การจำแนกสายพันธุ์ถั่วพิสตาชิโอ โดยใช้เทคนิคการเรียนรู้เชิงลึก

CLASSIFICATION OF PISTACHIO SPECIES USING DEEP LEARNING TECHNIQUES

ณิชารีย์ ศรีบูจันดี

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2566

การจำแนกสายพันธุ์ถั่วพิสตาชิโอ โดยใช้เทคนิคการเรียนรู้เชิงลึก

ณิชารีย์ ศรีบูจันดี

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2566

CLASSIFICATION OF PISTACHIO SPECIES USING DEEP LEARNING TECHNIQUES

NICHAREE SRIBUCHANDEE

A Master's Project Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2023

สารนิพนธ์

เรื่อง

การจำแนกสายพันธุ์ถั่วพิสตาชิโอ โดยใช้เทคนิคการเรียนรู้เชิงลึก

ของ

ณิชารีย์ ศรีบูจันดี

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(อาจารย์ ดร.เรืองศักดิ์ ตระกูลพุทธิรักษ์)

..... ประธาน

(อาจารย์ ดร.สุทธิพงศ์ ธัชยพงษ์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริสรรพ เหล่าหะเกียรติ)

ชื่อเรื่อง	การจำแนกสายพันธุ์ถั่วพิสตาชิโอ โดยใช้เทคนิคการเรียนรู้เชิงลึก
ผู้วิจัย	ณิชารีย์ ศรีบูจันดี
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2566
อาจารย์ที่ปรึกษา	อาจารย์ ดร. เรืองศักดิ์ ตระกูลพุทธิรักษ์

ปัจจุบันการนำเข้าถั่วพิสตาชิโอในไทย ซึ่งมูลค่าของถั่วพิสตาชิโอนั้นจะแตกต่างกันตามแต่ละสายพันธุ์ จึงจำเป็นต้องมีวิธีการและเทคโนโลยีใหม่ ๆ ในการจำแนกภาพสายพันธุ์พิสตาชิโอต่าง ๆ มูลค่าของถั่วพิสตาชิโอแต่ละสายพันธุ์ไม่เท่ากัน ขึ้นอยู่กับคุณภาพ ลักษณะทางกายภาพ และความนิยมในตลาด โดยสายพันธุ์ที่มีคุณภาพสูง จะมีความต้องการสูงในตลาด การพัฒนาวิธีการและเทคโนโลยีในการจำแนกสายพันธุ์จึงมีความสำคัญ เพื่อรักษามาตรฐานคุณภาพและมูลค่าของผลิตภัณฑ์ งานวิจัยนี้มีวัตถุประสงค์เพื่อการศึกษาการจำแนกสายพันธุ์ของถั่วพิสตาชิโอตามลักษณะทางกายภาพ โดยใช้ข้อมูลภาพสายพันธุ์ Kirmizi และ Siirt จากแหล่งข้อมูลสาธารณะ และเก็บข้อมูลภาพสายพันธุ์ Kerman เพิ่มเติม ซึ่งแบ่งออกเป็นข้อมูลภาพ Kirmizi จำนวน 1,232 รูป และ Siirt จำนวน 916 รูป และ Kerman จำนวน 1,056 รูป ในงานวิจัยนี้ศึกษาแบบจำลองการเรียนรู้เชิงลึกทั้งหมด 5 ประเภท ได้แก่ แบบจำลองที่มีการเรียนรู้ตั้งแต่เริ่มต้น (CNN from Scratch), แบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-Trained Model) VGG16, VGG19, ResNet50 และ EfficientNetB0 โดยใช้ถ่ายทอดจากข้อมูลภาพ ImageNet รวมถึงการปรับแต่งแบบจำลอง (Fine Tuning) และปรับค่าพารามิเตอร์ต่าง ๆ (Hyperparameters) เพื่อให้ได้แบบจำลองที่มีประสิทธิภาพสูงที่สุด จากผลการศึกษาพบว่าแบบจำลอง ResNet50 เป็นแบบจำลองที่มีประสิทธิภาพสูงสุด โดยมีค่าความถูกต้องอยู่ที่ 0.9812 รองมาเป็นแบบจำลอง VGG16 VGG19 EfficientNetB0 และ CNN from Scratch ตามลำดับ ซึ่งมีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9781, 0.9750, 0.9625 และ 0.9219 ตามลำดับ สำหรับการวิจัยในอนาคต มีข้อเสนอในการเพิ่มจำนวนข้อมูลภาพและการปรับแต่งแบบจำลอง (Tuning Hyperparameters) โดยใช้เทคนิคต่าง ๆ เช่น Grid Search หรือ Bayesian Optimization เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมที่สุดสำหรับแบบจำลอง

คำสำคัญ : ถั่วพิสตาชิโอ, การจำแนกภาพ, การเรียนรู้เชิงลึก, แบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อน

Title	CLASSIFICATION OF PISTACHIO SPECIES USING DEEP LEARNING TECHNIQUES
Author	NICHAREE SRIBUCHANDEE
Degree	MASTER OF SCIENCE
Academic Year	2023
Thesis Advisor	Lecturer Ruangsak Trakunphutthirak , Ph.D.

Nowadays, the consistent and growing trend in the consumption of pistachios necessitates the development of new methods and technologies for image classification of different pistachio varieties. The value of each pistachio variety varies, depending on the quality, physical characteristics, and market popularity. Pistachios, particularly certain varieties, are considered high-quality and expensive due to their high demand in the market. The development of methods and technologies for species classification is crucial in maintaining product quality standards and value. The objective of this research is to develop a deep learning model for classifying pistachio cultivars based on physical characteristics such as shell color, size, and shape, using image data of the 1,232 images of Kirmizi and 916 images of Siirt, from public sources, along with additional image data of the 1,056 images of Kerman which were collected by the researcher and the five types of models were employed: a basic Convolutional Neural Network (CNN) model trained from scratch, and four pre-trained models: VGG16, VGG19, ResNet50, and EfficientNetB0, each transferring weights from ImageNet. Fine-tuning and hyperparameter adjustments were made to achieve optimal efficiency. The results indicated that the ResNet50 model had the highest efficiency, with an accuracy of 0.9812, followed by the VGG16, VGG19, EfficientNetB0, and CNN from Scratch models, in that order, with accuracies of 0.9781, 0.9750, 0.9625, and 0.9219, respectively. For future research, there are suggestions to increase the number of image data and to fine-tune the models (Hyperparameter Tuning) using various techniques such as Grid Search or Bayesian Optimization to find the most suitable parameters for the model.

Keyword : Pistachio, Image classification, Deep learning, Pre-trained model

กิตติกรรมประกาศ

การวิจัยฉบับนี้สำเร็จลุล่วงได้ดีด้วยการสนับสนุนและคำแนะนำจากอาจารย์ดร. เรืองศักดิ์ ตระกูลพุทธิรักษ์ และคณาจารย์จากหลักสูตรวิทยาการข้อมูล ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ รวมทั้งการสนับสนุนจากบัณฑิตวิทยาลัยของมหาวิทยาลัยที่ทำให้การนำเสนอผลงานวิจัยนี้เป็นไปได้อย่างราบรื่น ผู้วิจัยขอแสดงความขอบคุณอย่างสูงต่อทุกท่านที่มีส่วนร่วมให้ความรู้ คำแนะนำ และการสนับสนุนตลอดการทำวิจัยฉบับนี้

ณิชารีย์ ศรีบูจันดี

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาของงานวิจัย

ถั่วพิสตาชิโอ (Pistachio) มีชื่อวิทยาศาสตร์ว่า Pistacia Vera เป็นไม้ยืนต้นในวงศ์ Anacardiaceae (กลุ่มเดียวกับมะม่วง และมะม่วงหิมพานต์) มีลักษณะเป็นไม้ยืนต้นสูงประมาณ 5-10 เมตร ใบเป็นใบประกอบแบบขนนก มีใบย่อยประมาณ 5-9 ใบ ดอกมีขนาดเล็ก สีขาวหรือสีเหลือง ออกเป็นช่อตามซอกใบ และผลเป็นฝักแบนยาวประมาณ 5-10 เซนติเมตร ภายในฝักมีเมล็ดเดี่ยว โดยผลดิบจะมีสีเขียว และเมื่อสุกผลจะกลายเป็นสีแดงหรือสีเหลือง (Wikipedia, 2024) ดังภาพประกอบ 1 จากนั้นผลจะแตกออกทำให้เห็นเมล็ดสีครีมภายในผลซึ่งเป็นเปลือกแข็งของถั่วพิสตาชิโอ (Shell) และเมื่อเปลือกแข็งเปิดออกจะเห็นเปลือกหุ้มเมล็ดถั่วพิสตาชิโอ (Seed Coat) หุ้มเนื้อถั่วพิสตาชิโอ (Kernel) ที่อยู่ข้างใน (Biodels, 2021) ดังภาพประกอบ 2

ภาพประกอบ 1 แสดงลักษณะลำต้น ใบ ดอก และผลของถั่วพิสตาชิโอ

ที่มา: https://www.gardenia.net/plant/pistacia-vera

ภาพประกอบ 2 แสดงส่วนประกอบทางกายภาพของถั่วพิสตาชิโอ

ที่มา: (Biodels, 2021)

ถั่วพิสตาชิโอเป็นพืชตระกูลถั่วชนิดหนึ่งที่ได้รับความนิยมทั่วโลก มีคุณค่าทางโภชนาการสูง โดยเนื้อถั่วพิสตาชิโอจะมีแหล่งไขมันดีเป็นกรดไขมันไม่อิ่มตัว (Unsaturated Fatty Acid) ประมาณ 50-60% ซึ่งเป็นสารอาหารจำเป็นสำหรับโภชนาการของมนุษย์ อีกทั้งยังเป็นสินค้าเกษตรที่มีราคาสูง (Wikipedia, 2024)

ตาราง 1 การผลิตถั่วพิสตาชิโอทั่วโลกในปี พ.ศ. 2564

ประเทศ	ปริมาณผลผลิต (ตัน)	ร้อยละของผลผลิต
สหรัฐอเมริกา	523,900	57
อิหร่าน	135,000	15
ตุรกี	119,355	13
จีน	78,818	9
ซีเรีย	43,104	5
อื่น ๆ	15,541	1
รวม (ทั่วโลก)	**915,718**	**100**

ที่มา: (Wikipedia, 2024)

จากตาราง 1 แสดงการผลิตถั่วพิสตาชิโอทั่วโลกในปี พ.ศ. 2564 โดยมีผลผลิตประมาณ 0.9 ล้านตัน โดยสหรัฐอเมริกาและอิหร่านเป็นประเทศผู้ผลิตรายใหญ่สุด คิดรวมกันเป็นร้อยละ 72 ของผลผลิตทั้งหมด รองลงมาคือตุรกี จีน และซีเรีย ตามลำดับ

ในงานวิจัยนี้ได้ศึกษาการจำแนกสายพันธุ์ถั่วพิสตาชิโอทั้งหมด 3 สายพันธุ์ ดังตาราง 2 และมีรายละเอียด ดังนี้

1. ถั่วพิสตาชิโอสายพันธุ์ Kirmizi เป็นสายพันธุ์ถั่วพิสตาชิโอที่ได้รับความนิยมมากที่สุดในตุรกี มีลักษณะเด่นคือมีเปลือกสีแดง เมล็ดสีเขียวเข้มหรือสีเขียวอมม่วง (Solo, 2022b) ดังภาพประกอบ 3 โดยมีลักษณะทางกายภาพ ดังนี้

- รูปร่างเปลือกแข็ง: มีลักษณะเล็กและเรียวยาว ความยาวประมาณ 2-3 เซนติเมตร ความกว้างประมาณ 1-1.5 เซนติเมตร
- เปลือกหุ้มเมล็ด: มีสีแดงสด และกะเทาะออกได้ง่าย
- เนื้อ: มีเนื้อสีเขียวเข้มหรือสีเขียวอมม่วง

ภาพประกอบ 3 แสดงถั่วพิสตาชิโอ สายพันธุ์ Kirmizi

ที่มา: (Solo, 2022b)

2. ถั่วพิสตาชิโอสายพันธุ์ Siirt เป็นสายพันธุ์ถั่วพิสตาชิโอที่มีต้นกำเนิดมาจากตุรกี มีลักษณะเด่นคือมีเปลือกสีแดงอมชมพู เมล็ดสีเขียวอมน้ำตาล (Solo, 2022b) ดังภาพประกอบ 4 โดยมีลักษณะทางกายภาพ ดังนี้

- รูปร่างเปลือกแข็ง: มีลักษณะค่อนข้างกลม ความยาวประมาณ 2-3เซนติเมตร ความกว้างประมาณ 1-1.5 เซนติเมตร แต่มีลักษณะกลมกว่าถั่วพิสตาชิโอสายพันธุ์ Kirmizi
- เปลือกหุ้มเมล็ด: มีสีแดงอมชมพู และกะเทาะออกได้ง่าย
- เนื้อ: มีเนื้อสีเขียวอมน้ำตาล

ภาพประกอบ 4 แสดงถั่วพิสตาชิโอ สายพันธุ์ Siirt

ที่มา: (Solo, 2022b)

3. ถั่วพิสตาชิโอสายพันธุ์ Kerman เป็นสายพันธุ์ที่มีการผลิตสูงสุดในสหรัฐอเมริกา มีลักษณะเด่นคือมีเปลือกสีแดงอมน้ำตาล เมล็ดสีเขียวอมน้ำตาล (Solo, 2022a) ดังภาพประกอบ 5 โดยมีลักษณะทางกายภาพ ดังนี้

- รูปร่างเปลือกแข็ง: มีลักษณะเล็กและเรียวยาว ความยาวประมาณ 2-2.5 เซนติเมตร ความกว้างประมาณ 1-1.5 เซนติเมตร
- เปลือกหุ้มเมล็ด: มีสีแดงอมน้ำตาล และกะเทาะออกได้ง่าย
- เนื้อ: มีเนื้อสีเขียวอ่อน

ภาพประกอบ 5 แสดงถั่วพิสตาชิโอ สายพันธุ์ Kerman

ที่มา: (Solo, 2022a)

ตาราง 2 สรุปลักษณะทางกายภาพของถั่วพิสตาชิโอ สายพันธุ์ Kirmizi, Siirt และ Kerman

ลักษณะทางกายภาพ	Kirmizi	Siirt	Kerman
สีของเปลือกหุ้มเมล็ด	สีแดงสด	สีแดงอมชมพู	สีแดงอมน้ำตาล
สีของเนื้อ	สีเขียวเข้มหรือสีเขียวอมม่วง	สีเขียวอมน้ำตาล	สีเขียวอ่อน
ขนาดความยาวเมล็ด	2-3 เซนติเมตร	2-3 เซนติเมตร	1.5-2.5 เซนติเมตร
รูปร่างเปลือกแข็ง	เล็กและเรียวยาว	เล็กและเรียวยาว	ค่อนข้างกลม

1.2 **ความมุ่งหมายของงานวิจัย**

ในการวิจัยนี้ผู้วิจัยได้ตั้งความมุ่งหมายไว้ดังนี้

1. เพื่อศึกษาแบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-Trained Model) โดยใช้วิธีการถ่ายโอนการเรียนรู้ (Transfer Learning) และแบบจำลองที่มีการเรียนรู้ตั้งแต่เริ่มต้น (CNN from Scratch)

2. เพื่อสร้างแบบจำลองสำหรับการจำแนกสายพันธุ์ถั่วพิสตาชิโอ 3 สายพันธุ์ (Kirmizi, Siirt และ Kerman) โดยใช้ข้อมูลภาพ

3. เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองที่สร้างขึ้นบนพื้นฐานโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) โดยใช้โครงสร้างที่แตกต่างกัน รวมทั้งการปรับค่าพารามิเตอร์ต่าง (Fine-tuning Hyperparameters) เพื่อให้ได้แบบจำลองที่มีประสิทธิภาพสูงสุด

1.3 **ความสำคัญของงานวิจัย**

ถั่วพิสตาชิโอนั้นมีหลากหลายสายพันธุ์ ซึ่งมีลักษณะแตกต่างกัน การจำแนกสายพันธุ์ของถั่วพิสตาชิโอจึงมีความสำคัญต่อการควบคุมคุณภาพการเกษตรและงานวิจัยทางวิทยาศาสตร์ วิธีการจำแนกสายพันธุ์แบบดั้งเดิมยังมีความผิดพลาดเนื่องจากต้องอาศัยแรงงานมนุษย์ งานวิจัย

นี้จึงมุ่งหวังที่จะใช้ระบบประมวลผลอัตโนมัติด้วยการเรียนรู้เชิงลึก (Deep Learning) โดยเป็นแบบจำลองการเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-Trained Model) โดยใช้วิธีการถ่ายโอนการเรียนรู้ (Transfer Learning) เพื่อจำแนกและวิเคราะห์สายพันธุ์ของถั่วพิสตาชิโอตามลักษณะทางภาพ เช่น สีของเปลือก ขนาด และรูปร่าง ได้อย่างรวดเร็วและมีประสิทธิภาพ การนำเทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) มาใช้ในการจำแนกสายพันธุ์ถั่วพิสตาชิโอจึงมีแนวโน้มที่จะได้รับความนิยมเพิ่มขึ้น เนื่องจากสามารถเรียนรู้จากข้อมูลจำนวนมากและสามารถระบุรูปแบบที่ซับซ้อนได้ ซึ่งจะช่วยเพิ่มประสิทธิภาพและลดข้อผิดพลาดในการจำแนกสายพันธุ์ถั่วพิสตาชิโอได้

