



## การพัฒนาตัวแบบการเรียนรู้เชิงลึกสำหรับการตรวจจับโรคอ้อยจากภาพถ่ายใบอ้อย

### Development of a Deep Learning Model for Sugarcane Disease Detection Using Leaf Images

สิรภัทร พวงทับทิม<sup>1</sup> พัชรมัย ทองดี<sup>1</sup> ธนพล ตั้งชูพงศ์<sup>1\*</sup>

Siraphat Phuangthapthim<sup>1</sup>, Phatcharamai Thongdee<sup>1</sup>, Thanaphon Tangchoopong<sup>1\*</sup>

<sup>1</sup>สาขาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น

<sup>1</sup>Department of Computer Science, College of Computing, Khon Kaen University.

#### ข้อมูลบทความ

##### ประวัติบทความ

รับเมื่อ: 22 พฤษภาคม 2568

แก้ไขเมื่อ: 5 กันยายน 2568

ตอบรับเมื่อ: 8 กันยายน 2568

เผยแพร่ออนไลน์:

22 กันยายน 2568

##### คำสำคัญ

การเรียนรู้เชิงลึก

การตรวจหาโรคอ้อย

EfficientNetB0

MobileNetV2

การประมวลผลภาพถ่าย

\*ผู้ประพันธ์บรรณกิจ

อีเมล:

[thanaphon@kku.ac.th](mailto:thanaphon@kku.ac.th)

(ธนพล ตั้งชูพงศ์)

#### บทคัดย่อ

การวิจัยนี้มีเป้าหมายเพื่อพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning) สำหรับการตรวจหาและจำแนกโรคในอ้อย 4 ชนิดที่พบในประเทศไทย ได้แก่ โรคใบต่างในอ้อย โรคเหี่ยวเน่าแดง โรคราสนิม และโรคใบไหม้ โดยใช้ภาพถ่ายใบอ้อยเป็นข้อมูลนำเข้า กระบวนการวิจัยเริ่มต้นจากการเตรียมข้อมูลภาพ การเพิ่มความหลากหลายของชุดข้อมูลผ่านกระบวนการเสริมข้อมูล (Data Augmentation) และเพื่อเปรียบเทียบประสิทธิภาพการฝึกสอนโมเดล EfficientNetB0 และ MobileNetV2 ผลการทดลองชี้ให้เห็นว่า EfficientNetB0 มีความแม่นยำสูงสุด ด้วยค่า F1-Score เท่ากับ 0.95 นอกจากนี้ยังได้ศึกษาปัจจัยด้านความสว่างและความเบลของภาพที่มีต่อประสิทธิภาพการทำนายของโมเดล พบว่าทั้งสองปัจจัยไม่มีผลกระทบต่อความสามารถในการทำนาย แสดงให้เห็นถึงความทนทานของโมเดลต่อสภาพแวดล้อมที่เปลี่ยนแปลง งานวิจัยนี้จึงมีศักยภาพในการนำไปประยุกต์ใช้สำหรับการตรวจสอบและจำแนกโรคในอ้อยเพื่อสนับสนุนการจัดการโรคในภาคการเกษตรได้อย่างมีประสิทธิภาพ

## ARTICLE INFO

## Article History

Received: 22 May 2025

Revised: 5 September 2025

Accepted:

8 September 2025

Available online:

22 September 2025

## Keywords:

Deep Learning  
Sugarcane Disease  
Detection  
EfficientNetB0  
MobileNetV2  
Image Processing

## \*Corresponding author

## Email address:

thanaphon@kku.ac.th

(T. Tangchoopong)

## ABSTRACT

This research aims to develop a deep learning model for detecting and classifying four major sugarcane diseases found in Thailand: sugarcane mosaic disease, red rot, rust, and leaf scorch. The study utilizes sugarcane leaf images as input data. The development process includes data preparation, dataset diversification through data augmentation, and training models using EfficientNetB0 and MobileNetV2 to compare performance. The experimental results show that EfficientNetB0 achieved the highest accuracy with an F1-Score of 0.95. Furthermore, an analysis of the effects of image brightness and blur on model performance reveals that these factors do not impact prediction accuracy, demonstrating the model's robustness under varying environmental conditions. This research highlights the potential of applying the developed model for effective sugarcane disease detection and classification, supporting disease management in the agricultural sector.

## 1. บทนำ

อ้อย (Sugarcane) เป็นพืชเศรษฐกิจที่มีความสำคัญอย่างยิ่งต่อระบบเศรษฐกิจของประเทศไทย โดยมีปริมาณผลผลิตสูงเป็นอันดับ 4 ของโลก [14] อย่างไรก็ตาม การแพร่ระบาดของโรคเป็นปัจจัยคุกคามหลักที่ส่งผลกระทบต่ออย่างมีนัยสำคัญต่อเสถียรภาพทางการผลิต ทั้งในมิติของปริมาณและคุณภาพของผลผลิต ซึ่งนำไปสู่ความเสียหายทางเศรษฐกิจต่อภาคเกษตรกรรมของประเทศ โดยโรคที่พบการระบาดบ่อยครั้งและมีความรุนแรง อาทิ โรคเหี่ยวแดง (Red Rot) โรคใบด่าง (Sugarcane Mosaic Virus) และโรคสนิม (Rust) ด้วยเหตุนี้ การพัฒนาเครื่องมือสำหรับวินิจฉัยโรคอ้อยที่รวดเร็วและแม่นยำจึงถือเป็นความท้าทายที่จำเป็นอย่างเร่งด่วน

งานวิจัยฉบับนี้จึงนำเสนอแนวทางการจำแนกโรคอ้อยโดยการบูรณาการเทคโนโลยีการประมวลผลภาพ (Image Processing) เข้ากับเทคนิคการเรียนรู้เชิงลึก (Deep Learning) โดยมุ่งเน้นการคัดเลือกแบบจำลอง (Model) ที่มีขนาดกะทัดรัดและประมวลผลได้อย่างรวดเร็ว ได้แก่ EfficientNetB0 และ MobileNetV2 เนื่องจากสถาปัตยกรรมทั้งสองถูกพัฒนาขึ้นเพื่อการใช้งานบนอุปกรณ์ที่มีทรัพยากรจำกัด (Resource-constrained Devices) เช่น อุปกรณ์เคลื่อนที่ ซึ่งจะช่วยให้เกษตรกรสามารถนำเทคโนโลยีไปใช้ตรวจหาโรคได้ ณ แหล่งเพาะปลูก นำไปสู่การจัดการและควบคุมโรคได้อย่างมีประสิทธิภาพและทันต่อสถานการณ์

## 2. วิธีดำเนินการวิจัย

## 2.1 ศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้จะกล่าวถึงทฤษฎีพื้นฐานและแนวคิดที่สำคัญซึ่งเป็นรากฐานของงานวิจัยฉบับนี้ประกอบด้วยหลักการจำแนกประเภทข้อมูลสถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชัน และเกณฑ์การวัดประสิทธิภาพของแบบจำลอง

