

ประสิทธิภาพการจำแนกโรคพืชที่แสดงลักษณะทางใบด้วยวิธีโครงข่ายประสาทเทียม
แบบคอนโวลูชันร่วมกับอัลกอริทึมการเพิ่มประสิทธิภาพและฟังก์ชันกระตุ้น
**Efficiency of plant diseases classification by convolutional
neural network
with optimization algorithm and activation function**

อนัญญา พรหมโคตร

A-nanya Promkot

สาขาวิชาการแพทย์แผนไทย คณะทรัพยากรธรรมชาติ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน

*Department of Thai traditional medicine, Faculty of Natural Resources, Rajamangala University of
Technology Isan*

Received: February 6, 2024; Revised: June 25, 2024; Accepted: June 28, 2024; Published: June 30, 2024

ABSTRACT – Plant diseases are a problem that has a huge impact on farmers. Detection of plant diseases at an early stage to effectively control the spread of germs. Therefore, it plays an important role in the agricultural industry. However, traditional approach requires extensive knowledge of the expert. It is expensive and requires a lot of labor. Nowadays, with the advancement of information technology, machine learning and deep learning has been applied to automatic identification of plant disease. Currently, convolutional neural network methods is a method that has been recognized for its efficiency in image classification. The objective of this research is to find appropriate values for the ResNet50 method with optimization algorithms, including AdaDelta, AdaGrad, Adam, RMSprop, and SGD, and activation functions including ReLU, Sigmoid, and Tanh, for plant disease classification by using plant leave image. Evaluated performance of plant disease classification by using the PlantVillage dataset. The results showed that the ResNet50 method with RMSprop optimizer and Sigmoid activation function gave the highest Accuracy value of 0.94, Precision value of 0.94, Recall 0.93, and F -measure is equal to 0.93. Therefore, it can be concluded that in selecting a model for learning to achieve the most efficient results should consider additional factors including algorithms to increase the efficiency of the model and stimulation functions

KEYWORDS: Image Classification, Image Processing, Convolution Neural Network, Plant Disease

บทคัดย่อ -- โรคพืชเป็นปัญหาที่สร้างผลกระทบต่อเกษตรกรเป็นอย่างมาก การตรวจหาโรคพืชตั้งแต่ระยะเริ่มต้น เพื่อควบคุมการแพร่กระจายของเชื้อโรครวมถึงประสิทธิภาพ จึงมีบทบาทสำคัญในอุตสาหกรรมเกษตร อย่างไรก็ตามการตรวจหาดังกล่าวต้องใช้ความรู้เกี่ยวกับเชื้อโรครวมถึงความรู้ที่กว้างขวาง มีค่าใช้จ่ายสูงและต้องใช้แรงงานมาก ปัจจุบันด้วยความก้าวหน้าทางด้านเทคโนโลยีสารสนเทศ จึงได้มีการประยุกต์ใช้ประโยชน์จากการเรียนรู้ของเครื่อง และวิธีการการเรียนรู้เชิงลึก สำหรับการระบุลักษณะโรคพืชที่แสดงอาการทางใบพืชอย่างอัตโนมัติ ปัจจุบันวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional neural network) เป็นวิธีที่ได้รับการยอมรับถึงประสิทธิภาพการจำแนกรูปภาพ การวิจัยครั้งนี้มีวัตถุประสงค์การค้นหาค่าที่เหมาะสมสำหรับแบบจำลองวิธีเรียนรู้เชิงลึก ResNet50 สำหรับการจำแนกรูปภาพโรคพืช

ที่แสดงลักษณะทางใบ ซึ่งเป็นวิธีที่พัฒนาจากพื้นฐานวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน ที่ทำงานร่วมกับอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล ประกอบด้วย AdaDelta AdaGrad Adam RMSprop และ SGD และการใช้ฟังก์ชันกระตุ้น (Activation function) ได้แก่ ReLU, Sigmoid และ Tanh ทดสอบประสิทธิภาพการจำแนกโรคพืชด้วยชุดข้อมูล PlantVillage ผลการทดลองพบว่า วิธี ResNet50 ที่ใช้ Optimizer RMSprop และฟังก์ชันกระตุ้น Sigmoid ให้ค่าความถูกต้อง (Accuracy) สูงสุดคือ 0.94 ค่าความแม่นยำ (Precision) 0.94 ค่าอัตราการจำแนกถูกต้องเมื่อข้อมูลเป็นจริง (Recall) 0.93 และค่า F-measure เท่ากับ 0.93 ดังนั้นจึงสรุปได้ว่าการพิจารณาเลือกแบบจำลองสำหรับการเรียนรู้เพื่อให้ได้ผลลัพธ์ที่มีประสิทธิภาพสูงสุดควรพิจารณาเลือกปัจจัยเพิ่มเติม ได้แก่ อัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล และฟังก์ชันกระตุ้น เป็นต้น