1.4 ขอบเขตงานวิจัย

ผู้วิจัยได้กำหนดขอบเขตของงานวิจัยไว้ดังต่อไปนี้

1. ข้อมูลที่ใช้ในงานวิจัยนี้มีเป็นข้อมูลภาพที่แบ่งออกเป็น 3 สายพันธุ์ ได้แก่ Kirmizi, Siirt และ Kerman โดยเป็นข้อมูลภาพที่ได้จาก Public Dataset จากงานวิจัยของ D. Singh (Singh และคณะ, 2022) ซึ่งเป็นถั่วพิสตาชิโอจำนวน 2 สายพันธุ์ คือ Kirmizi และ Siirt อีกทั้งทางผู้วิจัยได้เก็บข้อมูลภาพถั่วพิสตาชิโอ สายพันธุ์ Kerman เพิ่มเติมอีกด้วย

2. ข้อมูลภาพที่ใช้เป็นแบบมีป้ายกำกับ (Labeling) หรือ แบบจำลองการเรียนรู้แบบ Supervised Learning รวมทั้งเป็นข้อมูลภาพแบบ Balanced Dataset

3. แบบจำลองที่ใช้ศึกษามีทั้งหมด 5 ประเภท เป็นแบบจำลองพื้นฐานโครงข่ายประสาทเทียมแบบคอนโวลูชันที่มีการเรียนรู้ตั้งแต่เริ่มต้น (CNN from Scratch Model) และแบบจำลองที่มีการเรียนรู้แบบถ่ายโอนการเรียนรู้ (Pre-Trained Model) อีก 4 ประเภท ได้แก่ VGG16, VGG19, ResNet50 และ EfficientNetB0 โดยถ่ายทอดจากข้อมูลภาพ ImageNet Weight และใช้ภาษา Python รวมทั้งใช้ TensorFlow และ Keras Library ด้วย

4. ปรับค่าพารามิเตอร์ต่าง ๆ (Hyperparameters) เพื่อให้ได้แบบจำลองที่มีประสิทธิภาพสูงที่สุด

5. เปรียบเทียบประสิทธิภาพของแบบจำลอง โดยใช้การประเมินแบบจำลอง (Model Evaluation) จากการแบ่งหมวดหมู่ (Confusion Matrix) และวัดประสิทธิภาพด้วย Performance Metrics ได้แก่ ค่าความถูกต้อง (Accuracy), ค่าความแม่นยำ (Precision), ค่าความไว (Recall) และค่า F-1 Score

1.5 กรอบแนวคิดงานวิจัย

งานวิจัยนี้ศึกษาและนำเสนอเครื่องมือที่ช่วยในการจำแนกสายพันธุ์ถั่วพิสตาชิโอ ตามลักษณะทางกายภาพ ด้วยการจำแนกภาพ (Image Classification) โดยการใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) เพื่อสร้างแบบจำลองด้วยโครงสร้างต่าง ๆ แล้วนำมาเปรียบเทียบประสิทธิภาพ รวมถึงการปรับค่าพารามิเตอร์ต่าง ๆ (Hyperparameters) เพื่อให้ได้แบบจำลองที่มีความแม่นยำและมีประสิทธิภาพมากขึ้น ที่สามารถเป็นเครื่องมือที่สำคัญสำหรับผู้ผลิต ผู้แปรรูป หรือนักวิจัยได้

บทที่ 2
ทบทวนวรรณกรรม

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อต่อไปนี้

1. การจำแนกประเภทภาพ (Image Classification)
2. โครงข่ายประสาทเทียม (Artificial Neural Network)
    - โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network)
    - การเรียนรู้เชิงลึก (Deep Learning)
    - การเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-trained Deep Learning)
3. งานวิจัยที่เกี่ยวข้อง

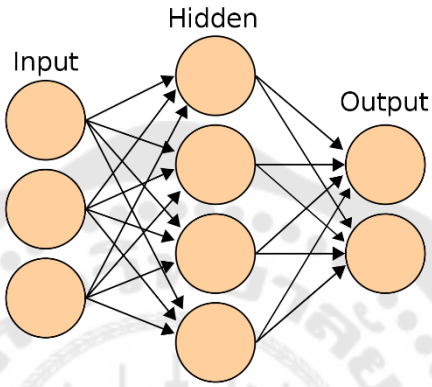
2.1 การจำแนกประเภทภาพ (Image Classification)

การจำแนกประเภทภาพ (Image Classification) เป็นงานคอมพิวเตอร์วิทัศน์ (Computer Vision) ประเภทหนึ่งที่ใช้เพื่อจำแนกวัตถุหรือคุณสมบัติต่าง ๆ ในภาพออกเป็นหมวดหมู่ต่าง ๆ โดยปกติแล้ว แต่ละหมวดหมู่จะแสดงถึงวัตถุหรือคุณสมบัติที่แตกต่างกัน เช่น สุนัข แมว ต้นไม้ รถยนต์ กล้วย ใบไม้ เป็นต้น โดยใช้การเรียนรู้ของเครื่อง (Machine Learning) จะเรียนรู้รูปแบบจากข้อมูลการฝึก (Training Set) ซึ่งสามารถเรียนรู้ทั้งได้แบบ Supervised Learning ข้อมูลภาพที่มีป้ายกำกับ (Label) หรือ Unsupervised Learning ข้อมูลภาพที่ไม่มีป้ายกำกับ (No Label) ก็ได้

สำหรับงานวิจัยนี้การจำแนกประเภทภาพโดยใช้การเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-trained Deep Learning) และเป็นโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network) และเป็นข้อมูลภาพแบบ Supervised Learning

2.2 **โครงข่ายประสาทเทียม** (Artificial Neural Network)

แนวคิดเรื่องโครงข่ายประสาทเทียม เริ่มจากโครงสร้างพื้นฐานของแบบจำลอง Perceptron โดย McCulloch และ Pitts ที่เกิดขึ้นในปีค.ศ.1943 ซึ่งเป็นหนึ่งในโครงสร้างพื้นฐานของโครงข่ายประสาทเทียม (Jaroensri และ Srimontrinond, 2566)



ภาพประกอบ 6 แสดงโครงสร้างพื้นฐานของแบบจำลอง Perceptron

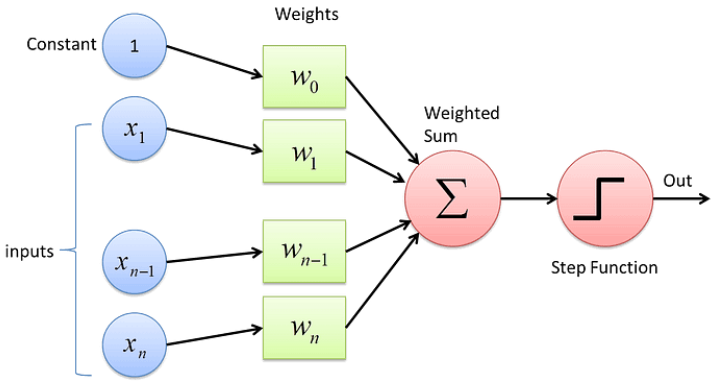
ที่มา: (Wikipedia, 2566)

จากภาพประกอบ 6 แสดงแบบจำลอง Perceptron มีหน่วยประมวลผล (Neuron) ประกอบด้วย 3 ส่วน ได้แก่ ชั้น Input, ชั้น Hidden ที่สามารถมีได้หลาย ๆ ชั้น และชั้น Output ซึ่งมีรายละเอียดแต่ละชั้น (Wikipedia, 2566) ดังนี้

1. Input Layer: เป็นชั้นที่หน่วยประมวลผล (Neuron) นำเข้าข้อมูลมาและส่งข้อมูลไปยังชั้นถัดไป โดยข้อมูลที่นำเข้าสามารถเป็นข้อมูลที่มีโครงสร้างหรือไม่มีโครงสร้างก็ได้ เช่น ภาพ, เสียง หรือตัวเลข เป็นต้น

2. Hidden Layers: เป็นชั้นประมวลผลหลักของโครงข่าย สามารถประกอบด้วยชั้น Hidden หลายชั้น (Multi-layer) และแต่ละชั้นจะประมวลผลข้อมูลจากชั้นก่อนหน้าและส่งผลลัพธ์ไปยังชั้นถัดไป ซึ่งช่วยให้โครงข่ายมีความซับซ้อนและสามารถเรียนรู้ลักษณะเฉพาะของข้อมูลได้ดีขึ้น

3. Output Layer: ชั้นสุดท้ายของโครงข่ายที่ประมวลผลข้อมูลจากชั้น Hidden สุดท้าย และส่งผลลัพธ์สุดท้ายออกมา โดยผลลัพธ์นี้ในงานวิจัยนี้เป็นการจำแนกประเภท



ภาพประกอบ 7 แสดงโครงสร้างแบบจำลองเพอร์เซปตรอน (Perceptron)
ที่มา : https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53

จากภาพประกอบ 7 เป็นโครงสร้างแบบจำลองเพอร์เซปตรอน (Perceptron) อย่างง่ายที่มีชั้น Hidden เพียง 1 ชั้น โดยกำหนดหน่วยประมวลผล (Neuron) ที่เป็นข้อมูลที่ป้อนเข้ามา (Input) และ w เป็นเวกเตอร์น้ำหนัก และค่าคงที่เท่ากับ 1 เป็นค่าไบแอส (Bias) ที่แบบจำลองจะเรียนรู้ขึ้นมาในระหว่างการฝึกฝน โดยจะคำนวณแบบ Dot Product ระหว่างค่า x และ w แล้วบวกค่า b ที่เป็นค่าไบแอส จากนั้นจะนำค่าที่ได้เข้าฟังก์ที่ไม่เป็นเส้น (σ) เช่น Sigmoid หรือ ReLU เป็นต้น และเมื่อชั้นหนึ่งคำนวณค่า f(x) ได้แล้วก็จะส่งต่อค่า f(x) เข้าสู่ชั้นถัดไปเรื่อย ๆ จนถึงชั้นสุดท้ายที่เป็นชั้น Output โดยแสดงฟังก์ชันของ x (Input), w (Weight) และ b (Bias) ในแต่ละชั้น ดังสมการที่ 2.1 (Jaroensri และ Srimontrinond, 2566)

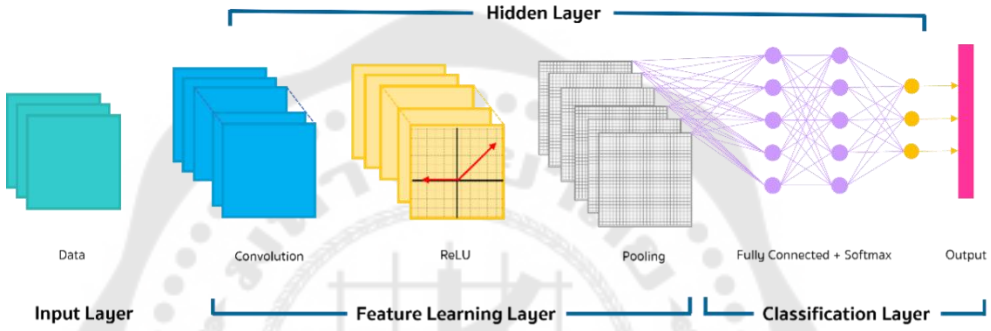
$$f(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) \tag{2.1}$$

2.2.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network)

Convolution Neural Network (CNN) คือ ระบบประมวลผลข้อมูลที่ใช้ในการแยกภาพและข้อมูลที่มีโครงสร้าง เช่น รูปภาพ โดยนิยามความสัมพันธ์ของข้อมูลด้วยการใช้คอนโวลูชัน (Convolution) ที่ช่วยในการหาคุณลักษณะหลักของข้อมูล (Feature Map) และลดมิติข้อมูลลง เช่น ในการตรวจจับลายนิ้วมือ หรือจดหมายบนรูปภาพ เป็นต้น นอกจากนี้ CNN ยังมีการใช้งาน

ในงานที่เกี่ยวข้องกับการประมวลผลข้อมูลทางเสียงและวิดีโอ เช่น การจดจำเสียงพูดหรือการค้นหาวิดีโอในเนื้อหาทางอินเทอร์เน็ตอีกด้วย

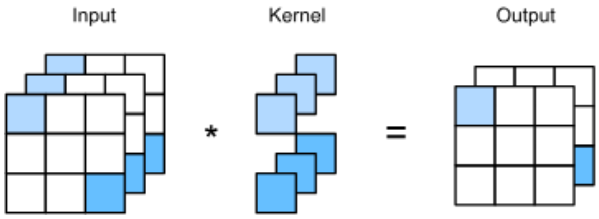
แบบจำลอง CNN มีการสร้างโครงข่ายของชั้นที่ประมวลผลข้อมูลโดยการคอนโวลูชัน และการกำหนดค่าน้ำหนัก (Weight) ในแต่ละชั้นเพื่อจำลองลักษณะพิเศษของข้อมูล รวมถึงมีชั้นที่ช่วยลดขนาดข้อมูลด้วยการสกัดข้อมูลสำคัญ (Feature Learning Layers) และใช้ในการจำแนกแยกหมวดหมู่ข้อมูล (รัสรินทร์ เมธาเฉลิมพัฒน์, 2565) ดังภาพประกอบ 8 โดยแบ่งตามลักษณะของชั้นออกเป็น 5 ประเภท ได้แก่



ภาพประกอบ 8 แสดงโครงสร้างของ Convolution Neural Network (CNN)

ที่มา: (รัสรินทร์ เมธาเฉลิมพัฒน์, 2565)

1. Convolutional Layer เป็นชั้นที่มีการทำ Convolution โดยการคำนวณทางคณิตศาสตร์แบบ Dot Product ของพื้นที่ส่วนย่อยของภาพ (Input) กับเคอร์นอล (Kernal) โดยเคอร์นอลคือเมทริกซ์ขนาดเล็กที่มีขนาดเท่ากับพื้นที่ส่วนย่อยของภาพ ทำหน้าที่เป็นตัวกรองที่จะดึงเอาคุณลักษณะเฉพาะบางอย่างออกมา เช่น ขอบ เส้น ความคมชัด เป็นต้น เพื่อให้ได้ Output ที่เป็นคุณลักษณะหลักของข้อมูล (Feature Map) ออกมา ดังภาพประกอบ 9



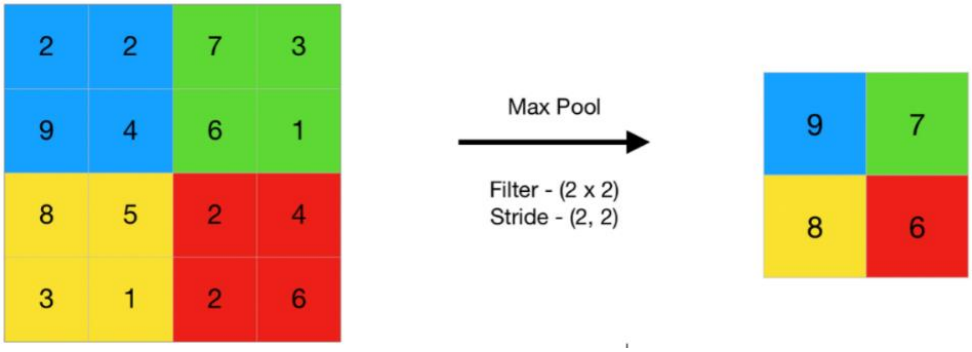
ภาพประกอบ 9 แสดงการทำ Convolution

ที่มา: https://d2l.ai/chapter_convolutional-neural-networks/channels.html

2. ReLU Layer (Rectified Linear Unit Layer) เป็นชั้นที่ทำหน้าที่เป็นฟังก์ชันการกระทำ (Activation Function) ซึ่งเป็นความสัมพันธ์ที่ไม่เป็นเชิงเส้น (Nonlinear) และช่วยในการแก้ปัญหา Gradient Vanishing ในระหว่างฝึกฝนข้อมูล Gradient ของ Activation Function ค่อย ๆ ลดลงจนเป็นศูนย์ และเมื่อแบบจำลองถูกปรับแต่งไปเรื่อย ๆ จะทำให้แบบจำลองไม่สามารถเรียนรู้ความสัมพันธ์ที่ซับซ้อนได้ โดยสมการของ ReLU Function แสดงดังสมการที่ 2.2 (รัสรินทร์ เมธาเฉลิมพัฒน์, 2565)