## 2.1.1 ทฤษฎีที่เกี่ยวข้อง

(1) การจำแนกประเภท (Classification) เป็นกระบวนการสำคัญในสาขาการเรียนรู้ของเครื่อง (Machine Learning) และวิทยาการข้อมูล (Data Science) ซึ่งมีวัตถุประสงค์เพื่อสร้างแบบจำลอง (Model) ที่สามารถกำหนดหมวดหมู่หรือคลาส (Class) ให้กับข้อมูลที่ไม่เคยพบมาก่อน (Unseen Data) โดยอาศัยการเรียนรู้รูปแบบ (Pattern) จากชุดข้อมูลที่มีการระบุคำตอบที่ถูกต้องไว้แล้ว [1] แบบจำลองจะเรียนรู้ความสัมพันธ์ระหว่างคุณลักษณะ (Features) ของข้อมูลกับคลาสเป้าหมาย เพื่อให้สามารถทำนายคลาสของข้อมูลใหม่ได้อย่างแม่นยำ

(2) โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN) คือ สถาปัตยกรรมโครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network) ประเภทหนึ่ง ซึ่งได้รับการออกแบบมาเพื่อประมวลผลข้อมูลที่มีลักษณะเป็นกริด (Grid-like Data) เช่น รูปภาพ โดยเฉพาะ โครงสร้างหลักประกอบด้วยชั้นนำเข้า (Input Layer) ชั้นซ่อน (Hidden Layers) ซึ่งมีชั้นคอนโวลูชัน (Convolutional Layer) เป็นหัวใจสำคัญ และชั้นผลลัพธ์ (Output Layer) จุดเด่นของ CNN คือความสามารถในการสกัดคุณลักษณะ (Feature Extraction) เชิงพื้นที่ของภาพได้อย่างเป็นลำดับขั้น ตั้งแต่คุณลักษณะพื้นฐาน (เช่น เส้นขอบ, สี) ไปจนถึงคุณลักษณะที่ซับซ้อน (เช่น รูปร่าง, วัตถุ) ทำให้ CNN ถูกนำมาประยุกต์ใช้อย่างกว้างขวางและประสบความสำเร็จอย่างสูงในงาน

ด้านวิทัศน์คอมพิวเตอร์ (Computer Vision) เช่น การจำแนกประเภทภาพ และการตรวจจับวัตถุ [2]

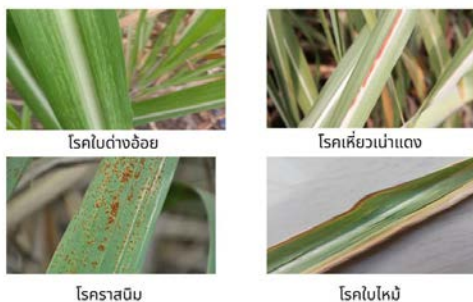
ในงานวิจัยนี้ได้คัดเลือกสถาปัตยกรรม CNN สมัยใหม่ที่มีประสิทธิภาพสูงและเหมาะสำหรับการประยุกต์ใช้บนอุปกรณ์ที่มีทรัพยากรจำกัด จำนวน 2 สถาปัตยกรรม ได้แก่

EfficientNet B0 เป็นสถาปัตยกรรมที่ได้รับการพัฒนาโดย Google Research โดยมีแนวคิดหลักคือการปรับขนาดของแบบจำลองอย่างสมดุลผ่านเทคนิคการปรับขนาดแบบผสม (Compound Scaling) ซึ่งเป็นการปรับทั้งความลึก (Depth) ความกว้าง (Width) และความละเอียดของภาพ (Resolution) ไปพร้อมกัน ทำให้แบบจำลองมีขนาดเล็ก ใช้ทรัพยากรในการประมวลผลน้อย แต่ยังคงไว้ซึ่งความแม่นยำในระดับสูง จึงเหมาะอย่างยิ่งสำหรับงานที่ต้องการประสิทธิภาพและการตอบสนองที่รวดเร็ว เช่น บนอุปกรณ์เคลื่อนที่ [12]

MobileNetV2 เป็นสถาปัตยกรรมที่ถูกพัฒนาต่อยอดมาจาก MobileNet โดยออกแบบมาเพื่อการใช้งานบนอุปกรณ์เคลื่อนที่และอุปกรณ์ฝังตัว (Embedded Systems) โดยเฉพาะ จุดเด่นคือการใช้โครงสร้างแบบ Inverted Residuals และ Linear Bottlenecks ซึ่งช่วยลดจำนวนพารามิเตอร์และลดภาระการคำนวณได้อย่างมีนัยสำคัญ ส่งผลให้แบบจำลองทำงานได้รวดเร็วและใช้หน่วยความจำน้อย แม้จะมีความแม่นยำอาจด้อยกว่าแบบจำลองขนาดใหญ่ แต่ยังคงอยู่ในเกณฑ์ที่ยอมรับได้สำหรับการใช้งานจริง [13]

(3) เมทริกซ์ความสับสน (Confusion Matrix) เป็นเครื่องมือมาตรฐานสำหรับใช้วัดและประเมินประสิทธิภาพของแบบจำลองการจำแนกประเภท โดยแสดงผลในรูปแบบตารางที่เปรียบเทียบระหว่างค่าคลาสจริง (Actual Class) กับค่าคลาสที่แบบจำลองทำนาย (Predicted Class) ตารางนี้ประกอบด้วยองค์ประกอบหลัก 4 ส่วน คือ ค่าบวกจริง (True Positive: TP), ค่าบวกปลอม (False Positive: FP), ค่าลบปลอม (False Negative: FN) และค่าลบจริง (True Negative: TN) ข้อมูลจากเมทริกซ์ความสับสนช่วยให้สามารถวิเคราะห์ความผิดพลาดของแบบจำลองได้อย่างละเอียด และเป็นพื้นฐานในการคำนวณตัวชี้วัดประสิทธิภาพเชิงปริมาณที่สำคัญ ได้แก่ ความแม่นยำ (Accuracy), ความเที่ยง (Precision), ความระลึก (Recall) และ F1-score [3]

(4) ขอบเขตของโรคอ้อยที่ศึกษา งานวิจัยนี้มุ่งเน้นการจำแนกโรคอ้อย



ภาพที่ 1 ตัวอย่างภาพใบอ้อยที่เป็นโรค

ที่ส่งผลกระทบต่อผลผลิตและพบได้บ่อยในภาคการเกษตรจำนวน 4 ชนิด ได้แก่ โรคใบด่าง (Mosaic) โรคเหี่ยวเห่าแดง (Red Rot) โรคราสนิม (Rust) และโรคใบไหม้ (Yellow Leaf) [4] โดยลักษณะอาการของแต่ละโรคที่ปรากฏบนใบอ้อยแสดงดังภาพที่ 1

(5) การเสริมข้อมูล (Augmentation) เป็นเทคนิคที่ใช้ในการสร้างข้อมูลฝึกสอนสังเคราะห์ขึ้นจากข้อมูลต้นฉบับผ่านกระบวนการดัดแปลงเชิงเรขาคณิตและเชิงสี (Geometric and Color Transformations) เช่น การหมุน การเลื่อน และการปรับความสว่าง วัตถุประสงค์หลักคือเพื่อเพิ่มความหลากหลายและความทนทาน (Robustness) ของแบบจำลอง และเพื่อป้องกันภาวะการเรียนรู้เกินพอดี (Overfitting) โดยเฉพาะในกรณีที่ชุดข้อมูลมีจำนวนจำกัด [5]

