การวัดผลโมเดลในการเรียนรู้ของเครื่อง (Machine Learning) เป็นกระบวนการสำคัญเพื่อให้แน่ใจว่าโมเดลสามารถทำงานได้ดีและตอบสนองความต้องการของผู้ใช้งานได้อย่างเหมาะสม มีค่าและมาตรการต่าง ๆ ที่ใช้ในการวัดผลดังนี้

- 1) ความแม่นยำ (Accuracy) คือการวัดว่าโมเดลทำนายผลได้ถูกต้องกี่ครั้งจากทั้งหมดที่ทำ การทำนาย ใช้บ่อยในปัญหาการจัดประเภท (Classification) เช่น การทำนายว่าอีเมลเป็นสแปมหรือไม่

2) ความถูกต้อง (Precision) คือการวัดว่าในจำนวนครั้งที่โมเดลทำนายว่าเป็นบวก โมเดลทำนายถูกต้องกี่ครั้ง ใช้เมื่อเราต้องการลดจำนวนการทำนายผลบวกที่ผิดพลาด เช่น ในการตรวจหา มะเร็ง เราอยากให้การทำนายว่ามีมะเร็งเป็นสิ่งที่ถูกต้อง

3) การเรียกกลับ (Recall) คือการวัดว่าในจำนวนครั้งที่จริงๆ แล้วเป็นบวก โมเดลสามารถทำนายได้ถูกต้องกี่ครั้ง ใช้เมื่อเราต้องการให้โมเดลสามารถตรวจจับทุกกรณีที่เป็นบวกได้ เช่น ในการตรวจหา มะเร็ง เราต้องการให้โมเดลจับทุกเคสที่มีมะเร็ง

4) ค่า F1 Score เป็นค่าที่ผสมผสานระหว่างความถูกต้องและการเรียกกลับ เพื่อให้ได้ค่าที่แสดงประสิทธิภาพของโมเดลโดยรวม ใช้เมื่อเราต้องการสมดุลระหว่างการทำนายผลบวกที่ถูกต้อง และการตรวจจับกรณีที่เป็นบวก

5) พื้นที่ใต้กราฟ ROC (AUC - Area Under the Curve) ใช้แสดงความสามารถของโมเดล ในการแยกแยะระหว่างคลาสต่างๆ ค่า AUC แสดงพื้นที่ใต้กราฟ ROC ยิ่งค่า AUC สูงเท่าไร โมเดลก็ยิ่งมีความสามารถในการแยกแยะระหว่างคลาสได้ดีขึ้น

6) ค่าเฉลี่ยของค่าผิดพลาดแบบสัมบูรณ์ (MAE - Mean Absolute Error) เป็นการวัดความแตกต่างเฉลี่ยระหว่างค่าที่ทำนายกับค่าจริง ใช้ในปัญหาการถดถอย (Regression) เช่น การทำนายราคาบ้าน

7) ค่าเฉลี่ยของค่าผิดพลาดยกกำลังสอง (MSE - Mean Squared Error) เป็นการวัดความแตกต่างยกกำลังสองเฉลี่ยระหว่างค่าที่ทำนายกับค่าจริง ใช้ในปัญหาการถดถอย เช่นเดียวกับ MAE แต่ให้ความสำคัญกับค่าผิดพลาดใหญ่ๆ มากกว่า

การเลือกใช้ค่าหรือมาตรการใดในการวัดผลขึ้นอยู่กับประเภทของปัญหาและความต้องการของผู้ใช้งาน ค่าเหล่านี้ช่วยให้เราประเมินและปรับปรุงโมเดลได้อย่างมีประสิทธิภาพ