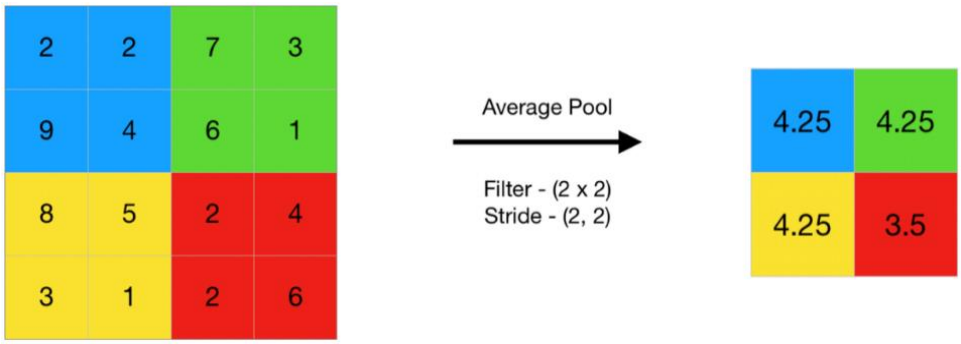
$$\mathrm{ReLU}(x) = \max(0, x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \tag{2.2}$$

3. Pooling Layer เป็นชั้นที่ทำหน้าที่ลดขนาดของ Feature Map ที่ถูกสร้างขึ้นจากชั้น Convolution Layer ก่อนหน้าซึ่งจะมีขนาดใหญ่และซับซ้อน เพื่อลดความซับซ้อนของแบบจำลองและไม่ให้แบบจำลองมีความจำเพาะกับข้อมูลที่ใช้ฝึกฝนมากเกินไป และไม่สามารถปรับตัวกับข้อมูลที่ไม่เคยเห็นมาก่อนได้ (Overfitting) พารามิเตอร์ที่ต้องกำหนด (Hyperparameters) คือ สเต็ปการสแกน (Stride) และ ขนาดของ Pool โดย Pooling Layer แบ่งได้เป็น 2 ประเภท ได้แก่ Max Pooling (Output คือค่าสูงสุด) ดังภาพประกอบ 10 และ Average Pooling (Output คือค่าเฉลี่ย) ดังภาพประกอบ 11



ภาพประกอบ 10 แสดงการทำ Max Pooling

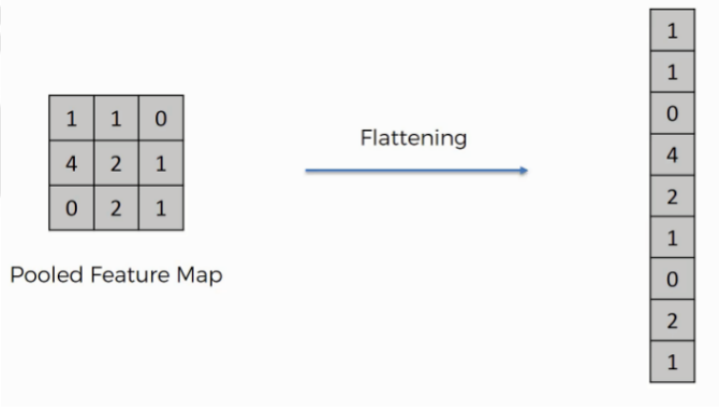
ที่มา: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/



ภาพประกอบ 11 แสดงการทำ Average Pooling

ที่มา: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

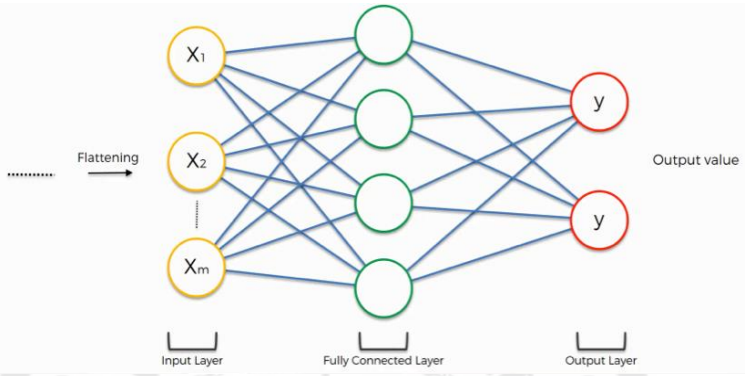
4. Flatten Layer ทำหน้าที่แปลง Input ให้เป็น Output ที่เป็นเวกเตอร์ (vector) 1 มิติ ดังภาพประกอบ 12 โดยทั่วไปจะใช้หลังจาก Convolutional Layer หรือ Pooling Layer เพื่อให้ข้อมูลสามารถส่งต่อไปยังชั้นต่อ ๆ ไปได้



ภาพประกอบ 12 แสดงการทำ Flattening ได้ Output แบบเวกเตอร์ 1 มิติ

ที่มา: https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening

5. Fully Connected Layer มักจะอยู่ชั้นสุดท้ายของโครงข่าย โดยทำหน้าที่เป็นชั้นที่เชื่อมต่อกับชั้น Output ที่มีจำนวน Node เท่ากับประเภทข้อมูลภาพที่ต้องการจำแนก โดยทั่วไป จะเลือกใช้ Activation Function เป็นแบบ Sigmoid หรือ Softmax อย่างใดอย่างหนึ่ง ดังภาพประกอบ 13 แต่ในงานวิจัยนี้เลือกใช้ Activation Function เป็นแบบ Softmax เท่านั้น

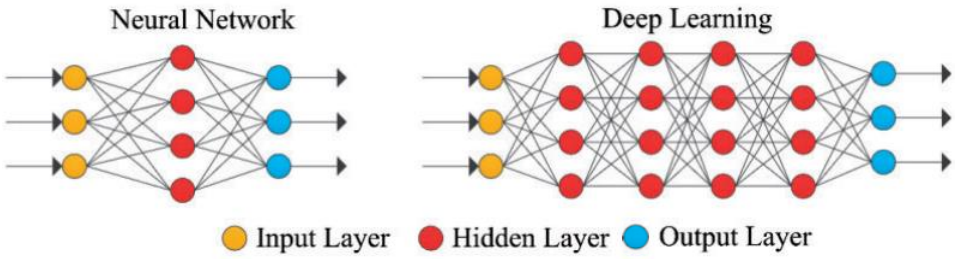


ภาพประกอบ 13 แสดง Fully Connected Layer ที่ค่า Output ทั้งหมด 2 ค่า

ที่มา: https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-4-full-connection

2.2.2 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก (Deep Learning) คือการเรียนรู้ที่ใช้แบบจำลองโครงข่ายประสาทเทียม (Neural Network) ที่มีจำนวนชั้นหลาย ๆ ชั้น (Deep) ในการเรียนรู้และสร้างความเข้าใจในข้อมูล โดยใช้เทคนิคการเรียนรู้แบบมีผู้สอน (Supervised Learning) หรือไม่มีผู้สอน (Unsupervised Learning) เพื่อให้แบบจำลองสามารถจำแนกและประมวลผลข้อมูลได้เอง โดยไม่ต้องระบุคุณสมบัติของข้อมูลแต่ละชุด แบบจำลองโครงข่ายประสาทเทียมที่มีจำนวนชั้นมากจะมีความสามารถในการจำแนกและวิเคราะห์ข้อมูลได้ดียิ่งขึ้น และสามารถเรียนรู้ความหมายและความสัมพันธ์ระหว่างข้อมูลได้เป็นอย่างดี



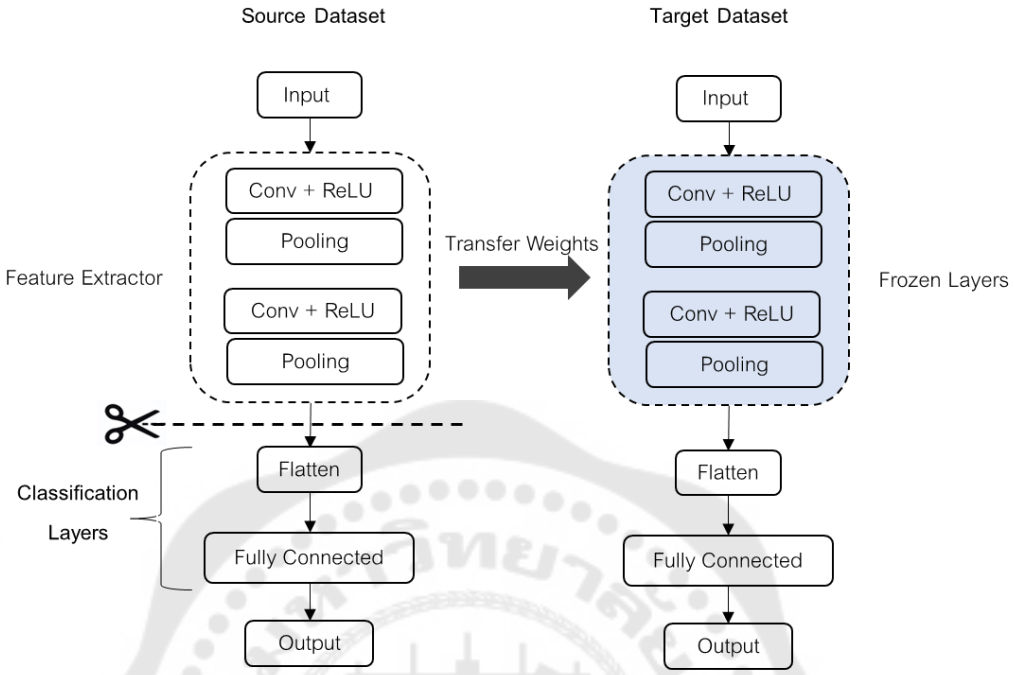
ภาพประกอบ 14 แสดงโครงสร้างพื้นฐานของแบบจำลอง Perceptron

ที่มา: (Wanli และ Dongping, 2018)

จากภาพประกอบ 14 โครงสร้างพื้นฐานของแบบจำลอง Perceptron ซึ่งเป็นหนึ่งในโครงสร้างพื้นฐานของโครงข่ายประสาทเทียม (Artificial Neural Network) โดยมีหน่วยประมวลผล (Neuron) ประกอบด้วย 3 ส่วน ได้แก่ ชั้น Input, ชั้น Hidden ที่สามารถมีได้หลาย ๆ ชั้น และชั้น Output โดยและเรียกแบบจำลองมีที่ชั้น Hidden ได้หลาย ๆ ชั้น ว่าแบบจำลอง Multi-layer Perceptron (MLP) หรือ Deep Learning ในการเรียนรู้เชิงลึกในงานวิจัยนี้จะใช้ซอฟต์แวร์ที่เป็น Open-source ได้แก่ TensorFlow และ Keras ซึ่งเป็นซอฟต์แวร์ชั้นสูงสำหรับการสร้างและการประมวลผลแบบจำลองโครงข่ายประสาทเทียม

2.2.3 การเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-trained Deep Learning)

แบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) ที่ถูกฝึกสอนด้วยข้อมูลจำนวนมากและสกัดคุณลักษณะของแบบจำลอง (Feature Extraction) แล้วนำมาปรับใช้ในงานที่มีความคล้ายคลึงกัน โดยการถ่ายโอนการเรียนรู้ (Transfer Learning) และเมื่อคุณลักษณะของแบบจำลอง (Feature Extractors) ถูกถ่ายโอน (Transfer Weight) มาที่แบบจำลองใหม่ จะเรียกชั้นที่ได้จาก Pre-Trained Model ว่า Frozen Layers ดังภาพประกอบ 15 อีกทั้งการถ่ายโอนการเรียนรู้ยังช่วยประหยัดเวลาและทรัพยากร โดยในงานวิจัยนี้ใช้ชุดข้อมูลจาก ImageNet เป็นข้อมูลภาพในการถ่ายโอนการเรียนรู้ ซึ่งในงานวิจัยนี้ได้ศึกษาแบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อนทั้งหมด 4 แบบจำลอง ได้แก่ VGG16, VGG19, ResNet50 และ EfficientNetB0



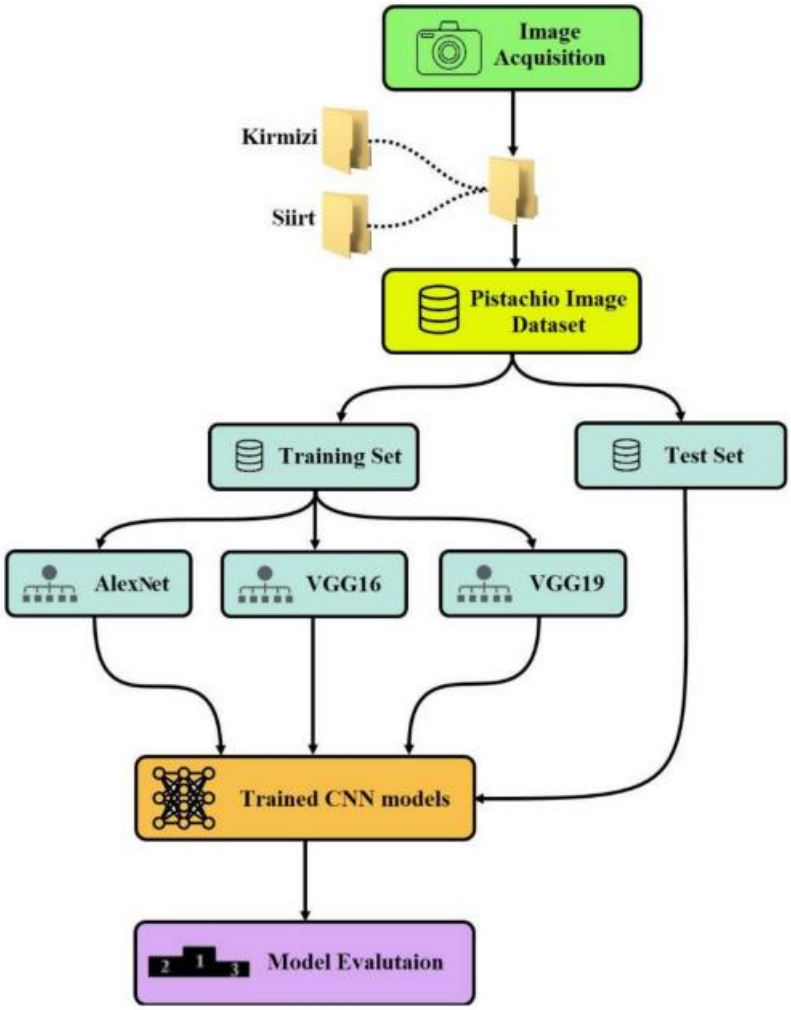
ภาพประกอบ 15 แสดงการถ่ายโอนการเรียนรู้ (Transfer Learning)

ที่มา: https://livebook.manning.com/book/transfer-learning-in-action/chapter-1/v-1/66

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 งานวิจัยเรื่อง Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models (Singh และคณะ, 2022)

งานวิจัยนี้ศึกษาการจำแนกประเภทภาพของเมล็ดพิสตาชิโอ 2 สายพันธุ์คือ Kirmizi และ Siirt โดยใช้แบบจำลองการเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-trained Deep Learning Models) ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) ทั้งหมด 3 ประเภท ได้แก่ AlexNet, VGG16 และ VGG19 มีชุดข้อมูลทั้งหมด 2,148 รูป ซึ่งแบ่งออกเป็นสายพันธุ์ Kirmizi จำนวน 2,132 รูป และสายพันธุ์ Siirt จำนวน 916 รูป และมีขั้นตอนการดำเนินงานวิจัย ดังภาพประกอบ 16



ภาพประกอบ 16 แสดงแผนภาพขั้นตอนการดำเนินการของงานวิจัย

ที่มา: (Singh และคณะ, 2022)

ตาราง 3 แสดงเวลาที่ใช้ในการเรียนรู้ (Training Time) ของแบบจำลองต่าง ๆ

	AlexNet	VGG16	VGG19
เวลาที่ใช้ไป	17 นาที 7 วินาที	90 นาที 28 วินาที	99 นาที 0 วินาที

ที่มา: (Singh และคณะ, 2022)

ตาราง 4 แสดงค่า Performance Metrics ของแบบจำลองต่าง ๆ

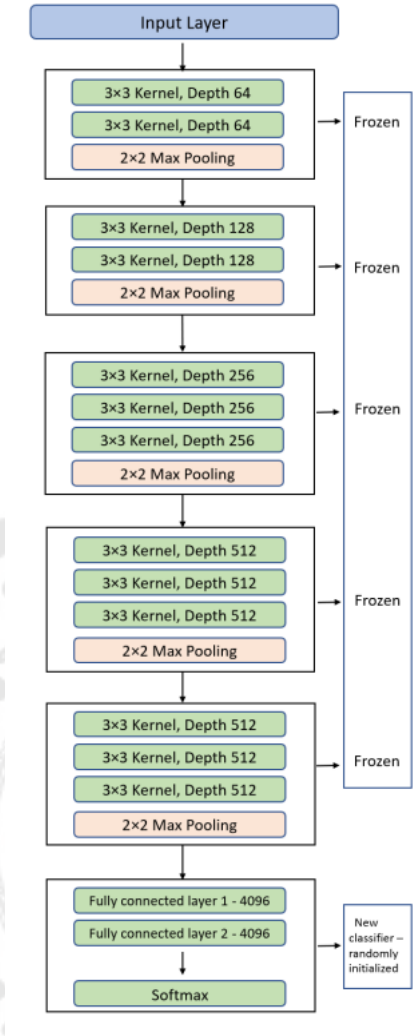
Performance Metrics	AlexNet	VGG16	VGG19
ค่าความถูกต้อง (Accuracy)	0.9442	0.9884	0.9814
ค่าความแม่นยำ (Precision)	0.9150	0.9828	0.9742
ค่า F-1 Score	0.9496	0.9884	0.982
ค่าความไว (Sensitivity)	0.9869	0.9956	0.9913
ค่าความจำเพาะ(Specificity)	0.8955	0.9801	0.9701