คำสำคัญ: การจำแนกภาพ, การประมวลผลภาพ, โครงข่ายประสาทเทียมแบบคอนโวลูชัน, โรคพืช

1. บทนำ

ด้วยสถานการณ์โลกในปัจจุบันมนุษย์ต้องเผชิญกับปัญหาและความท้าทายที่ส่งผลต่อการดำรงชีวิตมากมาย ได้แก่ การเปลี่ยนแปลงสภาพภูมิอากาศและสิ่งแวดล้อม อุบัติการณ์โรคระบาด ความรุนแรงในการแข่งขันทางเศรษฐกิจ และอัตราจำนวนประชากรโลกที่เพิ่มขึ้นอย่างรวดเร็ว โดยคาดการณ์ว่าในปี พ.ศ. 2593 จำนวนประชากรโลกจะเพิ่มขึ้นเป็น 10 พันล้านคน และส่งผลต่อปริมาณความต้องการทางด้านอาหารสูงเพิ่มมากขึ้น [1] อุตสาหกรรมเกษตรทั้งระดับรายย่อยและรายใหญ่ต่างพยายามผลิตสินค้าเกษตรเพื่อตอบสนองความต้องการของผู้บริโภคที่จำนวนเพิ่มมากขึ้น แต่อย่างไรก็ตามเกษตรกรผู้ปลูกพืชต้องเผชิญกับปัญหาโรคพืช ซึ่งผลกระทบต่อผลผลิตที่ได้ออกจำหน่ายเป็นอย่างมาก ศัตรูพืชและโรคพืชทำให้ผลผลิตสูญเสียมากกว่า 20 พันล้านดอลลาร์สหรัฐ [2] โดยแบคทีเรียไวรัส และเชื้อราเป็นสาเหตุของการติดเชื้อโรค จากปัจจัยและโรคที่หลากหลายเหล่านี้ จึงเป็นความท้าทายอย่างยิ่งสำหรับเกษตรกรในการตรวจหาและควบคุมการติดเชื้อโรคเหล่านี้ ในปัจจุบันเกษตรกรส่วนใหญ่ยังคงใช้วิธีตรวจหาโรคพืชด้วยตาเปล่าโดยใช้แรงงาน [3] ซึ่งต้องใช้ความรู้ความเชี่ยวชาญเกี่ยวกับโรคพืชและค่าใช้จ่ายสูง

จากปัญหาที่กล่าวมาข้างต้นจึงมีงานวิจัยมากมายที่ประยุกต์เทคโนโลยีปัญญาประดิษฐ์สำหรับการติดตามกระบวนการผลิตอย่างอัตโนมัติ วิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน เป็นหนึ่งในวิธีที่นิยมสำหรับงานประมวลผลภาพมีความสามารถในการสกัดคุณสมบัติที่เกี่ยวข้องอย่างอัตโนมัติ มี

งานวิจัยที่ผ่านมาได้แสดงประสิทธิภาพการจำแนกโรคพืชโดยเรียนรู้จากลักษณะรูปภาพใบพืช ด้วยวิธีการต่างๆ [4] และแสดงประสิทธิภาพของแต่ละวิธี สำหรับงานวิจัยนี้ ผู้วิจัยมีวัตถุประสงค์ในการค้นหาที่เหมาะสมสำหรับแบบจำลองวิธีเรียนรู้เชิงลึก ResNet50 สำหรับการจำแนกรูปภาพโรคพืชที่แสดงลักษณะทางใบ ซึ่งเป็นวิธีที่พัฒนาจากพื้นฐานวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน โดยนำมาทำงานร่วมกับอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล 5 วิธี ประกอบด้วย AdaDelta AdaGrad Adam RMSprop และ SGD พร้อมทั้งเปรียบเทียบการใช้ฟังก์ชันกระตุ้น (Activation Function) ได้แก่ ReLU, Sigmoid และ Tanh ทดสอบประสิทธิภาพการจำแนกโรคพืชด้วยชุดข้อมูล PlantVillage ซึ่งเป็นชุดข้อมูลรูปภาพใบพืชที่สมบูรณ์และใบพืชที่มีอาการของโรคพืช