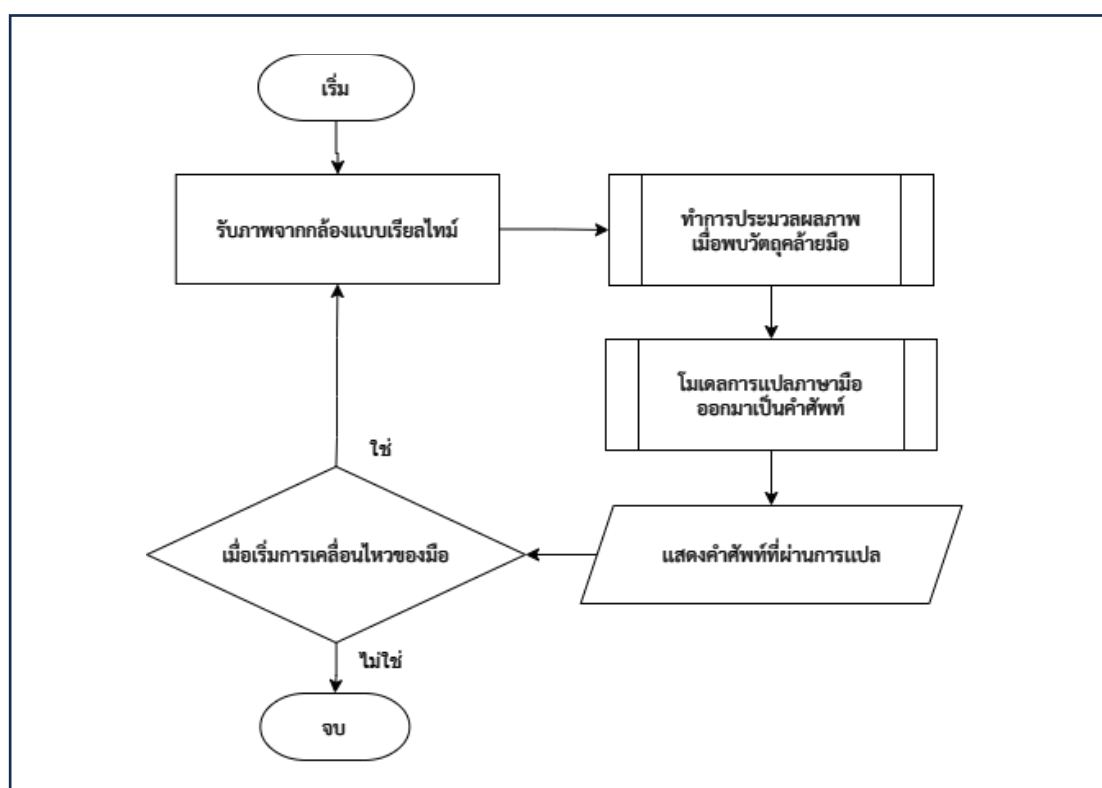
บทที่ 3

ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานของปัญญาประดิษฐ์ หุ่นมือ การจำแนกภาษามือที่ใช้ในชีวิตประจำวัน แบบเรียลไทม์ ซึ่งสามารถแบ่งขั้นตอนการดำเนินงาน ดังนี้

- 3.1 หลักการทำงานของระบบการจำแนกภาษามือและการแปลความหมาย
- 3.2 รวบรวมชุดข้อมูลคำศัพท์และประโยคภาษามือที่ใช้ในชีวิตประจำวัน
- 3.3 การเตรียมชุดข้อมูลก่อนนำเข้าโมเดล YOLOv8
 - 3.3.1 จัดทำรูปภาพของท่าทางตามคำศัพท์
 - 3.3.2 นำเข้าเฟรมเวิร์ค Roboflow เพื่อทำ Label ให้กับรูป
 - 3.3.3 ออกแบบชุดข้อมูลการเทรน
- 3.4 การเทรนชุดข้อมูลผ่านโมเดล YOLOv8 ใน Google Colab
- 3.5 พัฒนาโปรแกรมการจำแนกภาษามือแบบเรียลไทม์ ด้วยโมเดลที่ผ่านการเทรนแล้ว

3.1 หลักการทำงานของระบบจำแนกภาษามือและแปลความหมาย



รูปที่ 3.1 ผังงานโดยรวมของระบบจำแนกภาษามือและแปลความหมาย

จากรูปที่ 3.1 จากกล้องจะทำการรับภาพแต่ละเฟรมเข้ามาแบบเรียลไทม์และทำการประมวลผลภาพทุกเฟรมที่เข้ามาตลอดเวลา เมื่อพบท่าทางหรือมือที่ตรงกับคลาสใดในโมเดลที่ผ่านการเทรนข้อมูลมาแล้ว จะได้ผลลัพธ์ของคำศัพท์ของท่าทางนั้น ๆ สุดท้ายคำศัพท์ที่เข้ามาจะแสดงเป็นประโยคที่สามารถสื่อสารออกมาเป็นประโยคได้อย่างถูกต้องและเข้าใจ

3.2 รวบรวมชุดข้อมูลคำศัพท์และประโยคภาษามือที่ใช้ในชีวิตประจำวัน

ตารางที่ 3.1 บทการสนทนาทั่วไป

ที่	บทการสนทนาทั่วไป
1	สวัสดี
2	สบายดีไหม
3	วันนี้เป็นยังไงบ้าง
4	ทำอะไรอยู่
5	อย่างไร
6	ตอนนี้ก็เผลอแล้ว
7	มาทำอะไร
8	รอสักครู่
9	ไม่เป็นไร
10	เครื่องถ่ายเอกสารอยู่ไหน
11	ฉันชอบทำงานกับคุณ
12	วันนี้เขาไม่อยู่

ตารางที่ 3.2 การเสนอความช่วยเหลือ

ที่	การเสนอความช่วยเหลือ
1	มีอะไรให้ฉันช่วยไหม
2	คุณอยากได้อะไร
3	คุณอยากให้ฉันทำอะไร
4	ฉันช่วยทำงานไหม
5	ฉันช่วยคุณได้

ตารางที่ 3.3 การเสนอความคิดเห็น

ที่	การเสนอความคิดเห็น
1	คิดอย่างไร
2	เห็นด้วยกับฉันไหม
3	ฉันเห็นด้วย
4	ฉันไม่เห็นด้วย
5	นั่นเป็นความคิดที่ดี
6	ฉันคิดเหมือนกัน
7	ฉันรู้สึกเหมือนกัน