ที่มา: (Singh และคณะ, 2022)

จากตาราง 3 พบว่าแบบจำลอง AlexNet ใช้เวลาในการฝึกสอนข้อมูลน้อยที่สุด ตามด้วยแบบจำลอง VGG16 และแบบจำลอง VGG19 โดยใช้เวลาไป 17 นาที 7 วินาที และ 90 นาที 28 วินาที และ 99 นาที ตามลำดับ และจากตาราง 4 การประเมินประสิทธิภาพของแบบจำลองต่าง ๆ จะพบว่าแบบจำลองที่ใช้โครงสร้างแบบ AlexNet ใช้เวลาในการฝึกข้อมูลน้อยที่สุด เนื่องจากเป็นแบบจำลองที่มีความซับซ้อนน้อยที่สุด และแบบจำลองที่ใช้โครงสร้างแบบ VGG16 ได้ค่าความถูกต้อง (Accuracy) สูงที่สุดอยู่ที่ 0.9884 ตามด้วยแบบจำลองที่ใช้โครงสร้างแบบ VGG19 และแบบจำลองที่ใช้โครงสร้างแบบ AlexNet ตามลำดับ

2.3.2 งานวิจัยเรื่อง Transfer Learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images (Tammina, 2019)

งานวิจัยนี้ใช้การเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-trained Deep Learning Models) ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) แบบ VGG16 และใช้เทคนิค Transfer Learning จากชุดข้อมูล ImageNet เพื่อสร้างแบบจำลองที่สามารถเรียนรู้ สำหรับการจำแนกภาพได้เอง โดยการทำ Feature Extraction จากแบบจำลอง Pre-trained Deep Learning และถ่ายโอนการเรียนรู้มาที่แบบจำลองใหม่เป็น Frozen Layers ดังภาพประกอบ 17 ข้อดีของการใช้ Transfer Learning จะช่วยประหยัดเวลาในการฝึกข้อมูลแบบจำลองและประหยัดทรัพยากรอีกด้วย โดยศึกษาแบบจำลองที่มีการฝึกฝนข้อมูลและปรับพารามิเตอร์เท่านั้น (Training and Validation) และนำ Feature Extractor ที่ได้ไปประยุกต์ใช้กับงานอื่น ๆ ที่ลักษณะคล้ายคลึงกัน เพื่อจำแนกข้อมูลประเภทใหม่ (New Classifier) ได้อย่างมีประสิทธิภาพ



ภาพประกอบ 17 แสดงโครงสร้างแบบจำลอง VG16 ที่ได้รับการถ่ายโอนการเรียนรู้

ที่มา: (Tammina, 2019)

ตาราง 5 การเปรียบเทียบค่าความถูกต้อง (Accuracy) ของแบบจำลองที่ใช้เทคนิคแตกต่างกัน

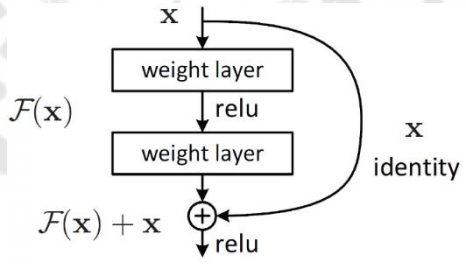
Model	Training Accuracy	Validation Accuracy
Basic CNN	98.20%	72.40%
Fine Tuning CNN with Image Augmentation	81.30%	79.20%
Fine Tuning CNN with Pre-trained VGG16 Model and Image Augmentation	86.50%	95.40%

ที่มา: (Tammina, 2019)

จากตาราง 5 พบว่าแบบจำลองที่เป็นแบบ Basic CNN และแบบจำลอง Fine Tuning CNN with Image Augmentation นั้นมีค่าความถูกต้องในการฝึกฝนข้อมูล (Training Accuracy) สูงกว่า ค่าความถูกต้องในการตรวจสอบแบบจำลอง (Validation Accuracy) ซึ่งเป็นเพราะเกิด Overfitting ทำให้แบบจำลองทำนายได้ถูกต้องลดลง ส่วนแบบจำลอง Fine tuning CNN with Pre-trained VGG16 Model and Image Augmentation นั้นได้ค่าความถูกต้องในการตรวจสอบแบบจำลอง (Validation Accuracy) สูงสุดอยู่ที่ 95.40% เนื่องมีการทำ Pre-trained VGG16 และการเพิ่มข้อมูลภาพ (Image Augmentation) ทำให้แบบจำลองนั้นได้ฝึกฝนข้อมูลภาพที่หลายหลากมากกว่าจึงได้แบบจำลองที่มีประสิทธิภาพมากกว่าด้วย

2.3.3 งานวิจัยเรื่อง Deep Residual Learning for Image Recognition (He และคณะ, 2016)

งานวิจัยนี้นำเสนอการจำแนกภาพโดยการใช้ Residual Neural Network (ResNet) ซึ่งเป็นโครงข่ายประสาทเทียมลึก (Deep Neural Network) โดยใช้แบบจำลอง ResNet ที่มีโครงสร้างแตกต่างกันหลายรูปแบบ ได้แก่ ResNet34, ResNet50, ResNet101 และResNet152 รวมทั้งใช้เทคนิค Residual Learning และเทคนิคการเพิ่มข้อมูลภาพ (Image Data Generator) มาช่วยพัฒนาประสิทธิภาพของแบบจำลอง



ภาพประกอบ 18 แสดงเทคนิคการเรียนรู้แบบ Residual Learning

ที่มา: (He และคณะ, 2016)

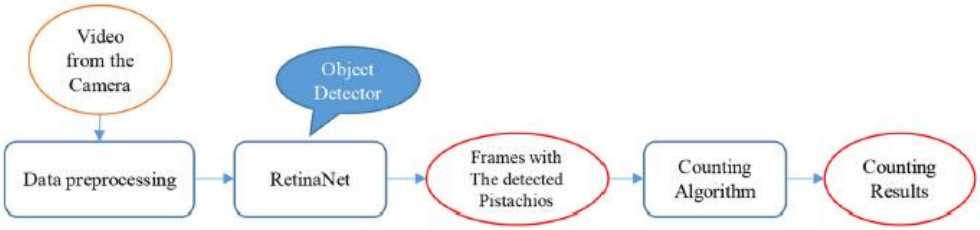
จากภาพประกอบ 18 แสดงเทคนิคการเรียนรู้แบบ Residual Learning หรือ A Building Block ซึ่งเป็นเทคนิคการเรียนรู้เชิงลึกที่ช่วยให้แบบจำลองสามารถเรียนรู้ได้ลึกขึ้นและมีประสิทธิภาพมากขึ้น เทคนิคนี้ใช้แนวคิดของการเชื่อมต่อข้ามชั้น (Skip Connection) ซึ่งจะช่วยให้แบบจำลองสามารถเรียนรู้ฟีเจอร์ที่ระดับต่ำ (Low-Level Features) และฟีเจอร์ที่ระดับสูง (High-

Level Features) ได้อย่างมีประสิทธิภาพ และเป็นแบบ Element-Wise Addition คือการบวก Feature map จากชั้นก่อนหน้าเข้าด้วยกันเพื่อคำนวณฟีเจอร์ใหม่ โดยการเชื่อมต่อข้ามชั้น (Skip Connection) จะช่วยให้แบบจำลองสามารถรับข้อมูลจากชั้นก่อนหน้านี้ได้โดยตรงโดยไม่ต้องผ่านชั้นกลางทั้งหมด ช่วยให้แบบจำลองการเรียนรู้ได้ดีขึ้น และไม่ต้องเพิ่มจำนวนพารามิเตอร์ของแบบจำลองมากเกินไป อีกทั้งยังช่วยลดปัญหา Vanishing Gradient อีกด้วย

2.3.4 **งานวิจัยเรื่อง** Detecting and Counting Pistachios based on Deep Learning **(**Rahimzadeh **และ** Attar, 2022)

งานวิจัยนี้ได้ดำเนินการใช้ตัวตรวจจับวัตถุ RetinaNet ซึ่งใช้โครงข่ายประสาทเทียมลึก (Deep Neural Network) แบบ ResNet ดังภาพประกอบ 19 และใช้ชุดข้อมูลในการตรวจจับถั่วพิสตาชิโอในเฟรมวิดีโอที่เป็นวิดีโอทั้งหมด 6 รายการ มีระยะเวลารวม 167 วินาที และถั่วพิสตาชิโอที่ที่ป้ายกำกับ 3,927 เมล็ด สามารถตรวจจับและนับถั่วพิสตาชิโอที่มีเปลือกเปิดและเปลือกปิดได้

ในการฝึกสอนแบบจำลองสำหรับตัวตรวจจับวัตถุ RetinaNet นั้น จะแบ่งข้อมูลออกเป็น 5 ชุด และใช้แบบจำลองทั้งหมด 3 ประเภท ได้แก่ ResNet50, ResNet152 และVGG16 โดยที่แบบจำลอง ResNet50 ได้ผลลัพธ์ค่าเฉลี่ยความแม่นยำ (mAP) สูงสุดอยู่ที่ 91.87%



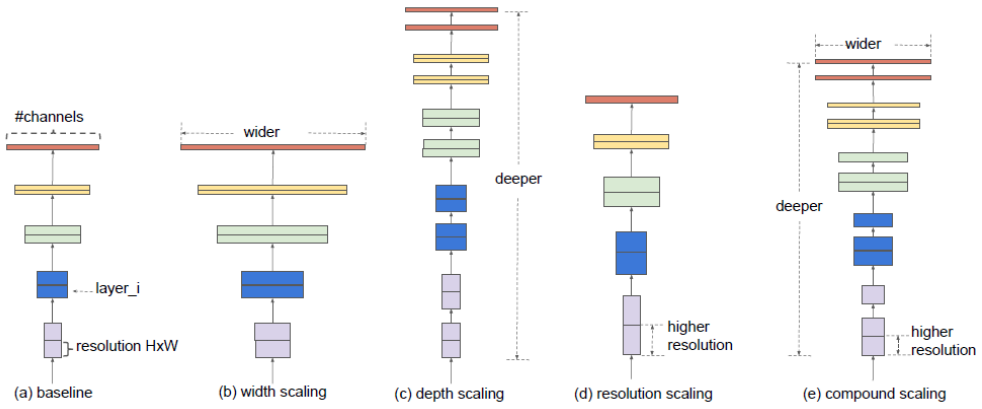
ภาพประกอบ 19 แสดงกระบวนการทำงานในการตรวจจับและนับถั่วพิสตาชิโอ

ที่มา: (Rahimzadeh และ Attar, 2022)

ข้อจำกัดของงานวิจัยนี้คือ ถั่วพิสตาชิโอเป็นวัตถุที่เคลื่อนไหวและมักจะหมุนตัวบนสายพานลำเลียง และจากมุมมองของกล้องอาจทำให้ตรวจจับผิดพลาดได้ เช่น ถั่วพิสตาชิโอที่เปลือกเปิดและถูกหมุนให้ด้านมีเปลือกเปิดกลับด้าน ทำให้ดูเหมือนกับถั่วพิสตาชิโอที่เปลือกปิดได้ และอาจจะปรากฏเป็นถั่วพิสตาชิโอที่เปลือกเปิดอีกครั้งในการตรวจจับรอบอื่น ๆ อีกด้วย

2.3.5 งานวิจัยเรื่อง EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (Tan และ Le, 2019)

งานวิจัยนี้ศึกษาการปรับโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) แบบ EfficientNet โดยใช้การถ่ายโอนการเรียนรู้ (Transfer Learning) จากชุดข้อมูล ImageNet และการปรับขนาดแบบจำลองแบบผสม (Compound Scaling) ด้วยการปรับขนาดทั้ง 3 มิติของแบบจำลอง คือความกว้าง (Width), ความลึก (Depth) และความละเอียด (Resolution) โดยการปรับค่า Compound Coefficient หรือค่าอัตราส่วนที่แน่นอน ตั้งแต่ 0 ถึง 7 โดย ซึ่งเรียกชื่อแบบจำลองตามค่า Compound Coefficient เป็น EfficientNetB0 จนถึง EfficientNetB7



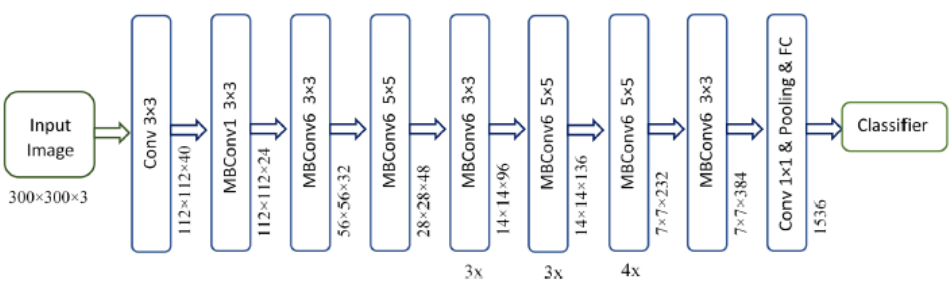
ภาพประกอบ 20 แสดงภาพการปรับขนาดของแบบจำลองต่าง ๆ (Model Scaling)

ที่มา: (Tan และ Le, 2019)

จากภาพประกอบ 20 การปรับขนาดของแบบจำลองต่าง ๆ (Model Scaling นั้นจะเห็นว่า (a) เป็นแบบจำลองพื้นฐานที่ไม่ได้มีการปรับขนาด, (b) เป็นการปรับขนาดความกว้าง (Width) ของแบบจำลองเท่านั้น, (c) เป็นการปรับขนาดความลึก (Depth) ของแบบจำลองเท่านั้น, (d) เป็นการปรับขนาดความละเอียด (Resolution) ของแบบจำลองเท่านั้น และ (e) เป็นการปรับขนาดแบบผสม (Compound Scaling) ที่งานวิจัยนี้นำเสนอ คือปรับขนาดทั้งสามมิติ คือความกว้าง (Width), ความลึก (Depth) และความละเอียด (Resolution) โดยใช้อัตราส่วนคงที่ (Compound Coefficient)

2.4.6 **งานวิจัยเรื่อง** Cultivar Identification of Pistachio Nuts in Bulk Mode through EfficientNet Deep Learning Model (Soleimanipour **และคณะ**, 2022)

งานวิจัยนี้นำเสนอการประยุกต์ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) แบบ EfficientNetB3 (ค่า Compound Coefficient เท่ากับ 3) โดยมีโครงสร้างแบบจำลองดังภาพประกอบ 21 เพื่อจำแนกประเภทถั่วพิสตาชิโอทั้งหมด 4 สายพันธุ์ โดยมีการเพิ่มข้อมูลภาพ (Data Augmentation) และการปรับค่าพารามิเตอร์ต่างของแบบจำลอง (Fine Tuning) ผลลัพธ์งานวิจัยได้ค่าความถูกต้อง (Accuracy) อยู่ที่ 98.00%, ค่าเฉลี่ยความแม่นยำ (Average Precision) อยู่ที่ 96.73%, ค่า Recall อยู่ที่ 96.70% และค่า F1-Score อยู่ที่ 96.67% แต่ข้อจำกัดงานวิจัยนี้คือไม่ได้ดำเนินการเปรียบเทียบประสิทธิภาพแบบจำลองกับแบบจำลองโครงสร้างอื่น ๆ



ภาพประกอบ 21 แสดงโครงสร้างแบบจำลอง EfficientNetB3

ที่มา: (Soleimanipour และคณะ, 2022)

ในบทนี้ผู้วิจัยได้ทบทวนวรรณกรรมที่เกี่ยวข้องและได้เลือกศึกษาแบบจำลองทั้งหมด 5 ประเภท ได้แก่ แบบจำลอง CNN ที่มีการเรียนรู้จากเริ่มต้น (CNN from Scratch) และแบบจำลองการเรียนรู้เชิงลึกที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-trained Deep Learning Model) อีก 4 ประเภท ได้แก่ VGG16, VGG19, ResNet50 และ EfficientNetB0 โดยจะนำแบบจำลองดังกล่าวรวมทั้งเทคนิคการเตรียมข้อมูลภาพ (Pre-processing) มาประยุกต์ใช้กับข้อมูลภาพของถั่วพิสตาชิโอทั้ง 3 สายพันธุ์ที่ใช้ในงานวิจัยนี้

บทที่ 3
วิธีดำเนินการวิจัย

ในการวิจัยครั้งนี้ ผู้วิจัยได้ดำเนินการตามขั้นตอนดังนี้

1. แผนการดำเนินงานวิจัย
2. กระบวนการทำงานของแบบจำลอง
3. การเก็บรวบรวมข้อมูล (Image Acquisition)
4. การเตรียมข้อมูล (Pre-Processing)
5. การสร้างแบบจำลอง (Modeling)
6. การประเมินผลแบบจำลอง (Model Evaluation)