(6) การตรวจสอบไขว้ (Cross-validation) เป็นกระบวนการทางสถิติที่ใช้ในการประเมินประสิทธิภาพของแบบจำลอง เพื่อวัดความสามารถในการทำงานกับข้อมูลใหม่ (Generalization) เทคนิคนี้จะแบ่งชุดข้อมูลออกเป็นหลายส่วนย่อย (Folds) และทำการฝึกสอนและทดสอบแบบจำลองสลับกันไปในแต่ละส่วน ทำให้การประเมินผลมีความน่าเชื่อถือและลดความเอนเอียงจากการแบ่งข้อมูลเพียงครั้งเดียว [6]

(7) ตัวดำเนินการลาปลาซ (Laplacian Operator) เป็นเทคนิคทางคณิตศาสตร์ในการประมวลผลภาพที่ใช้วัดอัตราการเปลี่ยนแปลงความเข้มของพิกเซล ซึ่งสามารถนำมาประยุกต์ใช้เพื่อตรวจจับขอบ (Edge Detection) หรือประเมินระดับความคมชัด (Sharpness) ของภาพได้ [7] ในงานวิจัยนี้ได้ใช้ตัวดำเนินการลาปลาซเป็นเกณฑ์ในการคัดกรองคุณภาพของภาพถ่าย

(8) แผนภาพกล่อง (Boxplot) เป็นเครื่องมือทางสถิติเชิงพรรณนาที่ใช้แสดงการกระจายตัวของข้อมูลเชิงปริมาณในรูปแบบกราฟ ซึ่งช่วยให้สามารถวิเคราะห์ค่าทางสถิติที่สำคัญได้อย่างรวดเร็ว เช่น ค่ามัธยฐาน (Median) ควอร์ไทล์ (Quartiles) และค่าผิดปกติ (Outliers) จึงนิยมใช้ในการเปรียบเทียบการกระจายตัวของข้อมูลระหว่างกลุ่มต่างๆ [8]

### 2.1.2 งานวิจัยที่เกี่ยวข้อง

จากการทบทวนวรรณกรรมพบว่ามีการนำเทคโนโลยีปัญญาประดิษฐ์มาประยุกต์ใช้ในการตรวจจับโรคพืชอย่างแพร่หลาย ตัวอย่างเช่น งานวิจัยของ Sarala และคณะ [9] ได้นำเสนอการใช้แบบจำลองโครงข่ายประสาทแบบคอนโวลูชัน (CNN) สถาปัตยกรรม VGG19 และ ResNet50 เพื่อวิเคราะห์ลักษณะทางกายภาพของอ้อยและจำแนกโรค ซึ่งแสดงให้เห็นถึงศักยภาพของแบบจำลองการเรียนรู้เชิงลึกในการตรวจหาโรคตั้งแต่ระยะเริ่มต้น

Narmilan Amarasingam และคณะ [10] ประยุกต์ใช้โดรน (UAV) ร่วมกับเทคนิค Deep Learning เพื่อตรวจจับโรคใบขาวในไร่อ้อย ซึ่งเป็นโรคสำคัญที่ส่งผลกระทบต่อผลผลิต นักวิจัยได้ใช้ภาพถ่าย RGB จากโดรนเพื่อเป็นข้อมูลนำเข้าสำหรับแบบจำลอง Deep

Learning ที่สร้างขึ้น เพื่อให้ระบบสามารถระบุตำแหน่งและขอบเขตของการระบาดของโรคในภาพได้อย่างแม่นยำและรวดเร็ว

นอกจากนี้ Karthik Krishnan และคณะ [11] ได้พัฒนาระบบ WhatsApp Bot สำหรับจำแนกโรคพืช โดยผลการทำงานของอัลกอริทึม HSV Filtering, MobileNetV2 และ YOLOv8 เข้าด้วยกัน ทำให้ผู้ใช้สามารถส่งภาพถ่ายของพืชที่เป็นโรคเพื่อให้ระบบวิเคราะห์และแจ้งผลกลับได้อย่างรวดเร็ว โดยแบบจำลองที่ใช้มีความแม่นยำสูงถึง 95.66%

งานวิจัยเหล่านี้เป็นเครื่องยืนยันถึงประสิทธิภาพของการประยุกต์ใช้เทคโนโลยีการประมวลผลภาพและปัญญาประดิษฐ์ในภาคการเกษตรและเป็นแนวทางสำคัญในการพัฒนางานวิจัยฉบับนี้ โดยเฉพาะอย่างยิ่งการเลือกใช้แบบจำลองที่มีขนาดเล็กและประสิทธิภาพสูง เช่น MobileNetV2 เพื่อให้สามารถนำไปใช้งานบนอุปกรณ์เคลื่อนที่ได้จริง

## 2.2 ขั้นตอนการสร้างโมเดลจำแนกโรคอ้อย

กระบวนการสร้างโมเดลได้รับการออกแบบอย่างเป็นระบบ ดังแสดงในภาพที่ 2 ซึ่งประกอบด้วย 2 ขั้นตอนหลัก ได้แก่ 1) การเตรียมชุดข้อมูล (Dataset Preparation) และ 2) การฝึกสอนและประเมินผลแบบจำลอง (Model Training and Evaluation) โดยมีรายละเอียดในแต่ละขั้นตอนดังต่อไปนี้

### 2.2.1 การจัดเตรียมข้อมูลสำหรับการทดลอง

ขั้นตอนการเตรียมชุดข้อมูลเป็นรากฐานสำคัญเพื่อให้แบบจำลองสามารถเรียนรู้ได้อย่างมีประสิทธิภาพ โดยมีกระบวนการดังนี้

(1) แหล่งข้อมูล (Data Source) งานวิจัยนี้ใช้ชุดข้อมูลภาพสาธารณะ (Public Dataset) จากงานวิจัยของ Daphal และคณะ [15] ซึ่งเป็นชุดข้อมูลที่รวบรวมภาพถ่ายใบอ้อยที่สอดคล้องกับขอบเขตการศึกษา

(2) การประมวลผลข้อมูลขั้นต้น (Data Preprocessing) ชุดข้อมูลภาพทั้งหมดจำนวน 2,521 ภาพ ถูกนำมาจัดหมวดหมู่และประมวลผลเบื้องต้น โดยแบ่งออกเป็น 5 คลาส ได้แก่ ใบอ้อยปกติ (Healthy) 1 คลาส และใบอ้อยที่เป็นโรค 4 คลาส (ดังแสดงในตารางที่ 1) จากนั้นจึงดำเนินการแปลงป้ายกำกับ (Label) ให้อยู่ในรูปแบบตัวเลข และปรับขนาดของภาพทุกภาพให้เป็น 224x224 พิกเซล เพื่อให้มีขนาดมิติที่สอดคล้องกัน

2.2.2 การออกแบบการทดลองและกระบวนการฝึกสอน (Experimental Design and Training Process)