ตารางที่ 3.4 การตอบสนองสื่อสาร

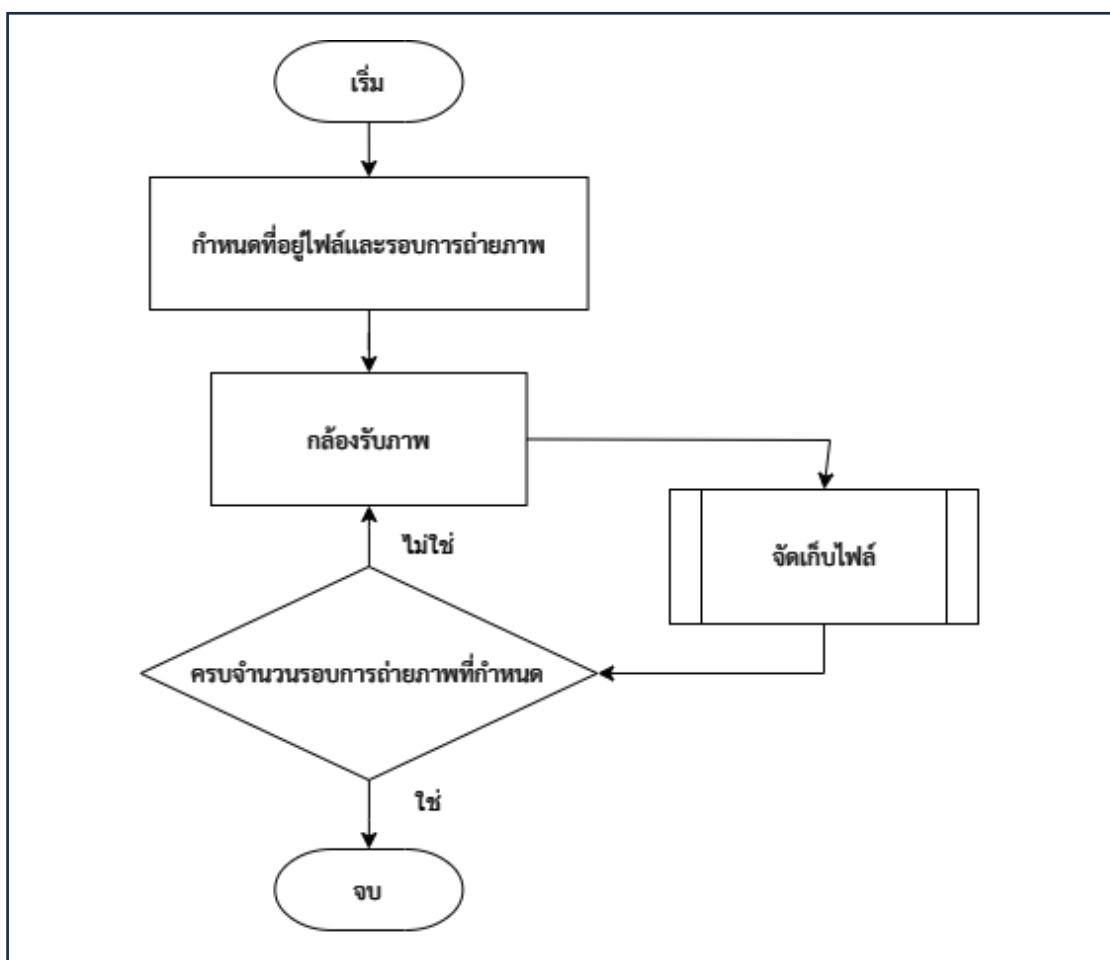
ที่	การตอบสนองสื่อสาร
1	ฉันเข้าใจ
2	ฉันไม่เข้าใจ
3	ฉันไม่เป็นไร
4	ฉันไม่สบาย
5	ฉันทำได้
6	ฉันทำไม่ได้
7	งานของคุณเยี่ยมมาก
8	ได้เลย, ยินดี
9	ขอบคุณ

สรุป จำนวนประโยคทั้งหมดรวม 33 ประโยค และจำนวนคำศัพท์ 50 คำ

3.3 การเตรียมชุดข้อมูลก่อนนำเข้าโมเดล YOLOv8

การเตรียมชุดข้อมูลเป็นขั้นตอนสำคัญที่มีผลต่อประสิทธิภาพของการทำ Object Detection โดยใช้โมเดล YOLOv8 ซึ่งเป็นหนึ่งในโมเดลที่ได้รับความนิยมในการตรวจจับวัตถุด้วยการใช้เทคนิคการเรียนรู้เชิงลึก (deep learning) ขั้นตอนที่สำคัญในการเตรียมชุดข้อมูลก่อนนำเข้าโมเดล YOLOv8 มีดังนี้

3.3.1 จัดทำรูปภาพของท่าทางตามคำศัพท์



รูปที่ 3.2 ผังงานโดยรวมการทำงานของระบบจัดทำรูปภาพ

จากรูปที่ 3.2 เป็นการทำงานของโปรแกรมถ่ายภาพอัตโนมัติตามจำนวนคำศัพท์ที่กำหนดและจำนวนรอบการถ่าย โดยทำให้การเก็บข้อมูลรูปภาพท่าทางตามคำศัพท์มีความสะดวกมากขึ้นทั้งในการตั้งชื่อไฟล์ และเก็บลงโฟลเดอร์แยกแต่ละคำศัพท์ ทำให้สามารถจัดเก็บข้อมูลรูปภาพได้อย่างสะดวกรวดเร็วมากยิ่งขึ้น

