



แอปพลิเคชันจำแนกสายพันธุ์ระบบองเพชร

Application for Cactus Specie Recognition

นายกฤษณพงษ์ ปันติ รหัส 60360210

นายเจษฎา ดวงปัญญาสว่าง รหัส 60360791

นายวสิษฐ์พล อุ่นทา รหัส 60364249

ปริญนานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2563



ใบรับรองปริญญาบัตร

ชื่อหัวข้อโครงการ แอปพลิเคชันจำแนกสายพันธุ์ระบบองเพชร

ผู้ดำเนินโครงการ นายกฤษณพงษ์ ปันติ รหัส 60360210

นายเจษฎา ดวงปัญญาสว่าง รหัส 60360791

นายวสิษฐ์พล อุ่นاث รหัส 60364249

ที่ปรึกษาโครงการ อาจารย์รัฐภูมิ วรรณุสาสน์

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2563

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญาบัตรฉบับนี้เป็นส่วนหนึ่งของ
การศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ

(อาจารย์รัฐภูมิ วรรณุสาสน์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ริยะมงคล)

.....กรรมการ

(รองศาสตราจารย์ ดร. พงศ์พันธ์ กิจสนโยธิน)

ชื่อหัวข้อโครงการ	แอปพลิเคชันจำแนกสายพันธุ์ระบบองเพชร		
ผู้ดำเนินโครงการ	นายกฤษณพงษ์ ปันติ	รหัส	60360210
	นายเจษฎา ดวงปัญญาสว่าง	รหัส	60360791
	นายวสิษฐ์พล อุ่นاث	รหัส	60364249
ที่ปรึกษาโครงการ	อาจารย์รัฐภูมิ วรรณสาสน์		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2563		

บทคัดย่อ

โครงการนี้นำเสนอการพัฒนาแอปพลิเคชันสำหรับจำแนกสายพันธุ์ระบบองเพชร โดยใช้เทคโนโลยีการเรียนรู้เชิงลึกในการจักภาพถ่ายต้นระบบองเพชร และทำการทดลองเพื่อเปรียบเทียบความแม่นยำของโมเดลการเรียนรู้เชิงลึกอีก 3 โมเดล ได้แก่ EfficientDet-D5, Faster-RCNN Resnet-50 และ YOLOv5x โดยใช้ชุดข้อมูลที่ถ่ายจากต้นระบบองเพชรจริงทั้งหมด 15 สายพันธุ์และนำไปผ่านกระบวนการเพิ่มภาพจนได้ภาพทั้งหมดที่ 7,500 ภาพ นำไปฝึกฝนและทดสอบจนได้โมเดลที่สามารถจำแนกสายพันธุ์ระบบองเพชรได้ความแม่นยำ 45% 76% และ 67% ตามลำดับ และเพื่อให้ผู้ใช้งานได้สะดวกจึงนำโมเดลที่ได้ไปพัฒนาเป็นแอปพลิเคชันบนโทรศัพท์สมาร์ทโฟนโดยใช้เครื่องมือที่สามารถพัฒนาแอปพลิเคชันบนหลายแพลตฟอร์ม

Project title	Application for Cactus Specie Recognition		
Name	Mr. Gritsanapong Punti	ID. 60360210	
	Mr. Jessada Duangpanyasawang	ID. 60360791	
	Mr. Wasitphon Unat	ID. 60364249	
Project advisor	Mr. Rattapoom Waranusast		
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic year	2020		

Abstract

This project presents the development of application for classifying cactus species by using deep learning technology from cactus images. In this work, we tested and compared the accuracies of 3 deep learning models; EfficientDet-D5, Faster-RCNN Resnet-50, and YOLOv5x using an image dataset that was composed of images from 15 cactus species, with augmented to a total of 7,500 images. The models were trained and tested with the dataset and yielded the accuracies of 45%, 76%, and 67%, respectively. For users' convenience, a smartphone application has been developed using tools that support cross-platform technology.

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ของอาจารย์ที่ปรึกษา โครงการคือ อาจารย์รัฐวุฒิ วรรณสาสน์ ซึ่งได้สละเวลาในการให้คำแนะนำ ให้คำปรึกษา และเสนอแนวทางแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้น รวมถึงการติดตาม และให้คำแนะนำในการเขียนบทความมาโดยตลอด พร้อมด้วยอาจารย์คณะกรรมการทั้ง 2 ท่านคือ ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล และ รองศาสตราจารย์ ดร.พงศ์พันธ์ กิจสนายேธิน ที่ให้คำแนะนำในการเขียนโครงการ

คณะผู้จัดทำบทความจึงขอขอบคุณทุก ๆ ท่านที่มีส่วนร่วมในการให้ความรู้และคำปรึกษาต่าง ๆ ที่เกี่ยวกับบทความนี้ จนสำเร็จลุล่วงในที่สุด

ผู้ดำเนินโครงการ

กฤษณพงษ์ ปันติ

เจษฎา ดวงปัญญาสว่าง

วสิษฐ์พล อุ่นاث

มีนาคม 2564

สารบัญ

ใบรับรองปริญญาอนิพนธ์	ก
บทคัดย่อ	ข
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ฉ
สารบัญรูปภาพ	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตการทำงาน	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 แผนการดำเนินงาน	3
1.7 รายละเอียดงบประมาณตลอดโครงการ	5
บทที่ 2 หลักการและทฤษฎีที่เบื้องต้น	6
2.1 ทฤษฎีที่เกี่ยวข้อง	6
2.1.1 ปัญญาประดิษฐ์ (Artificial Intelligence: AI)	6
2.1.1.1 การคิดแบบมนุษย์	6
2.1.1.2 การกระทำแบบมนุษย์	6
2.1.1.3 การคิดอย่างมีเหตุผล	7

2.1.1.4 การทำงานอย่างมีเหตุผล.....	7
2.1.2 การเรียนรู้ของเครื่องจักร (Machine Learning)	7
2.1.2.1 การเรียนรู้แบบมีผู้สอน (Supervised learning)	7
2.1.2.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning).....	7
2.1.3 โครงข่ายประสาทเทียม (Artificial Neuron Network: ANN)	8
2.1.4 การเรียนรู้เชิงลึก (Deep learning).....	9
2.1.5 การตรวจจับวัตถุ (Object Detection)	9
2.1.6 โครงข่ายประสาทเทียมแบบconvโอลูชัน (Convolution Neural Network: CNN).10	
2.1.7 Faster-RCNN	11
2.1.7.1 Feature maps.....	11
2.1.7.2 Region Proposal Network (RPN)	11
2.1.7.3 การจำแนกประเภทของวัตถุ.....	11
2.1.8 EfficientDet.....	12
2.1.9 Resnet (Deep Residual Network)	14
2.1.10 YOLO (You Only Look Once).....	15
2.1.11 ฐานข้อมูล (Database).....	16
2.1.11.1 JSON (JavaScript Object Notation).....	16
2.1.12 เฟรมเวิร์คสำหรับพัฒนาแอปพลิเคชัน (React Native)	18
2.1.13.1 ชุดพัฒนาซอฟต์แวร์ (Expo).....	18
2.1.13 Application Programming Interface (API).....	19
2.1.14 REST API	19
2.1.14.1 RESTful Web Services	20

2.1.15 สายพันธุ์ระบบองเพชร.....	20
2.1.15.1 หมวกสั้นΏราช (<i>Astrophytum myriostigma</i>)	20
2.1.15.2 ปราสาทนาฬีา (<i>Cereus peruvianus</i>).....	21
2.1.15.3 ไก่ตื๊อก (<i>Haworthia coarctata</i>).....	21
2.1.15.4 กระบองทอง (<i>Echinocactus leninghausii</i>).....	22
2.1.15.5 แม่มอรุชานุ (<i>Mammillaria bucareliensis</i>).....	22
2.1.15.6 ดาวล้อมเดือน (<i>Echinopsis calochalora</i>)	23
2.1.15.7 นูปต้า (<i>Echinopsis subdenudata</i>).....	23
2.1.15.8 บานเนียนมัม (<i>Gymnocalycium baldianum</i>)	24
2.1.15.9 มิหานิ (<i>Gymnocalycium mihanovichii</i>).....	24
2.1.15.10 แม่มกลวยหอม (<i>Mammillaria longimamma</i>)	25
2.1.15.11 หนามพิกุล (<i>Mammillaria scrippsiana</i>)	25
2.1.15.12 ลูกแมว (<i>Mammillaria bocasana</i>)	26
2.1.15.13 ชนนากขาว (<i>Mammillaria plumosa</i>).....	26
2.1.15.14 นิ่วทอง (<i>Mammillaria elongata</i>)	27
2.1.15.15 หุ่มิกกี้เม้าส์ (<i>Opuntia microdasys</i>).....	27
2.2 งานวิจัยที่เกี่ยวข้อง.....	28
บทที่ 3 วิธีดำเนินโครงการ.....	37
3.1 ภาพรวมของระบบ	37
3.2 การออกแบบส่วนประสาน และการพัฒนาแอปพลิเคชัน	38
3.2.1 การออกแบบส่วนประสาน (User Interface Design)	38
3.3 การออกแบบและพัฒนาระบบส่วนหลัง	40

3.3.1 ระบบส่วนหลัง.....	40
3.3.2 ระบบประมวลผลภาพ	41
3.3.2.1 การตรวจหากรอบของเพชร.....	41
3.4 การพัฒนาระบบทรัจจับกรอบของเพชร	42
3.4.1 การเตรียมชุดข้อมูล	42
3.4.1.1 รูปภาพกรอบของเพชร.....	42
3.4.1.2 การปรับแต่งรูปภาพ (Image augmentation)	43
3.4.1.3 การกำกับรูปภาพ (Image labeling).....	43
3.4.2 การเรียนรู้เชิงลึก (Deep learning).....	44
บทที่ 4 ผลการทดลองและวิเคราะห์.....	45
4.1 ผลการทดสอบส่วนแสดงผล.....	45
4.1.1 การทดสอบส่วนแสดงผลการเลือกที่มาข้อมูล	45
4.1.2 การทดสอบส่วนแสดงผลการเลือกถ่ายรูปภาพ	46
4.1.3 การทดสอบส่วนแสดงผลการเลือกอัพโหลดรูปภาพ	47
4.2 ผลการทดสอบการรับส่งข้อมูลรูปภาพ.....	48
4.3 ผลการทดสอบระบบจำแนกสายพันธุ์กรอบของเพชร	49
บทที่ 5	53
5.1 สรุปผลการทดลอง	53
5.2 วิจารณ์ผลการทดลอง	53
5.3 แนวทางการนำโครงงานไปพัฒนาต่อในอนาคต	54
5.4 ข้อเสนอแนะ	54

สารบัญตาราง

ตารางที่ 1.1 ตารางแสดงแผนการดำเนินงาน	3
ตารางที่ 2.1 รายชื่อพรรณไม้ในชุดข้อมูล PNE	28
ตารางที่ 2.2 รายละเอียดการแบ่งชุดข้อมูล	30
ตารางที่ 2.3 ผลการทดสอบการเรียนรู้เชิงลึกของชุดข้อมูล PNE, 102 Flower และ Folio	30
ตารางที่ 2.4 แสดงคุณสมบัติที่สกัดได้ใบไม้ชนิด (a).....	35
ตารางที่ 2.5 แสดงคุณสมบัติที่สกัดได้จากใบไม้ชนิด (b).....	35
ตารางที่ 2.6 แสดงประสิทธิภาพระหว่าง GLCM ในมุมองศาส่าง ๆ ที่ใช้ทดสอบ และ PCA	36
ตารางที่ 4.1 Confusion matrix ของ EfficientDet-D4	50
ตารางที่ 4.2 Confusion matrix ของ Faster-RCNN Resnet-50.....	50
ตารางที่ 4.3 Confusion matrix ของ YOLOv5.....	51
ตารางที่ 4.4 ผลการทดสอบโมเดลทั้ง 3 กับชุดข้อมูล	51

สารบัญรูปภาพ

รูปที่ 2.1 การทำงานของจุดเชื่อมประสาทเทียม	8
รูปที่ 2.2 การทำงานของเครือข่ายประสาทเทียม	9
รูปที่ 2.3 การทำงานของโครงข่ายประสาทเทียมแบบconvโอลูชัน LeNet-5	10
รูปที่ 2.4 การทำงานของ Faster-RCNN	12
รูปที่ 2.5 การทำงานของ FPN และ BiFPN	13
รูปที่ 2.6 โครงสร้างของ EfficientDet	13
รูปที่ 2.7 แสดงการทำงานของ Residual learning	14
รูปที่ 2.8 โครงสร้างของโมเดล Resnet 50	15
รูปที่ 2.9 การทำงานของ YOLO	16
รูปที่ 2.10 ตัวอย่างฐานข้อมูล JSON	17
รูปที่ 2.11 สถาปัตยกรรมของ React Native	18
รูปที่ 2.12 ลักษณะการติดต่อสื่อสารของ API	19
รูปที่ 2.13 หมากังหราช (Astrophytum myriostigma)	20
รูปที่ 2.14 ปราสาทนางฟ้า	21
รูปที่ 2.15 ไก่ตีอก	21
รูปที่ 2.16 กระบอกทอง	22
รูปที่ 2.17 แมมอรุชาມ	22
รูปที่ 2.18 ดาวล้อมเดือน	23
รูปที่ 2.19 นูปต้า	23
รูปที่ 2.20 บานเยียนม	24
รูปที่ 2.21 มิยาโน	24
รูปที่ 2.22 แมมกลวยหอม	25
รูปที่ 2.23 หนามพิกุล	25
รูปที่ 2.24 ลูกแมว	26
รูปที่ 2.25 ขนนกขาว	26

รูปที่ 2.26 นิ้วทอง	27
รูปที่ 2.27 หุ่มิกกี้เม้าส์.....	27
รูปที่ 2.28 ตัวอย่างรูปภาพของชุดข้อมูล 102 Flower.....	29
รูปที่ 2.29 ตัวอย่างของชุดข้อมูล Folio.....	29
รูปที่ 2.30 ตัวอย่างชุดข้อมูล BJFU100.....	32
รูปที่ 2.31 เปรียบเทียบโครงสร้างบล็อกแบบปกติกับแบบบล็อกคอนเวด.....	32
รูปที่ 2.32 ตัวอย่างโครงสร้างของ Resnet 26 ชั้น	33
รูปที่ 2.33 แสดงความแม่นยำต่อจำนวนที่ใช้ทรัพน และความแม่นยำต่อโครงสร้างชั้นของ Resnet	33
รูปที่ 2.34 ตัวอย่างของชุดข้อมูล Flavia	34
รูปที่ 2.35 ผลการทดสอบ CNN โครงสร้าง Resnet กับชุดข้อมูล Flavia.....	34
รูปที่ 2.36 ตัวอย่างข้อมูลและชนิดของพันธุ์พืช	36
รูปที่ 3.1 ภาพรวมระบบของแอปพลิเคชันจำแนกสายพันธุ์ระบบองเพชร	37
รูปที่ 3.2 การออกแบบส่วนประสานสำหรับแอปพลิเคชันจำแนกสายพันธุ์ระบบองเพชรสำหรับผู้ใช้	38
รูปที่ 3.3 การทำงานของแอปพลิเคชันจำแนกสายพันธุ์ระบบองเพชร	39
รูปที่ 3.4 การทำงานของระบบส่วนหลัง	40
รูปที่ 3.5 การตรวจหากระบองเพชร	41
รูปที่ 3.6 ตัวอย่างภาพถ่ายต้นกระบองเพชรในสภาพแวดล้อมที่มีแสงแตกต่างกัน	42
รูปที่ 3.7 ตัวอย่างรูปที่ผ่านการยึด หาด หมุน กลับด้าน	43
รูปที่ 4.1 ส่วนแสดงผลการเลือกที่มาข้อมูล	45
รูปที่ 4.2 ส่วนแสดงผลการเลือกถ่ายรูปภาพ.....	46
รูปที่ 4.3 ส่วนแสดงผลการเลือกอัปโหลดรูปภาพ	47
รูปที่ 4.4 ส่วนแสดงผลการตรวจจับ.....	48
รูปที่ 4.5 การทดสอบนำรูปที่ส่งไปที่เครื่องบริการ และเครื่องบริการส่งข้อมูลกลับมา.....	49
รูปที่ 4.6 กราฟแสดงความแม่นยำเทียบกับเวลา	492

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบัน ผู้คนเริ่มหันมาให้สนใจในการปลูกต้นไม้กันอย่างแพร่หลาย หลังจากเกิดการแพร่ระบาดของไวรัสโคโรนาไวรัส COVID-19 ซึ่งเป็นต้นเหตุให้หลาย ๆ คนนั้น ต้องทำงานอยู่ในที่พักอาศัย หรือต้องหยุดงานไปเป็นช่วงเวลานาน ๆ นับหลายเดือน จึงทำให้คนโดยทั่วไปนั้นมีเวลาว่างในแต่ละวันเพิ่มมากขึ้น และเนื่องด้วยหลายปัจจัยหน้าที่ได้มีการแพร่กระจายของฝุ่นละอองขนาดเล็กไม่เกิน 2.5 ไมครอน (PM2.5) ซึ่งเป็นอันตรายต่อระบบการหายใจ จึงเป็นเหตุให้คร่าชีวิต คุณ เริ่มหันมาสนใจในการปลูกต้นไม้มากยิ่งขึ้น ตั้งแต่การปลูกต้นไม้เพื่อกรองฝุ่นภายในที่พักอาศัย จนไปถึงการปลูกเพื่อความสวยงาม

กระบองเพชร (Cactus) ก็เป็นอีกหนึ่งต้นไม้ที่นิยม เนื่องมาจากกระบวนการปลูกส่วนใหญ่จะปลูกและเลี้ยงดูได้ง่าย เพราะมีขนาดเล็ก และมีลักษณะเปลกตา เหมาะสำหรับการตกแต่งบ้านหรือที่พักอาศัย โดยที่ลักษณะโดยทั่วไปของกระบองเพชรที่คนส่วนใหญ่นิยมก็คือ ต้นไม้ที่มีหนามแหลมบริเวณลำต้นหรือใบแต่ในความเป็นจริงนั้น กระบองเพชรมีสายพันธุ์ที่ค่อนข้างหลากหลาย และในแต่ละสายพันธุ์ ก็ย่อมที่จะมีวิธีเลี้ยงดูที่แตกต่างกันออกไป และยังมีเรื่องของราคากองแต่ละสายพันธุ์ที่แตกต่างกัน ซึ่งทำให้เกิดเป็นปัญหาภัยคุกคาม คุณ ที่กำลังเริ่มต้นเลี้ยงกระบองเพชร คือไม่รู้ว่ากระบวนการปลูกที่ตนเองมี หรือกำลังที่จะซื้อนั้นเป็นสายพันธุ์อะไร จนถึงอย่างที่จะศึกษาข้อมูลกระบวนการปลูกที่พบที่น้ำตามสถานที่ทั่วไป เช่น ร้านกาแฟ ร้านอาหาร เป็นต้น เพื่อที่จะสามารถนำพาข้อมูลและวิธีในการเลี้ยงเพิ่มเติม

จากปัญหาในข้างต้นเป็นที่มาของการศึกษาการเรียนรู้เชิงลึก (Deep learning) เพื่อที่จะนำไปพัฒนาแอปพลิเคชันที่ใช้ในการจำแนกสายพันธุ์ของกระบองเพชร โดยนำรูปภาพของกระบองเพชรมาผ่านการประมวลผลแล้วแสดงชื่อสายพันธุ์ของกระบองเพชร เพื่อลดปัญหาที่เกิดขึ้นได้

1.2 วัตถุประสงค์ของโครงการ

เพื่อพัฒนาแอปพลิเคชันสำหรับจำแนกสายพันธุ์ของต้นกระบอกเพชรโดยใช้การเรียนรู้เชิงลึก (Deep learning) มาจำแนกสายพันธุ์โดยใช้รูปภาพ และเปรียบความแม่นยำของโมเดล