2. ทฤษฎีที่เกี่ยวข้อง

2.1 วิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน

วิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน หรือชื่อย่อ CNN คือประเภทของโครงข่ายประสาทเทียมที่พัฒนาขึ้น จัดอยู่ในกลุ่ม Bio-inspired มีการจำลองการมองเห็นของมนุษย์ที่มีการมองสิ่งต่างๆ เป็นพื้นที่ย่อยๆ หลังจากนั้นจึงนำข้อมูลพื้นที่ย่อยๆ มารวมกันเพื่อสรุปว่าสิ่งที่มองเห็นนั้นคืออะไร หลักการทำงานของวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน ประกอบด้วย 4 ชั้น [5] ดังนี้

2.1.1 Convolution layer

เป็นชั้นที่รับข้อมูลนำเข้าเพื่อสกัดให้ได้คุณสมบัติที่สำคัญจากรูปภาพ โดยแปลงข้อมูลให้อยู่ในรูปแบบ Image pixel กำหนด Feature detector หรือ Convolution kernel มาเพื่อสแกนเพื่อแยกองค์ประกอบต่างๆ ของรูปภาพ ได้แก่ ลักษณะรูปทรงของเส้นขอบ สีและความเข้มของสี โดยกำหนดขนาดรูปภาพให้เป็นสี่เหลี่ยมจัตุรัส ขนาด $W \times W \times M$ โดยที่ W คือความสูงและความกว้างของรูปภาพ และ M คือมิติของรูปภาพ เช่น ภาพสี RGB จะกำหนดให้ $M = 3$ เป็นต้น หลังจากนั้นคำนวณแบบ dot product และเก็บผลลัพธ์ไว้ที่ Feature map ในขั้นตอนนี้จะทำให้ทราบคุณสมบัติที่สำคัญที่ใช้จำแนกประเภทรูปภาพ

2.1.1 Pooling layer

เป็นขั้นตอนการลดมิติของข้อมูลใน Feature map โดยยังคงรายละเอียดสำคัญเอาไว้ วิธีการ Pooling มี 2 วิธีด้วยกัน ได้แก่ Max pooling และ Mean pooling เป็นต้น

2.1.3 Fully connected layer

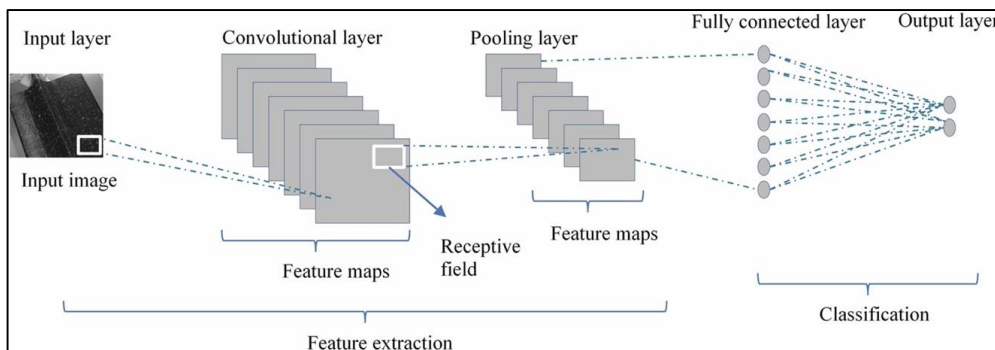
เป็นขั้นสุดท้ายของวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน ในขั้นนี้จะรับข้อมูล Feature map ซึ่งมีรูปแบบเป็นข้อมูลเวกเตอร์ 1 มิติ ที่ได้จากขั้นตอน Flattening หลังจากนั้นนำมาคำนวณด้วย Activate function ได้แก่ Softmax ReLU และ Tanh เป็นต้น ผลลัพธ์ที่ได้ในชั้น Fully connected layer นี้จะเป็นค่าความน่าจะเป็นของการจำแนกข้อมูลในแต่ละคลาส