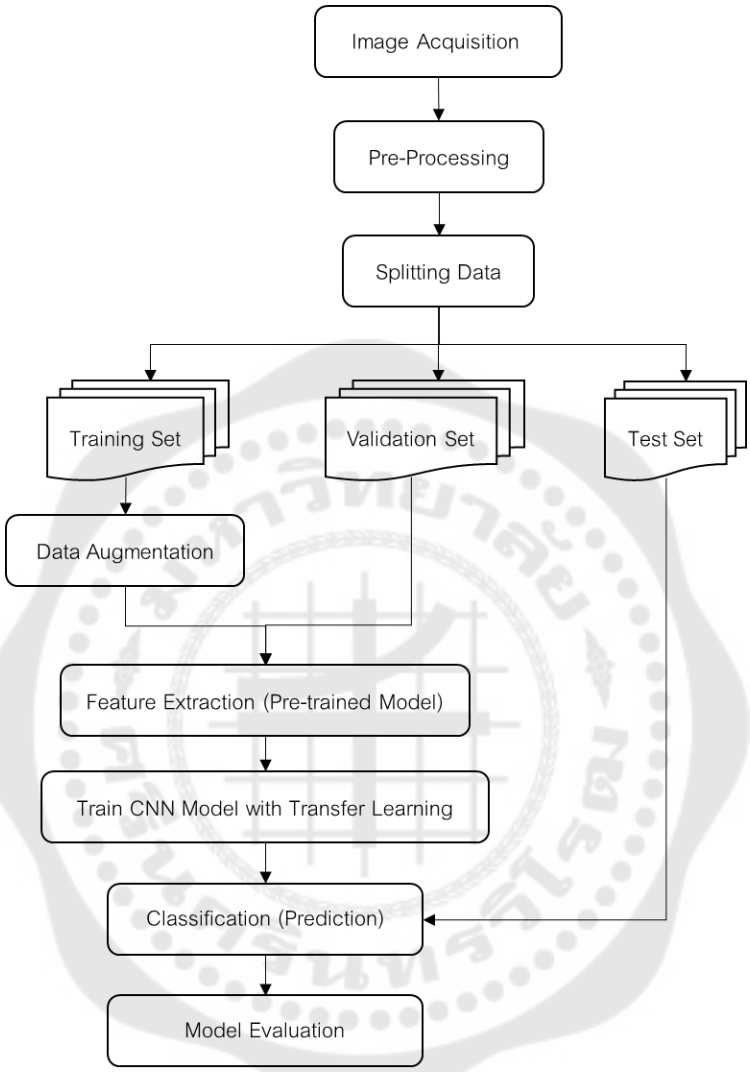
3.1 แผนการดำเนินงานวิจัย

ผู้วิจัยได้วางแผนการดำเนินงานวิจัย โดยเริ่มตั้งแต่เดือนสิงหาคม พ.ศ. 2566 จนถึงเดือนเมษายน พ.ศ. 2567 โดยดำเนินการตามขั้นตอนต่าง ๆ ดังตาราง 6

ตาราง 6 แผนการดำเนินงานวิจัย

ขั้นตอนการดำเนินงาน	พ.ศ. 2566					พ.ศ. 2567			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. วางแผนการทำงาน									
2. ศึกษาทฤษฎีและแนวคิดที่ใช้									
3. เก็บรวบรวมข้อมูล									
4. เตรียมข้อมูล									
5. สร้างแบบจำลอง									
6. ปรับปรุงประสิทธิภาพแบบจำลอง									
7. ประเมินผลแบบจำลอง									
8. สรุปผลและรายงานผล									

3.2 กระบวนการทำงานของแบบจำลอง

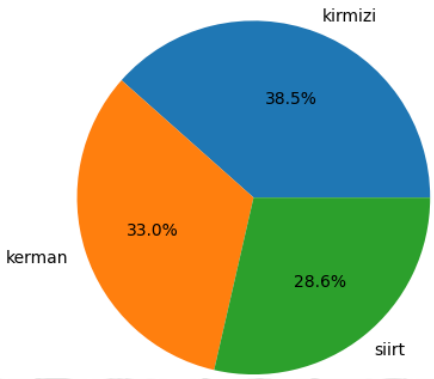


ภาพประกอบ 22 แสดงกระบวนการทำงานของแบบจำลอง

จากภาพประกอบ 22 ขั้นตอนในการวิจัยเริ่มจากการเก็บรวบรวมข้อมูล (Image Acquisition) จากนั้นเตรียมข้อมูลภาพ (Pre-Processing) และแบ่งข้อมูลภาพ (Splitting) โดยแบ่งข้อมูลออกเป็น 3 ส่วนเป็น Training Set, Validation Set และ Test Set ในอัตราส่วน 80:10:10 จากนั้นทำการเรียนรู้แบบจำลอง โดยใช้เทคนิคการถ่ายโอนเรียนรู้ (Transfer Learning) จากนั้นทดสอบการจำแนกประเภทของข้อมูลที่ใช้ทดสอบ (Test Set) และประเมินประสิทธิภาพแบบจำลอง (Model Evaluation) โดยจะอธิบายรายละเอียดแต่ละขั้นตอนในหัวข้อถัดไป

3.3 **การเก็บรวบรวมข้อมูล** (Image Acquisition)

ข้อมูลภาพทั้งหมดในงานวิจัยนี้อยู่ในรูปไฟล์ JPG จำนวน 3,204 รูป ซึ่งจำแนกออกเป็น 3 ประเภท ได้แก่ ถั่วพิสตาชิโอสายพันธุ์ Kirmizi, Siirt และ Kerman โดยที่ข้อมูลภาพสายพันธุ์ Kirmizi จำนวน 1,232 รูป และสายพันธุ์ Siirt จำนวน 916 รูป มาจากข้อมูลสาธารณะ (Singh และคณะ, 2022) และได้เก็บข้อมูลภาพเพิ่มเติม เป็นสายพันธุ์ Kerman จำนวน 1,056 รูป



ภาพประกอบ 23 แสดงแผนภูมิวงกลมแสดงเปอร์เซ็นต์ของสายพันธุ์ต่าง ๆ ในชุดข้อมูล

จากภาพประกอบ 23 แสดงแผนภูมิวงกลม (Pie Chart) ที่แสดงการแบ่งเปอร์เซ็นต์ของสายพันธุ์ต่าง ๆ ในชุดข้อมูล มีทั้งหมด 3 ประเภท ซึ่งเป็นชุดข้อมูลแบบ Balanced Dataset แสดงด้วยสีที่ต่างกัน ดังนี้

- สีน้ำเงิน หมายถึง สายพันธุ์ Kirmizi คิดเป็น 38.5% ของข้อมูลภาพทั้งหมด
- สีส้ม หมายถึง สายพันธุ์ Kerman คิดเป็น 33.0% ของข้อมูลภาพทั้งหมด
- สีเขียว หมายถึง สายพันธุ์ Siirt คิดเป็น 28.6% ของข้อมูลภาพทั้งหมด

3.4 **การเตรียมข้อมูล** (Pre-Processing)

3.4.1 Prepare Data

ดำเนินการย่อขนาดของรูปทั้งหมดให้มีขนาด 224×224 พิกเซล จากนั้นแบ่งชุดข้อมูลดังกล่าวเป็น Training Set, Validation Set และ Test Set ในอัตราส่วน 80:10:10 และทำการสร้างโฟลเดอร์ของข้อมูลภาพทั้งหมด 3 โฟลเดอร์ คือ Train, Test, Validation และในแต่ละโฟลเดอร์หลักจะมีโฟลเดอร์ย่อยอีกจำนวน 3 โฟลเดอร์ ตามจำนวนประเภทของสายพันธุ์ถั่วพิสตาชิโอ คือ Kirmizi, Siirt และ Kerman

3.4.2 Data Augmentation

การเพิ่มความหลากหลายของภาพและเพิ่มจำนวน หรือ ImageDataGenerator ใน TensorFlow และ Keras ให้กับชุดข้อมูลที่ใช้ในการฝึกสอน (Training Set) โดยดำเนินการดังนี้

- การสุ่มขยายหรือย่อภาพ (Zoom Range) เท่ากับ 0.2
- การเลื่อนภาพแนวนอนในช่วงสัดส่วนที่กำหนดของความกว้างของภาพ (Width Shift Range) เท่ากับ 0.1
- การเลื่อนภาพแนวตั้งในช่วงสัดส่วนที่กำหนดของความสูงของภาพ (Height Shift Range) เท่ากับ 0.1
- การหมุนภาพในช่วงขององศาที่กำหนด (Rotation Range) เท่ากับ 15 องศา
- การเติมพิกเซลที่อาจเกิดขึ้นระหว่างการแปลงภาพ (Fill Mode) โดยเลือกใช้ค่า Nearest ซึ่งเติมด้วยพิกเซลใกล้เคียง
- การพลิกภาพแนวนอน (Horizontal Flip)
- การพลิกภาพแนวตั้ง (Vertical Flip)
- ช่วงของความสว่างที่สามารถปรับเปลี่ยนได้ (Brightness Range) โดยเลือกค่า 0.8 ถึง 1.2 ซึ่งแสดงถึงการลดหรือเพิ่มความสว่างของภาพระหว่าง 80% ถึง 120% ของค่าความสว่างเดิม

3.5 การสร้างแบบจำลอง (Modeling)

ในงานวิจัยนี้ดำเนินการบน Google Colab โดยใช้ประเภท GPU คือ V100 และได้ทดลองกับแบบจำลองทั้งหมด 5 ประเภท ได้แก่ CNN from Scratch, VGG16, VGG19, ResNet50 และ EfficientNetB0 ซึ่งแต่ละแบบจำลองจะแตกต่างกันในส่วนของชั้นที่มีน้ำหนัก (Trainable Layers) และมีชั้นที่ไม่มีน้ำหนัก (Non-Trainable Layers) แต่ทุกแบบจำลองจะใช้ชั้นสำหรับการจำแนก (Classification Layers) ที่เหมือนกันทั้งหมด โดยใช้คำสั่งดังภาพประกอบ 24 ซึ่งจะเรียงลำดับชั้นและรายละเอียดดังนี้

- ชั้น Flatten
- ชั้น Fully Connected หรือ fc_1 (Dense) ที่มีจำนวนนิวรอน 4,096 นิวรอน และใช้ Activated function แบบ ReLU
- ชั้น Dropout หรือ dropout_1 โดยใช้อัตรา Dropout เท่ากับ 0.2

- ชั้น Fully Connected หรือ fc_2 (Dense) ที่มีจำนวนนิวรอน 4,096 นิวรอน และใช้ Activated function แบบ ReLU
- ชั้น Dropout หรือ dropout_2 โดยใช้อัตรา Dropout เท่ากับ 0.2
- ชั้น Output (Dense) ที่มีจำนวนนิวรอน 3 นิวรอน และใช้ Activated function แบบ Softmax

```
model = tf.keras.Sequential([
    extractor,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4096, activation='relu', name='fc_1'),
    tf.keras.layers.Dropout(0.2, name='dropout_1'),
    tf.keras.layers.Dense(4096, activation='relu', name='fc_2'),
    tf.keras.layers.Dropout(0.2, name='dropout_2'),
    tf.keras.layers.Dense(3, activation='softmax',name='output')
])
model.summary()
```

ภาพประกอบ 24 แสดงคำสั่งใช้สร้างชั้นสำหรับการจำแนก (Classification Layers)

โดยงานวิจัยนี้ได้ศึกษาแบบจำลองที่มีการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์ 2 ค่า คือ อัตราการเรียนรู้ (Learning Rate) และจำนวนตัวอย่างข้อมูล (Batch Size) ซึ่งมีรายละเอียดต่าง ๆ ดังนี้

- เครื่องมือเพิ่มประสิทธิภาพ (Optimizer) เลือกใช้ชนิด ADAM
- อัตราการเรียนรู้ของแบบจำลอง (Learning Rate) โดยใช้อัตราที่ 10^{-4} และ 10^{-5}
- อัตราส่วนของนิวรอนที่ไม่นำมาใช้ในแต่ละการฝึก เพื่อป้องกันการเกิด Overfitting (Dropout Rate) โดยใช้อัตราที่ 0.2
- Activation Functions ชั้น Fully Connected เลือกใช้ ReLU
- จำนวนรอบทั้งหมดที่ใช้ในการฝึกชุดข้อมูล (Epoch) โดยใช้จำนวน 10 รอบ
- ขนาดของกลุ่มข้อมูล (Batch Size) โดยใช้ขนาด 8, 16, 32 และ 64 ตามลำดับ

- จำนวนพารามิเตอร์ทั้งหมดของแต่ละแบบจำลอง สามารถดูข้อมูลเพิ่มเติมได้จากตาราง 13 ในส่วนของภาคผนวก

การสร้างแบบจำลองทั้งหมด 5 ประเภท มีดังนี้

1. แบบจำลอง CNN ที่มีการเรียนรู้จากเริ่มต้น (CNN from Scratch)
2. แบบจำลอง VGG16
3. แบบจำลอง VGG19
4. แบบจำลอง ResNet50
5. แบบจำลอง EfficientNetB0

3.5.1 **แบบจำลอง** CNN from Scratch

การสร้างแบบจำลอง CNN พื้นฐานในการฝึกสอนแบบจำลองตั้งแต่เริ่มต้น โดยรายละเอียดแต่ละชั้นมีดังนี้

- ชั้น Convolutional: ชั้นนี้มีทั้งหมด 3 ชั้น ที่ใช้เคอร์เนลขนาด 3×3 พร้อมฟังก์ชันการกระตุ้น ReLU และมีตัวกรอง (Filter) เพิ่มขึ้นเป็น 16, 32 และ 64 ตามลำดับ
- ชั้น Max Pooling (MaxPooling2D): มี 2 ชั้น ที่ใช้วินโดว์ขนาด 2x2 เพื่อลดขนาดของ Feature Map และลดปริมาณการคำนวณในชั้นถัดไป
- ชั้น Dropout: ใช้เพื่อป้องกันการ Overfitting ในช่วงการฝึกแบบจำลอง
- ชั้น Fully Connected (Dense): มี 3 ชั้นที่ปลายสุดของแบบจำลอง โดยสองชั้นแรกมีจำนวนนิวรอน 4,096 นิวรอนและชั้นสุดท้ายเป็นชั้น Output ที่มีจำนวนนิวรอน 3 นิวรอนตามจำนวนประเภทที่ต้องการจำแนก
- ชั้น Activation (ReLU): ฟังก์ชัน ReLU ใช้เป็นฟังก์ชันการกระตุ้นในชั้น Convolutional และ Fully Connected
- ชั้น Softmax: ใช้เป็นชั้นการกระตุ้นในชั้น Output เพื่อคำนวณความน่าจะเป็นของแต่ละประเภทที่จำแนก

3.5.2 **แบบจำลอง** VGG16

แบบจำลอง VGG16 เป็นหนึ่งในแบบจำลองที่มีการเรียนรู้มาก่อน (Pre-Trained Model) ทั้งหมด โดยทำการสกัดคุณลักษณะของแต่ละแบบจำลอง (Feature Extractor) จากนั้นถ่ายโอน (Transfer Weight) มาที่แบบจำลองใหม่ โดยเรียกชั้นที่ได้จาก Pre-Trained Model ว่า Frozen Layers และกำหนดชั้นสำหรับการจำแนก (Classification Layers) แบบจำลอง VGG16 ประกอบด้วยชั้นที่มีน้ำหนัก (Trainable Layers) ทั้งหมด 16 ชั้น โดยรายละเอียดของแต่ละชั้นมีดังนี้

- ชั้น Convolutional (Conv2D): ชั้นนี้มีทั้งหมด 13 ชั้น ที่ใช้เคอร์เนลขนาด 3×3 พร้อมฟังก์ชันการกระตุ้น ReLU และจะมีตัวกรอง (Filter) เพิ่มขึ้นเป็น 64, 128, 256, และ 512 ตามลำดับ
- ชั้น Max Pooling (MaxPooling2D): มี 5 ชั้น ที่ใช้วินโดว์ขนาด 2x2 เพื่อลดขนาดของ Feature Map และลดปริมาณการคำนวณในชั้นถัดไป
- ชั้น Dropout: ใช้เพื่อป้องกันการ Overfitting ในช่วงการฝึกแบบจำลอง
- ชั้น Fully Connected (Dense): มี 3 ชั้นที่ปลายสุดของแบบจำลอง โดยสองชั้นแรกมีจำนวนนิวรอน 4,096 นิวรอนและชั้นสุดท้ายเป็นชั้น Output ที่มีจำนวนนิวรอน 3 นิวรอนตามจำนวนประเภทที่ต้องการจำแนก
- ชั้น Activation (ReLU): ฟังก์ชัน ReLU ใช้เป็นฟังก์ชันการกระตุ้นในชั้น Convolutional และ Fully Connected
- ชั้น Softmax: ใช้เป็นชั้นการกระตุ้นในชั้น Output เพื่อคำนวณความน่าจะเป็นของแต่ละประเภทที่จำแนก