โปรแกรมที่ 3.1 จัดทำรูปภาพที่ใช้เทรนข้อมูลก่อนนำเข้าโมเดล

```

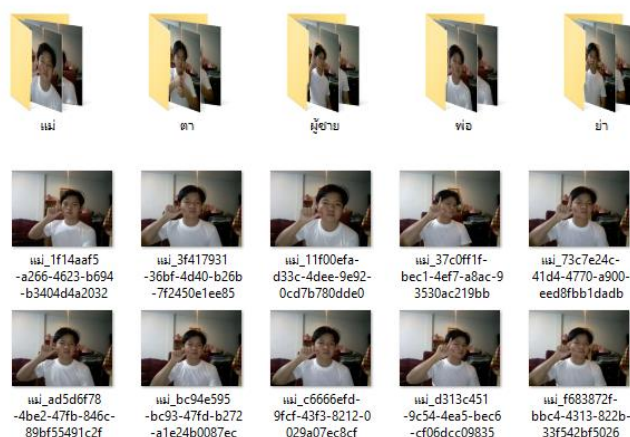
1. import cv2
2. import os
3. import time
4. import uuid
5. IMAGES_PATH = 'G:/Data_Images'
6. labels = ['dad', 'mom', 'child']
7. number_imgs = 15
8. for label in labels:
9.     os.makedirs(os.path.join(IMAGES_PATH, label), exist_ok=True)
10.    using os.makedirs
11.    cap = cv2.VideoCapture(0)
12.    print('Collecting images for {}'.format(label))
13.    time.sleep(5)
14.    for imgnum in range(number_imgs):
15.        ret, frame = cap.read()
16.        image_name = os.path.join(IMAGES_PATH, label, '{}_{}.jpg'.format(label,
            str(uuid.uuid4())))
17.        cv2.imwrite(image_name, frame)
18.        cv2.imshow('frame', frame)
19.        time.sleep(2)
20.        if cv2.waitKey(1) & 0xFF == ord('q'):
21.            break
22.    cap.release()
23. cv2.destroyAllWindows()

```

จากโปรแกรมที่ 3.1 สามารถอธิบายการทำงานได้ ดังนี้

- บรรทัดที่ 1 นำเข้าไลบรารี OpenCV สามารถใช้ฟังก์ชันเกี่ยวกับการประมวลผลภาพได้
- บรรทัดที่ 2 นำเข้าไลบรารี os เพื่อให้สามารถทำงานกับระบบไฟล์และไดเรกทอรี
- บรรทัดที่ 3 นำเข้าไลบรารี time เพื่อให้สามารถใช้ฟังก์ชันที่เกี่ยวกับการทำงานเวลา