2.2 Deep residual Network

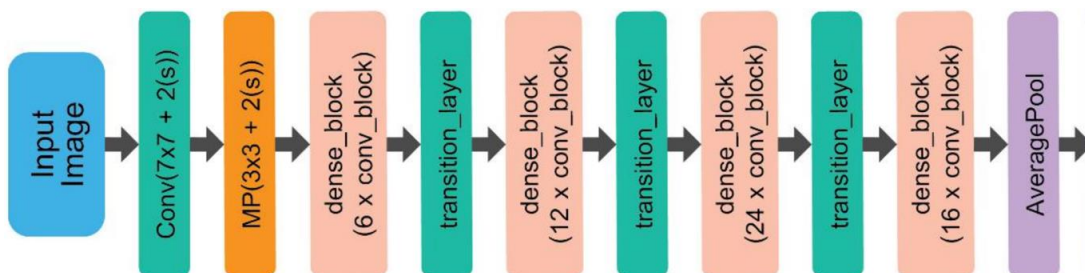
Deep residual Network หรือชื่อย่อ ResNet หรือ ResNet50 ถูกนำเสนอในปี ค.ศ. 2015 พัฒนาขึ้นเพื่อแก้ไขปัญหา Vanishing gradient โครงสร้างสถาปัตยกรรม ResNet50 ดังแสดงในรูปภาพที่ 2 ResNet50 มีการใช้ Residual block เป็นการนำผลลัพธ์ของชั้นก่อนหน้ามารวมกับผลลัพธ์ของชั้นที่กำลังทำการประมวลผลอยู่ ภายใน Residual block มีการใช้ Skip connection ช่วยดึงค่า Gradient จากชั้นก่อนหน้ามาใช้เพื่อลดเวลาในการเชื่อมกันของแต่ละชั้น ช่วยลดระยะเวลาในการประมวลผลโดยที่ประสิทธิภาพยังคงเดิม และนำเทคนิค Global average มาใช้ในชั้น Fully connected layer ทำให้ ResNet50 มีความลึก 50 ชั้น

2.3 อัลกอริทึมเพิ่มประสิทธิภาพ (Optimizer)

การปรับค่าน้ำหนักให้เหมาะสมเป็นเรื่องสำคัญในการประมวลผลด้วยวิธีโครงข่ายประสาทเทียม ทำให้การประมวลผลของแบบจำลองมีประสิทธิภาพยิ่งขึ้น โดยการปรับค่าพารามิเตอร์ ของโครงข่ายประสาทเทียมในระหว่างการศึกษาเพื่อหาค่า loss ลดลง อัลกอริทึมเพิ่มประสิทธิภาพสามารถนำมาทำงานร่วมกับเทคนิคอื่นๆ เพื่อปรับปรุงประสิทธิภาพของแบบจำลองให้ทำงานมีประสิทธิภาพมากยิ่งขึ้น [7]



รูปที่ 1. สถาปัตยกรรมวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน แหล่งที่มา: [5, 6]



รูปที่ 2. โครงสร้างสถาปัตยกรรม Deep residual Network

การเลือกใช้อัลกอริทึมเพิ่มประสิทธิภาพที่เหมาะสมขึ้นสำหรับแบบจำลองการเรียนรู้ของเครื่องนั้นขึ้นอยู่กับลักษณะของข้อมูลปัญหาที่ต้องการแก้ไข

2.3.1 SGD (Stochastic Gradient Descent)

SGD [7] [8] เป็นอัลกอริทึมเพิ่มประสิทธิภาพที่ใช้ลดค่าน้อยสุดของฟังก์ชันโดยการเคลื่อนที่ทีละขั้นไปยังค่าต่ำสุดของฟังก์ชัน เป็นรูปแบบหนึ่งของ Gradient Descent ที่ใช้การสุ่มข้อมูลย่อย (mini-batch) ของข้อมูลในแต่ละขั้นตอน วิธีการนี้ทำให้กระบวนการเพิ่มประสิทธิภาพเร็วขึ้นและมีประสิทธิภาพมากขึ้น โดยเฉพาะอย่างยิ่งสำหรับชุดข้อมูลขนาดใหญ่ ขั้นตอนการทำงานของวิธี SGD มีขั้นตอนที่สำคัญดังนี้

- การเริ่มต้น พารามิเตอร์: เริ่มต้นจากการสุ่มค่าพารามิเตอร์ของแบบจำลองซึ่งอาจเป็น ค่าน้ำหนัก (Weight) หรือค่าอคติ (bias)

- เลือก Mini-batch: เลือกส่วนย่อยเล็กๆ ของข้อมูลสำหรับการเรียนรู้โดยวิธีสุ่ม

- คำนวณ Gradient: คำนวณค่า Gradient ของฟังก์ชันสูญเสียโดยอิงจากพารามิเตอร์ของแบบจำลองโดยใช้ Mini-batch ที่เลือก

- ปรับพารามิเตอร์: ปรับพารามิเตอร์ของโมเดลในทิศทางตรงกันข้ามกับค่า Gradient ดังสมการ

$$\theta = \theta - \eta \cdot \nabla_{\theta} L(\theta; \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \quad (1)$$

โดยที่ η เป็น อัตราการเรียนรู้ และ $\nabla_{\theta} L(\theta; \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ คือ Gradient ของฟังก์ชันสูญเสีย L ที่คำนวณโดยใช้ mini-batch $\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$