1.3 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำความรู้ที่ได้จากการพัฒนาระบบนี้ไปต่อยอดในการพัฒนาระบบจำแนกสิ่งต่าง ๆ ได้โดยมีความแม่นยำ หรือออกแบบระบบที่สามารถอ่านข้อมูลจากภาพได้อัตโนมัติ เพื่อให้สอดคล้องกับเทคโนโลยีปัจจุบันที่เติบโตขึ้นอย่างรวดเร็ว

1.4 ขอบเขตการทำโครงการ

1.4.1 ใน การใช้งานแอปพลิเคชัน จะต้องเข้มต่อ กับระบบอินเทอร์เน็ตอยู่ตลอดการใช้งาน

1.4.2 ระบบจะสามารถแยกสายพันธุ์ของต้นกระบอกเพชรได้ทั้งหมดจำนวน 15 สายพันธุ์ โดยที่ต้นกระบอกเพชรนั้นไม่มีดอก

1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษาการใช้งานเอกสารต่าง ๆ

1.5.2 ศึกษาและออกแบบระบบโครงสร้างระบบโดยรวม

1.5.3 ศึกษาการสร้างแอปพลิเคชันด้วย React Native

1.5.4 ศึกษาการสร้างระบบ Back-end ด้วย Python Flask

1.5.5 ศึกษาการประมวลผลรูปภาพด้วย Open CV

1.5.6 ศึกษาการเรียนรู้เชิงลึก (Deep learning)

1.5.7 ออกแบบโครงสร้างระบบ

- 1.5.8 ออกแบบส่วนต่อประสานกับผู้ใช้แอปพลิเคชันกับระบบ Back-end
 - 1.5.9 ออกแบบโครงสร้างข้อมูล
 - 1.5.10 ออกแบบระบบสำหรับประมวลผลภาพ
 - 1.5.11 พัฒนาแอปพลิเคชันและระบบทั้งหมด
 - 1.5.12 รวมระบบทั้งหมดเข้าด้วยกัน
 - 1.5.13 ทดสอบการทำงานของแอปพลิเคชันและแก้ไขจุดบกพร่อง
 - 1.5.14 สรุปผลและจัดทำรายงาน

1.6 แผนการดำเนินงาน

ตารางที่ 1.1 ตารางแสดงแผนการดำเนินงาน

1.7 รายละเอียดงบประมาณตลอดโครงการ

1.7.1 ค่าระบบของเพชร 15 สายพันธุ์ ๆ ละ 1 ต้น	1,500 บาท
1.7.2 ค่าพิมพ์เอกสาร	1,000 บาท
1.7.3 ค่าใช้จ่ายอื่น ๆ	500 บาท
รวม	3,000 บาท (สามพันบาทถ้วน)

หมายเหตุ : ถ้าเฉลี่ยทุกรายการ

บทที่ 2

หลักการและทฤษฎีเบื้องต้น

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ปัญญาประดิษฐ์ (Artificial Intelligence: AI)

ปัญญาประดิษฐ์ (Artificial Intelligence) เป็นการนำความรู้ทางด้านวิทยาการคอมพิวเตอร์ และวิศวกรรมคอมพิวเตอร์มาพัฒนาระบบคอมพิวเตอร์ เช่น หุ่นยนต์ รถยนต์ไร้คนขับ เป็นต้น เพื่อให้สามารถทำงานให้ใกล้เคียงกับมนุษย์มากที่สุด ไม่ว่าจะเป็น ความคิด การตัดสินใจ หรือแม้แต่พฤติกรรมของมนุษย์ ส่งผลให้การพัฒนาระบบคอมพิวเตอร์นั้นมีศักยภาพที่มากขึ้นและมีความซับซ้อนมากขึ้นไป ด้วย ทำให้ปัญญาประดิษฐ์นั้นสามารถนำไปใช้ในงานที่ต้องการการตัดสินใจ หรือใช้ความคิดได้กว้างขวาง รวมถึงมีข้อผิดพลาดที่น้อยกว่าอีกด้วย ซึ่งสามารถแบ่งการทำงานของปัญญาประดิษฐ์ได้ 4 แบบ ดังนี้

2.1.1.1 การคิดแบบมนุษย์

คือปัญญาประดิษฐ์ที่ถูกพัฒนาขึ้นมาเพื่อให้สามารถคิดเองได้ โดยอาศัยหลักการพื้นฐานของการคิดของมนุษย์ จึงทำให้การทำงานไม่ว่าจะเป็น การแก้ปัญหาต่าง ๆ หรือการตัดสินใจ นั้น คล้ายคลึงกับมนุษย์

2.1.1.2 การกระทำแบบมนุษย์

คือการสร้างระบบที่อาศัยการทำงานของมนุษย์เข้ามาช่วยพัฒนาทำให้การทำงานของปัญญาประดิษฐ์นั้น เมื่อนอกมนุษย์ เป็นอย่างมาก เช่น การแปลภาษาหรือเขียนบทความ หุ่นยนต์ ทำความสะอาด เป็นต้น

2.1.1.3 การคิดอย่างมีเหตุผล

เป็นการนำความรู้และการตัดสินใจของมนุษย์มาพัฒนาปัญญาประดิษฐ์ โดยใช้เหตุผลมากำหนดการกระทำของปัญญาประดิษฐ์ เช่น หลักตรรกศาสตร์ สิ่ติเป็นต้น เพื่อให้ปัญญาประดิษฐ์สามารถคำนวณความเป็นไปได้ต่าง ๆ และเลือกตัดสินใจได้ถูกต้อง

2.1.1.4 การทำงานอย่างมีเหตุผล

จากการคิดอย่างมีเหตุผลนำไปสู่การทำงานต่าง ๆ ของปัญญาประดิษฐ์นั้นมีเหตุผลมากขึ้น ซึ่งการคำนวณต่าง ๆ ที่มีประสิทธิภาพนั้นนำไปสู่การพัฒนา ตัวแทนปัญญา (Intelligent Agent) อีกด้วย

2.1.2 การเรียนรู้ของเครื่องจักร (Machine Learning)

การเรียนรู้ของเครื่องจักร (Machine learning) นั้นเป็นการนำปัญญาประดิษฐ์มาประยุกต์ทำให้คอมพิวเตอร์นั้นเกิดการเรียนรู้ด้วยตัวเองอย่างอัตโนมัติโดยอาศัยสถานการณ์ หรือความรู้ต่าง ๆ ที่ได้จากการเรียนรู้ ทำให้การทำงานนั้นมีประสิทธิภาพมากยิ่งขึ้น รวมไปถึงการลดความผิดพลาดของการทำงานอีกด้วย จึงทำให้คอมพิวเตอร์นั้นสามารถวางแผนและตัดสินใจแก้ปัญหาต่าง ๆ ได้ใกล้เคียงกับมนุษย์ ซึ่งการเรียนรู้ของเครื่องสามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ ๆ ดังนี้

2.1.2.1 การเรียนรู้แบบมีผู้สอน (Supervised learning)

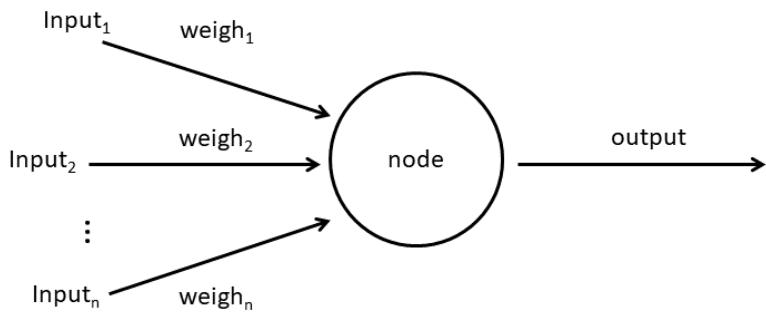
เป็นการพัฒนาโดยอาศัยข้อมูลจำนวนมาก และการรับข้อมูลผลลัพธ์จากมนุษย์ที่เปรียบเสมือนคำสอนไปปรับปรุง เพื่อที่จะให้ระบบนั้นเกิดการเรียนรู้ของความสัมพันธ์ของข้อมูล จึงทำให้สามารถแยกการทำงานได้ออกเป็นอีก 2 อย่างคือ การจำแนกประเภท (Classification) และการทดถอย (Regression)

2.1.2.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning)

เป็นการพัฒนาโดยอาศัยข้อมูลจำนวนมากที่ประกอบไปด้วยคุณลักษณะสำหรับการเรียนรู้ที่ถูกต้องเพียงอย่างเดียว

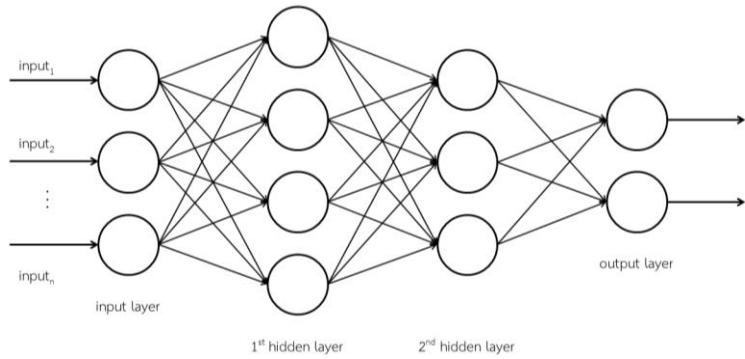
2.1.3 โครงข่ายประสาทเทียม (Artificial Neuron Network: ANN)

โครงข่ายประสาทเทียม (Artificial Neuron Network) นั้นเป็นการจำลองการทำงานของระบบประสาทในสมองของมนุษย์ นั่นคือหน่วยเซลล์ หรือเรียกว่า นิวรอน (Neuron) ที่เชื่อมกันด้วยจุดประสาทประสาท (Synapses) ซึ่งเป็นระบบประสาทที่ใช้ในการประมวลผลต่าง ๆ ของมนุษย์ แต่ในทางคอมพิวเตอร์นั้น โครงข่ายประสาทเทียมนั้นถูกสร้างขึ้นโดยอาศัยโมเดลทางคณิตศาสตร์ และโมเดลทางคอมพิวเตอร์ ทำให้มีความสามารถในการเรียนรู้ จดจำ และอื่น ๆ โดยการสร้างจุดเชื่อมต่าง ๆ (Node) ขึ้นมาซึ่งทำงานที่เป็นส่งต่อข้อมูลไปยังจุดอื่น ๆ ที่เชื่อมอยู่เมื่อมีการรับข้อมูลเข้ามา โดยแบ่งข้อมูลที่รับเข้ามาด้วยค่าน้ำหนักของเส้นที่เชื่อมกับจุดเชื่อมต่าง ๆ ดังรูปที่ 2.1



รูปที่ 2.1 การทำงานของจุดเชื่อมประสาทเทียม

ดังนั้นการทำงานของโครงข่ายประสาทเทียมจึงเป็นการทำงานที่เชื่อมต่อกันดังรูปที่ 2.2 นั่นคือรับข้อมูลเข้ามายังชั้นรับเข้า (input layer) และก็จะถูกส่งไปยังจุดเชื่อมประสาทชั้นซ่อน (hidden layer) ซึ่งการเชื่อมต่อนี้อาจจะเป็นการเชื่อมต่อแบบรูปแบบเต็ม (full connect) หรืออาจจะเป็นการเชื่อมต่อแค่บางส่วนก็ได้ (partial connect) ซึ่งก็อาจมีชั้นซ่อน (hidden layer) มากกว่า 1 ได้ ซึ่งทุกชั้นจะเชื่อมต่อกันผ่านจุดเชื่อมประสาท (node) และส่งข้อมูลออกไปที่ชั้นส่งออก (output layer)



รูปที่ 2.2 การทำงานของเครือข่ายประสาทเทียม

2.1.4 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึก (Deep learning) เป็นส่วนหนึ่งของการเรียนรู้ของเครื่อง (Machine learning) ซึ่งประกอบไปด้วย 2 ประเด็นใหญ่ ๆ คือโครงข่ายประสาทเทียม (Artificial Neuron Network) ที่เชื่อมต่อกันโดยไม่เป็นชิ้นและมีหลายชั้น และการเรียนรู้ของเครื่องแบบมีผู้สอนหรือไม่มีผู้สอนหรือกิ่งมีผู้สอนก็ได้

การทำงานของการเรียนรู้เชิงลึกนั้นมีพื้นฐานจากการใช้ประสาทเทียม (Neural Network) เข้ามาเชื่อมต่อกันเป็นชั้น ๆ เกิดเป็นโครงข่ายประสาทเทียมย่อยที่เชื่อมต่อกันไปอีกชั้นนึง เพื่อให้เกิดความเข้าใจความหมายของข้อมูล เช่น รูปภาพหนึ่งสามารถแยกออกเป็นจุดพิกเซลช่องๆ ได้ว่าเป็นเส้นเวกเตอร์หนึ่งของประสาทเทียมต่อหนึ่งจุดพิกเซล ซึ่งการที่จะเป็นหนึ่งภาพได้ก็จะเชื่อมต่อเส้นเวกเตอร์ด้วยจุดเชื่อมประสาท (Node) หลาย ๆ ชั้นเข้าด้วยกันจึงเกิดเป็นหนึ่งภาพที่เกิดการเรียนรู้ของเครื่องจักรซึ่งคำว่าเชิงลึกนั้นมีความหมายมากจากการที่มีการเชื่อมต่อกันของโครงข่ายประสาทเทียมหลาย ๆ ชั้นเข้าด้วยกัน ทำให้มีประสิทธิภาพมากขึ้นด้วย

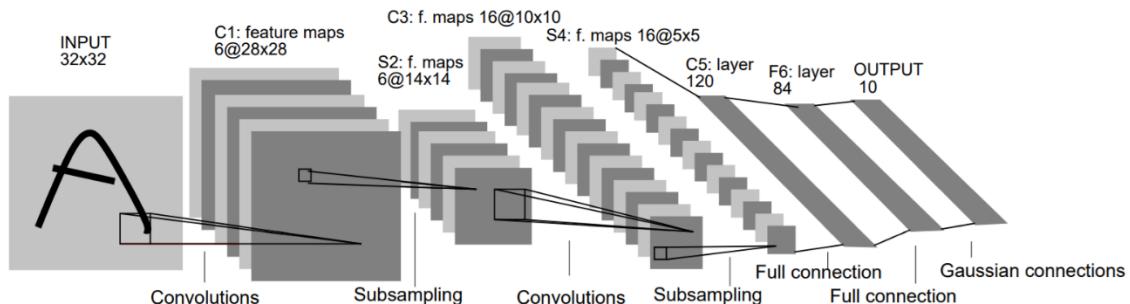
2.1.5 การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุ (Object Detection) คือการนำเอาความรู้ทางด้านการประมวลผลภาพ (Image Processing) และปัญญาประดิษฐ์ (Artificial Intelligence) เข้าด้วยกัน เพื่อทำให้คอมพิวเตอร์

นั้นสามารถรับรูปภาพเข้าไปแล้วนำไปประมวลผลจนสามารถแยกแยะสิ่งของหรือสิ่งมีชีวิตในรูปภาพหรือวิดีโอด้วย ซึ่งสามารถมาทำได้หลายวิธี แต่วิธีที่นิยมกันได้แก่การตีกรอบให้กับวัตถุ (Bounding Box) หรือการ tessellate กับวัตถุ (Segmentation) ซึ่งสามารถนำไปประยุกต์กับการทำงานได้หลากหลายไม่ว่าจะเป็นการตรวจจับใบหน้า (Face recognition) การอ่านตัวอักษรในภาพ (OCR) เป็นต้น

2.1.6 โครงข่ายประสาทเทียมแบบคอนволูชัน (Convolution Neural Network: CNN)

เป็นลักษณะการทำงานของการเรียนรู้เชิงลึก (Deep learning) โดยโครงสร้างของโครงข่ายประสาทเทียมแบบคอนволูชัน มีทั้งหมด 5 ชั้นเรียกว่า LeNet-5 ซึ่งถูกเขียนต่อ กันอย่างสมบูรณ์ (Fully-Connection) โดยจะมีชั้นที่ทำงานแบบคอนволูชัน (Convolution layer) ซึ่งจะถูกคำนวณหาคุณลักษณะพิเศษ (Feature map) ซึ่งจะทำงานคู่กับการพูลิ่ง (Pooling) เพื่อลดขนาดของเมทริกในการคำนวณลงดังรูปที่ 2.3 จึงทำให้โครงข่ายประสาทเทียมนั้นถูกนำไปใช้ในการเรียนรู้เชิงลึกเพื่อศึกษาข้อมูลประเภทรูปภาพ



รูปที่ 2.3 การทำงานของโครงข่ายประสาทเทียมแบบคอนволูชัน LeNet-5

ที่มา (Yann LeCun 1998)

2.1.7 Faster-RCNN

Faster-RCNN เป็นแนวคิดที่ถูกพัฒนาต่อจาก RCNN และ Fast-RCNN ซึ่งถูกออกแบบมาเพื่อแก้ปัญหาของการทำ Object Detection เช่นความช้าในการทำงานของ RCNN และปรับปรุงความเร็วในการประมวลผลให้เร็วกว่า Fast-RCNN ซึ่งสามารถแยกการทำงานได้ดังนี้

2.1.7.1 Feature maps

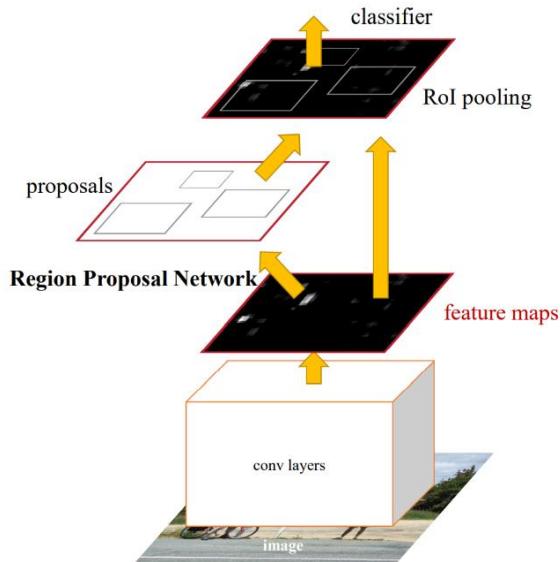
จะรับข้อมูลรูปที่ประมวลผลแล้วจาก convolution layer มาทำการสกัดข้อมูลต่างๆ ของภาพที่รับเข้ามาแล้วส่งออกไปให้กับ Region Proposal Network และ Region of Interest pooling

2.1.7.2 Region Proposal Network (RPN)

จะประมวลผลข้อมูลที่ได้มาจากการรับรูปเข้ามาประมวลผลแล้วทำการตีกรอบว่าส่วนไหนของรูปภาพน่าจะมีวัตถุอยู่ ซึ่งกรอบสี่เหลี่ยมที่ Region Proposal Network ได้ประมวลผลนั้นจะเรียกว่า Anchor box โดยการสร้าง Anchor box ให้ครอบคลุมพื้นที่ของ Feature maps ในตำแหน่งต่างๆ ที่รับเข้ามาประมวลผล แล้วจึงส่งข้อมูลออกไปเป็นเมทริกซ์

2.1.7.3 การจำแนกประเภทของวัตถุ

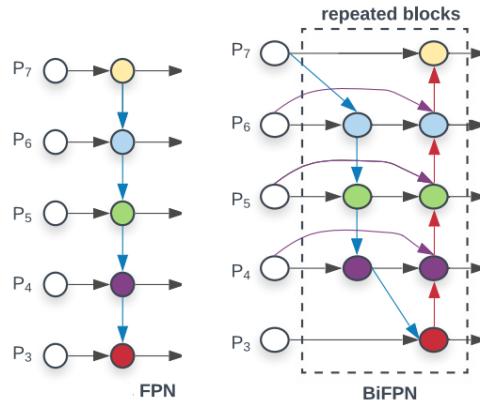
ซึ่งจะนำเอาข้อมูลที่ได้จาก feature maps และ RPN มาประมวลผลด้วย Region of Interest (RoI) pooling เพื่อให้ได้ Feature Vector ที่มีขนาดคงที่เพื่อให้สามารถจำแนกประเภทของวัตถุได้แม่นยำมากยิ่งขึ้น