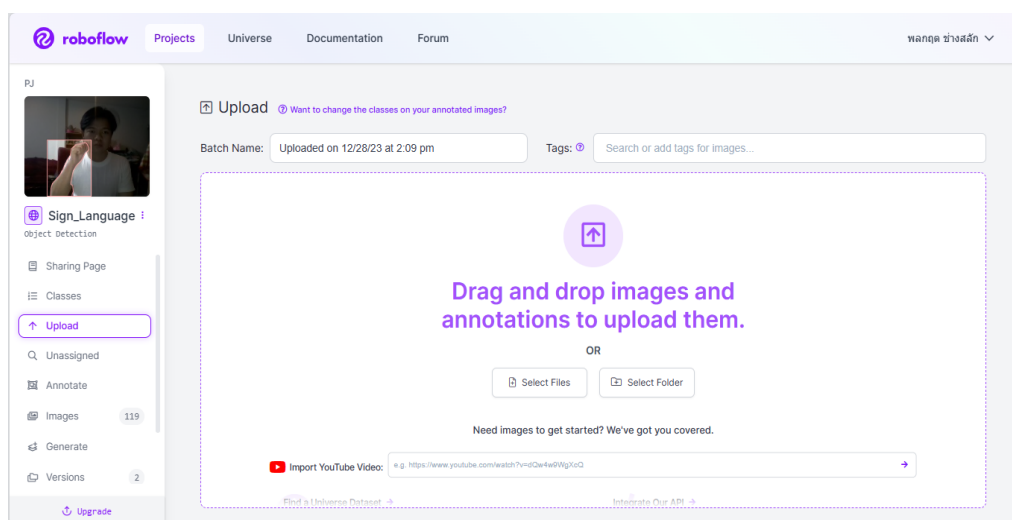
- บรรทัดที่ 4 นำเข้าไลบรารี uuid เพื่อให้สามารถสร้าง UUID
- บรรทัดที่ 5 กำหนดตัวแปรและกำหนดค่าเป็น 'G:/Data_Images' ซึ่งเป็นที่เก็บรูปภาพ
- บรรทัดที่ 6 กำหนดรายการ labels ซึ่งเก็บข้อมูลเกี่ยวกับป้ายกำกับของรูปภาพ
- บรรทัดที่ 7 กำหนดตัวแปรและกำหนดค่าเป็น 15 ซึ่งเป็นจำนวนรูปภาพที่ต้องการจะดึงมาในแต่ละป้ายกำกับ
- บรรทัดที่ 8-23 ใช้ลูป for เพื่อวนลูปทุกรายการใน labels โดยทำงานต่อไปตามลำดับ:
- บรรทัดที่ 9 ใช้ os.makedirs เพื่อสร้างไดเรกทอรีที่จะใช้เก็บรูปภาพของแต่ละป้ายกำกับถ้าไดเรกทอรีนี้มีอยู่แล้วก็จะไม่ทำการสร้างใหม่
- บรรทัดที่ 11 เปิดการทำงานกล้องด้วย cv2.VideoCapture(0) โดย 0 หมายถึงการใช้กล้องต่ออยู่กับคอมพิวเตอร์
- บรรทัดที่ 12 แสดงข้อความบอกว่ากำลังรวบรวมรูปภาพสำหรับป้ายกำกับที่กำลังทำงาน
- บรรทัดที่ 13 หน่วงเวลา 5 วินาที เพื่อให้ผู้ใช้มีเวลาเตรียมตัว
- บรรทัดที่ 14-21 ใช้ลูป for เพื่อรวบรวมรูปภาพจากกล้อง
- บรรทัดที่ 15 ใช้ cap.read() เพื่ออ่านภาพจากกล้อง
- บรรทัดที่ 16 สร้างชื่อไฟล์รูปภาพโดยใช้ UUID เพื่อป้องกันชื่อที่ซ้ำกัน
- บรรทัดที่ 17 ใช้ cv2.imwrite เพื่อบันทึกรูปภาพลงในไดเรกทอรีที่กำหนด
- บรรทัดที่ 18 ใช้ cv2.imshow เพื่อแสดงรูปภาพที่ถูกจับไว้
- บรรทัดที่ 19 หน่วงเวลา 2 วินาที
- บรรทัดที่ 20-21 ตรวจสอบว่าผู้ใช้กด 'q' หรือไม่ ถ้ากดให้หยุดการทำงาน
- บรรทัดที่ 22 ปิดการทำงานของกล้อง
- บรรทัดที่ 23 ปิดหน้าต่างทั้งหมดที่เปิดขึ้นมาด้วย cv2.destroyAllWindows()



รูปที่ 3.3 จากโปรแกรมจะได้ภาพที่มีชื่อ และถูกเก็บลงในโฟลเดอร์ที่กำหนด

3.3.2 นำเข้าเฟรมเวิร์คโรโบโฟลว์เพื่อกำหนดวัตถุให้กับรูปภาพ

การกำหนด bounding box หรือการระบุตำแหน่งของ object ที่ต้องการทำนายให้รูปภาพ และบันทึกข้อมูลนี้ในรูปแบบที่โมเดลรองรับ โดยเฟรมเวิร์ค Roboflow เป็นเครื่องมือที่ช่วยในการกำหนด Label Image ให้กับรูปภาพได้อย่างสะดวกและรวดเร็วการกำหนด Label Image ให้กับรูปภาพมีความสำคัญอย่างยิ่งสำหรับการสร้างโมเดลการรู้จำภาพ เนื่องจากข้อมูล Label Image จะเป็นข้อมูลที่ใช้ฝึกสอนโมเดลให้สามารถระบุ object ต่าง ๆ ในรูปภาพได้อย่างถูกต้องแม่นยำ



รูปที่ 3.4 การอัปโหลดรูปภาพไปยัง Roboflow



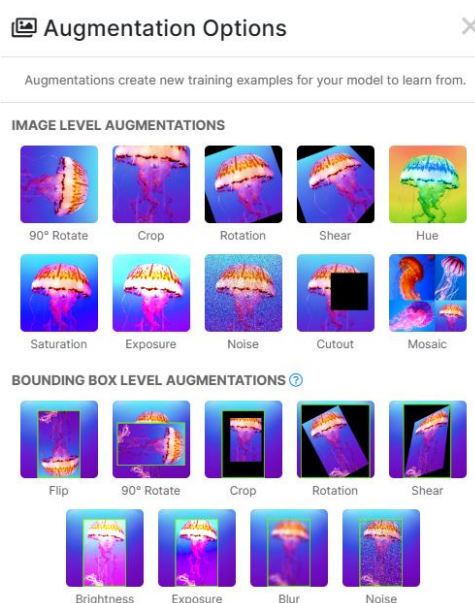
รูปที่ 3.5 ขั้นตอนการกำหนด Bounding Box ให้กับภาพ