- ทำซ้ำจนกว่าจะบรรจบกัน
- การบรรจบกัน: อัลกอริทึมจะบรรจบกันเมื่อการเปลี่ยนแปลงของพารามิเตอร์น้อยลงอย่างเพียงพอหรือเมื่อฟังก์ชันสูญเสียไม่ถึงระดับที่แน่นอน

2.3.2 RMSprop (Root Mean Square Propagation)

RMSprop [7] เป็นวิธีการที่ออกแบบมาเพื่อแก้ปัญหาการลดลงอย่างรวดเร็วของอัตราการเรียนรู้ (learning rate) ของวิธี SGD หลักการของวิธี RMSprop คือการรักษาค่าเฉลี่ยเคลื่อนที่กำลังสอง (Squared gradient) ของพารามิเตอร์แต่ละตัว ทำให้กระบวนการเรียนรู้มีความเร็วขึ้น มีขั้นตอนดังนี้

- การคำนวณ Gradient: คำนวณ Gradient ของฟังก์ชันสูญเสีย L ตามพารามิเตอร์ θ สำหรับ Mini-batch ที่สุ่มเลือก

$$\mathbf{g}_t = \nabla_{\theta} L_{\theta}(\theta) \quad (2)$$

- การอัปเดตค่าเฉลี่ยเคลื่อนที่ของ Squared gradient $\mathbf{E}[g^2]_t$ โดยใช้ค่าเฉลี่ยที่สะสมมาจาก Gradient ที่ผ่านมา

$$\mathbf{E}[g^2]_t = \gamma \mathbf{E}[g^2]_{t-1} + (1 - \gamma) \mathbf{g}_t^2 \quad (3)$$

โดยที่ γ เป็นพารามิเตอร์ที่กำหนดอัตราลดลงของค่าเฉลี่ยเคลื่อนที่ โดยส่วนใหญ่นิยมกำหนดค่า $\gamma = 0.9$

- การปรับพารามิเตอร์: ปรับพารามิเตอร์ θ โดยใช้อัตราการเรียนรู้ที่ปรับตามค่าเฉลี่ยเคลื่อนที่ของ Squared gradient

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbf{E}[g^2]_t + \epsilon}} \mathbf{g}_t \quad (4)$$

โดยที่ η คืออัตราการเรียนรู้ และ ϵ คือค่าคงที่ขนาดเล็กที่เพิ่มเข้าไปเพื่อป้องกันการหารด้วยศูนย์ นิยมกำหนดค่า $\epsilon = 10^{-8}$

2.3.3 AdaGrad (Adaptive Gradient Algorithm)

AdaGrad [7] เป็นอัลกอริทึมการเพิ่มประสิทธิภาพที่พัฒนาขึ้นเพื่อปรับอัตราการเรียนรู้สำหรับพารามิเตอร์แต่ละตัว ในระหว่างการเรียนรู้ของแบบจำลอง ซึ่งช่วยให้การฝึกมีประสิทธิภาพมากขึ้น มีขั้นตอนดังนี้

- การคำนวณ Gradient: คำนวณ Gradient ของฟังก์ชันสูญเสีย L ตามพารามิเตอร์ θ สำหรับ Mini-batch ที่สุ่มเลือก ดังสมการที่ 2

- การสะสม Squared gradient: สะสมค่า Squared gradient สำหรับแต่ละพารามิเตอร์

$$\mathbf{G}_{t,ii} = \mathbf{G}_{t-1,ii} + \mathbf{g}_{t,i}^2 \quad (5)$$

โดยที่ \mathbf{G} เป็นเมทริกซ์การสะสมของ Squared gradient และ $\mathbf{g}_{t,i}$ เป็น Gradient ของพารามิเตอร์ i ในเวลา t

- การปรับพารามิเตอร์: ปรับค่าพารามิเตอร์ θ โดยใช้อัตราการเรียนรู้ที่ปรับตามการสะสมของ Squared gradient

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{\mathbf{G}_{t,ii} + \epsilon}} \mathbf{g}_{t,i} \quad (6)$$

2.3.4 Adam (Adaptive Moment Estimation)

Adam [7] เป็นอัลกอริทึมการเพิ่มประสิทธิภาพที่ได้รับ ความนิยมมากในงานการเรียนรู้เชิงลึก เนื่องจากมีประสิทธิภาพ สูงและสามารถปรับอัตราการเรียนรู้แบบอัตโนมัติได้ วิธีที่รวม ข้อดีของวิธี AdaGrad และ วิธี RMSProp เข้าด้วยกัน โดยใช้ ค่าเฉลี่ยเคลื่อนที่ของ Gradient และ Squared gradient มีขั้นตอน ดังนี้