รูปที่ 2.4 การทำงานของ Faster-RCNN
ที่มา (Ren, et al. 2017)

2.1.8 EfficientDet

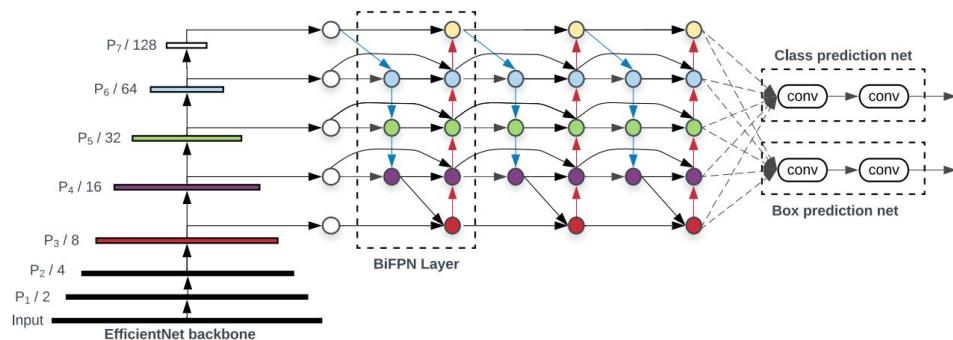
EfficientDet นั้นถูกนำเสนอในงานวิจัย EfficientDet: Scalable and Efficient Object Detection โดย EfficientDet นั้นมีพื้นฐานจาก EfficientNet ซึ่งเป็นโครงข่ายแบบคอนโวลูชัน ในงานวิจัยนั้นต้องการที่จะได้ประสิทธิภาพและความแม่นยำของโมเดล โดยการเพิ่มมิติของโครงข่ายทั้งในความกว้างและความลึก รวมไปถึงขนาดของภาพ ซึ่งวิธีการที่ใช้ใน EfficientNet นั้นเรียกว่า Compound Scaling นอกจากนี้ EfficientDet ยังใช้ BiFPN (Bidirectional Feature Fusion Network) ซึ่งเป็นคุณลักษณะของโครงข่ายแบบพิวชัน (Feature Fusion Network) ที่ทำหน้าที่รวมคุณลักษณะที่สกัดได้จากความละเอียดที่แตกต่างกัน โดยพัฒนาต่อมาจากโครงข่ายคุณลักษณะแบบพิรัมิด (Feature Pyramid Network: FPN) โดยการเพิ่มเส้นทางจากจุดเชื่อมด้านบนลงไปด้านล่าง และเส้นเชื่อมจากจุดเริ่มต้นไปยังจุดผลลัพธ์ คล้ายกับวิธี Skip Connection ของ Resnet จากนั้นจึงเชื่อมจากจุดด้านล่างขึ้นมาข้างบน ดังรูปที่ 2.5



รูปที่ 2.5 การทำงานของ FPN และ BiFPN

ที่มา (Mingxing, et al. 2020)

โครงสร้างของ EfficientDet นั้นจึงประกอบด้วย EfficientNet ที่ทำหน้าที่เป็นแกนหลักในการสกัดรูปภาพให้ได้ขนาดต่าง ๆ แล้วจึงเชื่อมต่อกับ BiFPN เพื่อสกัดเอาคุณลักษณะของรูปภาพ แล้วจึงส่งข้อมูลไปทำนายวัตถุและตำแหน่งของวัตถุดังรูปที่ 2.6

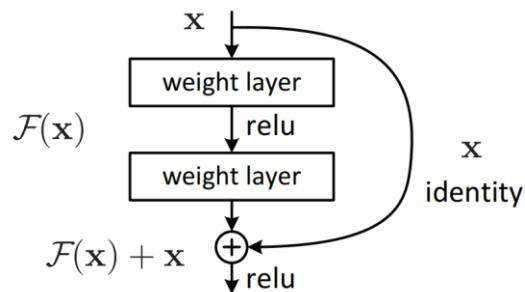


รูปที่ 2.6 โครงสร้างของ EfficientDet

ที่มา (Mingxing, et al. 2020)

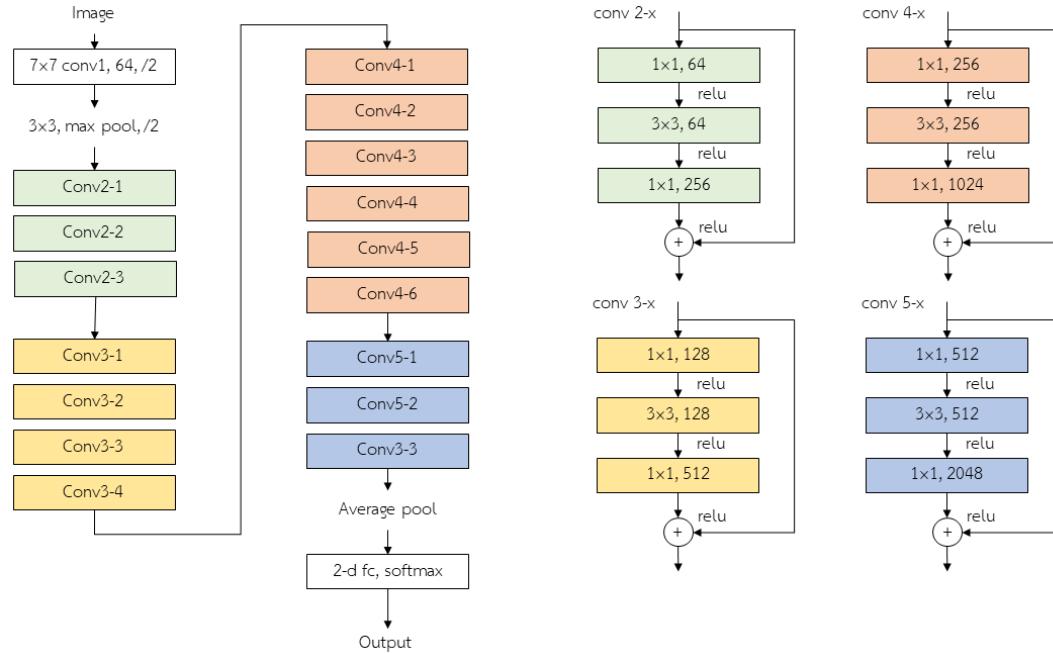
2.1.9 Resnet (Deep Residual Network)

Resnet เป็นแนวคิดที่ถูกเสนอในงานวิจัย Deep residual learning for image recognition เพื่อแก้ไขปัญหาการล่าระดับสีแล้วมีสีหายไป (Vanishing Gradient) จากการที่ตัวโครงข่ายนั้นมีจำนวนชั้นที่ลึกมากถึง 152 ชั้น โดยการเพิ่มทางลัดเข้าไปในโมเดล (Skip connection) ซึ่งจะข้ามโครงข่ายที่ทำงานเหมือนกัน ทำให้ผลลัพธ์ของโมเดลเป็น $F(x) + x$ ซึ่งโมเดลนี้ถูกเรียกว่า Residual learning ดังรูปที่ 2.7



รูปที่ 2.7 แสดงการทำงานของ Residual learning
ที่มา (Kaiming, et al. 2015)

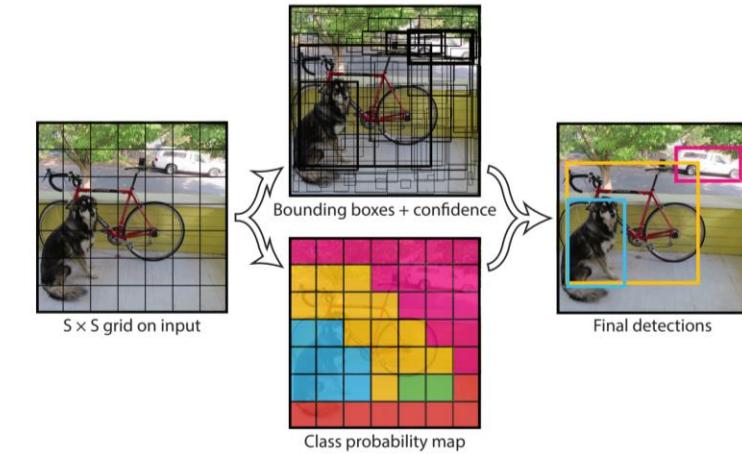
จากการทำ Residual learning นี้จะทำให้ Resnet นั้นทำงานได้รวดเร็วรวมถึงไม่เปลี่ยนประสิทธิภาพของการทำงานลง โดยโครงสร้างของ Resnet นั้นจะแบ่งเป็น 4 กลุ่มโครงข่ายขนาดใหญ่ และรวมของจำนวนโครงข่ายทั้งหมดจะถูกเรียกเป็นชื่อของโมเดล เช่น Resnet 50 จะมีขนาดของกลุ่มโครงข่ายเป็น [3, 4, 6, 3] ซึ่งจะได้เป็น $1 + (3 \times 3) + (4 \times 3) + (6 \times 3) + (3 \times 3) + 1 = 50$ ดังรูปที่ 2.8 จะแสดงถึงโครงสร้างโครงข่ายของ Resnet 50



รูปที่ 2.8 โครงสร้างของโมเดล Resnet 50

2.1.10 YOLO (You Only Look Once)

YOLO เป็นวิธีการที่ใช้ตรวจจับวัตถุได้ในชั้นเดียว (Single-Stage Detector: SSD) โดยแนวคิดของ YOLO คือการจำแนกวัตถุว่าเป็นวัตถุชนิดใด (Classification) และหาตำแหน่งของวัตถุโดยการตีกรอบขอบวัตถุ (Bounding Box) ไปพร้อม ๆ กัน ดังรูปที่ 2.9 จึงทำให้ YOLO นั้นมีความรวดเร็วสูงแต่ก็แลกมาด้วยความแม่นยำที่ต่ำลงไปด้วย ซึ่ง YOLO นั้นถูกพัฒนาต่อมาเรื่อย ๆ จนมาถึง YOLOv5



รูปที่ 2.9 การทำงานของ YOLO

ที่มา (Joseph, et al 2016)

2.1.11 ฐานข้อมูล (Database)

ฐานข้อมูล คือกลุ่มของข้อมูลที่นำมาเก็บไว้ด้วยกันในลักษณะต่างๆ ที่มีความสัมพันธ์กัน โดยผู้ใช้สามารถเรียกใช้ข้อมูล แก้ไข เพิ่ม หรือลบข้อมูลได้ตามต้องการ โดยลักษณะการเข้าถึงข้อมูลนั้นขึ้นอยู่ กับระบบฐานข้อมูลที่ผู้ใช้เลือกใช้ตามความถนัดหรือความสะดวก

2.1.11.1 JSON (JavaScript Object Notation)

คือรูปแบบของฐานข้อมูลชนิดหนึ่ง โดยสามารถเข้าถึงได้ในทุกภาษาของการเขียนโปรแกรม โดยเป็นการเก็บข้อมูลในภาษา JavaScript จัดเก็บแบบ JSON Object โดยฐานข้อมูลแบบ JSON นั้น สามารถเก็บลักษณะของข้อมูลได้ดังนี้

- Number: ตัวเลข (ตัวเลขจำนวนจริงทั้งหมด)
- String: Unicode ตัวอักษรที่เข้ารหัสแบบ Unicode โดยจะใช้เครื่องหมายคำพูด (Double-Quote: "") เป็นตัวบ่งบอก เช่น "Hello"
- Boolean: True หรือ False
- Array: ชุดของข้อมูล เป็นข้อมูลชนิดใดก็ได้ โดยใช้งานลีบสีเหลี่ยม

(Square-Bracket: []) เป็นตัวบ่งบอก และใช้ลูกน้ำ (,) เป็นตัวแบ่งข้อมูล เช่น [Val1, Val2, Val3]

- Object: ชุดข้อมูลที่เป็นคู่ลักษณะแบบ Key-Value ใช้สัญลักษณ์ ปีกกา ({}) เป็นตัวบ่งบอก ใช้ลูกน้ำ (,) เป็นตัวแบ่งชุดข้อมูล และใช้ทวิภาค (:) เช่น {Key1:Value1, Key2:Value2}
- Null: ค่าว่าง ใช้คำว่า null แทน

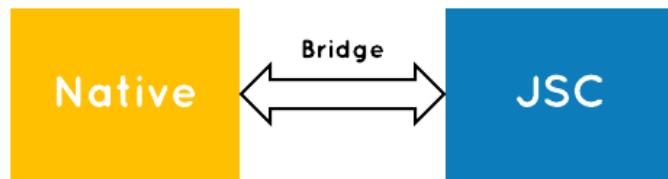
```
{
  "type": "office",
  "number": "646 555-4567"
},
{
  "type": "mobile",
  "number": "123 456-7890"
},
"children": [],
"spouse": null
}

{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "work",
      "number": "917 555-1234"
    }
  ]
}
```

รูปที่ 2.10 ตัวอย่างฐานข้อมูล JSON
พี่มา (Mange 2019)

2.1.12 เฟรมเวิร์คสำหรับพัฒนาแอปพลิเคชัน (React Native)

React Native เป็น JavaScript Framework ตัวหนึ่ง โดยมีผู้พัฒนาคือ Facebook ซึ่งจะช่วยให้เราสามารถทำการเขียน Mobile Application แบบ Cross platform ได้ ทำให้เราเขียนโค้ดเพียงครั้งเดียวที่สามารถสร้าง Application ทั้งของ Android OS และ iOS



รูปที่ 2.11 สถาปัตยกรรมของ React Native

ในส่วนสถาปัตยกรรมของ React Native จะมีการทำงานระหว่าง Native ทำงานร่วมกับ JavaScriptCore โดยที่ JavaScriptCore ก็มีอยู่ทั้งทางฝั่ง Android OS และ iOS ทาง React Native ก็จะทำการ bundle ตัว JavaScriptCore เข้าไปใน package โดยทั้งฝั่งของ Native และ JavaScriptCore จะสื่อสารผ่าน Bridge ซึ่งทำหน้าที่เป็นตัวเชื่อม ซึ่งใน React Native จะมีการสร้าง JSON รูปแบบ จำเพาะที่ทำงานสื่อสารกันในสองส่วนนี้ โดยลักษณะจะเป็น แบบ Serialized Messages

2.1.13.1 ชุดพัฒนาซอฟต์แวร์ (Expo)

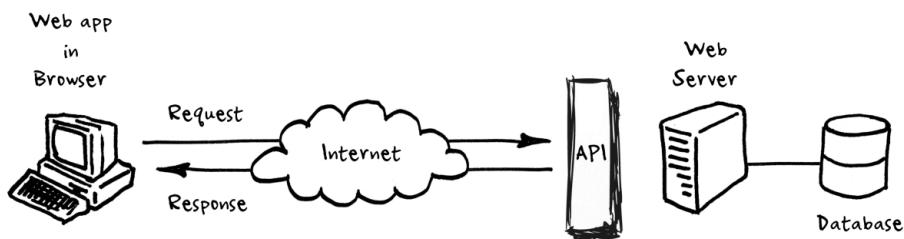
ชุดพัฒนาซอฟต์แวร์ (Software Development Kit, SDK) หมายถึง ชุดเครื่องมือที่สามารถใช้ในการพัฒนาซอฟต์แวร์ หรือแอปพลิเคชันที่กำหนดเป้าหมายไปยังแพลตฟอร์มเฉพาะ โดยที่ SDK นั้นประกอบด้วยเครื่องมือไลบรารีเอกสารประกอบและโค้ดตัวอย่างที่จะช่วยโปรแกรมเมอร์ในการพัฒนาแอปพลิเคชัน

Expo เป็น SDK ชุดหนึ่งที่เข้ามาช่วยให้การพัฒนาแอปพลิเคชัน ด้วย react-native ซึ่งเป็นมิตรมากขึ้นหรือง่ายขึ้นด้วยการจัดการสิ่งต่าง ๆ ที่จำเป็นในการทำงานให้กับ React-native เช่น Camera, Map, ImagePicker, Font, Permission, Video, Push Notification เป็นต้น โดยไม่ต้องเข้า

น้ำจัดการเขียน Native Module อีก และมีการพ่วงนามด้วยเครื่องมือที่ช่วยในการดู log การทำงาน ร่วมไปถึงการทดสอบแอปพลิเคชันผ่านอุปกรณ์จริงโดยไม่ต้องใช้สายเชื่อมต่อผ่านลิงก์ หรือคิวอาร์โค้ด

2.1.13 Application Programming Interface (API)

API คือชุดคำสั่งที่เปิดใช้งานการส่งข้อมูลระหว่างซอฟต์แวร์หนึ่งไปยังซอฟต์แวร์อื่น ๆ โดยอาจจะมีเงื่อนไขการแลกเปลี่ยนข้อมูล



รูปที่ 2.12 ลักษณะการติดต่อสื่อสารของ API
ที่มา (What is API: Definition, Types, Specifications, Documentation, 2019)

2.1.14 REST API

REST มาจากคำว่า Representational state transfer โดยเป็นสถาปัตยกรรมหนึ่งที่อยู่ภายใต้มาตรฐานการเว็บ (Web Protocol) เพื่อใช้ในการสร้างบริการเว็บ (Web Service)

โดยปัจจัยหลักที่ทำให้เกิดสถาปัตยกรรมนี้นั้นคือ

- ประสิทธิภาพในการตีตอบ
- มีความสามารถในการปรับขยายได้ทำให้สามารถรองรับส่วนประกอบจำนวนมาก
- ความเรียบง่ายของการเข้าถึงที่เหมือนกัน

- ความสามารถในการปรับเปลี่ยนส่วนประกอบเพื่อตอบสนองความต้องการที่เปลี่ยนแปลง (แม้ในขณะที่แอปพลิเคชันทำงานอยู่)
- การมองเห็นการสื่อสารระหว่างส่วนประกอบต่าง ๆ ผ่านผู้ให้บริการ
- ง่ายต่อการโยกย้ายในส่วนของโปรแกรมและข้อมูล
- ความน่าเชื่อถือในการต้านทานความล้มเหลวในระบบ

2.1.14.1 RESTful Web Services

เป็นการอนุญาตให้ระบบต่าง ๆ ขอรับเข้าถึงข้อมูลบนเว็บไซต์ โดยการโต้ตอบจะอยู่ภายใต้ REST โดยการข้อเข้าถึง จะส่งคำขอไปยัง URL ของบริการนั้น ๆ และทำการรับข้อมูลการตอบกลับในรูปแบบการตอบกลับต่าง ๆ ไปใช้ต่อไป

2.1.15 สายพันธุ์ระบบองเพชร

2.1.15.1 หมวกสังฆราช (*Astrophytum myriostigma*)

ลักษณะที่โดดเด่นคือลำต้นเป็นทรง 5 เหลี่ยม สีเขียวอ่อน มีสันพู 5-6 พู มีหนาม เป็นกระჯุกขนาดเล็ก ๆ สีขาวตามตุ่มเนินหนาม ดังรูปที่ 2.13



รูปที่ 2.13 หมวกสังฆราช (*Astrophytum myriostigma*)

2.1.15.2 ปราสาทนางฟ้า (*Cereus peruvianus*)

ลักษณะทางพฤกษศาสตร์คือ รูปร่างเป็นทรงกระบอก 5 เหลี่ยม สีเขียวเข้ม แต่ละแนวเหลี่ยมจะมีหนามสีขาวเล็ก ๆ และแตกกอขึ้นเป็นกลุ่ม ดังรูปที่ 2.14



รูปที่ 2.14 ปราสาทนางฟ้า

2.1.15.3 ไก่ต็อก (*Haworthia coarctata*)

ลักษณะทางพฤกษศาสตร์ ใบเรียวยั่วแหลม ออกเวียนรอบต้น กว้าง 7 – 8 มิลลิเมตร ยาว 2.5 – 3 เซนติเมตร แผ่นใบหนาห่อรูมเข้าหากันอยู่ในกล่องใบ สีเขียวเข้ม มีเต้มสีขาวพาดขวางตลอดใบ ดังรูปที่ 2.15