ในงานวิจัยนี้ได้ใช้แบบจำลอง VGG16 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) ต่อชั้น Convolutional หรือชั้น Dense ที่มีอยู่ ซึ่งน้ำหนักที่ได้จากการฝึกฝนบนชุดข้อมูล ImageNet ถูกใช้โดยตรง โดยไม่มีการเปลี่ยนแปลงใด ๆ
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune) โดยปรับแต่งกลุ่มชั้นที่ 5 (Convolutional Block 5) ซึ่งเป็นกลุ่มชั้น Convolutional สุดท้ายให้เป็นชั้นที่มีน้ำหนัก (Trainable Layers) ก่อนจะเข้าชั้นสำหรับการจำแนก (Classification Layers)

3.5.3 **แบบจำลอง** VGG19

แบบจำลอง VGG19 ประกอบด้วยชั้นที่มีน้ำหนัก (Trainable Layers) ทั้งหมด 19 ชั้น โดยรายละเอียดของแต่ละชั้นมีดังนี้

- ชั้น Convolutional (Conv2D): ชั้นนี้มีทั้งหมด 16 ชั้นที่ใช้เคอร์เนลขนาด 3x3 พร้อมฟังก์ชันการกระตุ้น ReLU และมีตัวกรอง (Filter) เพิ่มขึ้นเป็น 64, 128, 256, และ 512 ตามลำดับ โดยจะเรียงซ้อนกันหลายชั้นเพื่อจับคู่ลักษณะเด่นที่ซับซ้อนขึ้น
- ชั้น Max Pooling (MaxPooling2D): มี 5 ชั้น ที่ใช้วินโดว์ขนาด 2x2 เพื่อลดขนาดของ Feature Map และลดปริมาณการคำนวณในชั้นถัดไป
- ชั้น Dropout: ใช้เพื่อป้องกันการ Overfitting ในช่วงการฝึกแบบจำลอง
- ชั้น Fully Connected (Dense): มี 3 ชั้นที่ปลายสุดของแบบจำลอง โดยสองชั้นแรกมีจำนวนนิวรอน 4,096 นิวรอนและชั้นสุดท้ายเป็นชั้น Output ที่มีจำนวนนิวรอน 3 นิวรอนตามจำนวนประเภทที่ต้องการจำแนก
- ชั้น Activation (ReLU): ฟังก์ชัน ReLU ใช้เป็นฟังก์ชันการกระตุ้นในชั้น Convolutional และ Fully Connected
- ชั้น Softmax: ใช้เป็นชั้นการกระตุ้นในชั้น Output เพื่อคำนวณความน่าจะเป็นของแต่ละประเภทที่จำแนก

ในงานวิจัยนี้ได้ใช้แบบจำลอง VGG19 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Fune) ต่อชั้น Convolutional หรือชั้น Dense ที่มีอยู่ ซึ่งน้ำหนักที่ได้จากการฝึกฝนบนชุดข้อมูล ImageNet ถูกใช้โดยตรง โดยไม่มีการเปลี่ยนแปลงใด ๆ
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune) โดยปรับแต่งกลุ่มชั้นที่ 5 (Convolutional Block 5) ซึ่งเป็นกลุ่มชั้น Convolutional สุดท้ายให้เป็นชั้นที่มีน้ำหนัก (Trainable Layers) ก่อนจะเข้าชั้นสำหรับการจำแนก (Classification Layers)

3.5.4 แบบจำลอง ResNet50

แบบจำลอง ResNet50 เป็นหนึ่งในแบบจำลอง ResNet (Residual Networks) ประกอบด้วยชั้นที่มีน้ำหนัก (Trainable Layers) ทั้งหมด 50 ชั้น และมีชั้นที่ไม่มีน้ำหนัก (Non-Trainable Layers) เช่น ชั้น Batch Normalization เป็นต้น โดยรายละเอียดของแต่ละชั้นมีดังนี้

- ชั้น Convolutional (Conv2D): ชั้นแรกของแบบจำลองใช้ตัวกรองขนาดใหญ่ 7x7 สำหรับจับคู่ลักษณะเด่นระดับสูง (High-Level Features)
- ชั้น Max Pooling (MaxPooling2D): ชั้นนี้ตามหลังชั้น Convolutional แรก เพื่อลดขนาดของ Feature Maps
- ชั้น Residual Blocks: ประกอบด้วยชุดของชั้น Convolutional ที่มีการเชื่อมต่อแบบ Skip Connections ที่ช่วยให้ Gradient สามารถไหลผ่านแบบจำลองได้ดีขึ้น และช่วยป้องกันปัญหา Gradient Vanishing ซึ่งชั้น Residual Blocks จะต่อกันหลาย ๆ ชุด
- ชั้น Batch Normalization: ชั้นนี้ใช้ในตัว Residual Blocks ช่วยในการทำให้โครงสร้างเสถียรมากขึ้น และเร่งความเร็วในการฝึกฝน
- ชั้น Activation Function (ReLU): ใช้เป็นฟังก์ชันการกระตุ้นหลังจากชั้น Convolutional และ Batch Normalization
- ชั้น Average Pooling (AveragePooling2D): ใช้ในตอนท้ายของแบบจำลองก่อนชั้น Output เพื่อลดขนาด Feature Map และเตรียมสำหรับการจำแนก
- ชั้น Dropout: ใช้เพื่อป้องกันการ Overfitting ในช่วงการฝึกแบบจำลอง
- ชั้น Fully Connected (Dense): มี 3 ชั้นที่ปลายสุดของแบบจำลอง โดยสองชั้นแรกมีจำนวนนิวรอน 4,096 นิวรอนและชั้นสุดท้ายเป็นชั้น Output ที่มีจำนวนนิวรอน 3 นิวรอน ตามจำนวนประเภทที่ต้องการจำแนก
- ชั้น Softmax: ใช้เป็นชั้นการกระตุ้นในชั้น Output เพื่อคำนวณความน่าจะเป็นของแต่ละประเภทที่จำแนก

ในงานวิจัยนี้ได้ใช้แบบจำลอง ResNet50 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) ต่อชั้น convolutional หรือชั้น dense ที่มีอยู่ ซึ่งน้ำหนักที่ได้จากการฝึกฝนบนชุดข้อมูล ImageNet ถูกใช้โดยตรง โดยไม่มีการเปลี่ยนแปลงใด ๆ

2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune) โดยปรับแต่ง 20 ชั้นสุดท้ายให้เป็นชั้นที่มีน้ำหนัก (Trainable Layers) ก่อนจะเข้าชั้นสำหรับการจำแนก (Classification Layers)

3.5.5 **แบบจำลอง** EfficientNetB0

แบบจำลอง EfficientNetB0 เป็นแบบจำลองที่มีขนาดเล็กที่สุด ในกลุ่มแบบจำลอง EfficientNet ที่ถูกปรับขนาด โดยใช้วิธี Compound Scaling แต่เนื่องจากแบบจำลองถูกออกแบบมาเพื่อให้สามารถปรับขนาดได้ในสามมิติได้ ทั้งความกว้าง (Width), ความลึก (Depth) และความละเอียด (Resolution) ทำให้ไม่สามารถระบุจำนวนชั้นที่มีน้ำหนัก (Trainable Layers) แน่นอนได้ โดยรายละเอียดของแต่ละชั้นมีดังนี้

- ชั้น Convolutional: มีชั้น Convolutional หลายชั้นที่ใช้ในการสกัดคุณลักษณะของรูปภาพ เพื่อสร้างการเรียนรู้เกี่ยวกับรายละเอียดและลักษณะต่าง ๆ ของภาพ
- ชั้น Batch Normalization: ปรับเทียบค่าภายในข้อมูล เพื่อช่วยให้โครงข่ายเสถียรและเร่งความเร็วในการฝึกสอนแบบจำลอง
- ชั้น Depth Wise Separable Convolutions: รวมการทำงานของชั้น Convolutional ที่แยกการทำงานตามความลึกของช่องสี (Channel) และชั้น Point Wise Convolution ที่นำข้อมูลจากชั้น Depth Wise มารวมกัน เพื่อลดความซับซ้อนและประหยัดทรัพยากรในการคำนวณ
- ชั้น Global Average Pooling: ชั้นนี้ตามหลังชั้น Convolutional แรกเพื่อลดขนาดของ Feature Maps
- ชั้น Swish Activation: ฟังก์ชันการกระตุ้นที่ใช้ในแบบจำลอง EfficientNet เป็นการปรับปรุงเวอร์ชันของ ReLU ในชั้นต่าง ๆ เพื่อเพิ่มการเรียนรู้และการสังเคราะห์ลักษณะของภาพ
- ชั้นSqueeze and Excitation (SE) Blocks: ชุดของชั้นที่ช่วยปรับขนาดค่า Channel-wise ใน Feature Map เพื่อเน้นหรือลดความสำคัญของลักษณะเด่น
- ชั้น Dropout: ใช้เพื่อป้องกันการ Overfitting ในช่วงการฝึกแบบจำลอง
- ชั้น Fully Connected (Dense): มี 3 ชั้นที่ปลายสุดของแบบจำลอง โดยสองชั้นแรกมีจำนวนนิวรอน 4,096 นิวรอนและชั้นสุดท้ายเป็นชั้น Output ที่มีจำนวนนิวรอน 3 นิวรอนตามจำนวนประเภทที่ต้องการจำแนก

- ชั้น Softmax: ใช้เป็นชั้นการกระตุ้นในชั้น Output เพื่อคำนวณความน่าจะเป็นของแต่ละประเภทที่จำแนก

ในงานวิจัยนี้ได้ใช้แบบจำลอง EfficientNetB0 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) ต่อชั้น Convolutional หรือชั้น Dense ที่มีอยู่ ซึ่งน้ำหนักที่ได้จากการฝึกฝนบนชุดข้อมูล ImageNet ถูกใช้โดยตรง โดยไม่มีการเปลี่ยนแปลงใด ๆ

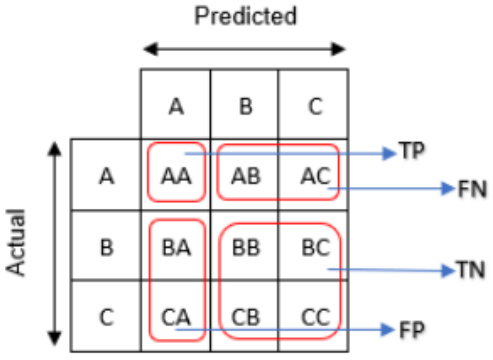
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune) โดยปรับแต่ง 20 ชั้นสุดท้าย ยกเว้นชั้น Batch Normalization ให้เป็นชั้นที่มีน้ำหนัก (Trainable Layers) ก่อนจะเข้าชั้นสำหรับการจำแนก (Classification Layers)

3.6 การประเมินผลแบบจำลอง (Model Evaluation)

ในการประเมินผลแบบจำลองโครงสร้างต่าง ๆ เพื่อดูประสิทธิภาพในการจำแนกข้อมูลภาพ รวมถึงเวลาที่ใช้ในการเรียนรู้ (Training Time) ผู้วิจัยได้กำหนดตัววัดประสิทธิภาพของแบบจำลองโดยใช้ Performance Metrics ได้แก่ ค่าความถูกต้อง (Accuracy), ค่าความแม่นยำ (Precision), ค่าความไว (Recall) และค่า F-1 Score

3.6.1 Confusion Matrix

Confusion Matrix คือ ตารางที่แสดงจำนวนข้อมูลที่แบบจำลองทำนายได้อย่างถูกต้องและไม่ถูกต้อง โดยงานวิจัยนี้เป็นแบบ Multiclass Classification ซึ่งจำแนกข้อมูล 3 ประเภท และสามารถหาค่า True Positive (TP), False Positive (FP), True Negative (TN) และ False Negative (FN) ได้จากผลรวมจำนวนข้อมูลที่แสดงไว้ตามภาพประกอบ 25



ภาพประกอบ 25 แสดง Confusion Matrix ขนาด 3×3

ที่มา: (Wabang และคณะ, 2022)

3.6.2 ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้อง คือ สัดส่วนของข้อมูลที่แบบจำลองทำนายได้อย่างถูกต้องต่อข้อมูลทั้งหมด แสดงดังสมการที่ 3.1

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.1}$$

3.6.3 ค่าความแม่นยำ (Precision)

ค่าความแม่นยำ คือ สัดส่วนของข้อมูลที่แบบจำลองทำนายเชิงบวกที่ถูกต้อง (TP) ต่อข้อมูลเชิงบวกทั้งหมด แสดงดังสมการที่ 3.2

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

3.6.4 ค่าความไว (Sensitivity) หรือ Recall หรือ True Positive Rate (TPR)

ค่าความไว หรือ Recall คือ สัดส่วนของข้อมูลที่แบบจำลองทำนายเชิงบวกที่ถูกต้อง (TP) ต่อข้อมูลเชิงบวกจริงที่ตรวจพบได้อย่างถูกต้อง ใช้ในการวัดว่าแบบจำลองสามารถตรวจจับกรณีเชิงบวกทั้งหมดได้ดีเพียงใด (TPR) แสดงดังสมการที่ 3.3

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

3.6.5 ค่า F-1 Score

F-1 Score คือ ค่าเฉลี่ยฮาร์มอนิกของ Precision และ Recall ใช้ในการวัดประสิทธิภาพโดยรวมของแบบจำลอง แสดงดังสมการที่ 3.4

$$\text{F-1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{3.4}$$

ในงานวิจัยนี้ประกอบด้วย 4 ขั้นตอนหลัก ดังนี้ 1. การเก็บรวบรวมข้อมูล (Image Acquisition) โดยใช้ข้อมูลภาพถั่วพิสตาชิโอ 3 สายพันธุ์ 2. การเตรียมข้อมูล (Pre-Processing) ปรับขนาดภาพเป็น 224×244 พิกเซล และปรับสีพื้นหลังของข้อมูลภาพก่อนส่งข้อมูลเข้าแบบจำลอง 3. การสร้างแบบจำลอง (Modeling) ทั้งหมด 5 ประเภท และ 4. การประเมินผลแบบจำลอง (Model Evaluation) ตามลำดับ

บทที่ 4
ผลการดำเนินการวิจัย

ในการวิจัยศึกษาการสร้างแบบจำลองเพื่อจำแนกสายพันธุ์ถั่วพิสตาชิโอ 3 สายพันธุ์ (Kirmizi, Siirt และ Kerman) ที่ใช้ข้อมูลภาพ โดยใช้เทคนิคการเรียนรู้เชิงลึก ผู้วิจัยได้ดำเนินการวิจัยโดยการศึกษาตามขบวนการและขั้นตอนต่าง ๆ ตลอดจนการประเมินประสิทธิภาพของแบบจำลอง เพื่อให้สอดคล้องกับสมมติฐานที่กำหนดไว้ โดยกำหนดการทดลองไว้ดังนี้

1. แบบจำลอง CNN from Scratch
2. แบบจำลอง VGG16
3. แบบจำลอง VGG19
4. แบบจำลอง ResNet50
5. แบบจำลอง EfficientNetB0

4.1 แบบจำลอง CNN from Scratch

การเปรียบเทียบผลลัพธ์จากการใช้แบบจำลอง CNN from Scratch โดยมีการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์ 2 ค่า คือ อัตราการเรียนรู้ (Learning Rate) และขนาดของกลุ่มข้อมูล (Batch Size) ได้ผลการดำเนินการดังตาราง 7

ตาราง 7 การเปรียบเทียบประสิทธิภาพของแบบจำลอง CNN from Scratch

Learning Rate	Batch Size	Precision	Recall	F-1 Score	Accuracy
10^{-4}	8	0.9034	0.9000	0.9003	0.9000
	16	0.8966	0.8969	0.8966	0.8969
	32	0.8903	0.8906	0.8895	0.8906
	64	0.8965	0.8969	0.8965	0.8969
10^{-5}	8	0.8796	0.8750	0.8754	0.8750
	16	0.9092	0.9094	0.9092	0.9094
	32	0.8719	0.8719	0.8721	0.8719
	64	0.9218	0.9219	0.9218	0.9219

จากตาราง 7 พบว่าแบบจำลอง CNN from Scratch มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 64 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9219 และใช้เวลาในการฝึกข้อมูล 1,297.10 วินาที หรือ 21 นาที 38 วินาที ซึ่งสามารถดูข้อมูลเพิ่มเติมได้จากตาราง 14 ในส่วนของภาคผนวก

4.2 **แบบจำลอง** VGG16

ในงานวิจัยนี้ได้ใช้แบบจำลอง VGG16 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune)
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune)

การเปรียบเทียบผลลัพธ์จากการใช้แบบจำลอง VGG16 ทั้ง Non-Fine Tune และ Fine Tune โดยมีการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์ 2 ค่า คือ อัตราการเรียนรู้ (Learning Rate) และขนาดของกลุ่มข้อมูล (Batch Size) ได้ผลการดำเนินการ ดังตาราง 8