- การคำนวณ Gradient: คำนวณ Gradient ของฟังก์ชัน สูญเสีย L ตามพารามิเตอร์ θ สำหรับ Mini-batch ที่สุ่มเลือก ดัง สมการที่ 2

- การคำนวณค่าเฉลี่ยเคลื่อนที่ของ Gradient: ใช้ค่าเฉลี่ย เคลื่อนที่ของ Gradient m_t

$$m_t = \beta_1 m_t + (1 - \beta_1) g_t \quad (7)$$

โดยที่ β_1 เป็นพารามิเตอร์ที่กำหนดอัตราการลดลงของ ค่าเฉลี่ยเคลื่อนที่

- การคำนวณค่าเฉลี่ย Squared gradient v_t

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (8)$$

โดยที่ β_2 เป็นพารามิเตอร์ที่กำหนดอัตราการลดลงของค่าเฉลี่ย เคลื่อนที่

- การปรับค่าเบี่ยงเบน (Bias correction):

เนื่องจาก m_t และ v_t มีค่าเบี่ยงเบนเริ่มต้นในช่วงแรก จะต้องปรับค่าเหล่านี้

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (9)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (10)$$

- การคำนวณการเปลี่ยนแปลงของพารามิเตอร์:

คำนวณการเปลี่ยนแปลงของพารามิเตอร์ θ :

$$\Delta \theta_t = - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (11)$$

โดยที่ η เป็นอัตราการเรียนรู้ และ ϵ เป็นค่าคงที่ขนาดเล็กที่เพิ่ม เข้าไปเพื่อป้องกันการหารด้วยศูนย์

- การปรับพารามิเตอร์: ปรับพารามิเตอร์ θ โดยใช้การ เปลี่ยนแปลงที่คำนวณได้

$$\theta_{t+1} = \theta_t + \Delta \theta_t \quad (12)$$

2.3.5 AdaDelta

AdaDelta [7] เป็นอัลกอริทึมที่พัฒนาขึ้นเพื่อแก้ไข ปัญหาบางประการของวิธี AdaGrad โดยเฉพาะการลดลงอย่าง รวดเร็วของอัตราการเรียนรู้เมื่อการสะสมของ Squared gradient มีค่ามากขึ้น AdaDelta ปรับอัตราการเรียนรู้แบบอัตโนมัติ สำหรับพารามิเตอร์แต่ละตัวโดยไม่ต้องกำหนดค่าเริ่มต้นของ อัตราการเรียนรู้ (Initial learning rate) และรักษาค่าเฉลี่ยเคลื่อนที่ ของการเปลี่ยนแปลงพารามิเตอร์และ Gradient เพื่อรักษาอัตรา การเรียนรู้ให้มีความสมดุลมากขึ้น มีขั้นตอน ดังนี้

- การคำนวณ Gradient: คำนวณ Gradient ของฟังก์ชัน สูญเสีย L ตามพารามิเตอร์ θ สำหรับ Mini-batch ที่สุ่มเลือก ดัง สมการที่ 2

- การอัปเดตค่าเฉลี่ยเคลื่อนที่ของ Squared gradient: อัปเดตค่าเฉลี่ยของ Squared gradient

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho) g_t^2 \quad (13)$$

โดยที่ ρ เป็นพารามิเตอร์ที่กำหนดอัตราการลดลงของค่าเฉลี่ย เคลื่อนที่ โดยส่วนใหญ่จะนิยามกำหนดค่า $\rho = 0.9$

- คำนวณค่าเฉลี่ยการเปลี่ยนแปลงของพารามิเตอร์

$$E[\Delta \theta^2]_t = \rho E[\Delta \theta^2]_{t-1} + (1 - \rho) (\Delta \theta_t)^2 \quad (14)$$

- การปรับพารามิเตอร์: ปรับพารามิเตอร์ θ โดยใช้การ เปลี่ยนแปลงที่คำนวณได้ ดังสมการที่ 12

2.4 ฟังก์ชันกระตุ้น (Activation function)

ฟังก์ชันกระตุ้น เป็นองค์ประกอบที่สำคัญของ แบบจำลองการเรียนรู้เชิงลึกที่ช่วยเพิ่มความไม่เชิงเส้นให้กับ แบบจำลอง และทำให้สามารถเรียนรู้รูปแบบที่ซับซ้อนได้ ฟังก์ชันกระตุ้นที่นิยมในปัจจุบันได้แก่ Sigmoid ReLU และ Tanh [9] ดังนั้นงานวิจัยนี้จึงนำทั้งสามวิธีมาใช้ในการทดลอง