รูปที่ 2.15 ไก่ต็อก

2.1.15.4 กระบองทอง (*Echinocactus leninghausii*)

ลักษณะทางพฤกษาศาสตร์คือ ลำต้นเป็นทรงกระบอกสูง ลำตันมีสีเขียวเข้ม มีหนามรอบต้นสีเหลืองทอง ดังรูปที่ 2.16



รูปที่ 2.16 กระบองทอง

2.1.15.5 แมมอธุรานุ (Mammillaria bucareliensis)

ลักษณะทางพฤกษาศาสตร์คือ มีผิวสีเขียวเข้ม มีปุ่ยขันสีขาวบริเวณปลายเนินหนามและระหว่างพุ บางต้นอาจมีหนามเหลืออยู่เล็กน้อย เมื่ออายุมากจะแตกกอ ดังรูปที่ 2.17



รูปที่ 2.17 แมมอธุรานุ

2.1.15.6 ดาวล้อมเดือน (*Echinopsis calochalora*)

ลักษณะที่โดดเด่นคือลำต้นเป็นสีเขียว มีหนามแหลม ลำตันกลม มักแตกหน่อที่โคนต้นเป็นจำนวนมาก เมื่อมีดอก ดอกจะมีลักษณะเป็นสีขาว ระบะของเพชรชนิดนี้แตกหน่อได้ไว ดังรูปที่ 2.18



รูปที่ 2.18 ดาวล้อมเดือน

2.1.15.7 นูปต้า (*Echinopsis subdenudata*)

ลักษณะที่โดดเด่นคือลำต้นมีสีเขียวเข้ม รอบ ๆ มีหนามสีขาว ที่โคนห่านมีลักษณะเป็นปุยขนสีขาว เมื่อออกรากจะมีดอกสีขาวขนาดใหญ่และก้านดอกยาว ดังรูปที่ 2.19



รูปที่ 2.19 นูปต้า

2.1.15.8 บานเนียนัม (*Gymnocalycium baldianum*)

ลักษณะที่โดดเด่นคือลำต้นกลมแบบมีสีเข้มเข้มหรือน้ำตาลดำ มีหนามขนาดใหญ่
อยู่เป็นกลุ่มๆ รอบต้น และเมื่อออกดอกดอกมีความงามดีมากสีแดงสด ดังรูปที่ 2.20



รูปที่ 2.20 บานเนียนัม

2.1.15.9 มิยาโน (*Gymnocalycium mihanovichii*)

ลักษณะที่โดดเด่นคือลำต้นเป็นแฉก มีสีเขียวเข้ม บางทีอาจจะมีลักษณะเป็นด่าง
บ้าง มีหนามแหลมตั้งขึ้น มีร่องพลุที่ลึก ลักษณะชั้นรอยหยักคล้ายเชือกรอง เมื่อมีดอก ดอกจะมีสีชมพูอ่อน
ดังรูปที่ 2.21



รูปที่ 2.21 มิยาโน

2.1.15.10 แมมกล้ายหอม (*Mammillaria longimamma*)

ลักษณะที่โดดเด่นคือลำต้นมีคล้ายผลกล้ายหารายๆ ผลอยู่ร่วมกัน เป็นสีเขียว ที่ปลายแต่ละผลจะมีหนาม โดยทั่วไปเมื่อออกรากมีดออกสีเหลืองหรือสีขาว ดังรูปที่ 2.22



รูปที่ 2.22 แมมกล้ายหอม

2.1.15.11 หนามพิกุล (*Mammillaria scrippsiana*)

ลักษณะที่โดดเด่นคือลำต้นค่อนข้างกลมสีเขียวเข้ม มีหนารอบต้น และมีปุยขนสีขาวแทรกอยู่ระหว่างแต่ละกลุ่มหนาม เมื่อออกรากมีดออกสีชมพู ดังรูปที่ 2.23



รูปที่ 2.23 หนามพิกุล

2.1.15.12 ลูกแมว (*Mammillaria bocasana*)

ถักรูปที่ 2.24 ลักษณะที่โดดเด่นคือลำต้นมีขนสีขาวคล้ายขนแมวปลอกคลุมทั่วลำต้น และเมื่อออกดอกก้มมีดอกสีชมพูเป็นส่วนใหญ่ ดังรูปที่ 2.24



รูปที่ 2.24 ลูกแมว

2.1.15.13 ขนนกขาว (*Mammillaria plumosa*)

ถักรูปที่ 2.25 ลักษณะที่โดดเด่นคือลำต้นข้าวกลม บริเวณที่ควรเป็นหนามจะเป็นแพขนสีขาวคล้ายกับขนนกปกคลุมทั่วทั้งต้น ดังรูปที่ 2.25



รูปที่ 2.25 ขนนกขาว

2.1.15.14 นิ้วทอง (*Mammillaria elongata*)

ลักษณะที่โดดเด่นคือลำต้นยาวคล้ายนิ้ว มีสีเหลืองทอง จึงได้ชื่อว่า “นิ้วทอง” ให้ดอกสีขาวครีม มักออกดอกในช่วงฤดูหนาว ดังรูปที่ 2.26



รูปที่ 2.26 นิ้วทอง

2.1.15.15 หมิกกีเม้าส์ (*Opuntia microdasys*)

ลักษณะที่โดดเด่นคือลำต้นกลมหรือแบน มีจุดนามธรรมกระจายอยู่ทั่วผิว และจะแตกหน่อออกจากตุ่มห่านมลายเป็นต้นใหม่ต่อไปเรื่อย ๆ ดังรูปที่ 2.27



รูปที่ 2.27 หมิกกีเม้าส์

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยของจักรินทร์ สนุกแสน และโอลิฟาริก สุรินตี (2018) ได้ศึกษาการใช้ CNN เพื่อนำมาจำแนกพรรณไม้ที่มีอยู่ในธรรมชาติ โดยได้นำ LeNet-5, AlexNet, GoogLeNet และ VGGNet ทั้งหมด 4 แบบ เพื่อเปรียบเทียบการประมวลผลข้อมูลพรรณไม้ทั้งหมด 3 ชุดข้อมูลได้แก่

ชุดข้อมูล PNE ที่ประกอบไปด้วยพรรณไม้จำนวน 10 ชนิด ชนิดละ 300 รูปภาพ ซึ่งรายชื่อของพรรณไม้แสดงดังตารางที่ 2.1

ตารางที่ 2.1 รายชื่อพรรณไม้ในชุดข้อมูล PNE

ลำดับ	ชื่อพรรณไม้	ชื่อพรรณไม้ภาษาไทย
1	Fairy Rose	กุหลาบหนู
2	Glory Tree	นางແຍ້ມ
3	Antigonia	พวงชมพุ
4	Jasminum Auriculatum	พุธชาด
5	Arabian Jasmine	มะลิลา
6	Wild Water Plum	โนกพวง
7	Plumeria	ลีลาวดี
8	Rangoon Creeper	ເລື່ບມືອນາງ
9	Cape of Good Hope	วาสนา
10	Dwarf Ylang-Ylang Shrub	ສ່າເຫຼັກປັຕານ

ชุดข้อมูล 102 Flower ที่ประกอบไปด้วยรูปภาพของพรรณไม้จำนวนกว่า 102 สายพันธุ์โดยมีรูปภาพสายพันธุ์ละ 40-250 รูปภาพ รวมทั้งหมดกว่า 8,189 รูปภาพ ซึ่งแสดงดังตัวอย่างในรูปที่ 2.28



รูปที่ 2.28 ตัวอย่างรูปภาพของชุดข้อมูล 102 Flower

ชุดข้อมูล Folio ประกอบไปด้วยใบไม้ทั้งหมด 32 ชนิด ซึ่งเก็บรวบรวมชนิดละ 20 ใบ ซึ่งมีรูปภาพทั้งหมด 640 รูปภาพ ซึ่งแสดงดังตัวรูปที่ 2.29



รูปที่ 2.29 ตัวอย่างของชุดข้อมูล Folio

ซึ่งชุดข้อมูลทั้ง 3 จะถูกนำมาใช้เพื่อทดสอบประสิทธิภาพของการเรียนรู้เชิงลึก (Deep learning) ซึ่งแสดงรายละเอียดดังตารางที่ 2.2

ตารางที่ 2.2 รายละเอียดการแบ่งชุดข้อมูล

ชุดข้อมูล	จำนวนชนิด	จำนวนรูปภาพ	จำนวนรูปที่ใช้เทรน	จำนวนรูปที่ใช้ทดสอบ
PNE	10	3,000	2,400	600
102 Flower	102	8,189	6,564	1,625
Folio	32	637	510	127

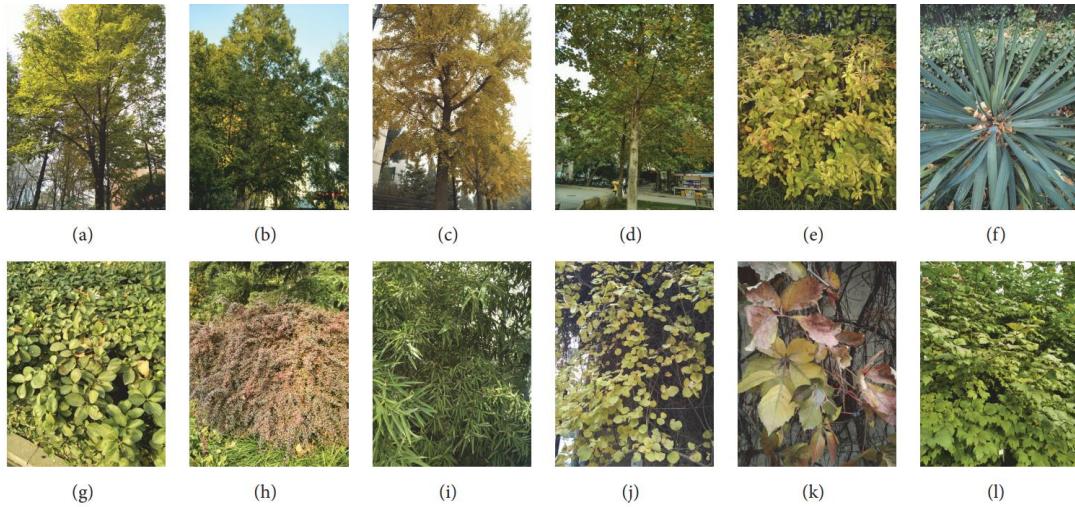
จากการศึกษาพบว่า GoogLeNet นั้นมีการประสิทธิภาพสูงสุดในการประมวลผลข้อมูล PNE และ 102 Flower และยังใช้เวลาในการเรียนรู้เร็วกว่า VGGNet แต่ทั้งนี้โครงสร้างแบบ VGGNet นั้นกลับมีความถูกต้องของชุดข้อมูล Folio มากกว่า ดังตารางที่ 2.3 จึงทำให้หากต้องการที่จะสร้างโมเดลเพื่อจำแนกพรรณไม้ในธรรมชาตินั้นควรจะใช้โครงสร้างแบบ GoogLeNet

ตารางที่ 2.3 ผลการทดสอบการเรียนรู้เชิงลึกของชุดข้อมูล PNE, 102 Flower และ Folio

ชนิดของชุดข้อมูล	โครงสร้าง CNN	ความถูกต้องของจำนวนรอบที่ใช้เทรน/เวลาที่ใช้เทรน			
		1,000 รอบ	เวลาที่ใช้	10,000 รอบ	เวลาที่ใช้
PNE	LeNet-5	88.50±0.14	~1.3h	89.17±2.13	~14h
	AlexNet	96.67±0.23	~1.5h	97.33±2.18	~6.5h
	GoogLeNet	99.04±0.11	~4.3h	99.17±1.75	~1.9d

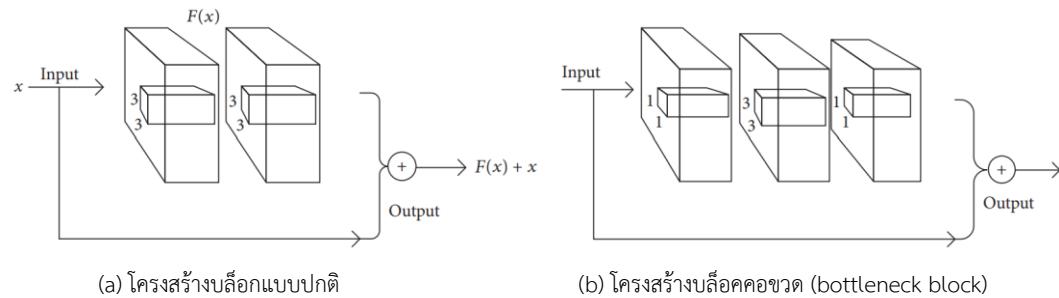
	VGGNet	97.17 ± 0.64	$\sim 20h$	97.33 ± 0.85	$\sim 6.7d$
102 Flower	LeNet-5	33.19 ± 0.27	$\sim 4.4h$	31.42 ± 1.12	$\sim 1.16d$
	AlexNet	66.06 ± 1.74	$\sim 6.3h$	66.38 ± 2.46	$\sim 1.6d$
	GoogLeNet	74.98 ± 1.59	$\sim 1.3d$	78.89 ± 0.69	$\sim 4.4d$
	VGGNet	62.60 ± 0.87	$\sim 1.19d$	64.12 ± 2.74	$\sim 17d$
Folio	LeNet-5	84.25 ± 0.51	$\sim 20m$	74.02 ± 0.95	$\sim 2.5h$
	AlexNet	85.04 ± 0.48	$\sim 30m$	73.23 ± 0.81	$\sim 1.4h$
	GoogLeNet	87.40 ± 1.54	$\sim 3.5h$	82.68 ± 2.41	$\sim 7.2h$
	VGGNet	91.85 ± 3.31	$\sim 3.5h$	87.49 ± 1.54	$\sim 1.8d$

งานวิจัยของ Yu Sun, Yuan Liu, Guan Wang และ Haiyan Zhang (2017) ได้ศึกษาการจำแนกพรรณพืชที่เป็นรูปถ่ายในสภาพแวดล้อมธรรมชาติโดยใช้ชุดข้อมูลรูปภาพ BJFU100 ซึ่งมีพันธุ์ไม้ 100 ชนิด ซึ่งรูปภาพกว่า 10,000 รูปภาพ ทั้งหมดเป็นรูปที่ถ่ายด้วยสมาร์ทโฟน ดังตัวอย่างในรูปที่ 2.30 มาทดสอบ

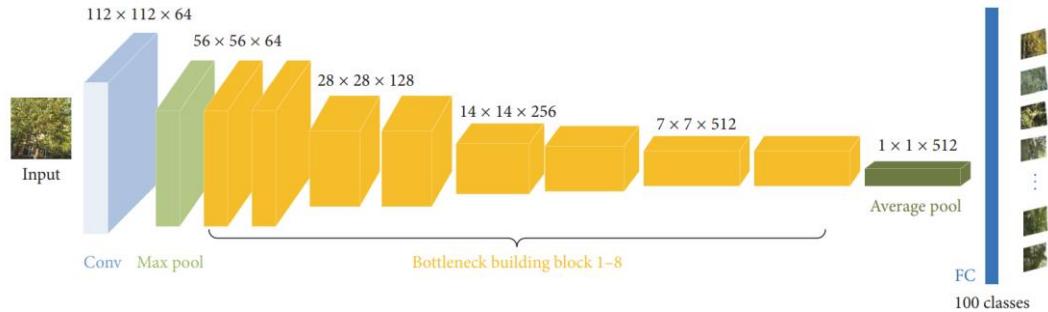


รูปที่ 2.30 ตัวอย่างชุดข้อมูล BJFU100

พวกเข้าได้ใช้ CNN ที่มีโครงสร้างแบบ Resnet โดยได้ทดลองกับจำนวนชั้น 18, 26, 34 และ 50 แต่เนื่องจากโครงสร้างปกตินั้นมีปัญหาในการแยกแยะการไล่ระดับสีของใบไม้ในพืช จึงจำเป็นที่จะใช้ Resnet เนื่องจากมีการเพิ่มชั้นที่ใช้เรียนรู้ข้อมูลเพิ่มซึ่งถูกเรียกว่าบล็อกคอขวด(bottleneck block) ทำให้สามารถทำงานได้ดีขึ้น ดังรูปที่ 2.31 แสดงโครงสร้างของบล็อกแบบปกติกับบล็อกคอขวด ซึ่งโครงสร้างทั้งหมดของ Resnet 26 ชั้น แสดงดังตัวอย่างในรูปที่ 2.32

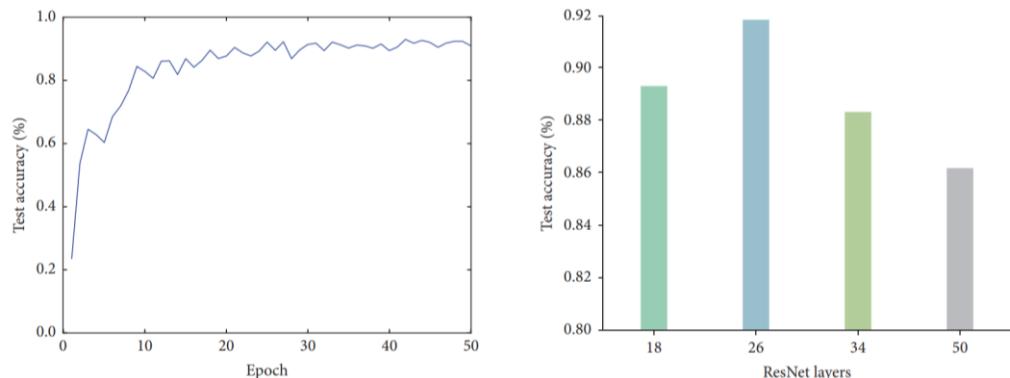


รูปที่ 2.31 เปรียบเทียบโครงสร้างบล็อกแบบปกติกับแบบบล็อกคอขวด



รูปที่ 2.32 ตัวอย่างโครงสร้างของ Resnet 26 ชั้น

จากการทดลองได้แบ่งข้อมูลออกเป็น 2 ส่วนคือ ข้อมูลสำหรับการเรียนรู้จำนวน 80% และข้อมูลสำหรับการทดสอบโมเดลจำนวน 20% จากการศึกษาพบว่าโครงสร้าง Resnet ที่มีจำนวน 26 ชั้น นั้นมีความถูกต้องมากที่สุดที่ 91.78% ดังรูปที่ 2.33

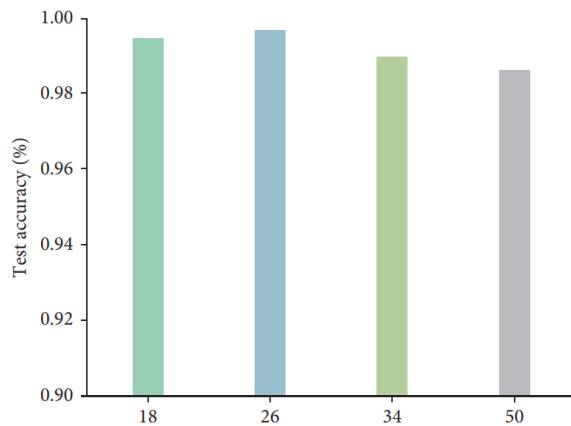


รูปที่ 2.33 แสดงความแม่นยำต่อจำนวนที่ใช้เท่าน และความแม่นยำต่อโครงสร้างชั้นของ Resnet

นอกจากนี้พากเข้ายังได้นำโมเดล Resnet นี้ไปทดสอบกับชุดข้อมูล Flavia ซึ่งเป็นข้อมูลใบไม้ 32 ชนิด รูปภาพทั้งหมด 1,907 รูปภาพ ดังตัวอย่างรูปที่ 2.34 โดยได้แบ่งข้อมูลเป็น 2 ส่วนคือ ข้อมูลสำหรับการเรียนรู้จำนวน 80% และข้อมูลสำหรับการทดสอบโมเดลจำนวน 20% มีความถูกต้องถึง 99.65% ดังรูปที่ 2.35