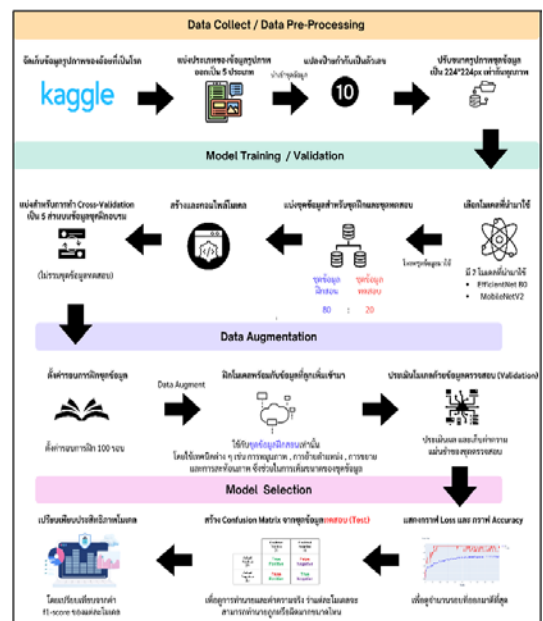
การวิจัยนี้ได้คัดเลือกแบบจำลองสถาปัตยกรรม Convolutional Neural Network (CNN) ที่มีประสิทธิภาพสูง 2 แบบจำลอง ได้แก่ EfficientNetB0 และ MobileNetV2 เพื่อนำมา

เปรียบเทียบประสิทธิภาพ โดยกระบวนการทดลองถูกออกแบบเป็นลำดับขั้นดังนี้

(1) การเปรียบเทียบประสิทธิภาพเบื้องต้น แบบจำลองทั้งสองถูกฝึกสอนเป็นจำนวน 50 รอบ โดยยังไม่ใช้เทคนิคการเพิ่มข้อมูล เพื่อประเมินประสิทธิภาพพื้นฐานของแต่ละสถาปัตยกรรม

(2) การประยุกต์ใช้เทคนิคการเพิ่มข้อมูล นำเทคนิคการเพิ่มข้อมูล (Data Augmentation) ซึ่งประกอบด้วย การหมุน การเลื่อน การเฉือน การซูม และการพลิกภาพ มาใช้ร่วมกับการตรวจสอบไขว้แบบ K-Fold (k=5) เพื่อปรับปรุงประสิทธิภาพและลดภาวะการเรียนรู้เกินพอดี (Overfitting)

(3) การวิเคราะห์ผลกระทบของพารามิเตอร์ ทำการทดลองเพิ่มเติมกับแบบจำลอง EfficientNetB0 ซึ่งให้ผลดีที่สุด เพื่อวิเคราะห์ผลกระทบของขนาดชุดข้อมูล (ที่สัดส่วน 40%, 60%, 80%, และ 100%) และผลของการเพิ่มจำนวนรอบการฝึกสอนเป็น 100 รอบ



ภาพที่ 2 ขั้นตอนการสร้างโมเดล

ตารางที่ 1 ชุดข้อมูลโรคอ้อย [15]

ประเภทข้อมูล	คำอธิบาย	จำนวนข้อมูล
Healthy	ใบอ้อยปกติ	522
Mosaic	โรคใบด่างอ้อย	462
Red Rot	ใบอ้อยที่เป็นโรคเหี่ยวเฉาแดง	518
Rust	โรคราสนิม	514
Yellow	โรคใบไหม้	505

(1) การเปรียบเทียบประสิทธิภาพเบื้องต้น แบบจำลองทั้งสอง ถูกฝึกสอนเป็นจำนวน 50 รอบ โดยยังไม่ใช้เทคนิคการเพิ่มข้อมูล เพื่อประเมินประสิทธิภาพพื้นฐานของแต่ละสถาปัตยกรรม

(2) การประยุกต์ใช้เทคนิคการเพิ่มข้อมูล นำเทคนิคการเพิ่มข้อมูล (Data Augmentation) ซึ่งประกอบด้วย การหมุน การเลื่อน การเฉือน การซูม และการพลิกภาพ มาใช้ร่วมกับการตรวจสอบไขว้แบบ K-Fold (k=5) เพื่อปรับปรุงประสิทธิภาพและลดภาวะการเรียนรู้เกินพอดี (Overfitting)

(3) การวิเคราะห์ผลกระทบของพารามิเตอร์ ทำการทดลองเพิ่มเติมกับแบบจำลอง EfficientNetB0 ซึ่งให้ผลดีที่สุด เพื่อวิเคราะห์ผลกระทบของขนาดชุดข้อมูล (ที่สัดส่วน 40%, 60%, 80%, และ 100%) และผลของการเพิ่มจำนวนรอบการฝึกสอนเป็น 100 รอบ

### 2.2.3 เกณฑ์การประเมินประสิทธิภาพ (Performance Evaluation Metrics)

ประสิทธิภาพของแบบจำลองถูกวัดผลโดยใช้ตัวชี้วัดมาตรฐาน ได้แก่ ความแม่นยำ (Accuracy), ความเที่ยง (Precision), ความระลึก (Recall) และ F1-Score ซึ่งเป็นค่าเฉลี่ยฮาร์โมนิกของ Precision และ Recall นอกจากนี้ยังใช้เมทริกซ์ความสับสน (Confusion Matrix) เพื่อวิเคราะห์รูปแบบความผิดพลาดในการจำแนกประเภทของแต่ละคลาส

## 3. ผลการวิจัยและอภิปรายผล

### 3.1 ประสิทธิภาพเบื้องต้นของแบบจำลอง

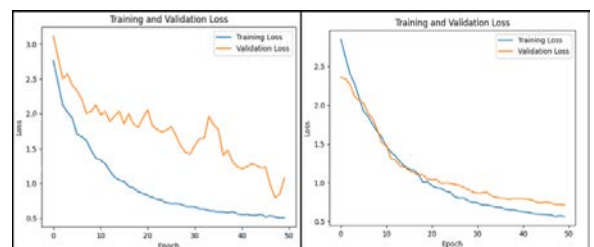
จากการฝึกสอนเบื้องต้น โดยใช้รอบฝึกจำนวน 50 รอบ และไม่ใช้เทคนิคการเพิ่มชุดข้อมูล พบว่า EfficientNetB0 มีประสิทธิภาพสูงกว่า MobileNetV2 อย่างชัดเจน โดยมีค่า F1-score เท่ากับ 0.91 ขณะที่ MobileNetV2 มีค่า F1-score เท่ากับ 0.88 (ตารางที่ 2) กราฟค่าความสูญเสีย (ภาพที่ 3) ยังแสดงให้เห็นว่า MobileNetV2 มีแนวโน้มเกิด Overfitting ซึ่งต่างจาก EfficientNetB0 ที่เรียนรู้ได้อย่างมีประสิทธิภาพ

จากภาพที่ 3 กราฟฝั่งซ้ายคือกราฟที่แสดงผลการฝึกโมเดล MobileNetV2 พบว่าเส้นสีน้ำเงิน (ข้อมูลฝึกสอน) มีค่าความแม่นยำสูงกว่าเส้นสีส้ม (ข้อมูลตรวจสอบ) ซึ่งบ่งชี้ว่าโมเดลเรียนรู้จากข้อมูลฝึกสอนมากเกินไปจนเกิดการเรียนรู้มากเกินไปทำให้โมเดลไม่สามารถทำงานได้ดีเมื่อประเมินกับข้อมูลใหม่