ตาราง 8 การเปรียบเทียบประสิทธิภาพของแบบจำลอง VGG16

Model	Learning Rate	Batch Size	Precision	Recall	F-1 Score	Accuracy
Non-Fine Tune	10^{-4}	8	0.9205	0.9156	0.9161	0.9156
		16	0.9471	0.9469	0.9467	0.9469
		32	0.9658	0.9625	0.9623	0.9625
		64	0.9596	0.9594	0.9594	0.9594
	10^{-5}	8	0.9627	0.9625	0.9625	0.9625
		16	0.9562	0.9563	0.9562	0.9563
		32	0.9564	0.9563	0.9561	0.9563
		64	0.9711	0.9688	0.9685	0.9688
Fine Tune	10^{-4}	8	0.9724	0.9719	0.9718	0.9719
		16	0.9689	0.9688	0.9688	0.9688
		32	0.9656	0.9656	0.9656	0.9656
		64	0.9377	0.9313	0.9299	0.9313
	10^{-5}	8	0.9781	0.9781	0.9781	0.9781
		16	0.9688	0.9688	0.9688	0.9688
		32	0.9719	0.9719	0.9719	0.9719
		64	0.947	0.9469	0.9469	0.9469

จากตาราง 8 อธิบายผลการดำเนินการได้ดังนี้

1. ในกรณีที่เป็นแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 64 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9688

2. ในกรณีที่เป็นแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 8 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9781

เมื่อทำการเปรียบเทียบแบบจำลองทั้ง 2 รูปแบบ พบว่าแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพดีกว่าแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9781และใช้เวลาในการฝึกข้อมูล 1,326.32 วินาที หรือ 22 นาที 7 วินาที ซึ่งสามารถดูข้อมูลเพิ่มเติมได้จากตาราง 15 ในส่วนของภาคผนวก

4.3 **แบบจำลอง** VGG19

ในงานวิจัยนี้ได้ใช้แบบจำลอง VGG19 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune)
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune)

การเปรียบเทียบผลลัพธ์จากการใช้แบบจำลอง VGG19 ทั้ง Non-Fine Tune และ Fine Tune โดยมีการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์ 2 ค่า คือ อัตราการเรียนรู้ (Learning Rate) และขนาดของกลุ่มข้อมูล (Batch Size) ได้ผลการดำเนินการ ดังตาราง 9

ตาราง 9 การเปรียบเทียบประสิทธิภาพของแบบจำลอง VGG19

Model	Learning Rate	Batch Size	Precision	Recall	F-1 Score	Accuracy
Non-Fine Tune	10^{-4}	8	0.9596	0.9594	0.9595	0.9594
		16	0.9659	0.9656	0.9657	0.9656
		32	0.9657	0.9656	0.9655	0.9656
		64	0.9533	0.9531	0.9532	0.9531
	10^{-5}	8	0.9689	0.9688	0.9688	0.9688
		16	0.9689	0.9688	0.9688	0.9688
		32	0.9571	0.9563	0.9561	0.9563
		64	0.9692	0.9688	0.9687	0.9688
Fine Tune	10^{-4}	8	0.9264	0.9250	0.9252	0.9250
		16	0.9765	0.9750	0.9748	0.9750
		32	0.9753	0.9750	0.9749	0.9750
		64	0.9563	0.9563	0.9563	0.9563
	10^{-5}	8	0.9695	0.9688	0.9686	0.9688
		16	0.9594	0.9563	0.9557	0.9563
		32	0.9783	0.9781	0.9781	0.9781
		64	0.9657	0.9656	0.9656	0.9656

จากตาราง 9 อธิบายผลการดำเนินการได้ดังนี้

1. ในกรณีที่เป็นแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 64 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9688

2. ในกรณีที่เป็นแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-4} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 16 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9750

เมื่อทำการเปรียบเทียบแบบจำลองทั้ง 2 รูปแบบ พบว่าแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพดีกว่าแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9750 และใช้เวลาในการฝึกข้อมูล 1,251.77 วินาที หรือ 20 นาที 52 วินาที ซึ่งสามารถดูข้อมูลเพิ่มเติมได้จากตาราง 16 ในส่วนของภาคผนวก

4.4 แบบจำลอง ResNet50

ในงานวิจัยนี้ได้ใช้แบบจำลอง ResNet50 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune)
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune)

การเปรียบเทียบผลลัพธ์จากการใช้แบบจำลอง ResNet50 ทั้ง Non-Fine Tune และ Fine Tune โดยมีการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์ 2 ค่า คือ อัตราการเรียนรู้ (Learning Rate) และขนาดของกลุ่มข้อมูล (Batch Size) ได้ผลการดำเนินการ ดังตาราง 10

ตาราง 10 การเปรียบเทียบประสิทธิภาพของแบบจำลอง ResNet50

Model	Learning Rate	Batch Size	Precision	Recall	F-1 Score	Accuracy
Non-Fine Tune	10^{-4}	8	0.9758	0.9750	0.9749	0.9750
		16	0.9347	0.9281	0.9285	0.9281
		32	0.9660	0.9656	0.9657	0.9656
		64	0.9595	0.9594	0.9594	0.9594
	10^{-5}	8	0.9631	0.9625	0.9626	0.9625
		16	0.9641	0.9625	0.9629	0.9625
		32	0.9657	0.9656	0.9655	0.9656
		64	0.9659	0.9656	0.9657	0.9656
Fine Tune	10^{-4}	8	0.9750	0.9750	0.9750	0.9750
		16	0.9814	0.9812	0.9813	0.9812
		32	0.9674	0.9656	0.9653	0.9656
		64	0.9753	0.9750	0.9749	0.9750
	10^{-5}	8	0.9599	0.9594	0.9595	0.9594
		16	0.9505	0.95	0.9498	0.9500
		32	0.9688	0.9688	0.9688	0.9688
		64	0.9324	0.9313	0.9317	0.9312

จากตาราง 10 อธิบายผลการดำเนินการได้ดังนี้

1. ในกรณีที่เป็นแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-4} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 8 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9750

2. ในกรณีที่เป็นแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-4} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 16 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9812

เมื่อทำการเปรียบเทียบแบบจำลองทั้ง 2 รูปแบบ พบว่าแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพดีกว่าแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9812 และใช้เวลาในการฝึกข้อมูล 1,373.19 วินาที หรือ 22 นาที 54 วินาที ซึ่งสามารถดูข้อมูลเพิ่มเติมได้จากตาราง 17 ในส่วนของภาคผนวก

4.5 **แบบจำลอง** EfficientNetB0

ในงานวิจัยนี้ได้ใช้แบบจำลอง EfficientNetB0 โดยมีการใช้แบบจำลองใน 2 รูปแบบ ดังนี้

1. แบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune)
2. แบบจำลองที่ทำการปรับแต่ง (Fine Tune)

การเปรียบเทียบผลลัพธ์จากการใช้แบบจำลอง EfficientNetB0 ทั้ง Non-Fine Tune และ Fine Tune โดยมีการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์ 2 ค่า คือ อัตราการเรียนรู้ (Learning Rate) และ ขนาดของกลุ่มข้อมูล (Batch Size) ได้ผลการดำเนินการ ดังตาราง 11

ตาราง 11 การเปรียบเทียบประสิทธิภาพของแบบจำลอง EfficientNetB0

Model	Learning Rate	Batch Size	Precision	Recall	F-1 Score	Accuracy
Non-Fine Tune	10^{-4}	8	0.8715	0.7875	0.7779	0.7875
		16	0.9573	0.9531	0.9525	0.9531
		32	0.9266	0.9125	0.9125	0.9125
		64	0.9389	0.9344	0.9351	0.9344
	10^{-5}	8	0.8884	0.8469	0.8327	0.8469
		16	0.9348	0.9281	0.9288	0.9281
		32	0.9371	0.9313	0.9316	0.9313
		64	0.9594	0.9563	0.9565	0.9563
Fine Tune	10^{-4}	8	0.9327	0.9313	0.9317	0.9313
		16	0.9372	0.9250	0.9232	0.9250
		32	0.9246	0.9094	0.9093	0.9094
		64	0.9260	0.9125	0.9095	0.9125
	10^{-5}	8	0.9192	0.9062	0.9067	0.9062
		16	0.9308	0.9156	0.9130	0.9156
		32	0.9592	0.9594	0.9592	0.9594
		64	0.9628	0.9625	0.9625	0.9625

จากตาราง 11 อธิบายผลการดำเนินการได้ดังนี้

1. ในกรณีที่เป็นแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 64 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9563

2. ในกรณีที่เป็นแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพที่ดีที่สุดเมื่ออัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 64 โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9625

เมื่อทำการเปรียบเทียบแบบจำลองทั้ง 2 รูปแบบ พบว่าแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพดีกว่าแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune) โดยได้ค่าความถูกต้อง (Accuracy) สูงสุดที่ 0.9625 และใช้เวลาในการฝึกข้อมูล 1,262.13 วินาที หรือ 21 นาที 3 วินาที ซึ่งสามารถดูข้อมูลเพิ่มเติมได้จากตาราง 18 ในส่วนของภาคผนวก

ในบทนี้ได้แสดงผลลัพธ์จากการศึกษาวิจัย พบว่าแบบจำลอง ResNet50 เป็นแบบจำลองที่มีประสิทธิภาพสูงสุด โดยมีค่าความถูกต้องอยู่ที่ 0.9812 ตามด้วยแบบจำลอง VGG16 มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9781, แบบจำลอง VGG19 มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9750, แบบจำลอง EfficientNetB0 มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9625 และแบบจำลองที่มีการเรียนรู้ตั้งแต่เริ่มต้น มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9219 ตามลำดับ

บทที่ 5
สรุปผลการวิจัย อภิปรายผลการวิจัย และข้อเสนอแนะ

ในการวิจัยนี้ได้วัดประสิทธิภาพของแบบจำลองทั้ง 5 ประเภท ได้แก่ แบบจำลอง CNN from Scratch, VGG16, VGG19, ResNet50 และ EfficientNetB0 เพื่อนำมาเปรียบเทียบและสรุปผล โดยสามารถแบ่งหัวข้อในการสรุปผลได้ดังต่อไปนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการศึกษาและเปรียบเทียบประสิทธิภาพของแบบจำลองที่ใช้ในงานวิจัยนี้ ทั้งหมด 5 ประเภท ได้แก่ CNN from Scratch, VGG16, VGG19, ResNet50 และ EfficientNetB0 เพื่อศึกษาวิเคราะห์และหาไฮเปอร์พารามิเตอร์ที่เหมาะสมของแต่ละแบบจำลอง เพื่อสามารถนำผลลัพธ์ที่ได้ไปปรับใช้ให้เหมาะสมกับการจำแนกชุดข้อมูลภาพถั่วพิสตาชิโอ 3 สายพันธุ์ (Kirmizi, Siirt และ Kerman) ได้อย่างมีประสิทธิภาพสูงสุด โดยการเปรียบเทียบประสิทธิภาพของแต่ละแบบจำลองดังตาราง 12 และสามารถอธิบายสรุปผลการดำเนินการวิจัยได้ดังนี้

1. แบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-Trained Model) โดยใช้วิธีการถ่ายโอนการเรียนรู้ (Transfer Learning) มีประสิทธิภาพดีกว่า แบบจำลองที่มีการเรียนรู้ตั้งแต่เริ่มต้น (CNN from Scratch)

2. สำหรับแบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-Trained Model) พบว่าแบบจำลองที่ทำการปรับแต่ง (Fine Tune) มีประสิทธิภาพดีกว่าแบบจำลองที่ไม่ได้ทำการปรับแต่งเพิ่มเติม (Non-Fine Tune)

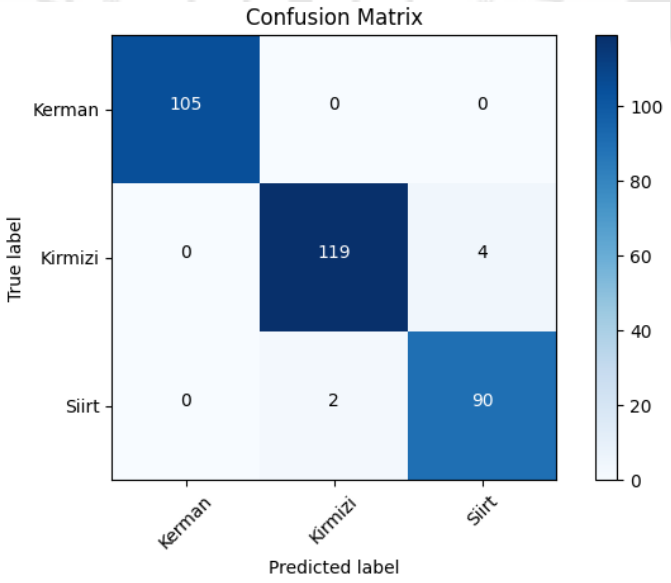
3. แบบจำลอง ResNet50 เป็นแบบจำลองที่มีประสิทธิภาพสูงสุด โดยมีค่าความถูกต้องอยู่ที่ 0.9812 ตามด้วยแบบจำลอง VGG16 มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9781, แบบจำลอง VGG19 มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9750, แบบจำลอง EfficientNetB0 มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9625 และแบบจำลองที่มีการเรียนรู้ตั้งแต่เริ่มต้น มีค่าความถูกต้อง (Accuracy) อยู่ที่ 0.9219 ตามลำดับ

ตาราง 12 การเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 5 ประเภท

Model	Learning Rate	Batch Size	Accuracy
ResNet50 with Fine Tune	10^{-4}	16	0.9812
VGG16 with Fine Tune	10^{-5}	8	0.9781
VGG19 with Fine Tune	10^{-4}	16	0.9750
EfficientNetB0 with Fine Tune	10^{-5}	64	0.9625
CNN from Scratch	10^{-5}	64	0.9219

5.2 อภิปรายผลการวิจัย

ในงานวิจัยนี้ได้ทำการวิเคราะห์ความแตกต่างทางกายภาพของถั่วพิสตาชิโอแต่ละสายพันธุ์ ที่ได้จากการทำนายผลลัพธ์จากข้อมูลภาพ Test Set โดยใช้แบบจำลอง ResNet50 ที่ทำการปรับแต่ง (Fine Tune) อัตราการเรียนรู้ (Learning Rate) เท่ากับ 10^{-5} และ ขนาดของกลุ่มข้อมูล (Batch Size) เท่ากับ 64 ซึ่งเป็นแบบจำลองที่มีประสิทธิภาพสูงสุด

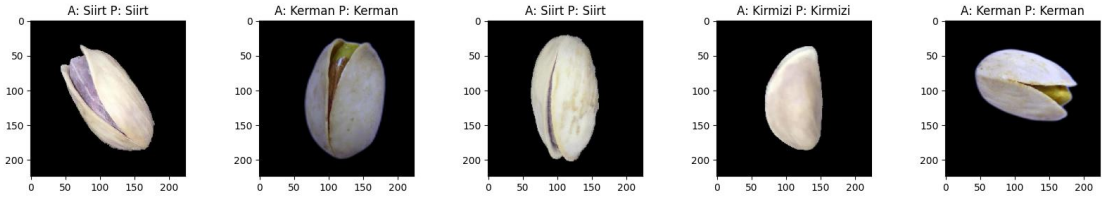


ภาพประกอบ 26 แสดง Confusion Matrix ที่ได้จากแบบจำลอง ResNet50

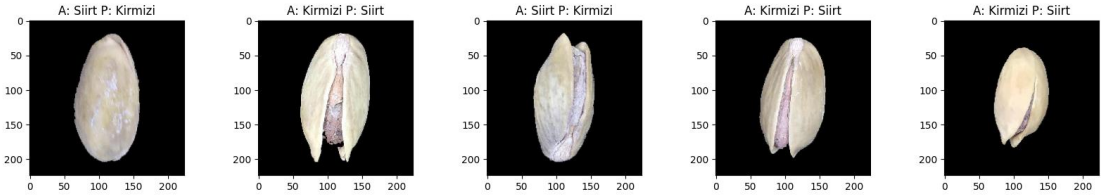
จากภาพประกอบ 26 สามารถอธิบายผลลัพธ์ที่ได้จากการทำนาย ดังนี้

- สายพันธุ์ Kerman มีจำแนกประเภทได้อย่างถูกต้อง 105 ภาพ หรือ 100% โดยไม่มีการจำแนกผิดพลาดสำหรับสายพันธุ์นี้
- สายพันธุ์ Kirmizi มีจำแนกประเภทได้อย่างถูกต้อง 119 ภาพ และมีการจำแนกผิดไปเป็นสายพันธุ์ Siirt 4 ภาพ
- สายพันธุ์ Siirt มีจำแนกประเภทได้อย่างถูกต้อง 90 ภาพ และมีการจำแนกผิดไปเป็นสายพันธุ์ Kirmizi 2 ภาพ