3.3.3 ออกแบบชุดข้อมูลการเทรน

Augmentation เป็นกระบวนการเพิ่มข้อมูลให้กับชุดข้อมูลภาพโดยใช้เทคนิคต่าง ๆ เช่น การพลิก การหมุน การย่อ/ขยาย และการบิดเบี้ยว วัตถุประสงค์ของการ Augmentation คือ การทำให้ชุดข้อมูลมีความหลากหลายมากขึ้น ซึ่งจะช่วยให้โมเดลการเรียนรู้ของเครื่องสามารถเรียนรู้ลักษณะของวัตถุต่าง ๆ ได้อย่างแม่นยำยิ่งขึ้น

โดยได้นำเทคนิคต่อไปนี้มาใช้ในชุดข้อมูล

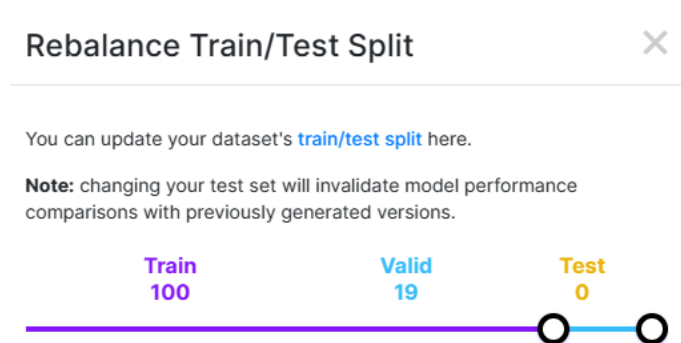
1. Flip : Horizontal
2. Rotate : -10° and $+10^\circ$
3. Grayscale : Apply to 100% of images
4. Brightness : Between 0% and +40%
5. Blur : Up to 2.5px



รูปที่ 3.6 การ Augmentation ที่มีใน Roboflow

ข้อมูลภาพที่ผ่านการทำ Image Labeling เป็นข้อมูล Training, Validation และ Test โดยการใช้ข้อมูลรูปจำนวนทั้งหมด -- รูป ด้วยอัตราส่วนดังนี้

- Training Dataset = 80% ของ Dataset (-- รูป)
- Validation Dataset = 20% ของ Training Dataset (-- รูป)
- Test Dataset = ทดสอบโดยกล้องแบบเรียลไทม์ และ ทำทางวิดีโอคำศัพท์



รูปที่ 3.7 การแบ่งอัตราส่วนชุดข้อมูลก่อนการเทรน

3.4 การเทรนชุดข้อมูลผ่านโมเดล YOLOv8 ใน Google Colab

ทำการติดตั้ง Ultralytics เพื่อเรียกใช้โมเดล yolov8x.pt ที่จะนำมาใช้เป็นโมเดลในการฝึกสอนและการทำการประเมินประสิทธิภาพของโมเดลหลังการเทรน

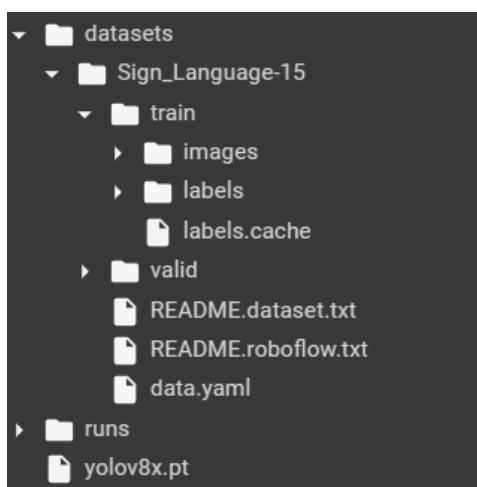
โปรแกรมที่ 3.2 ทำการติดตั้งไลบรารี และดาวน์โหลด dataset

```
1. !pip install ultralytics==8.0.20
2. import ultralytics
3. ultralytics.checks()
4. !mkdir {HOME}/datasets
5. %cd {HOME}/datasets
6. !pip install roboflow
7. from roboflow import Roboflow
8. rf = Roboflow(api_key="F1Un5ituuD1yUDZ6n3ub")
9. project = rf.workspace("pj-ovbgc").project("sign_language-4ix3j")
10. dataset = project.version(15).download("yolov8")
```

จากโปรแกรมที่ 3.2 สามารถอธิบายการทำงานได้ ดังนี้

- บรรทัดที่ 1 ใช้ติดตั้งไลบรารี Ultralytics เวอร์ชัน 8.0.20 โดยใช้คำสั่ง ! ใน Google Colab เพื่อให้สามารถใช้ Ultralytics ในโปรเจกต์
- บรรทัดที่ 2 นำเข้าไลบรารี Ultralytics เพื่อให้สามารถใช้ฟังก์ชันและคลาสต่าง ๆ ที่ Ultralytics มีได้