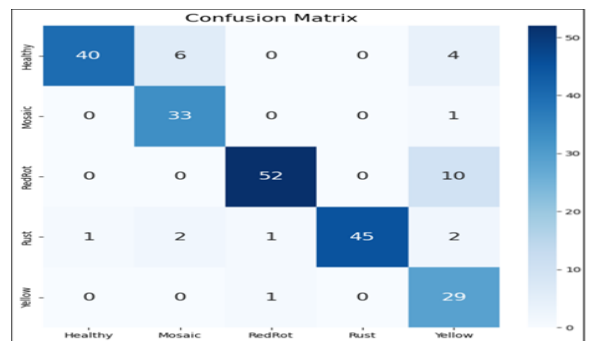
กราฟฝั่งขวาคือการแสดงผลการฝึกโมเดล EfficientNetB0 พบว่าเส้นสีน้ำเงิน (ข้อมูลฝึกสอน) และเส้นสีส้ม (ข้อมูลตรวจสอบ) จะเห็นได้ว่าค่าความสูญเสียของทั้งสองเส้นลดลงอย่างต่อเนื่อง ถือว่าโมเดลฝึกสอนได้ดี และชุดข้อมูลฝึกสอนทำงานได้ดีกว่าชุดข้อมูลตรวจสอบ

ภาพที่ 4 เป็นผลลัพธ์เมทริกซ์ความสับสนของโมเดล MobileNetV2 แทนที่คือค่าจริง แทนนอนคือค่าทำนาย จะเห็นได้ว่าโมเดลทำงานได้ดีในคลาส Mosaic และ Yellow แต่มีความผิดพลาดในการทำนายคลาส Healthy RedRot

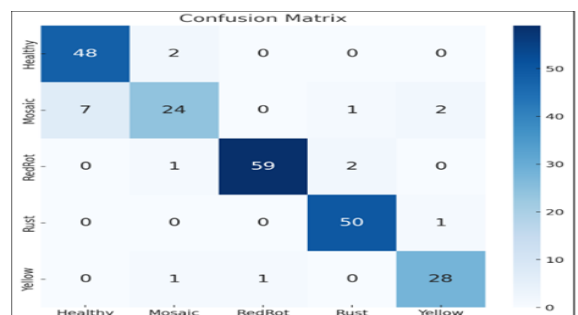
ภาพที่ 5 เป็นผลลัพธ์เมทริกซ์ความสับสนของโมเดล EfficientNetB0 แทนที่คือค่าจริง แทนนอนคือค่าทำนาย จะเห็นได้ว่าโมเดลทำงานได้ดีในคลาส RedRot และ Healthy แต่มีความผิดพลาดในการทำนายคลาสที่ซับซ้อน เช่น Mosaic จากผลลัพธ์ของเมทริกซ์ความสับสนของทั้งสองโมเดล สามารถนำมาคำนวณประสิทธิภาพของโมเดลได้แสดงในตารางที่ 2 แสดงให้เห็นว่า EfficientNet-B0 มีค่าF1-Score มากที่สุด เท่ากับ 0.91



ภาพที่ 3 กราฟแสดงค่าความสูญเสียของโมเดล MobileNetV2 และ EfficientNetB0



ภาพที่ 4 ผลลัพธ์เมทริกซ์ความสับสนของโมเดล MobileNetV2



ภาพที่ 5 ผลลัพธ์เมทริกซ์ความสับสนของโมเดล EfficientNetB0

### 3.2 ประสิทธิภาพหลังการใช้เทคนิคการเพิ่มข้อมูล

เมื่อนำเทคนิคการเพิ่มข้อมูลมาใช้ พบว่าประสิทธิภาพของ EfficientNetB0 ได้รับการปรับปรุงอย่างมีนัยสำคัญ โดยมีค่า F1-score เพิ่มขึ้นเป็น 0.94 ในทางกลับกัน ประสิทธิภาพของ MobileNetV2 ไม่มีการเปลี่ยนแปลง (ตารางที่ 3)

จากภาพที่ 6 กราฟฝั่งซ้ายคือ กราฟที่แสดงผลการฝึกโมเดล MobileNetV2 พบว่าค่าความสูญเสียของทั้งสองเส้นลดลงอย่างต่อเนื่อง และไม่เกิดการเรื้อรังมากเกินไป เมื่อเทียบกับก่อนการใช้เทคนิคการเสริมข้อมูล กราฟฝั่งขวาคือ กราฟที่แสดงผลการฝึกโมเดล EfficientNetB0 ค่าความสูญเสียของทั้งสองเส้นลดลงอย่างต่อเนื่อง ถือว่าโมเดลฝึกสอนได้ดี ซึ่งได้แสดงผลลัพธ์เมตริกซ์ความสับสนของโมเดล MobileNetV2 จากการใช้เทคนิคการเสริมข้อมูลดังภาพที่ 7

จากภาพที่ 7 แสดงผลลัพธ์เมตริกซ์ความสับสนของโมเดล MobileNetV2 แกนตั้งคือค่าจริง แกนนอนคือค่าทำนาย จะเห็นได้ว่าหลังจากที่นำเทคนิคเพิ่มข้อมูลมาใช้ คลาส RedRot ที่มีการผิดพลาดที่ทำนายเป็น Yellow เป็นส่วนใหญ่มีการทำนายถูกต้องมากขึ้น แต่ไม่ได้ดีขึ้นมากนักเพราะยังมีการทำนายผิดในคลาส Mosaic ที่เพิ่มมากขึ้น

ภาพที่ 8 เป็นผลลัพธ์เมตริกซ์ความสับสนของโมเดล EfficientNetB0 แกนตั้งคือค่าจริง แกนนอนคือค่าทำนาย จะเห็นได้ว่าหลังจากที่นำเทคนิคการเสริมข้อมูลมาใช้ โมเดลทำงานได้ดีขึ้นในคลาส Mosaic โดยทำนายภาพถูกต้องมากขึ้น

จากตารางที่ 3 พบว่าหลังจากการใช้เทคนิคการเสริมข้อมูล โมเดล MobileNetV2 ค่า F1-Score ไม่มีการเพิ่มขึ้น ในขณะที่โมเดล EfficientNet-B0 มีค่า F1-Score เพิ่มขึ้นอย่างเห็นได้ชัด

### 3.3 ผลการวิเคราะห์ประสิทธิภาพของ EfficientNetB0

ในส่วนนี้เป็นการวิเคราะห์ปัจจัยต่างๆ ที่มีผลต่อประสิทธิภาพของแบบจำลอง EfficientNetB0 เพื่อหาเงื่อนไขที่ดีที่สุดในการทำงาน

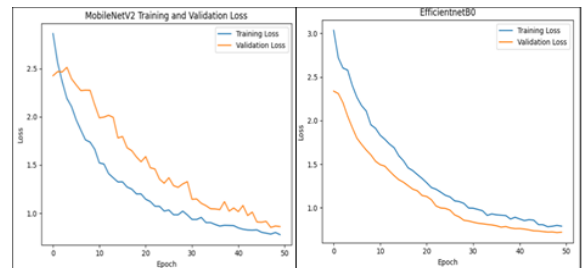
#### 3.3.1 ผลกระทบของขนาดชุดข้อมูลฝึกสอน