การที่แบบจำลองสามารถจำแนกสายพันธุ์ Kerman ได้อย่างถูกต้อง 100% อาจมีสาเหตุจากเป็นเก็บข้อมูลภาพใหม่เพิ่มเติมในงานวิจัยนี้ แต่สำหรับสายพันธุ์ Kirmizi และ Siirt เป็นข้อมูลภาพที่มาจากข้อมูลสาธารณะ โดยผู้วิจัยคนเดียวกัน (Singh และคณะ, 2022) ทำให้มีความแตกต่างในขั้นตอนการเก็บรวบรวมข้อมูล (Image Acquisition) เช่น คุณภาพของภาพ, ความคมชัด, แสงสว่าง, และความแตกต่างของสีในภาพถ่ายที่อาจส่งผลต่อการจำแนกของแบบจำลอง อีกสาเหตุหนึ่งคือสายพันธุ์ Kerman มีลักษณะทางกายภาพที่แตกต่างกับอีก 2 สายพันธุ์อย่างชัดเจน โดยเฉพาะสีของเนื้อถั่วมีสีเขียวอ่อน และรูปร่างเมล็ดค่อนข้างกลม ตามรายละเอียดลักษณะทางกายภาพของถั่วพิสตาชิโอ ดังตาราง 2 ซึ่งอาจเป็นสาเหตุที่ให้การจำแนกสายพันธุ์ Kerman มีความถูกต้อง 100% ได้ ส่วนสายพันธุ์ Kirmizi และ Siirt มีลักษณะทางกายภาพคล้ายคลึงกันทำให้มีข้อมูลภาพบางส่วนที่แบบจำลองไม่สามารถจำแนกประเภทได้ ดังภาพประกอบ 27 และ 28



ภาพประกอบ 27 แสดงตัวอย่างภาพถั่วพิสตาชิโอที่แบบจำลองทำนายได้ถูกต้อง



ภาพประกอบ 28 แสดงตัวอย่างภาพถั่วพิสตาชิโอที่แบบจำลองทำนายผิดพลาด

ภาพที่แบบจำลองทำนายไม่ถูกต้อง ส่วนใหญ่เป็นภาพที่อาจเกิดจากสาเหตุจากการวางตำแหน่งและมุมมอง ทำให้เห็นสีของเปลือกหุ้มเมล็ดไม่ชัดเจน หรือขนาดและสีของเนื้อถั่วที่เปลี่ยนแปลง ซึ่งสร้างความสับสนให้กับแบบจำลองในทำนายได้ อาจต้องดำเนินการปรับแต่งหรือการเพิ่มข้อมูลภาพแต่ละมุมมอง เพื่อให้แบบจำลองมีการเรียนรู้ที่ดีขึ้น โดยอธิบายความแตกต่างของแต่ละสายพันธุ์ได้ดังนี้

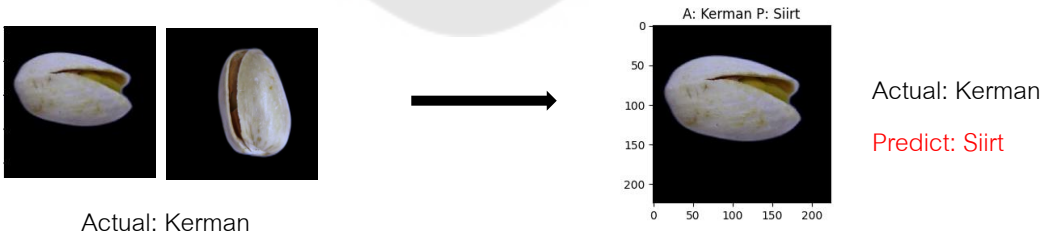
1. สายพันธุ์ Kirmizi เป็นตัวอย่างภาพที่ใช้ในการเรียนรู้ (ภาพซ้าย) โดยเมื่อนำแบบจำลองไปใช้ทำนายภาพ พบว่าทำนายผิดเป็นสายพันธุ์ Siirt (ภาพขวา) ดังภาพประกอบ 29 หากพิจารณาการที่แบบจำลองทำนายผิดพลาด อาจเนื่องมาจากรูปร่าง สีของเปลือกและสีของเปลือกหุ้มเมล็ดมีสีอมน้ำตาล มุมของภาพด้านที่เป็นปากเมล็ด ซึ่งมีความใกล้เคียงกับลักษณะของสายพันธุ์ Siirt รวมทั้งการไม่เห็นสีของเนื้อถั่วภายใน จึงทำให้การทำนายภาพนี้เป็นสายพันธุ์ Siirt ได้

ภาพประกอบ 29 แสดงภาพถั่วพิสตาชิโอสายพันธุ์ Kirmizi ที่ถูกทำนายผิดเป็นสายพันธุ์ Siirt

2. สายพันธุ์ Siirt เป็นตัวอย่างภาพที่ใช้ในการเรียนรู้ (ภาพซ้าย) โดยเมื่อนำแบบจำลองไปใช้ทำนายภาพ พบว่าทำนายผิดเป็นสายพันธุ์ Kirmizi (ภาพขวา) ดังภาพประกอบ 30 หากพิจารณาการที่แบบจำลองทำนายผิดพลาด อาจเนื่องมาจากรูปร่าง และสีของเปลือกหุ้มเมล็ดเป็นน้ำตาลอมม่วง รวมทั้งมุมของภาพด้านที่เป็นปากเมล็ด ซึ่งมีความใกล้เคียงกับลักษณะของสายพันธุ์ Kirmizi จึงทำให้การทำนายผิดพลาดได้

ภาพประกอบ 30 แสดงภาพถั่วพิสตาชิโอสายพันธุ์ Siirt ที่ถูกทำนายผิดเป็นสายพันธุ์ Kirmizi

3. สายพันธุ์ Kerman เป็นตัวอย่างภาพที่ใช้ในการเรียนรู้ (ภาพซ้าย) โดยเมื่อนำแบบจำลองไปใช้ทำนายภาพ พบว่าทำนายผิดเป็นสายพันธุ์ Siirt (ภาพขวา) ดังภาพประกอบ 31 หากพิจารณาการที่แบบจำลองทำนายผิดพลาด อาจเนื่องมาจากมุมในการถ่ายภาพ ทำให้รูปร่างและสีเปลือกหุ้มเมล็ดใกล้เคียงกับสายพันธุ์ Siirt และ Kirmizi แต่เมื่อพิจารณาจากสีเนื้อของถั่วพิสตาชิโอของรูปนี้ออกสีเขียวอมน้ำตาล ซึ่งใกล้เคียงกับสายพันธุ์ Siirt มากกว่า จึงทำให้การทำนายผิดพลาดไปเป็นสายพันธุ์ Siirt ได้



ภาพประกอบ 31 แสดงภาพถั่วพิสตาชิโอสายพันธุ์ Kerman ที่ถูกทำนายผิดเป็นสายพันธุ์ Siirt

5.3 ข้อเสนอแนะ

ในงานวิจัยนี้ผู้วิจัยได้มีข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาต่อไปไว้ ดังนี้

1. การเพิ่มข้อมูลภาพถั่วพิสตาชิโอที่มีความหลากหลายมากขึ้น เพื่อช่วยให้แบบจำลองสามารถเรียนรู้ลักษณะเฉพาะของแต่ละสายพันธุ์ได้ดีขึ้น เช่น การเพิ่มจำนวนภาพถั่วพิสตาชิโอจาก 1,000 ภาพ เป็น 2,000 ภาพ เพื่อครอบคลุมสายพันธุ์และมุมมองที่หลากหลายมากขึ้น

2. การปรับปรุงโครงสร้างแบบจำลอง การเพิ่มหรือปรับแต่งชั้นของแบบจำลอง เช่น การเพิ่มชั้น Convolutional หรือการใช้แบบจำลอง Pre-Trained อื่น ๆ เพื่อปรับปรุงประสิทธิภาพของแบบจำลองในการจำแนก

3. การคำนึงถึงความหลากหลายทางกายภาพ โดยพิจารณาความหลากหลายทางกายภาพและพันธุกรรมของสายพันธุ์ถั่วพิสตาชิโอ โดยเฉพาะอย่างยิ่งจากสายพันธุ์ที่มาจากต่างภูมิภาคกัน

4. การปรับแต่งแบบจำลอง (Tuning Hyperparameters) โดยใช้เทคนิคอื่น ๆในการปรับแต่งแบบจำลองเช่น Grid Search หรือ Bayesian Optimization เพื่อหา Hyperparameters ที่เหมาะสมที่สุด

5. การปรับค่า Hyperparameters ต่าง ๆ เช่น การปรับค่า Dropout ที่ 0.2 ถึง 0.5 หรือการใช้เทคนิค Weight Decay L1/L2 regularization เพื่อลดการ Overfitting หรือการปรับค่า Learning Rate ช่วงที่กว้างขึ้น 0.001 ถึง 0.1 หรือการปรับค่า Epoch เพื่อเพิ่มความเร็วในการฝึกฝน เป็นต้น

6. การวิจัยในอนาคตควรพิจารณาการใช้งานแบบจำลองที่ได้รับการเรียนรู้ข้อมูลมาก่อน (Pre-Trained Model) เพื่อจำแนกลักษณะเฉพาะของถั่วพิสตาชิโอ เช่น การตรวจสอบถั่วที่มีเปลือกปิดซึ่งไม่สามารถเปิดได้ง่าย ซึ่งความสำคัญต่อการประเมินคุณภาพผลิตภัณฑ์และส่งผลต่อกระบวนการผลิตและการบริโภค การประยุกต์ใช้เทคโนโลยีจำแนกอัตโนมัติจะเพิ่มประสิทธิภาพในการคัดเลือกและปรับปรุงกระบวนการผลิตในอุตสาหกรรมมากขึ้น

บรรณานุกรม

Biodels. (2021). Pistachio Kernels. http://biodels.com/content/pistachio-kernels

He, K., Zhang, X., Ren, S., และ Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778. https://doi.org/10.1109/CVPR.2016.90

Jaroensri, P., และ Srimontrinond, M. (2566, 3 กุมภาพันธ์). ประวัติและหลักการของเครือข่ายประสาทเทียม (Artificial Neural Networks – ANNs). Big Data Thailand. https://bdi.or.th/big-data-101/neural-network/

Rahimzadeh, M., และ Attar, A. (2022). Detecting and counting pistachios based on deep learning. Iran Journal of Computer Science, 5(1), 69-81. https://doi.org/10.1007/s42044-021-00090-6

Singh, D., Taspinar, Y. S., Kursun, R., Cinar, I., Koklu, M., Ozkan, I. A., และ Lee, H.-N. (2022). Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models. Electronics, 11(7), 981. https://doi.org/10.3390/electronics11070981

Soleimanipour, A., Azadbakht, M., และ Rezaei Asl, A. (2022). Cultivar identification of pistachio nuts in bulk mode through EfficientNet deep learning model. Journal of Food Measurement and Characterization, 16(4), 2545-2555. https://doi.org/10.1007/s11694-022-01367-5

Solo, D. (2022a). American pistachio industry: Borne from a single Iranian seed. Khoshbin Group. https://ratinkhosh.com/american-pistachios/

Solo, D. (2022b). Turkish pistachio industry: A slow rising supplier. Khoshbin Group. https://ratinkhosh.com/turkish-pistachios/

Tammina, S. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications, 9(10), 143-150. https://doi.org/10.29322/IJSRP.9.10.2019.p9420

Tan, M., และ Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 36th International Conference on Machine Learning (ICML 2019). https://doi.org/https://doi.org/10.48550/arXiv.1905.11946

Wabang, K., Nurhayati, O. D., และ Farikhin. (2022). Application of The Naïve Bayes Classifier Algorithm to Classify Community Complaints. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 2022, 872-876.

Wanli, X., และ Dongping, D. (2018). Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention. Journal of Educational Computing Research, 57(4), 1-23. https://doi.org/10.1177/0735633118757015

Wikipedia. (2024, April 28). Pistachio. https://en.wikipedia.org/wiki/Pistachio

Wikipedia. (2566, 1 กรกฎาคม). โครงร่างประสาทเทียม. https://th.wikipedia.org/wiki/โครงข่ายประสาทเทียม

รัสรินทร์ เมธาเฉลิมพัฒน์. (2565, 22 สิงหาคม). การประยุกต์ใช้ Machine Learning กับงานในภาคอุตสาหกรรม (ตอนที่ 1). NECTEC : National Electronics and Computer Technology Center. https://www.nectec.or.th/news/news-public-document/machine-learning-manufact-1.html

ภาคผนวก

ตาราง 13 แสดงจำนวนพารามิเตอร์ทั้งหมดของแบบจำลองต่าง ๆ

Model		Trainable Parameters	Non-Trainable Parameters	Total Parameters
CNN from Scratch		399,008,739 (1.49 GB)	0 (0.00 Byte)	399,008,739 (1.49 GB)
VGG16	Non-Fine Tune	119,558,147 (456.08 MB)	14,714,688 (56.13 MB)	134,272,835 (512.21 MB)
	Fine Tune	126,637,571 (483.08 MB)	7,635,264 (29.13 MB)	
VGG19	Non-Fine Tune	119,558,147 (456.08 MB)	20,024,384 (76.39 MB)	139,582,531 (532.47 MB)
	Fine Tune	128,997,379 (492.09 MB)	10,585,152 (40.38 MB)	
ResNet50	Non-Fine Tune	427,839,491 (1.59 GB)	23,587,712 (89.98 MB)	451,427,203 (1.68 GB)
	Fine Tune	436,770,819 (1.63 GB)	14,656,384 (55.91 MB)	
EfficientNetB0	Non-Fine Tune	273,698,819 (1.02 GB)	4,049,571 (15.45 MB)	277,748,390 (1.03 GB)
	Fine Tune	275,041,587 (1.02 GB)	2,706,803 (10.33 MB)	

ตาราง 14 แสดงเวลาที่ใช้ในการเรียนรู้ และค่า Validation Accuracy ของแบบจำลอง CNN from Scratch

Learning Rate	Batch Size	Training Time (sec)	Validation Accuracy
10^{-4}	8	1,075.59	0.9031
	16	1,068.54	0.9156
	32	1,154.60	0.9031
	64	1,084.45	0.9031
10^{-5}	8	1,135.08	0.8781
	16	1,103.46	0.9094
	32	1,139.33	0.9187
	64	1,297.10	0.9031

ตาราง 15 แสดงเวลาที่ใช้ในการเรียนรู้ และค่า Validation Accuracy ของแบบจำลอง VGG16

Model	Learning Rate	Batch Size	Training Time (sec)	Validation Accuracy
Non-Fine Tune	10^{-4}	8	1,261.65	0.9875
		16	1,388.10	0.9844
		32	1,589.49	0.9875
		64	1,868.32	0.9969
	10^{-5}	8	1,489.85	0.9969
		16	1,381.24	1.0000
		32	1,420.84	0.9906
		64	1,414.67	0.9812
Fine Tune	10^{-4}	8	1,367.17	0.9906
		16	1,739.86	0.9844
		32	1,293.74	0.9969
		64	1,599.71	0.9937
	10^{-5}	8	1,326.32	0.9969
		16	1,505.46	0.9969
		32	1,373.03	0.9937
		64	1,481.97	0.9875

ตาราง 16 แสดงเวลาที่ใช้ในการเรียนรู้ และค่า Validation Accuracy ของแบบจำลอง VGG19

Model	Learning Rate	Batch Size	Training Time (sec)	Validation Accuracy
Non-Fine Tune	10^{-4}	8	1,716.46	0.9875
		16	1,328.97	0.9906
		32	1,296.42	0.9906
		64	1,552.97	0.9875
	10^{-5}	8	1,226.35	0.9906
		16	1,240.47	0.9906
		32	1,292.83	0.9906
		64	1,217.19	0.9906
Fine Tune	10^{-4}	8	1,300.64	0.9594
		16	1,251.77	0.9906
		32	1,719.44	0.9969
		64	1,463.62	0.9906
	10^{-5}	8	1,144.87	0.9937
		16	1,950.72	1.0000
		32	1,436.65	0.9937
		64	1,542.48	0.9906

ตาราง 17 แสดงเวลาที่ใช้ในการเรียนรู้ และค่า Validation Accuracy ของแบบจำลอง ResNet50

Model	Learning Rate	Batch Size	Training Time (sec)	Validation Accuracy
Non-Fine Tune	10^{-4}	8	1,134.68	0.9844
		16	1,260.41	0.9563
		32	1,118.51	0.9688
		64	1,164.89	0.9656
	10^{-5}	8	1,173.28	0.9688
		16	1,649.23	0.9812
		32	1,548.37	0.9875
		64	1,280.15	0.9844
Fine Tune	10^{-4}	8	1,393.52	0.9750
		16	1,373.19	0.9906
		32	1,306.03	0.9875
		64	1,505.72	0.9875
	10^{-5}	8	1,571.76	0.9781
		16	1,252.70	0.9625
		32	1,373.52	0.9812
		64	1,132.92	0.9688

ตาราง 18 แสดงเวลาที่ใช้ในการเรียนรู้ และค่า Validation Accuracy ของแบบจำลอง EfficientNetB0

Model	Learning Rate	Batch Size	Training Time (sec)	Validation Accuracy
Non-Fine Tune	10^{-4}	8	1,156.82	0.9688
		16	1,411.58	0.9844
		32	1,119.44	0.9844
		64	1,200.24	0.9875
	10^{-5}	8	1,198.48	0.9844
		16	1,240.10	0.9875
		32	1,473.61	0.9906
		64	1,852.19	0.9906
Fine Tune	10^{-4}	8	1,397.74	0.9937
		16	1,360.20	0.9937
		32	1,678.89	0.9969
		64	1,342.39	0.9937
	10^{-5}	8	1,387.88	0.9937
		16	1,464.53	0.9937
		32	1,238.66	0.9750
		64	1,262.13	0.9937

ประวัติผู้เขียน