- บรรทัดที่ 3 ตรวจสอบสภาพแวดล้อมและข้อกำหนดของ Ultralytics เพื่อให้แน่ใจว่าทุกอย่างพร้อมใช้งาน
- บรรทัดที่ 4 สร้างไดเรกทอรีชื่อ datasets ในโฟลเดอร์ HOME โดยใช้คำสั่ง mkdir ซึ่งเป็นการสร้างไดเรกทอรีใหม่
- บรรทัดที่ 5 เปลี่ยนไดเรกทอรีทำงานไปที่ {HOME}/datasets โดยใช้คำสั่ง %cd
- บรรทัดที่ 6 ใช้ติดตั้งไลบรารี Roboflow เพื่อให้สามารถใช้งานร่วมกับ Roboflow API
- บรรทัดที่ 7 นำเข้าคลาส Roboflow จากไลบรารี Roboflow
- บรรทัดที่ 8 สร้างอ็อบเจกต์ Roboflow และกำหนด API key เพื่อทำการยืนยันตัวตน
- บรรทัดที่ 9 ระบุ workspace และ project ใน Roboflow โดยใช้ชื่อ workspace และ project
- บรรทัดที่ 10 ดาวน์โหลด dataset ที่เป็นเวอร์ชันที่ 15 จาก project และทำการเลือกใช้โมเดล YOLOv8 สำหรับการตรวจจับวัตถุ



รูปที่ 3.8 เมื่อดาวโหลดสำเร็จจะได้ไฟล์โมเดลสำหรับการฝึกสอนและชุดข้อมูล

หลังทำการติดตั้ง Ultralytics สามารถเรียกใช้โมเดล yolov8x.pt ที่จะนำมาใช้ในการฝึกสอนโมเดล และติดตั้ง Roboflow ที่ใช้ในการโหลดชุดข้อมูลด้วย API key ของเฟรมเวิร์คจะได้โฟลเดอร์ของการ Train และ Valid ชุดข้อมูล ในโฟลเดอร์จะประกอบด้วย images กับ labels ของรูปภาพที่จะนำมาประกอบการเทรนโมเดล ภายในไฟล์ data.yaml จะประกอบไปด้วยคลาสในแต่ละรูปภาพของชุดข้อมูล

โปรแกรมที่ 3.3 คำสั่งแทนข้อมูลด้วย Yolo8

```
1. !yolo task=detect mode=train model=yolov8x.pt data={dataset.location}/data.yaml
   epochs=30 imgsz=640 plots=True
```

จากโปรแกรมที่ 3.3 สามารถอธิบายการทำงานได้ ดังนี้

บรรทัดที่ 1 !yolo: ใช้เรียกโปรแกรม YOLO

task=detect: กำหนดให้ YOLO ทำหน้าที่ตรวจจับวัตถุ

mode=train: กำหนดให้ YOLO ทำการฝึก (training) โมเดล

model=yolov8x.pt: เลือกใช้ yolov8x.pt

data={dataset.location}/data.yaml: กำหนดที่อยู่ของไฟล์ data.yaml ที่ใช้ในการฝึกโมเดล

epochs=30: กำหนดจำนวนรอบการฝึก (epochs) เป็น 30

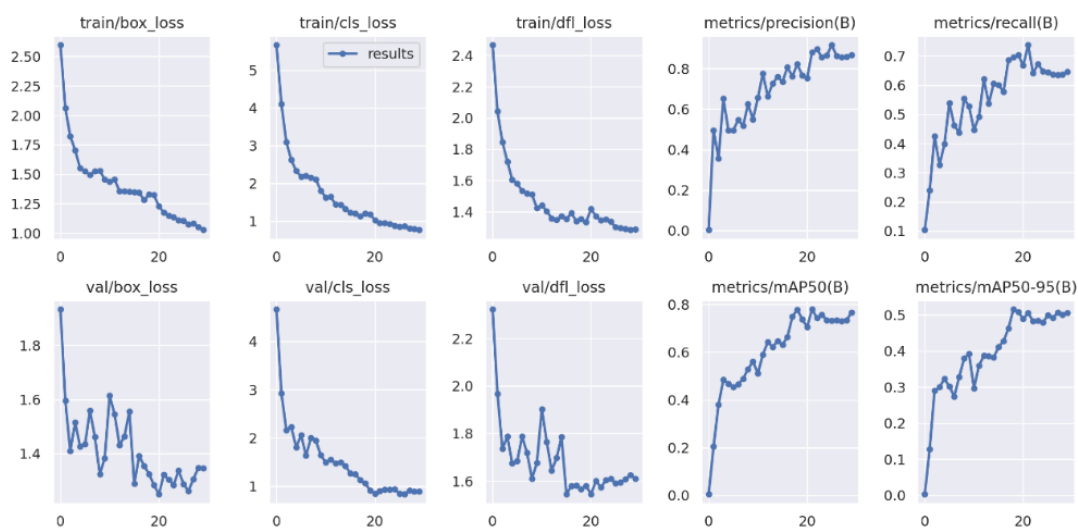
imgsz=640: กำหนดขนาดของรูปภาพที่ใช้ในการฝึกเป็น 640x640 pixels