เพื่อประเมินผลกระทบของขนาดชุดข้อมูล ผู้วิจัยได้ทดลองปรับสัดส่วนข้อมูลฝึกสอนเป็น 40%, 60%, 80% และ 100% ผลการทดลองดังตารางที่ 4 แสดงให้เห็นว่าประสิทธิภาพของแบบจำลองมีความสัมพันธ์โดยตรงกับปริมาณข้อมูล โดยชุดข้อมูลขนาด 100% ให้ค่า F1-score สูงที่สุดที่ 0.94 ซึ่งยืนยันได้ว่าปริมาณข้อมูลที่มากขึ้นช่วยให้แบบจำลองเรียนรู้และจำแนกประเภทได้ดียิ่งขึ้น ผลลัพธ์การเปรียบเทียบประสิทธิภาพโมเดลเมื่อทำการลดจำนวนข้อมูลรูปภาพสำหรับชุดข้อมูลฝึกสอน

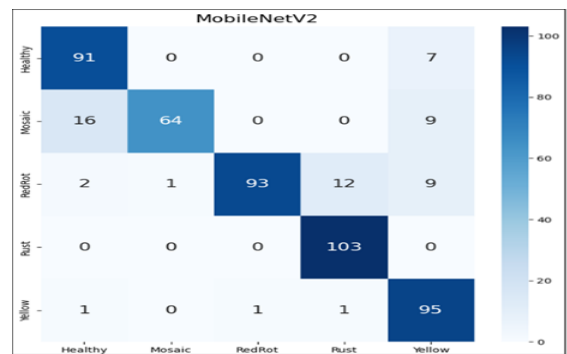
ผลลัพธ์เมตริกซ์ความสับสนของโมเดล EfficientNetB0 จากการใช้เทคนิคการเสริมข้อมูล

ตารางที่ 2 ตารางแสดงผลการฝึกสอนโมเดล

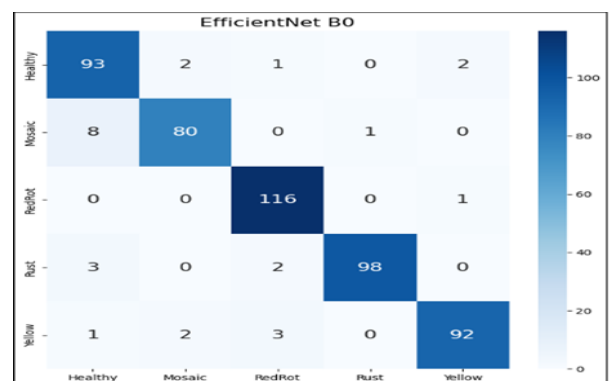
Pre-Trained	Accuracy (%)	Loss	Weighted avg.		
			Precision	Recall	F1-Score
Mobile NetV2	87.47%	0.89	0.90	0.87	0.88
Efficient Net-B0	92.09%	0.71	0.92	0.92	0.91



ภาพที่ 6 กราฟแสดงค่าความสูญเสียของโมเดล MobileNetV2 , EfficientNetB0



ภาพที่ 7 ผลลัพธ์เมตริกซ์ความสับสนของโมเดล MobileNetV2



ภาพที่ 8 ผลลัพธ์เมตริกซ์ความสับสนของโมเดล EfficientNetB0

3.3.2 ผลกระทบของจำนวนรอบการฝึกสอน

ผู้วิจัยได้ทดลองเพิ่มจำนวนรอบการฝึกสอนของ EfficientNetB0 จาก 50 รอบเป็น 100 รอบ เพื่อตรวจสอบว่าแบบจำลองสามารถเรียนรู้เพิ่มเติมได้หรือไม่ ผลการทดลองดังตารางที่ 5 พบว่า ค่า F1-score ของแบบจำลองเพิ่มขึ้นเป็น 0.95 ซึ่งเป็นค่าประสิทธิภาพสูงสุดที่ได้จากการทดลองทั้งหมด

ภาพที่ 9 แสดงให้เห็นว่าการเพิ่มจำนวนรอบค่าความสูญเสียลดลงต่อเนื่องเหมือนเดิมแต่เส้นของค่าทั้งสองใกล้เข้าหากันจนติดกันและกำลังจะคงที่แสดงให้เห็นว่าโมเดลเรียนรู้ได้ดี

ผลลัพธ์เมตริกซ์ความสับสนของโมเดล EfficientNetB0 การเพิ่มจำนวนรอบเป็น 100 รอบ

ภาพที่ 10 เป็นผลลัพธ์เมตริกซ์ความสับสนของโมเดล EfficientNetB0 แกนตั้งคือค่าจริง แกนนอนคือค่าทำนาย จะเห็นได้ว่าหลังจากเพิ่มจำนวนรอบการฝึกโมเดลทำงานได้ดีขึ้นในคลาส Healthy Yellow โดยทำนายภาพถูกต้องมากขึ้น

จากภาพที่ 11 สังเกตได้ว่าโมเดลมักทำนายคลาส Mosaic ผิดเป็นคลาส Healthy อยู่บ่อยครั้ง รวมถึงคลาสนอื่น ๆ ที่มีการทำนายผิดเช่นกัน ซึ่งเมื่อพิจารณาในภาพที่ 11 จะสังเกตได้ว่าภาพที่เป็นคลาส Mosaic ที่ถูกทำนายเป็นคลาส Healthy นั้นมีลักษณะใกล้เคียงกัน จึงทำให้โมเดลเกิดความสับสนและทำนายผิดพลาดได้ รูปภาพเหล่านี้แม้มองด้วยตาเปล่าก็ยากที่จะแยกออก

3.4 อภิปรายผลการทำนายที่ผิดพลาด

แม้ว่าแบบจำลอง EfficientNetB0 ที่ผ่านการปรับปรุงแล้วจะมีประสิทธิภาพสูง แต่ยังคงมีการทำนายที่ผิดพลาดเกิดขึ้น เพื่อทำความเข้าใจสาเหตุ ผู้วิจัยได้วิเคราะห์ปัจจัยด้านคุณภาพของภาพ ได้แก่ ค่าความสว่าง (Brightness) และค่าความเบลอ (Bluniness) [7]

จากภาพ 12 คือภาพการเปรียบเทียบค่าแสง แสดงให้เห็นว่าภาพที่คลาสนทำนายถูก มีค่าความสว่างกระจายตัวกว้างกว่า (IOR 110-140) และมีค่าสุดโต่ง (outliers) ชัดเจน ส่วนภาพในคลาสนที่ทำนายผิดมีช่วงการกระจายแคบลง เล็กน้อย โดยการกระจายตัวของค่าความสว่างของคลาสนที่โมเดลทำนายถูก และทำนายผิดไม่ได้แตกต่างกันอย่างมีนัยสำคัญ