2.4.1 ReLU (Rectified Linear Unit)

การคำนวณ ReLU ดังสมการต่อไปนี้ โดยกรณีที่ค่าของ x เป็นบวก จะให้ค่า x และกรณีที่ค่าของ x เป็นลบหรือมีค่าศูนย์ จะให้ค่า 0 ข้อดีของ ReLU คือคำนวณง่ายและรวดเร็ว แก้ไขปัญหาการหายไปของ Gradient (Vanishing gradient problem) และเพิ่มความไม่เชิงเส้นให้กับโมเดลได้ดี แต่มีข้อจำกัดคือ กรณีที่ค่า x เป็นลบตลอดเวลาจะไม่สามารถอัปเดตค่า Neuron ได้

$$f(x) = \max(0, x) \quad (15)$$

2.4.2 Sigmoid

เหมาะสำหรับการจำแนกข้อมูลประเภท Binary เพราะค่าที่คำนวณได้อยู่ในช่วง $[0,1]$ ซึ่งเป็นข้อดีเหมาะกับการใช้งานที่ต้องการความน่าจะเป็น และการจำแนกข้อมูล Binary การคำนวณดังแสดงในสมการต่อไปนี้ ข้อจำกัดคือ มีปัญหาการหายไปของ Gradient ทำให้การเรียนรู้ของแบบจำลองเป็นไปได้ช้า

$$f(x) = \frac{1}{1+e^x} \quad (16)$$

2.4.3 Tanh (Hyperbolic Tangent)

เป็นวิธีที่คล้ายกับวิธี Sigmoid แต่มีความแตกต่างคือค่าผลลัพธ์จะอยู่ในช่วง $[-1,1]$ การคำนวณดังแสดงในสมการต่อไปนี้

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (17)$$

การรวมการทำงานของวิธีโครงข่ายประสาทเทียม อัลกอริทึมเพิ่มประสิทธิภาพ และฟังก์ชันกระตุ้น ช่วยทำให้โครงข่ายมีประสิทธิภาพสูงขึ้นได้

3. งานวิจัยที่เกี่ยวข้อง

ในช่วงหลายปีที่ผ่านมาจนถึงปัจจุบันมีนักวิจัยที่ได้นำเสนองานวิจัยที่ใช้เทคนิคการประมวลผลภาพสำหรับงานจำแนกลักษณะรูปภาพมากมาย [4] เพื่อจำแนกความแตกต่าง หรือการระบุลักษณะความผิดปกติหรืออาการ โรคพืชอย่างอัตโนมัติ เพื่อสนับสนุนงานด้านเกษตรกรรมให้มีประสิทธิภาพ ช่วยลดความสูญเสียจากโรคพืชได้ ดังปรากฏในงานวิจัยโครงข่าย