plots=True: กำหนดให้แสดงกราฟผลลัพธ์ขณะฝึก

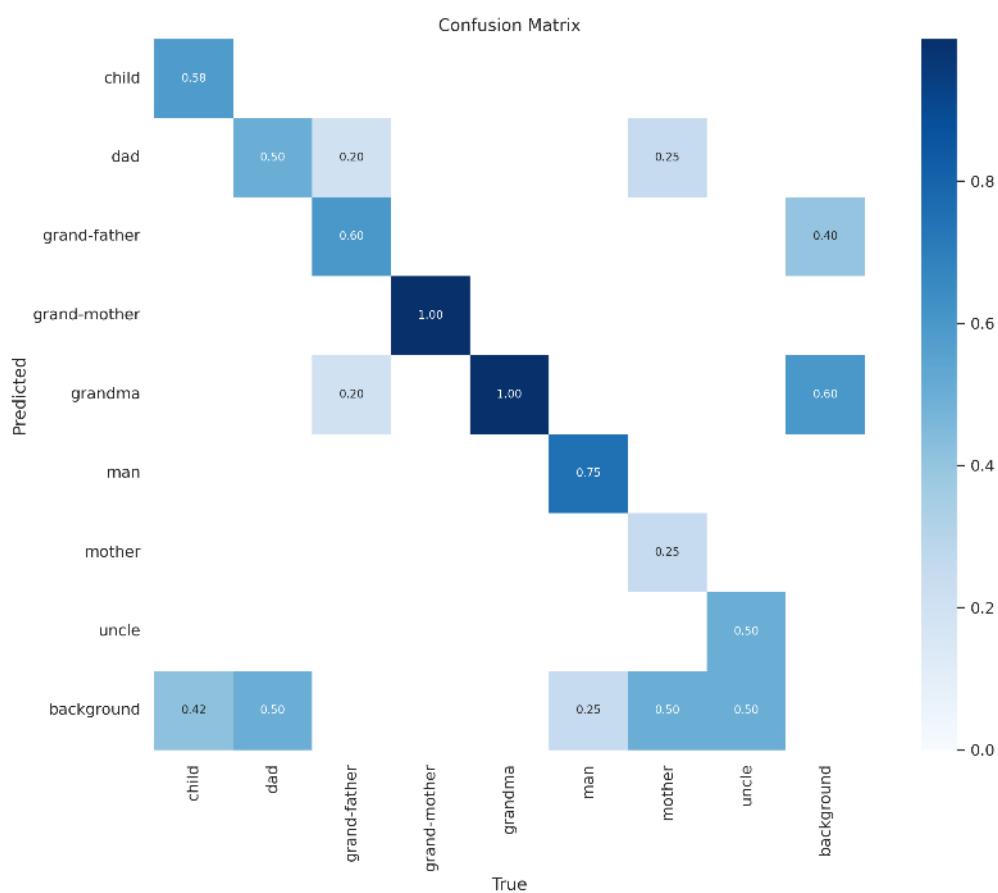
Model summary (fused): 268 layers, 68131272 parameters, 0 gradients, 257.4 GFLOPs						
Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	19	41	0.823	0.696	0.778	0.515
child	19	12	1	0.631	0.766	0.539
dad	19	4	0.502	0.5	0.622	0.386
grand-father	19	5	0.785	0.739	0.788	0.522
grand-mother	19	2	0.902	1	0.995	0.746
grandma	19	4	0.512	1	0.912	0.688
man	19	4	1	0.733	0.856	0.428
mother	19	4	1	0.464	0.746	0.322
uncle	19	6	0.88	0.5	0.539	0.49
Speed: 0.2ms pre-process, 22.8ms inference, 0.0ms loss, 1.1ms post-process per image						

รูปที่ 3.9 ตัววัดที่ใช้เพื่อประเมินคุณภาพและประสิทธิภาพของโมเดล

ตัวชี้วัดเหล่านี้ช่วยให้ทราบถึงประสิทธิภาพของโมเดลขณะทำการจำแนกประเภทของวัตถุในรูปภาพ ค่าของการประเมินประสิทธิภาพของโมเดลระบุวัตถุ และความแม่นยำในการทำนายคลาสต่าง ๆ สามารถนำค่าที่ได้มาปรับแก้ให้เข้ากันกับชุดข้อมูลที่สุด ทำให้การใช้งานโมเดลนี้มีประสิทธิภาพสูงสุด ค่าต่าง ๆ ที่ได้จากการเทรนมีดังต่อไปนี้



รูปที่ 3.10 กราฟแสดงค่าการสูญเสียและประสิทธิภาพของโมเดลในแต่ละรอบการฝึก



รูปที่ 3.11 ตาราง Confusion Matrix แสดงผลลัพธ์การทำนายของคลาสต่าง ๆ

เมื่อนำโมเดลมาทำการทดสอบด้วยรูปภาพหรือวิดีโอ นั้น โมเดลที่ได้รับการเทรนจะสามารถทำนายคลาสของการแปลภาษามือ จะแสดงผลลัพธ์กรอบสี่เหลี่ยมรอบท่าทางหรือมือและมี label กำหนดบอกชื่อของคลาสนั้น ๆ ดังรูปที่ 3.11



รูปที่ 3.12 ภาพตัวอย่างการทำนายคลาสของโมเดลที่ผ่านการเทรน

3.5 พัฒนาโปรแกรมการจำแนกภาษามือแบบเรียลไทม์ ด้วยโมเดลที่ผ่านการเทรนแล้ว