ภาพที่ 13 แสดงตัวอย่างภาพที่โมเดลทำนายผิดพลาดซึ่งสังเกตได้ว่าส่วนใหญ่จะเป็นภาพที่มีเงารวมด้วย แต่ค่าความเข้มแสงมาก หรือน้อยไม่ได้ส่งผลต่อการทำนายผิด ดังจะเห็นได้จากภาพที่ 12 และ 13

ภาพที่ 14 คือภาพการเปรียบเทียบค่าความเบลอ แสดงให้เห็นว่าภาพที่จัดประเภทถูกต้องมีค่าความเบลอต่ำกว่าค่ากลาง (~2000) และการกระจายต่ำ มี outliers บ้าง (>10,000) ขณะที่ภาพที่จัดประเภทผิดพลาดมีค่าความเบลอสูงค่ามาก กระจายตัวกว้าง และ outliers มากกว่า โดยภาพที่ 15 แสดงตัวอย่างภาพที่ทำนายผิดพร้อมค่าความเบลอ

ผลลัพธ์วิเคราะห์ค่าแสงและค่าความเบลอของภาพโดยใช้แผนภาพกล่อง พบว่าค่าแสงและค่าความเบลอของภาพไม่มีผลต่อการฝึกสอนโมเดล

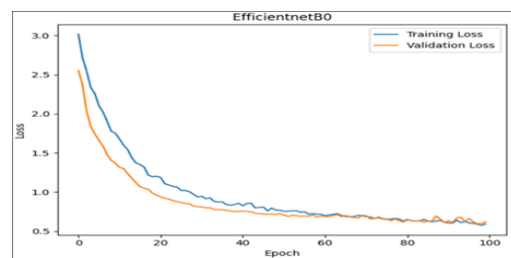
เท่าที่คาดไว้ ซึ่งจากการวิเคราะห์ภาพที่ถูกทำนายผิดพบว่าค่าแสงในส่วนของเงาจะมีผลกระทบบางส่วนต่อการจำแนกประเภท แต่ไม่ถึงกับส่งผลกระทบต่ออย่างมีนัยสำคัญ

ตารางที่ 3 ตารางแสดงผลการฝึกสอนโมเดลโดยการนำเทคนิคการเสริมข้อมูลมาใช้

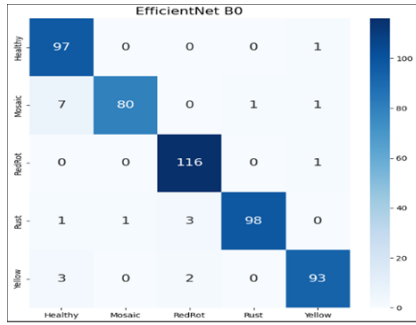
Data set size (%)	Accuracy (%)	Loss	Weighted avg.		
			Precision	Recall	F1-Score
Mobile NetV2	88.32%	0.89	0.89	0.88	0.88
Efficient Net-B0	94.41%	0.77	0.94	0.94	0.94

ตารางที่ 4 จำนวนข้อมูลรูปภาพกับประสิทธิภาพในการฝึกสอนตัวแบบ

Data set size (%)	Accuracy (%)	Loss	Weighted avg.		
			Precision	Recall	F1-Score
40 %	72.04%	1.30	0.76	0.74	0.73
60 %	87.40%	0.97	0.87	0.87	0.87
80 %	92.43%	0.87	0.90	0.90	0.90
100 %	94.41%	0.77	0.94	0.94	0.94



ภาพที่ 9 กราฟแสดงค่าความสูญเสียของโมเดล EfficientNetB0

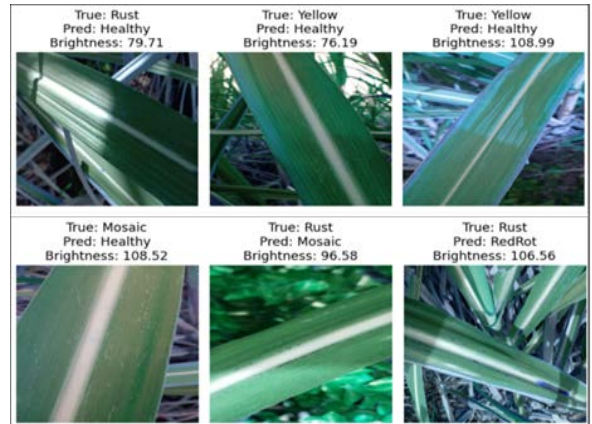


ภาพที่ 10 ผลลัพธ์เมทริกซ์ความสับสนของโมเดล EfficientNetB0

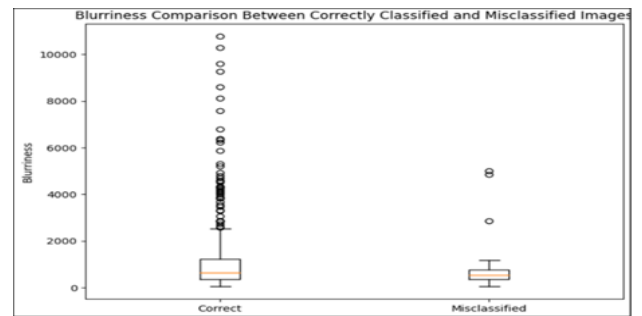
ตารางที่ 5 ผลลัพธ์การเพิ่มจำนวนรอบในการฝึกเป็น 100 รอบของโมเดล EfficientNetB0

Pre-Trained	Accuracy (%)	Loss	Weighted avg.		
			Precision	Recall	F1-Score
Efficient Net-B0	95.84%	0.71	0.96	0.95	0.95

ตารางที่ 5 การทดลองด้วยการเพิ่มจำนวนรอบการฝึกสอนเป็น 100 รอบ พบว่า EfficientNet-B0 มีค่า F1-Score เพิ่มขึ้นเท่ากับ 0.95



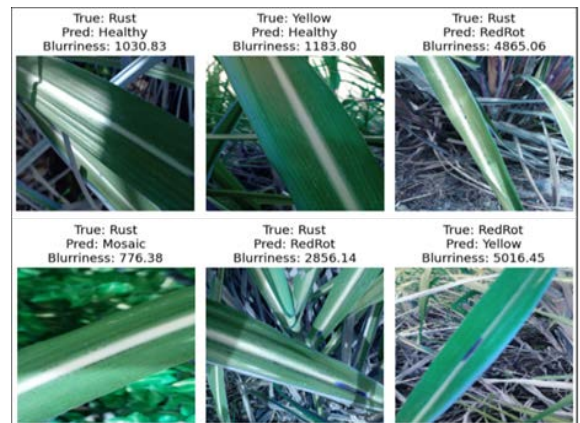
ภาพ 13 ตัวอย่างภาพที่โมเดลทำนายผิดพร้อมค่าแสง



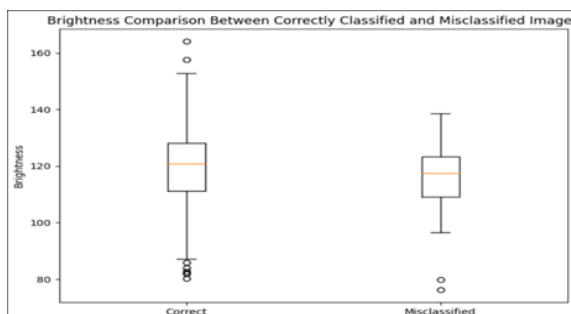
ภาพที่ 14 แผนภาพกล่องเปรียบเทียบค่าความเบลอระหว่างชุดข้อมูลที่ทำนายถูกและผิด