รูปที่ 2.34 ตัวอย่างของชุดข้อมูล Flavia



รูปที่ 2.35 ผลการทดสอบ CNN โครงสร้าง Resnet กับชุดข้อมูล Flavia

งานวิจัยของ Abdolvahab Ehsanirad (2010) ได้ศึกษาการจำแนกพันธุ์พืชด้วยใบไม้ โดยใช้การประมวลผลภาพใช้ 2 วิธีคือ Gray-Level Co-occurrence matrix (GLCM) และ Principal Component Analysis (PCA) โดย GLCM คือการเปรียบเทียบพิกเซลที่ใกล้เคียงของรูปที่แปลงเป็นภาพขาวดำซึ่งเป็นอัลกอริทึมสามารถที่จะสกัดเอากลุ่มสมบัติของใบไม้ ดังตัวอย่างในตารางที่ 2.4 และตารางที่ 2.5

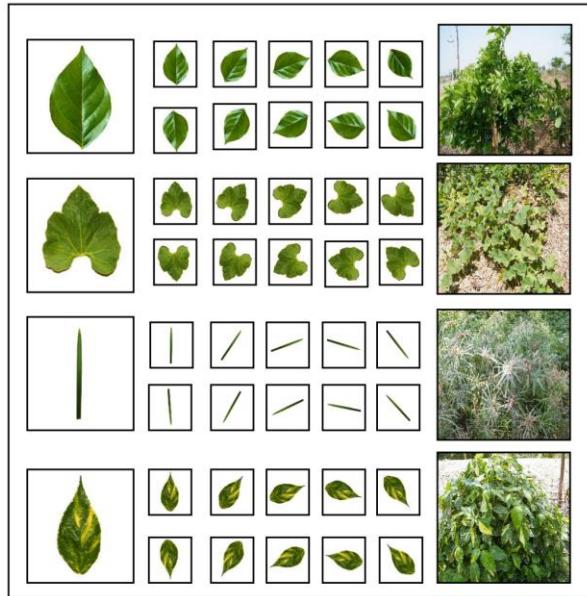
ตารางที่ 2.4 แสดงคุณสมบัติที่สกัดได้ไปไม้ชนิด (a)

มุม	สหสัมพันธ์อัตโนมัติ	เอนโทรปี	ความคงชัด	สหสัมพันธ์	ภาวะเอกพันธุ์
0°	45.5748	1.4311	0.3184	0.9638	0.6144
45°	45.2799	1.4928	0.4458	0.9496	0.6060
90°	45.6190	1.3886	0.2301	0.9738	0.6166
135°	45.2932	1.4716	0.4192	0.9526	0.6074

ตารางที่ 2.5 แสดงคุณสมบัติที่สกัดได้จากไปไม้ชนิด (b)

มุม	สหสัมพันธ์อัตโนมัติ	เอนโทรปี	ความคงชัด	สหสัมพันธ์	ภาวะเอกพันธุ์
0°	54.8540	0.8972	0.4361	0.9401	0.8214
45°	54.7371	0.9132	0.4438	0.9396	0.8199
90°	54.9797	0.8405	0.1845	0.9747	0.8267
135°	54.6610	0.9310	0.5961	0.9189	0.8172

ในการจำแนกพันธุ์พืชจำนวน 13 ชนิด ซึ่งได้ใช้ชุดข้อมูลมาทดสอบเป็นไปไม้ชนิดใหม่หรือไปไม้ที่ผิดรูปแบบมาทดสอบจำนวน 65 ชนิด และเรียนรู้ไปไม้ทั้งหมด 390 ใบ ดังรูปตัวอย่างที่ 2.37



รูปที่ 2.36 ตัวอย่างข้อมูลและชนิดของพันธุ์พืช

จากการวิจัยพบว่า PCA นั้นมีความถูกต้องที่ 98% และประสิทธิภาพมากกว่า GLCM ดังตารางที่ 2.6

ตารางที่ 2.6 แสดงประสิทธิภาพระหว่าง GLCM ในมุมองศาต่าง ๆ ที่ใช้ทดสอบ และ PCA

มุมองศา	ค่าเฉลี่ยความถูกต้องที่ทดสอบ (%)
0°	78.46
45°	49.23
90°	78.46
135°	70.76
PCA	98.46

ทฤษฎีและงานวิจัยที่เกี่ยวข้องที่ถูกกล่าวมาในบทนี้จะถูกนำไปพัฒนาและประยุกต์ใช้กับการพัฒนาแอปพลิเคชันจำแนกสายพันธุ์ของระบบทองเพชร เพื่อให้เป็นไปตามวัตถุประสงค์ โดยการแอปพลิเคชันนี้จะถูกกล่าวไว้ในบทต่อไป

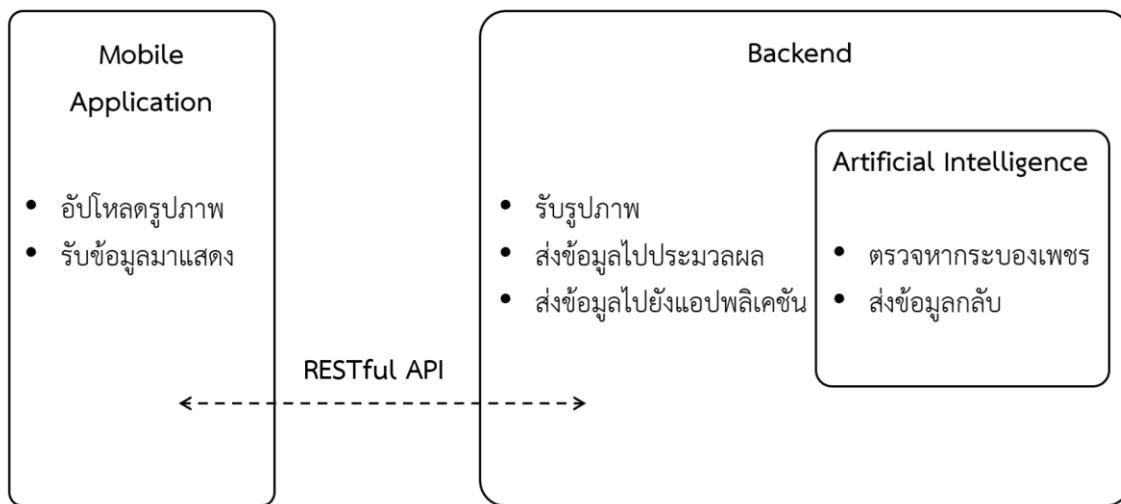
บทที่ 3

วิธีดำเนินโครงการ

ในบทนี้จะกล่าวถึงการนำทฤษฎีที่เกี่ยวข้องจากในบทที่ 2 มาประยุกต์ใช้ให้สอดคล้องกับการพัฒนาแอปพลิเคชันจำแนกสายพันธุ์ของเบ็ดเตล็ดโดยแต่ละส่วนมีรายละเอียด วิธีการดังต่อไปนี้

3.1 ภาพรวมของระบบ

ภาพรวมของระบบแอปพลิเคชันจำแนกสายพันธุ์ของเบ็ดเตล็ด โดยแบ่งออกเป็น 3 ส่วน คือ ส่วนของแอปพลิเคชัน หรือส่วนของผู้ใช้แอปพลิเคชัน เป็นส่วนสำคัญในการมีปฏิสัมพันธ์กับผู้ใช้ ส่วนของระบบจัดการที่จะค่อยทำหน้าที่สื่อสารระหว่างผู้ใช้และส่วนหลัก ด้วยการรับรูปภาพมาแปลงเป็นเบส 64 และจึงส่งข้อมูลต่อไปยังปัญญาประดิษฐ์ และส่วนสุดท้ายคือส่วนของปัญญาประดิษฐ์ที่จะนำข้อมูลจากการจัดการไปทำการประมวลผล และส่งข้อมูลกลับมาไปยังแอปพลิเคชันดังรูปที่ 3.1



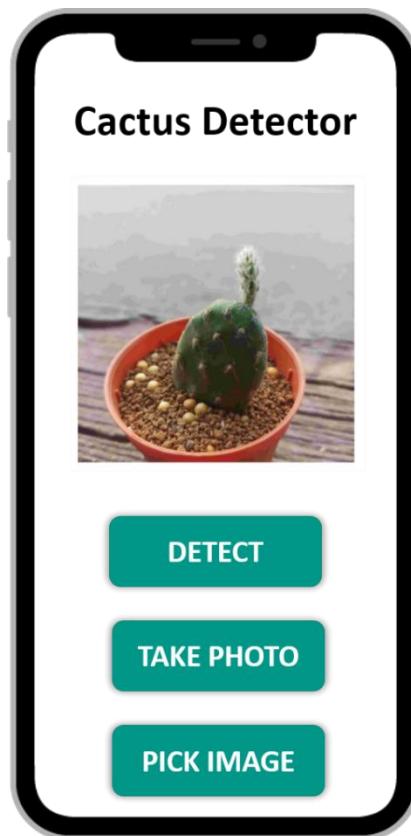
รูปที่ 3.1 ภาพรวมระบบของแอปพลิเคชันจำแนกสายพันธุ์ของเบ็ดเตล็ด

3.2 การออกแบบส่วนประสาน และการพัฒนาแอปพลิเคชัน

ในหัวข้อนี้จะอธิบายเกี่ยวกับ การออกแบบส่วนประสาน และการพัฒนาแอปพลิเคชัน เพื่อให้ตอบโจทย์ต่อการใช้งานต่าง ๆ โดยแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

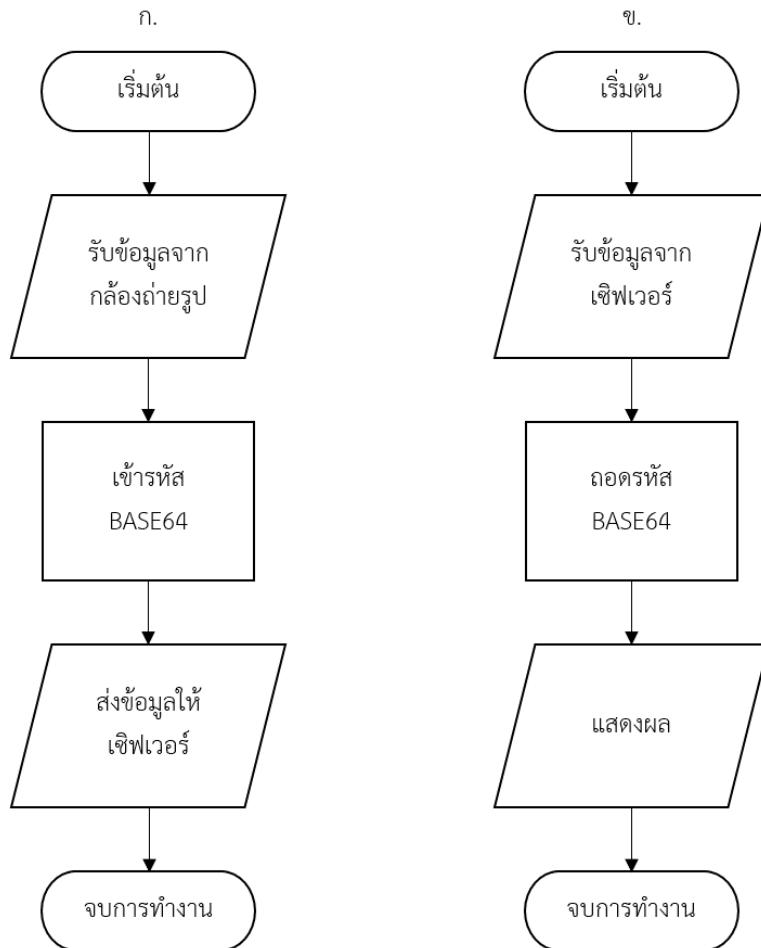
3.2.1 การออกแบบส่วนประสาน (User Interface Design)

การออกแบบส่วนประสานสำหรับแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชรสำหรับผู้ใช้ เป็นการออกแบบส่วนประสานเพื่อศูนย์กลางเอียงต่าง ๆ เสน่ห์ทางของการทำแอปพลิเคชัน การจำลองการเปลี่ยนหน้าระหว่างการใช้งานก่อนทำการพัฒนาจริง รวมถึงการซ่อมแซมออกแบบบริการต่าง ๆ ที่ต้องพัฒนา ดังรูปที่ 3.2



รูปที่ 3.2 การออกแบบส่วนประสานสำหรับแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชรสำหรับผู้ใช้

เมื่อแอปพลิเคชันรับรูปภาพจากผู้ใช้จากการถ่ายภาพหรืออัปโหลดรูปภาพ แอปพลิเคชันก็จะทำการเข้ารหัส BASE64 เพื่อส่งไปให้ยังเครื่องบริการส่งต่อไปให้ปัญญาประดิษฐ์ประมวลผล แล้วแอปพลิเคชันจะรับข้อมูลจากเครื่องบริการที่ส่งมาจากปัญญาประดิษฐ์แล้วมาถอดรหัส BASE64 เพื่อนำมาแสดงผลให้แก่ผู้ใช้ดังรูปที่ 3.3

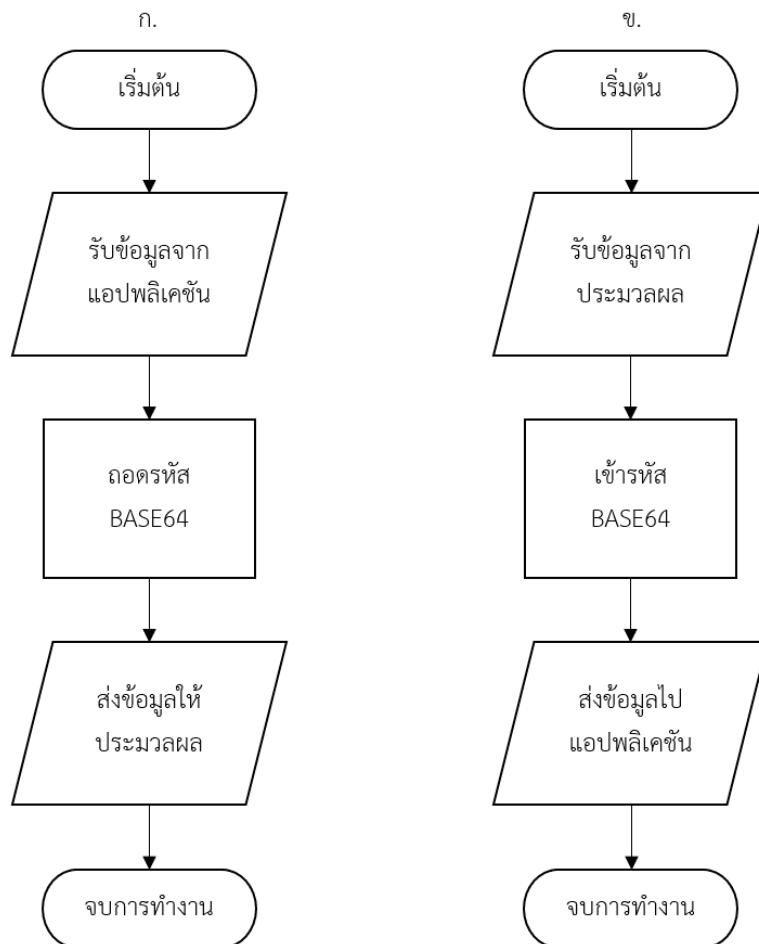


รูปที่ 3.3 การทำงานของแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชร

3.3 การออกแบบและพัฒนาระบบส่วนหลัง

3.3.1 ระบบส่วนหลัง

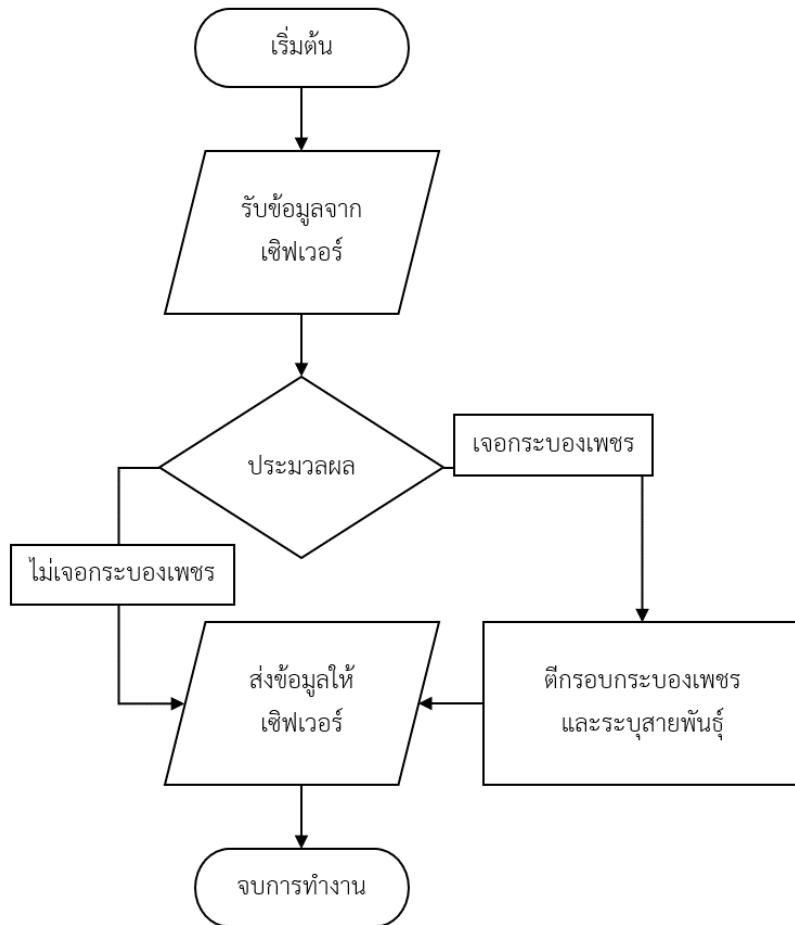
เมื่อแอปพลิเคชันทำการส่งข้อมูลมา จะทำการถอดรหัสข้อมูลจาก BASE64 ให้เป็นรูปภาพ จากนั้นจะส่งเข้าไปประมวลผลภาพ เมื่อประมวลผลภาพเสร็จ จะทำการเข้ารหัสข้อมูลภาพเป็นข้อมูล BASE64 และทำการส่งไปที่แอปพลิเคชันเพื่อแสดงผลให้ผู้ใช้ดังรูปที่ 3.4