ภาพที่ 11 ภาพตัวอย่างที่มีการทำนายผิด



ภาพที่ 15 ตัวอย่างภาพที่โมเดลทำนายผิดพร้อมค่าความเบลอ



ภาพที่ 12 แผนภาพกล่องเปรียบเทียบค่าความสว่างระหว่างชุดข้อมูลที่ทำนายถูกและผิด

#### 4. สรุปผล

ในการดำเนินโครงการวิจัยนี้ ผู้วิจัยได้พัฒนาโมเดลสำหรับการตรวจจับโรคในอ้อย โดยได้ทดลองใช้โมเดลที่แตกต่างกัน เพื่อเปรียบเทียบประสิทธิภาพในการแยกแยะระหว่างภาพที่มีโรคและภาพที่ไม่มีโรค จากการทดลองพบว่า EfficientNet B0 เป็นโมเดลที่ให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า F1-Score เท่ากับ 0.95 ก่อนทำการฝึกโมเดลในชุดข้อมูลอบรม ผู้วิจัยได้ใช้เทคนิคการเสริมข้อมูล เพื่อเพิ่มความหลากหลายให้กับชุดข้อมูลที่ใช้ฝึก โมเดลถูกฝึกด้วยการ

หมุนภาพ (rotation) การย้ายภาพ (width/height shift) การเอียงภาพ (shear) การซูม (zoom) และการพลิกภาพในแนวนอน (horizontal flip) เทคนิคเหล่านี้ทำให้โมเดลมีความสามารถในการเรียนรู้ และจำแนกประเภทภาพได้ดีขึ้นภายใต้สภาพแวดล้อมที่แตกต่างกัน ซึ่งทำให้ได้ผลลัพธ์ที่แม่นยำมากขึ้นเมื่อเทียบกับก่อนมีการเพิ่มข้อมูล เมื่อได้โมเดลดังกล่าวสามารถนำไปพัฒนาต่อเป็นระบบหรือโปรแกรมในรูปแบบต่างๆ เช่น เว็บไซต์ แอปพลิเคชัน หรือ เซพทอปสำหรับการตรวจจับโรคอ้อย ซึ่งจะช่วยอำนวยความสะดวกแก่เกษตรกรในการตรวจสอบและจัดการโรคในอ้อยได้อย่างมีประสิทธิภาพ

### กิตติกรรมประกาศ

ในการดำเนินโครงการครั้งนี้ ผู้วิจัยได้รับความกรุณาและการสนับสนุนจากบุคคลหลายท่าน จึงขอแสดงความขอบพระคุณมายังอาจารย์ทุกท่านที่ได้ถ่ายทอดความรู้และอบรมสั่งสอนตลอดระยะเวลาในการทำโครงการ โดยเฉพาะอย่างยิ่ง ขอขอบพระคุณอาจารย์ชนพล ตั้งชูพงศ์ ที่ปรึกษาโครงการที่ได้ให้คำแนะนำอันมีคุณค่า ตลอดจนแนวทางในการคิดวิเคราะห์และพัฒนาโมเดลอย่างเป็นระบบนอกจากนี้ ผู้วิจัยขอขอบคุณคุณพ่อและคุณแม่ที่เป็นกำลังใจที่สำคัญในการศึกษาและการทำโครงการมาโดยตลอด รวมถึงเพื่อน ๆ ที่ให้คำแนะนำ ช่วยเหลือ และเป็นแรงสนับสนุนในทุกขั้นตอนของโครงการท้ายที่สุดนี้ ขอขอบพระคุณวิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น ที่มอบโอกาสในการเรียนรู้และสนับสนุนสภาพแวดล้อมที่เอื้อต่อการศึกษาค้นคว้าและพัฒนาโครงการนี้ให้สำเร็จลุล่วงด้วยดี

### เอกสารอ้างอิง

- [1] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. The MIT Press.
- [2] Aggarwal, C. C. (2018). Neural Networks and Deep Learning: A Textbook. Springer.  
<http://doi.org/10.1007/978-3-319-94463-0>
- [3] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). New York, NY: Springer.  
<https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf>
- [4] มิตรพลโมเดิร์นฟาร์ม. (2020). การวินิจฉัยอ้อยเบื้องต้นจากอาการผิดปกติทางใบ. Retrieved July 25, 2024, from <https://www.mitrpholmodernfarm.com/news/2020/05/การวินิจฉัยอ้อยเบื้องต้น-จากอาการผิดปกติทางใบ>

- [5] Amazon Web Services. (2020). การเสริมข้อมูลคืออะไร. Retrieved July 25, 2024, from <https://aws.amazon.com/th/what-is/data-augmentation/>
- [6] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. New York, NY: Springer.
- [7] Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification (2nd ed.). John Wiley & Sons.
- [8] Big Data Institute. (2022). 4 เหตุผลที่จะทำให้คุณตกหลุมรัก Box Plot. Retrieved July 26, 2024, from <https://bdi.or.th/big-data-101/four-reasons-you-ll-love-box-plot/>
- [9] Sarala, R., SwethaVishnu, V. A., & Raji, T. (2024). Sugarcane disease detection along chatbot. International Journal of Engineering Research in Computer Science and Engineering, 11(3), [Page numbers not available]. Retrieved July 12, 2024, from <https://ijercse.com/article/March%207.pdf>
- [10] Amarasingam, N., Gonzalez, F., Salgadoe, A. S. A., Sandino, J., & Powell, K. (2022). Detection of White Leaf Disease in Sugarcane Crops Using UAV-Derived RGB Imagery with Existing Deep Learning Models. Remote Sensing, 14(23), 6137.
- [11] K. Krishnan O, N. Shaju, J. Joseph, G. Mohandas, N. CK and T. T, "Artificial Intelligence Based WhatsApp Bot for Plant Disease Detection," 2023 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2023, pp. 1-6,  
<https://doi.org/10.1109/i-PACT58649.2023.10434523>.
- [12] GeeksforGeeks. (2024). EfficientNet B0 to B7. Retrieved July 20, 2024, from <https://www.geeksforgeeks.org/efficientnet-architecture/>
- [13] Hollemans, M. (2018, April 22). MobileNet version 2. MachineThink. Retrieved July 17, 2024, from <https://machinethink.net/blog/mobilenet-v2/>
- [14] สำนักงานคณะกรรมการอ้อยและน้ำตาลทราย. (2024). คู่มือวินิจฉัยโรคอ้อย. Retrieved July 20, 2024, from <https://cspc3.ocsb.go.th/wp-content/uploads/2024/02/คู่มือวินิจฉัยโรคอ้อย.pdf>

- [15] Daphal, S., Koli, S., Koli, D., & Mane, P. (2022). *Sugarcane leaf disease dataset* [Data set]. Mendeley Data. Retrieved July 20, 2024, from <https://data.mendeley.com/datasets/9424skmnrk/1>