รูปที่ 3.4 การทำงานของระบบส่วนหลัง

3.3.2 ระบบประมวลผลภาพ

3.3.2.1 การตรวจหากรอบของเพชร



รูปที่ 3.5 การตรวจหากรอบของเพชร

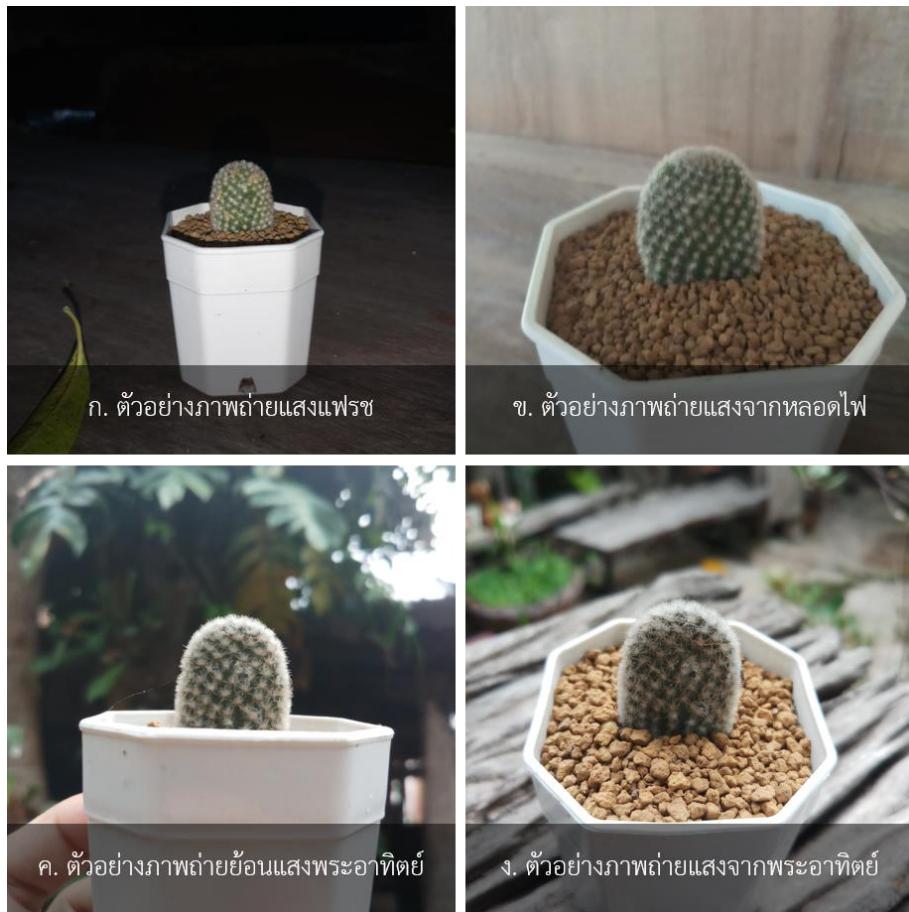
จากรูป 3.5 เมื่อเครื่องบริการมีการเรียกใช้การประมวลผลภาพ ระบบจะรับภาพเข้ามาแล้วทำการประมวลผลตรวจสอบว่าพบกรอบของเพชรในภาพหรือไม่ ถ้าหากว่าระบบตรวจเจอกรอบของเพชร ก็จะทำการตีกรอบรูปภาพในจุดที่มีกรอบของเพชรอยู่ รวมไปถึงบอกว่าเป็นสายพันธุ์ใด แล้วจึงส่งข้อมูลดังกล่าวกลับไปยังเครื่องบริการ แต่ถ้าหากระบบไม่สามารถตรวจจับกรอบของเพชรในภาพได้ ระบบจะส่งข้อมูลกลับไปยังเครื่องบริการเพื่อแจ้งผลว่าไม่พบกรอบของเพชร

3.4 การพัฒนาระบบตรวจสอบระบบของเพชร

3.4.1 การเตรียมชุดข้อมูล

3.4.1.1 รูปภาพระบบของเพชร

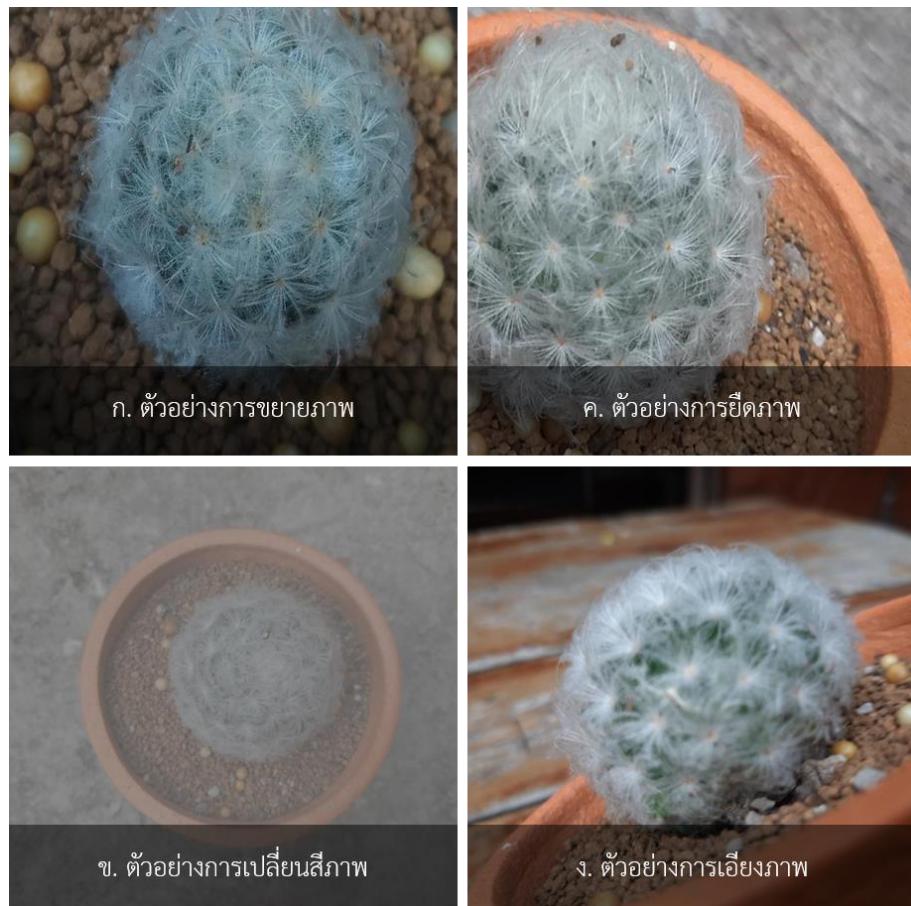
รูปภาพระบบของเพชรทั้งหมด 15 สายพันธุ์นั้นถูกถ่ายภาพโดยโทรศัพท์มือถือ ในสภาพแวดล้อมที่มีแสงแตกต่างกัน เช่น ถ่ายภาพในอาคาร มีแหล่งกำเนิดแสงเป็นหลอดไฟ หรือถ่ายกลางแจ้งแหล่งกำเนิดแสงเป็นดวงอาทิตย์ เป็นต้น โดยถ่ายภาพให้เห็นมุมต่าง ๆ ของต้นระบบของเพชร เช่น ด้านบน ด้านข้าง เป็นต้น ซึ่งต้นระบบของเพชรทั้งหมด 15 สายพันธุ์ถูกถ่ายรูปสายพันธุ์ละ 300 รูป รวมทั้งหมด 4500 รูป ดังรูปตัวอย่างที่ 3.6



รูปที่ 3.6 ตัวอย่างภาพถ่ายต้นระบบของเพชรในสภาพแวดล้อมที่มีแสงแตกต่างกัน

3.4.1.2 การปรับแต่งรูปภาพ (Image augmentation)

รูปภาพระบบของเพชรที่ได้จากการถ่ายภาพทั้งหมด 4500 รูป จะถูกปรับขนาดของรูปให้ด้านกว้างมีขนาด 960 พิกเซล ด้านยาวมีขนาด 1280 พิกเซล จากนั้นจะนำรูปภาพระบบของเพชรทั้งหมดไปสุ่มจำนวน 200 รูป จากสายพันธุ์ลักษณะ 300 รูป เพื่อทำการยืด หด หมุน สลับด้าน ดังรูปตัวอย่างที่ 3.7 จึงทำให้ได้รูปภาพระบบของเพชรทั้งหมดสายพันธุ์ลักษณะ 500 รูปทำให้รูปภาพทั้งหมดมี 7500 รูป



รูปที่ 3.7 ตัวอย่างรูปที่ผ่านการยืด หด หมุน กลับด้าน

3.4.1.3 การกำกับรูปภาพ (Image labeling)

จากรูปภาพสายพันธุ์ระบบของเพชรทั้งหมด 7500 รูปจะถูกกำกับรูปภาพ (label image) แต่ละสายพันธุ์โดยใช้ป้ายกำกับเป็นรหัสที่สร้างขึ้นสำหรับแต่ละสายพันธุ์ดังตารางที่ 3.1

ตารางที่ 3.1 รหัสที่สร้างขึ้นสำหรับสายพันธุ์ต่าง ๆ

รหัส	ชื่อสายพันธุ์
C001	หมากสังขราชน (Astrophytum myriostigma)
C002	ปราสาทนางฟ้า (Cereus peruvianus)
C003	ไก่ต็อก (Haworthia coarctata)
C004	กระบอกทอง (Echinocactus leninghausii)
C005	หนามพิกุล (Mammillaria scrippsiana)
C006	ลูกแมว (Mammillaria bocasana)
C007	ขนนกขาว (Mammillaria plumosa)
C008	นิ่วทอง (Mammillaria Elongata)
C009	หูมิกีเมร์ส (Opuntia microdasys)
C010	นูปต้า (Echinopsis subdenudata)
C011	มิ哈โน (Gymnocalycium mihanovichii)
C012	แม่มกลวยหอม (Mammillaria longimamma)
C013	ดาวล้อมเดือน (Echinopsis calochalora)
C014	บานเย็นนัม (Gymnocalycium baldianum)
C015	แม่มอรุชา�ุ (Mammillaria bucareliensis)

3.4.2 การเรียนรู้เชิงลึก (Deep learning)

รูปภาพกรอบของเพชรทั้งหมด 15 สายพันธุ์จะถูกนำมาเรียนรู้เชิงลึกจากโมเดลเครือข่ายประสาทเทียม 3 โมเดลที่แตกต่างกันได้แก่ EfficientDet-D5 Faster-RCNN Resnet-50 และ YOLOv5 ซึ่งทั้งสามโมเดลนั้นเป็นการเรียนรู้ต่อจากเดิม (Transfer learning) เพื่อปรับปรุงประสิทธิภาพของโมเดล โดยทำการเรียนรู้บนเครื่องคอมพิวเตอร์ที่มีรายละเอียดเป็น Intel(R) Xeon(R) CPU @ 2.00GHz มีขนาด RAM ที่ 24 GB และ GPU เป็น Nvidia Tesla P100 มีขนาด RAM ที่ 16 GB

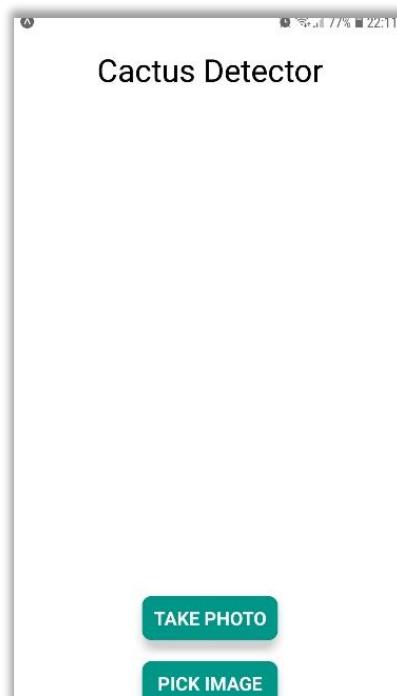
บทที่ 4

ผลการทดลองและวิเคราะห์

จากบทที่ 3 ได้กล่าวถึงการออกแบบและการพัฒนาระบบของแอปพลิเคชันจำแนกสายพันธุ์กระบอกเพชร ในบทนี้จะนำเสนอผลการทดสอบประสิทธิภาพของระบบดังกล่าวที่ได้พัฒนาขึ้น

4.1 ผลการทดสอบส่วนแสดงผล

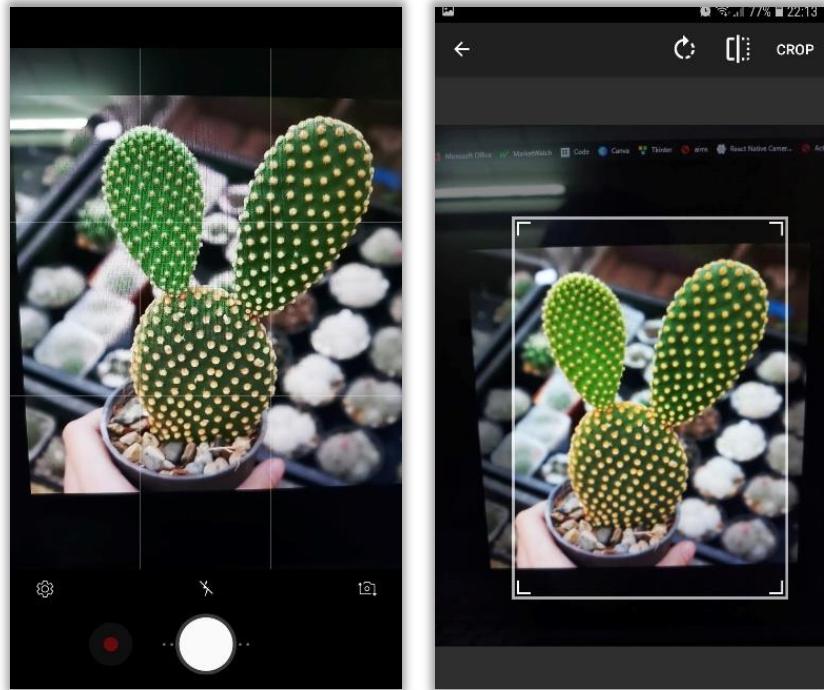
4.1.1 การทดสอบส่วนแสดงผลการเลือกที่มาข้อมูล



รูปที่ 4.1 ส่วนแสดงผลการเลือกที่มาข้อมูล

จากรูปที่ 4.1 ส่วนแสดงผลการเลือกที่มาข้อมูล จะมีปุ่มสำหรับให้ผู้ใช้เลือก เพื่อนำเข้าข้อมูลระหว่างถ่ายรูปภาพจากโทรศัพท์ (TAKE PHOTO) กับอัปโหลดรูปที่มีอยู่ (PICK IMAGE)

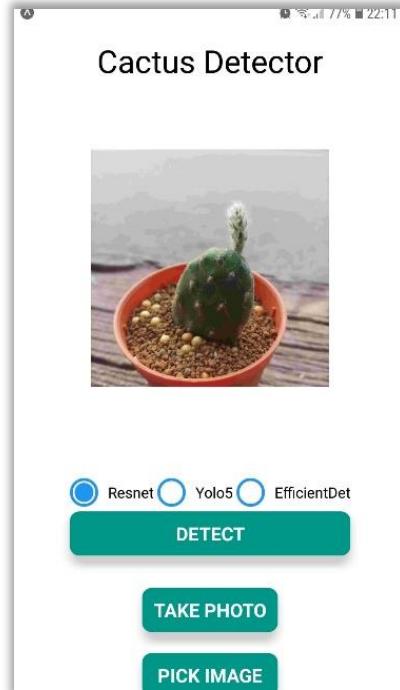
4.1.2 การทดสอบส่วนแสดงผลการเลือกถ่ายรูปภาพ



รูปที่ 4.2 ส่วนแสดงผลการเลือกถ่ายรูปภาพ

จากรูปที่ 4.2 ส่วนแสดงผลการเลือกถ่ายรูปภาพ เมื่อเลือกถ่ายรูปภาพจากโทรศัพท์ และพลิกเค้นจะส่งเข้าไปใช้งานกล้องของโทรศัพท์ และเราจะสามารถตัดรูปภาพให้เหลือเฉพาะส่วนที่เราต้องการตรวจจับสายพันธุ์ระบบบองเพชรได้ เพื่อให้ได้ประสิทธิภาพในการตรวจจับที่ดีที่สุด

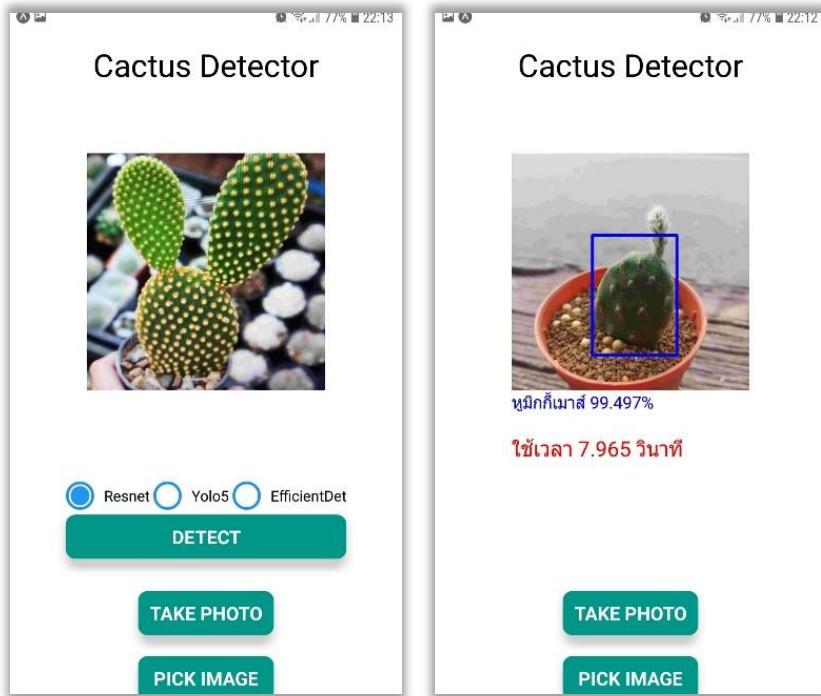
4.2.3 การทดสอบส่วนแสดงผลการเลือกอัปโหลดรูปภาพ



รูปที่ 4.3 ส่วนแสดงผลการเลือกอัปโหลดรูปภาพ

จากรูปที่ 4.3 ส่วนแสดงผลการเลือกอัปโหลดรูปภาพ เมื่อเลือกอัปโหลดรูปภาพจากโทรศัพท์ แอปพลิเคชันจะส่งเข้าไปใช้งานคลังรูปภาพของโทรศัพท์ เพื่อเลือกรูปที่ต้องการตรวจจับสายพันธุ์ระบบของเพชร

4.2.4 การทดสอบส่วนแสดงผลการตรวจจับ



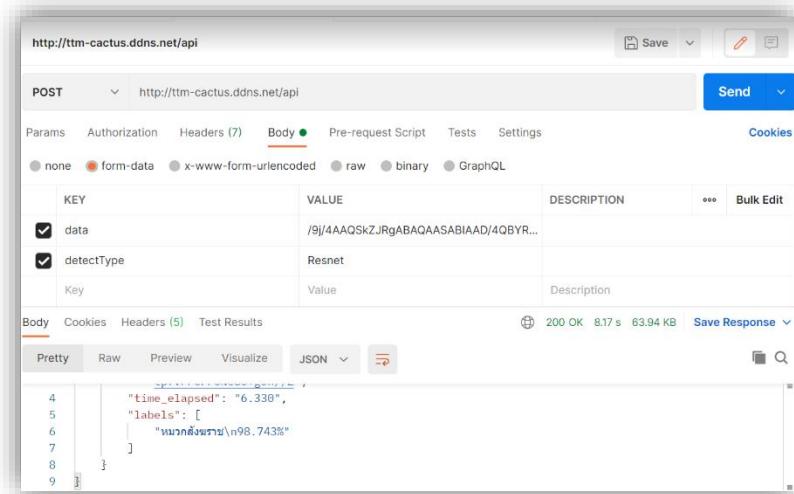
รูปที่ 4.4 ส่วนแสดงผลการตรวจจับ

จากรูปที่ 4.4 ส่วนแสดงผลการตรวจจับ ผู้ใช้สามารถเลือกโมเดลที่ใช้ตรวจจับสายพันธุ์ กระบวนการเพชร ได้ทั้งหมด 3 โมเดล ได้แก่ Resnet, YOLOv5 และ EfficientDet และเมื่อเลือกเสร็จแล้ว ก็กด “DETECT” และแอปพลิเคชันก็จะทำการตรวจจับ โดยเข้ารหัสรูปภาพให้เป็น base64 จากนั้น จึงส่งข้อมูล รูปที่เข้ารหัส และโมเดลที่ใช้ตรวจสอบ ไปยังเครื่องบริการโดยใช้ API ที่สร้างขึ้น และรอการตอบรับจากเครื่องบริการ เมื่อมีการตอบรับจะแสดงเวลาที่ใช้ในการค้นหา และร้อยละความถูกต้องของสายพันธุ์ที่ตรวจพบ รวมไปถึงแสดงชื่อสายพันธุ์ที่ตรวจพบ

4.2 ผลการทดสอบการรับส่งข้อมูลรูปภาพ

หัวข้อนี้จะเป็นการทดสอบการทำงานของ API ที่ทางแอปพลิเคชันทำการส่งข้อมูลไปที่เครื่องบริการ และเครื่องบริการทำการตอบกลับพร้อมข้อมูลมาที่แอปพลิเคชัน

4.2.1 ผลการทดสอบการส่งรูปเพื่อนำไปตรวจหากระ奔跑 และรับรูปและข้อมูลมาแสดงที่แอปพลิเคชัน



รูปที่ 4.5 การทดสอบนำรูปที่ส่งไปที่เครื่องบริการ และเครื่องบริการส่งข้อมูลกลับมา

จากรูป จะเห็นได้ว่า เมื่อทำการส่งรูปไปที่เครื่องบริการ โดยรูปนั้นได้เข้ารหัสแบบ base64 แล้ว เมื่อเครื่องบริการได้รับ ก็จะถอดรหัสให้กลับมาเป็นรูปภาพแล้วจึงนำไปตรวจจับระบบของเพชรในรูป โดยใช้โมเดลที่ถูกส่งมาด้วย หากตรวจจับระบบของเพชรได้ จะทำการส่งรูปที่ถูกตีกรอบ ซึ่งของต้นระบบของเพชรนั้น ๆ ที่สามารถตรวจสอบเจอ พร้อมทั้งร้อยละความถูกต้อง และเวลาที่ใช้ในการตรวจสอบรูปภาพ มาในลักษณะข้อมูลแบบ JSON

4.3 ผลการทดสอบระบบจำแนกสายพันธุ์ระบบของเพชร

จากบทที่ 3 ได้ทำการเทรนโมเดลจำนวน 3 อย่างได้แก่ EfficientDet-D5, Faster-RCNN Resnet-50 และ YOLOv5x จนได้โมเดลที่สามารถจำแนกสายพันธุ์ระบบของเพชรได้ จากนั้นจึงนำทั้ง 3 โมเดลนี้มาทดสอบจำแนกสายพันธุ์ระบบของเพชรจากชุดข้อมูลที่เตรียมไว้จำนวนสายพันธุ์ละ 10 รูป รวมทั้งหมด 150 รูป ซึ่งไม่เกี่ยวข้องกับชุดข้อมูลที่ใช้เทรนทำให้ได้ผลการทดลองดังตารางที่ 4.1, 4.2 และ 4.3

ตารางที่ 4.1 Confusion matrix ของ EfficientDet-D5

	Predicted Label															
	C001	C002	C003	C004	C005	C006	C007	C008	C009	C010	C011	C012	C013	C014	C015	None
True Label	C001	7	1	0	0	0	0	1	0	0	0	0	0	0	0	1
C002	0	0	3	0	0	0	0	2	5	0	0	0	0	0	0	0
C003	0	0	6	0	0	0	0	2	2	0	0	0	0	0	0	0
C004	1	0	0	5	0	3	0	0	1	0	0	0	0	0	0	0
C005	0	0	0	0	6	0	0	0	0	0	3	0	0	1	0	0
C006	0	0	0	0	0	8	1	0	0	0	1	0	0	0	0	0
C007	0	0	0	0	0	1	7	0	0	0	0	0	0	0	0	2
C008	1	0	1	3	1	0	0	4	0	0	0	0	0	0	0	0
C009	2	0	1	0	0	0	0	0	6	0	0	0	0	0	0	1
C010	3	0	0	0	0	0	0	0	0	3	1	0	0	1	2	0
C011	0	0	0	0	1	0	0	0	0	0	7	0	0	1	0	1
C012	0	0	0	0	6	0	0	0	2	0	0	1	0	1	0	0
C013	0	0	0	0	6	0	0	0	0	0	1	0	2	0	1	0
C014	0	0	0	0	1	0	0	1	0	0	2	0	0	5	0	1
C015	1	0	0	0	4	0	0	0	1	2	0	0	2	0	0	0

ตารางที่ 4.2 Confusion matrix ของ Faster-RCNN Resnet-50

	Predicted Label															
	C001	C002	C003	C004	C005	C006	C007	C008	C009	C010	C011	C012	C013	C014	C015	None
True Label	C001	7	0	0	0	0	0	0	1	0	0	0	0	0	2	0
C002	0	9	0	0	0	0	0	0	1	0	0	0	0	0	0	0
C003	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
C004	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
C005	0	0	0	0	9	1	0	0	0	0	0	0	0	0	0	0
C006	0	0	0	0	0	6	3	1	0	0	0	0	0	0	0	0
C007	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
C008	0	0	0	1	2	0	0	3	2	0	0	0	0	2	0	0
C009	2	0	0	0	0	0	0	0	6	0	0	0	1	0	1	0
C010	0	0	0	0	0	0	0	0	0	3	0	0	1	0	6	0
C011	0	0	0	0	0	0	0	0	0	0	7	0	0	1	2	0
C012	0	0	0	0	0	0	0	0	0	0	1	8	1	0	0	0
C013	0	0	0	0	0	0	0	0	0	0	1	0	9	0	0	0
C014	0	0	0	0	0	0	0	0	0	0	1	0	0	9	0	0
C015	0	1	0	0	0	0	0	0	0	1	0	0	0	0	8	0

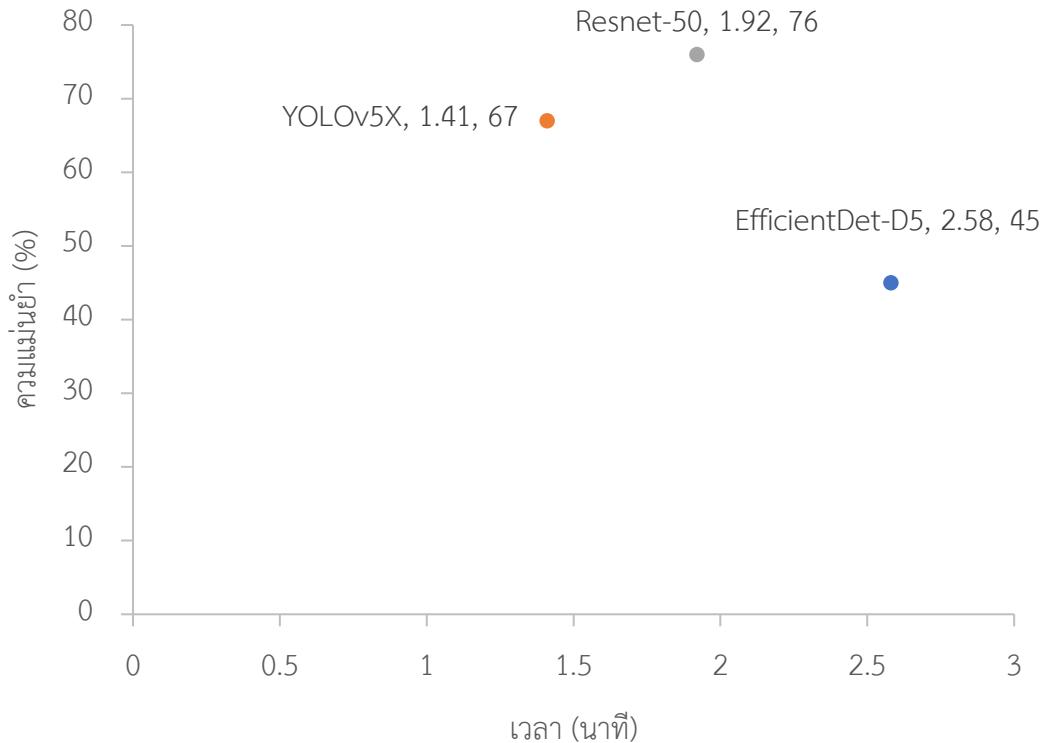
ตารางที่ 4.3 Confusion matrix ของ YOLOv5

		Predicted Label															
		C001	C002	C003	C004	C005	C006	C007	C008	C009	C010	C011	C012	C013	C014	C015	None
True Label	C001	9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
	C002	0	5	4	0	0	0	0	0	0	0	0	0	0	0	0	1
	C003	0	0	8	0	0	1	1	0	0	0	0	0	0	0	0	0
	C004	0	0	0	9	0	0	0	1	0	0	0	0	0	0	0	0
	C005	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	1
	C006	0	0	0	0	0	9	1	0	0	0	0	0	0	0	0	0
	C007	0	0	0	0	0	0	9	0	1	0	0	0	0	0	0	0
	C008	0	0	0	2	0	1	0	5	0	0	0	0	0	0	0	2
	C009	0	1	1	0	1	0	0	0	6	0	0	0	0	0	0	1
	C010	0	0	0	0	0	0	1	0	1	6	0	0	1	0	0	1
	C011	1	0	1	0	0	0	0	0	0	0	4	0	0	0	3	1
	C012	0	0	0	0	1	3	0	0	0	0	0	6	0	0	0	0
	C013	0	0	0	0	0	1	0	0	0	0	0	0	8	0	0	1
	C014	0	1	0	0	3	0	0	0	0	0	0	0	1	3	1	1
	C015	0	0	0	1	0	0	0	0	0	1	0	0	0	0	5	3

จากตารางแสดงผลการทดสอบของทั้ง 3 โมเดล ทำให้สามารถหาความถูกต้องของการจำแนกสายพันธุ์ระบบป้องเพชรและเวลาที่ใช้ในการจำแนกจากชุดข้อมูลได้ดังตารางที่ 4.4 และรูปภาพที่ 4.6

ตารางที่ 4.4 ผลการทดสอบโมเดลทั้ง 3 กับชุดข้อมูล

โมเดล	ความถูกต้อง (%)	เวลา (นาที)
EfficientDet-D5	45	2.58
Faster-RCNN Resnet-50	76	1.92
YOLOv5	67	1.41



รูปที่ 4.6 กราฟแสดงความแม่นยำเทียบกับเวลา

บทที่ 5

บทสรุปและข้อเสนอแนะ

แอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชร โดยมีจุดประสงค์เพื่อใช้การรู้เชิงลึกมาจำแนกสายพันธุ์ระบบของเพชรโดยใช้รูปภาพ และเปรียบเทียบความแม่นยำของโมเดลที่ได้ ซึ่งผัฒนาแอปพลิเคชันโดยใช้ React-Native เพื่อให้สามารถทำงานได้ทั้งระบบ Android และ iOS ซึ่งแลกเปลี่ยนข้อมูลด้วยระบบบริการ RESTful API รวมไปถึงการใช้โมเดลการเรียนรู้เชิงลึกจำนวน 3 อย่างได้แก่ EfficientDet-D5, Faster-RCNN Resnet-50 และ YOLOv5 เพื่อจำแนกสายพันธุ์ของต้นระบบของเพชรทั้ง 15 สายพันธุ์ด้วยรูปภาพ

5.1 สรุปผลการทดลอง

จากการพัฒนาแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชรด้วย React-Native นั้นสามารถให้ผู้ใช้เลือกถ่ายรูปภาพหรือใช้รูปที่มีในเครื่อง และเข้ารหัสรูปภาพให้อยู่ในรูป base64 พร้อมทั้งระบุโมเดลที่ใช้ในการตรวจสอบ จากนั้นจึงส่งข้อมูลไปยังเครื่องบริการและรอการตอบกลับ เมื่อเครื่องบริการได้รับข้อมูลจะนำรูปภาพมาถอดรหัส และจึงไปเรียกใช้โมเดลในการตรวจจับแล้วจึงนำผลที่ได้ส่งกลับไปยังแอปพลิเคชัน เนื่องจากต้องการเปรียบเทียบความแม่นยำของโมเดลทั้ง 3 อย่างได้แก่ EfficientDet-D4, Faster-RCNN Resnet-50 และ YOLOv5 จึงเปรียบเทียบด้วยการใช้ชุดข้อมูลที่ไม่ได้นำไปเทรนด้วยจำนวนสายพันธุ์ละ 10 รูป รวมทั้งหมด 150 รูป ทำให้ได้โมเดลที่มีความแม่นยำสูงสุดคือ Faster-RCNN Resnet-50 ที่ 76% รองลงมาเป็น YOLOv5x ที่ 67% และท้ายสุดคือ EfficientDet-D5 ที่ 45% และโมเดลที่ใช้เวลาน้อยสุดได้แก่ YOLOv5x ที่ 1.41 นาที รองลงมาเป็น Faster-RCNN Resnet-50 ที่ 1.92 นาที และสุดท้ายคือ EfficientDet-D5 ที่ 2.58 นาที

5.2 วิจารณ์ผลการทดลอง

จากการทดสอบแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชร พบร่วมกับแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชรนั้นจะมีประสิทธิภาพตามโมเดลที่เลือกใช้ ซึ่งโมเดลที่มีความแม่นยำสูงก็จะใช้เวลาในการตรวจจับนาน ในทางกลับกันโมเดลที่ความแม่นยำต่ำก็ใช้เวลาในการตรวจจับน้อย โดยรวมแล้ว

แอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชรสามารถทำงานได้ตามวัตถุประสงค์ และจุดอ่อนของตัวแอปพลิเคชันจำแนกสายพันธุ์ระบบของเพชรคือ ในบางมุมของรูปที่ใช้ในการตรวจจับนั้นอาจจะเกิดความคลาดเคลื่อนของสายพันธุ์ที่ตรวจจับไม่ตรงกับสายพันธุ์ที่นำมาทดสอบ เนื่องด้วยในบางสายพันธุ์ของระบบของเพชรมีลักษณะที่ค่อนข้างคล้ายกันค่อนข้างมาก

5.3 แนวทางการนำโครงการไปพัฒนาต่อในอนาคต

พัฒนาและออกแบบตัวแอปพลิเคชันให้มีความสวยงามและน่าใช้ สะดวกแก่ผู้ใช้มากยิ่งขึ้น และทำให้ตัวแอปพลิเคชันอยู่ในร้านค้าที่อยู่ในระบบปฏิบัติการนั้น ๆ โดยอาจจะเพิ่มการหารายได้โดยการใส่โฆษณาในตัวแอปพลิเคชัน และทำการปรับปรุงความแม่นยำของตัวตรวจสอบสายพันธุ์ของต้นระบบของเพชร โดยทำการเพิ่มชุดข้อมูลให้หลากหลายมากยิ่งขึ้น และอาจจะเพิ่มจำนวนของสายพันธุ์ต้นระบบของเพชรให้เยอะขึ้น เพื่อการตรวจสอบต้นระบบของเพชรที่หลากหลายขึ้น และเพิ่มประสิทธิภาพของเครื่องบริการ โดยอาจจะเช่าเครื่องบริการที่มีราคาสูงขึ้น เพื่อประสิทธิภาพในการตรวจสอบสายพันธุ์ต้นระบบของเพชรที่รวดเร็วขึ้น

5.4 ข้อเสนอแนะ

หากต้องการให้มีเดลนั้นมีความแม่นยำมากขึ้น สามารถที่จะเพิ่มชุดข้อมูลให้มากกว่านี้ รวมไปถึงการใช้รูปของต้นระบบของเพชรสายพันธุ์เดียวแต่ถ่ายจากหลายต้น และหลากหลายสภาพแสง จะทำให้สามารถเพิ่มความแม่นยำให้กับโมเดลได้

เอกสารอ้างอิง

ผศ. ดร. ชูพันธ์ รัตโนภาค. (2556). **ความรู้ทางปัญญาประดิษฐ์**. กรุงเทพมหานคร: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ผศ. ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล. (2556). **บทนำความฉลาดเชิงคำนวนสำหรับวิศวกรรมคอมพิวเตอร์**. เชียงใหม่: มหาวิทยาลัยเชียงใหม่.

Abdolvahab Ehsanirad. (2010). **Plant Classification Based on Leaf Recognition**. *International Journal of Computer Science and Information Security*. 78-81.

Girshick, Ross. (2015). **Fast R-CNN**. arXiv.

Jakkarin Sanuksan, Olarik Surinta. (2018). **Deep Convolutional Neural Networks for Plant Recognition in the Natural Environment**. Maha Sarakham: Mahasarakham.

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. (2016). **You Only Look Once: Unified, Real-Time Object Detection**. arXiv.

Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. (2015). **Deep Residual Learning for Image Recognition**. arXiv.

Koolapat S. (2018, Feb 25). **Architecture of React Native**. Retrieved from Medium:
<https://medium.com/@koolapatsirikamol/architecture-%e0%e1%e0-f0f818b0ca36>

Mange, P. (2019, May 15). **Javascript Object Notation (JSON)**. Retrieved from Medium:
<https://medium.com/@pakawatmange/javascript-object-notation-json-886824916823>

Mingxing Tan, Ruoming Pang, Quoc V. Le. (2020). **EfficientDet: Scalable and Efficient Object Detection**. arXiv.

Montha, S. (2019, June 1). **REST กับ RESTful API ต่างกันนะรึยัง**. Retrieved from Medium: <https://iamgique.medium.com/restful-api-กับ-rest-api-ต่างกันนะรึยัง-2c70c42990e3>

Preston-Mafham, K. (2007). **500 Cacti: Species and Varieties in Cultivation Paperback.** London: The Brown Reference Group plc.

Yann LeCun, Bottou Léon, Bengio Yoshua, Haffner Patrick. (1998). **Gradient-Based Learning Applied to Document Recognition.** arXiv.

Yu Sun, Yuan Liu, Guan Wang, Haiyan Zhang. (2017). **Deep Learning for Plant Identification in Natural Environment.** Hindawi.

What is API: Definition, Types, Specifications, Documentation. (2019, June 18).

Retrieved from altexsoft: <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>

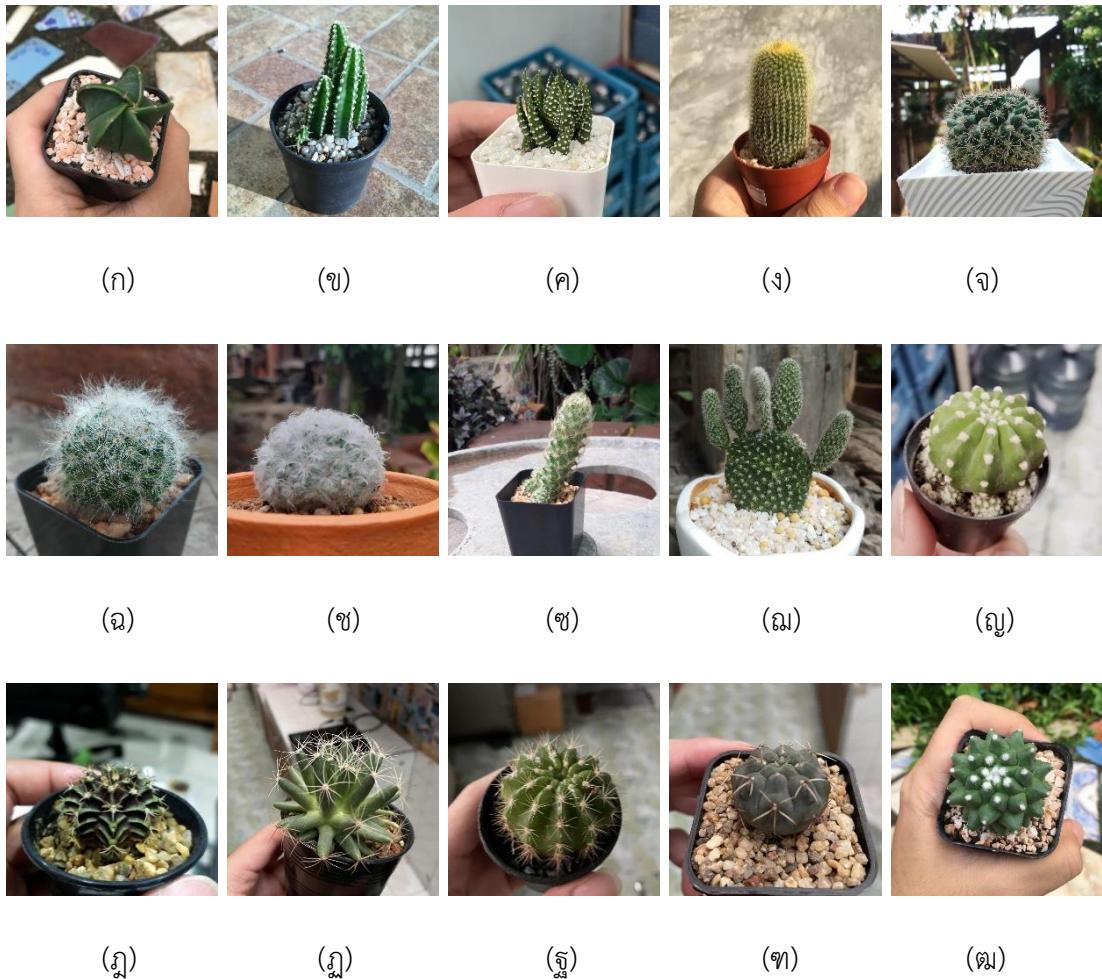
ภาคผนวก

ภาคผนวก ก.
ขั้นตอนการดำเนินการ

1. การพัฒนาโนเมเดลการเรียนรู้เชิงลึก

1.1 การเตรียมชุดข้อมูล

รูปภาพกรอบของเพชรทั้งหมด 15 สายพันธุ์นั้นถูกถ่ายภาพโดยโทรศัพท์มือถือ สายพันธุ์ละ 200 รูป รวมทั้งหมด 3000 รูป ดังรูปตัวอย่างที่ ก.1 ในสภาพแวดล้อมที่มีแสงแตกต่างกัน เช่น ถ่ายภาพในอาคาร มีแหล่งกำเนิดแสงเป็นหลอดไฟ หรือถ่ายกลางแจ้งแหล่งกำเนิดแสงเป็นดวงอาทิตย์ เป็นต้น โดยถ่ายภาพให้เห็นมุมต่าง ๆ ของต้นกระบองเพชร เช่น ด้านบน ด้านข้าง เป็นต้น ดังรูปที่ ก.1



รูปที่ ก.1 สายพันธุ์กระบองเพชรทั้งหมด

- (ก) หมวกสังฆราษฎร์ (ข) ปราสาทนางฟ้า (ค) ไก่ตีออก (ง) กระบองทอง (จ) หนามพิกุล (ฉ) แม่มลูกแมว (ช) แม่มขนนก
(ซ) นิ้วทอง (ฌ) หูมิกกี้เม้าส์ (ญ) นูปต้า (ญ) มิฮาโนะ (ญ) แม่มกล้ายหอม (ญ) ดาวล้อมเดือน (ຫ) บานียันม
(ນ) แม่มอڑุชา�



(ก)

(ข)

(ค)

(ง)

รูปที่ ก.2 ตัวอย่างภาพถ่ายต้นกระบองเพชรในสภาพแวดล้อมที่มีแสงแตกต่างกัน

(ก) ถ่ายในร่มมีแหล่งกำเนิดแสงธรรมชาติ (ข) ถ่ายในร่มแหล่งกำเนิดแสงจากหลอดไฟ (ค) ถ่ายกลางแจ้ง

(ง) ถ่ายในที่ร่มนอกอาคาร

รูปภาพกระบองเพชรที่ได้จากการถ่ายภาพทั้งหมด 3000 รูป จะถูกปรับขนาดของรูปให้ด้านกว้างมีขนาด 960 พิกเซล ด้านยาวมีขนาด 1280 พิกเซล จากนั้นจะนำรูปภาพกระบองเพชรทั้งหมดสายพันธุ์ละ 200 รูป ไปสู่สร้างรูปภาพใหม่จำนวน 300 รูป โดยใช้ Augmentor ซึ่งเป็นไลบรารีของ Python โดยกำหนดการตั้งค่าต่าง ๆ ดังรูปที่ ก.3 เพื่อทำการยืด หมุน สลับด้าน และการเพิ่มความเข้มของภาพ ดังรูปตัวอย่างที่ ก.4 จึงทำให้ได้รูปภาพกระบองเพชรทั้งหมดสายพันธุ์ละ 500 รูปทำให้รูปภาพทั้งหมดมี 7500 รูป

```
import Augmentor

while True:
    iPath = input('Input path> ')
    oPath = input('Output path> ')
    nSample = int(input('Sample> '))
    p = Augmentor.Pipeline(iPath, oPath) # temp
    p.random_contrast(0.4, 0.2, 0.5)
    p.rotate(0.75, 25, 25)
    p.random_distortion(probability=1, grid_width=4, grid_height=4, magnitude=8)
    p.flip_left_right(0.5)
    p.zoom(0.5, 1.1, 2)
    p.sample(nSample)
    print('\nDone. Run again?')
    c = input('y/n >')
    if c=='n':
        break
```

รูปที่ ก.3 การตั้งค่าสำหรับปรับแต่งรูปภาพ



(ก)

(ข)

(ค)

(ง)

รูปที่ ก.4 ตัวอย่างรูปที่ผ่านการยืด หมุน กลับด้าน และเพิ่มความเข้ม

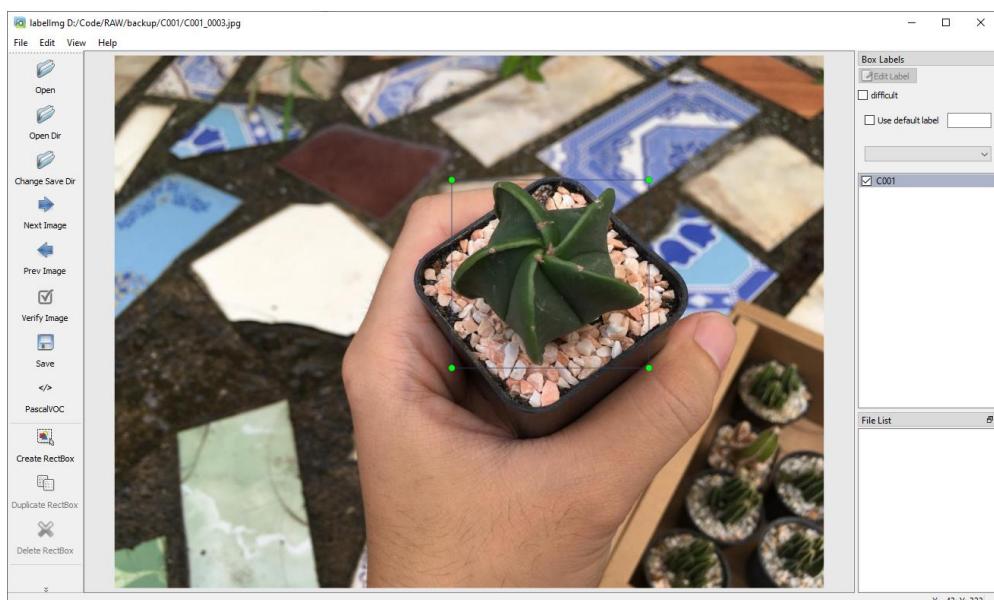
(ก) รูปที่ทำการยืด (ข) รูปที่ทำการหมุน (ค) รูปที่กลับด้าน (ง) รูปที่เพิ่มความเข้ม

รูปภาพของกระบองเพชรทั้งหมด 7500 ภาพจะถูกนำมาทำการกำกับรูปภาพ (label image) แต่ละสายพันธุ์ ซึ่งใช้ป้ายกำกับเป็นรหัสที่สร้างขึ้นสำหรับแต่ละสายพันธุ์ดังตารางที่ ก.1 โดยใช้ LabelImg ซึ่งเป็นไลบรารีของ Python ทำการกำกับรูปภาพดังรูปที่ ก.5

ตารางที่ ก.1 รหัสที่สร้างขึ้นสำหรับสายพันธุ์ต่าง ๆ

รหัส	ชื่อสายพันธุ์
C001	หมวดสังฆราชา (Astrophytum myriostigma)
C002	ปราสาทนางฟ้า (Cereus peruvianus)
C003	ไก่ต็อก (Haworthia coarctata)
C004	กระบองทอง (Echinocactus leninghausii)
C005	หนามพิกุล (Mammillaria scrippsiana)
C006	แม่มลูกแมว (Mammillaria bocasana)
C007	แม่มขนกวาง (Mammillaria plumosa)
C008	นิ่วทอง (Mammillaria Elongata)
C009	หูมิกกี้เม้าส์ (Opuntia microdasys)
C010	นูบต้า (Echinopsis subdenudata)

C011	มิชาน (Gymnocalycium mihanovichii)
C012	แม่มกล้ายหอม (Mammillaria longimamma)
C013	ดาวล้อมเดือน (Echinopsis calochalora)
C014	บานเนินนัม (Gymnocalycium baldianum)
C015	แม่มอธุรชาム (Mammillaria bucareliensis)



รูปที่ ก.5 การกำกับรูปภาพโดยใช้ LabelImg

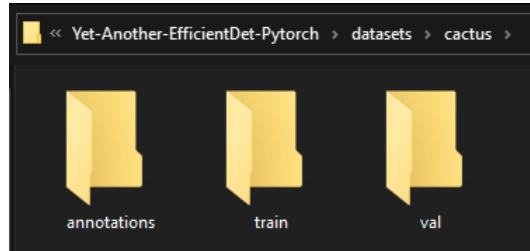
เนื่องจากว่าการเรียนรู้เชิงลึกของแต่ละโมเดลนั้นมีการใช้รูปแบบการกำกับรูปภาพที่แตกต่างกันโดยครั้งแรกของการกำกับรูปภาพจะเป็นในรูปแบบ PascalVOC จากนั้นจึงนำไฟล์กำกับรูปภาพที่มีนามสกุล .xml มาอ่านและแก้ไขไฟล์ให้อยู่ในรูปแบบของ YOLO โดยการเขียนโปรแกรมแปลงไฟล์ .xml เป็น .txt เพื่อใช้สำหรับการเรียนรู้เชิงลึกของ YOLOv5x จากนั้นทำการรวมไฟล์ .xml ให้อยู่ในรูป .csv เพื่อใช้สำหรับการเรียนรู้เชิงลึกของ Faster-RCNN Resnet-50 ถัดไปจึงนำไฟล์ .csv ไปแปลงให้เป็นให้อยู่ในรูปแบบของ COCO Dataset นั่นก็คือ .json เพื่อใช้สำหรับการเรียนรู้เชิงลึกของ EfficientDet

1.2 การเรียนรู้เชิงลึก

รูปภาพระบบของเพชรทั้งหมด 15 สายพันธุ์จะถูกนำไปเรียนรู้เชิงลึกจากโมเดลเครือข่ายประสาทเทียม 3 โมเดลที่แตกต่างกันได้แก่ EfficientDet-D5 Faster-RCNN Resnet-50 และ YOLOv5x ซึ่งทั้งสามโมเดลนั้นเป็นการเรียนรู้ต่อจากเดิม (Transfer learning) เพื่อเปรียบเทียบประสิทธิภาพของโมเดล โดยได้แบ่งข้อมูลออกเป็น 2 ส่วนได้แก่ ข้อมูลที่นำไปเรียนรู้ 80% ข้อมูลที่นำไปตรวจสอบ 20% จึงทำการเรียนรู้ใน Google Colab Pro ที่มีรายละเอียดเป็น Intel(R) Xeon(R) CPU @ 2.00GHz มีขนาด RAM ที่ 24 GB และ GPU เป็น Nvidia Tesla P100 มีขนาด RAM ที่ 16 GB

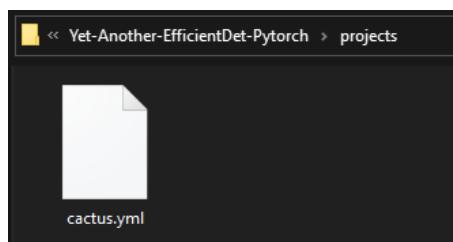
1.2.1 การเรียนรู้โมเดล EfficientDet

ทำการโคลนโค้งงาน Yet-Another-EfficientDet-Pytorch จาก GitHub จากนั้นจึงทำการอัปโหลดข้อมูลรูปภาพและการกำกับรูปภาพไปยังโฟลเดอร์ datasets ในโฟลเดอร์โค้งงาน โดยในโฟลเดอร์จะประกอบด้วย annotation สำหรับเก็บไฟล์กำกับรูปภาพ .json train สำหรับเก็บรูปภาพที่ใช้เรียนรู้ และ val สำหรับเก็บรูปภาพที่ใช้ตรวจสอบ ดังรูปภาพที่ ก.6



รูปที่ ก.6 แสดงการเตรียมข้อมูลสำหรับการเรียนรู้ของโมเดล EfficientDet

จากนั้นสร้างไฟล์ชื่อ cactus.yml ไว้ในโฟลเดอร์ projects ดังรูปที่ ก.7 โดยไฟล์จะประกอบด้วยชื่อโค้งงาน ชื่อโฟลเดอร์รูปที่ใช้เรียนรู้ ชื่อโฟลเดอร์ที่ใช้ตรวจสอบ ค่า anchors และชื่อกลุ่มที่ใช้ในการกำกับรูปภาพ ดังรูปที่ ก.8 จึงทำการเรียนรู้ใน Google Colab ดังรูปที่ ก.9



รูปที่ ก.7 แสดงการสร้างไฟล์ cactus.yml

```

project_name: cactus
train_set: train
val_set: val
num_gpus: 1

mean: [0.485, 0.456, 0.406]
std: [0.229, 0.224, 0.225]

anchors_scales: '[2 ** 0, 2 ** (1.0 / 3.0), 2 ** (2.0 / 3.0)]'
anchors_ratios: '[(1.0, 1.0), (1.4, 0.7), (0.7, 1.4)]'

obj_list: ['C001', 'C002', 'C003', 'C004', 'C005', 'C006', 'C007', 'C008', 'C009', 'C010', 'C011', 'C012',
'C013', 'C014', 'C015']

```

รูปที่ ก.8 แสดงรายละเอียดของไฟล์ cactus.yml

```

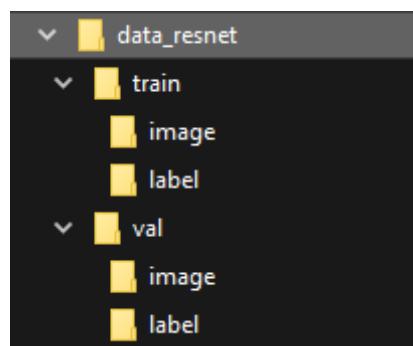
!git clone --depth 1 https://github.com/zylo117/Yet-Another-EfficientDet-Pytorch
%cd Yet-Another-EfficientDet-Pytorch
!pip install pycocotools numpy opencv-python tqdm tensorboard tensorboardX pyyaml webcolors matplotlib
! wget https://github.com/zylo117/Yet-Another-EfficientDet-Pytorch/releases/download/1.0/efficientdet-d5.pth -O weights/efficientdet-d5.pth
! python train.py -c 5 -p cactus --head_only False --lr 1e-3 --batch_size 1 --load_weights weights/efficientdet-d5.pth
--num_epochs 20 --save_interval 200

```

รูปที่ ก.9 แสดงการวิธีเรียนรู้ใน Google Colab ของ Yet-Another-EfficientDet-Pytorch

1.2.2 การเรียนรู้โมเดล Faster-RCNN Resnet-50

เตรียมไฟล์ข้อมูลสำหรับการเรียนรู้โดยอัปโหลดข้อมูลชื่อ data_resnet ซึ่งจะมีไฟล์เดอร์ train และ valid และในไฟล์เดอร์ train และ valid จะมีไฟล์เดอร์ image สำหรับเก็บรูปภาพ และไฟล์เดอร์ label สำหรับเก็บไฟล์กำกับรูปภาพ ดังรูปที่ ก.10 จากนั้นทำการติดตั้งไลบรารีของ python ชื่อ detecto และทำการเรียนรู้บน Google Colab ดังรูปที่ ก.11



รูปที่ ก.10 แสดงการเตรียมข้อมูลสำหรับการเรียนรู้ของโมเดล Faster-RCNN Resnet-50

```

!pip install detecto

from detecto import core, utils, visualize

utils.xml_to_csv('train/label/', 'train_labels.csv')
utils.xml_to_csv('val/label/', 'val_labels.csv')

name = ['C001', 'C002', 'C003', 'C004', 'C005', 'C006', 'C007', 'C008', 'C009', 'C010', 'C011', 'C012', 'C013',
'C014', 'C015']
model = core.Model.load(name)
dataset = core.Dataset('train_labels.csv', 'train/image/')
loader = core.DataLoader(dataset, batch_size=4)
val_dataset = core.Dataset('val_labels.csv', 'val/image/')
model.fit(loader, val_dataset, epochs=20, learning_rate=0.001, verbose=True)
model.save("model_resnet_50_e50.pth")

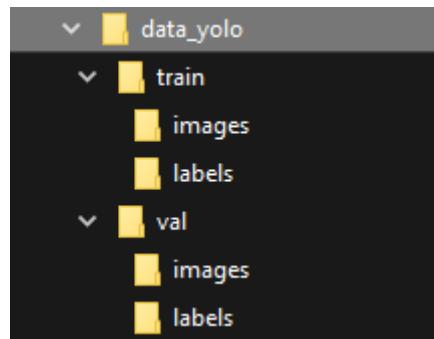
```

รูปที่ ก.11 แสดงการวิธีเรียนรู้ใน Google Colab ของ Detecto

1.2.3 การเรียนรู้โมเดล YOLOv5x

ทำการโคลนโค้ดงาน yolov5 จาก GitHub จากนั้นจึงทำการอัปโหลดข้อมูล

รูปภาพและการกำกับรูปภาพไปยังโฟลเดอร์ data_yolo ในโฟลเดอร์โค้ดงานโดยใน data_yolo จะประกอบด้วย train และ val ซึ่งในทั้งสองโฟลเดอร์จะประกอบด้วย images สำหรับเก็บรูปภาพ และ labels สำหรับเก็บไฟล์การกำกับรูปภาพดังรูปที่ ก.12



รูปที่ ก.12 แสดงการเตรียมข้อมูลสำหรับการเรียนรู้ของโมเดล YOLOv5

สร้างไฟล์ data.yaml สำหรับตั้งค่าต่าง ๆ ของโค้ดงานโดยมีรายละเอียดคือ
ที่อยู่ของข้อมูลสำหรับเรียนรู้ ที่อยู่ของข้อมูลสำหรับตรวจสอบ จำนวนข้อมูลที่กำกับ และชื่อกลุ่มที่ใช้
ในการกำกับรูปภาพ ดังรูปที่ ก.13 จากนั้นจึงทำการเรียนรู้ใน Google Colab ดังรูปที่ ก.14

```
train: ./data_yolo/train/images
val: ./data_yolo/valid/images

nc: 15
names: ['C001', 'C002', 'C003', 'C004', 'C005', 'C006', 'C007', 'C008', 'C009', 'C010', 'C011',
'C012', 'C013', 'C014', 'C015']
```

รูปที่ ก.13 แสดงรายละเอียดไฟล์ data.yaml

```
!git clone https://github.com/ultralytics/yolov5
!pip install -qr yolov5/requirements.txt

%cd yolov5

!python train.py --img 640 --batch 16 --epochs 50 --data './data.yaml' --cfg ./models/yolov5x.yaml --weights ''
--name yolov5x_results --cache
```

รูปที่ ก.14 แสดงการวิเครียนรูปใน Google Colab ของ YOLOv5

ประวัติผู้ดำเนินโครงการ



ชื่อ นายกฤษณพงษ์ ปันติ
ภูมิลำเนา เลขที่ 143/2 หมู่ 1 ตำบลหนองม่วงไข่ อำเภอ
หนองม่วงไข่ จังหวัดแพร่
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิริยาลัย
จังหวัดแพร่
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: gritsanapongp60@nu.ac.th



ชื่อ นายเจษฎา ดวงปัญญาสว่าง
ภูมิลำเนา เลขที่ 79-81-83 ถนนเจริญสุข ตำบลในเมือง
อำเภอเมืองกำแพงเพชร จังหวัดกำแพงเพชร
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนวชิรวิทยา
จังหวัดกำแพงเพชร
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: jessadad60@nu.ac.th



ชื่อ นายวิษณุ พล อุ่นpath
ภูมิลำเนา เลขที่ 313/4 หมู่ 11 ตำบลลังทอง อำเภอวังทอง
จังหวัดพิษณุโลก

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนเซนต์นิโกลาส

จังหวัดพิษณุโลก

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: wasitphonu60@nu.ac.th