ประสาทเทียมแบบคอนโวลูชันเป็นวิธีที่ได้รับความนิยมเนื่องจากประสิทธิภาพของแบบจำลอง งานของ [10] ีรวิวแบบจำลองการเรียนรู้ต่างๆ ที่นำมาใช้งานจำแนกโรคพืช และแสดงการทดลองเปรียบเทียบการจำแนกโรคพืชด้วยแบบจำลอง 23 วิธี ได้แก่ด้วยชุดข้อมูล PlantDoc ทำนายว่ารูปภาพใบพืชที่มีอาการของโรคพืชหรือไม่ [11] ได้ทดสอบประสิทธิภาพการจำแนกโรคพืชจากรูปภาพใบพืช ด้วยวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชัน ทดสอบด้วยชุดข้อมูล PlantVillage ซึ่งเป็นชุดข้อมูลรูปภาพใบพืชจำนวน 14 ชนิด จำแนกเป็นรูปภาพใบพืชที่สมบูรณ์และใบพืชที่มีเชื้อโรคจำนวน 38 คลาส ผลการทดสอบประสิทธิภาพแสดงค่าความถูกต้องในการจำแนกเท่ากับ 98.01% และ Test accuracy เท่ากับ 94.33% นอกจากนี้ยังมีงานลักษณะคล้ายกันที่ศึกษาเทคนิควิธีการเพิ่มประสิทธิภาพการจำแนกรูปภาพหนึ่งด้วยเทคนิคการเพิ่มรูปภาพ (Augmentation) ดังปรากฏในงานของ [12] และ [13] ทดสอบประสิทธิภาพการจำแนกโรคใบมะเขือเทศ ด้วยวิธีโครงข่ายประสาทเทียมแบบคอนโวลูชันร่วมกับวิธีการเพิ่มจำนวนรูปภาพ โดยคัดเลือกรูปภาพใบมะเขือเทศจากชุดข้อมูล PlantVillage จำนวน 18,160 รูปภาพ ขนาด 256 x 256 พิกเซล ผลลัพธ์การทดลองมีค่าความถูกต้องเฉลี่ยรวมเท่ากับ 87.96% ด้วยอัตราการเรียนรู้เท่ากับ 0.001 จำนวน รอบการเรียนรู้ (Epoch) 100 รอบ และงานของ [14] ได้ใช้เทคนิคการเพิ่มรูปภาพสำหรับแบบจำลองการเรียนรู้ AlexNet ResNet50 VGG16 Inception V3 และ EfficientNet ผลการทดลองสรุปว่าเทคนิคการเพิ่มรูปภาพสามารถเพิ่มประสิทธิภาพแบบจำลองการเรียนรู้ได้ นอกจากนี้ [15] ได้ประยุกต์วิธีโครงข่ายประสาทเทียมแบบคอนโวลูชันเพื่อจำแนกประเภทสมุนไพร โดยทดลองอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล 4 วิธี และเพิ่มประสิทธิภาพด้วยเทคนิคการเพิ่มจำนวนรูปภาพโดยการกลับและปรับความสว่างของรูปภาพ ผลการทดลองสรุปว่า การใช้ optimizer AdaDelta กับฟังก์ชันกระตุ้น Tanh มีค่าความถูกต้องสูงสุดเท่ากับ 0.937 [16] ได้ปรับปรุงประสิทธิภาพวิธี ResNet18 ในการจำแนกโรคพืชจากลักษณะ ใบแอปเปิ้ล ใช้ชุดข้อมูล PlantVillage จำนวน 300 รูป ด้วยเทคนิค Convolution block attention module หรือ CBAM เปรียบเทียบกับเทคนิค Random clipping หรือ RC ผลการทดลองแสดงประสิทธิภาพของวิธี ResNet18 ทำงานร่วมกับ RC มีค่าความถูกต้องสูงสุดเท่ากับ 97.2% จากการศึกษาวิจัยที่เกี่ยวข้องแม้ว่าจะมีรายงานแสดง

ประสิทธิภาพการจำแนกโรคพืชไว้แต่อย่างไรก็ตามการศึกษาเพื่อหาค่าหรือสภาพแวดล้อมที่เหมาะสมสำหรับการประเมินประสิทธิภาพการจำแนกโรคพืชยังคงเป็นประเด็นที่น่าสนใจ

4. วิธีดำเนินการวิจัย

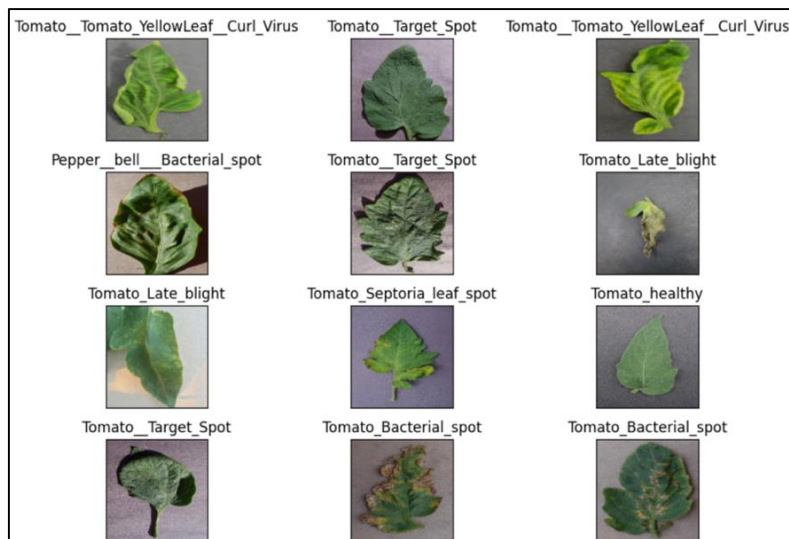
4.1 ชุดข้อมูลสำหรับการทดลอง

ในการดำเนินการทดลองผู้วิจัยใช้ข้อมูลรูปภาพใบพืชที่สมบูรณ์และใบพืชที่แสดงอาการโรคพืชทางใบจากชุดข้อมูล PlantVillage [17] ซึ่งเป็นชุดข้อมูลที่มีนักวิจัยนิยมใช้ในการทดสอบประสิทธิภาพการจำแนกโรคพืช [11] [13] [16] ชุดข้อมูล PlantVillage [17] ได้รับการเผยแพร่ในปี ค.ศ. 2015 ประกอบด้วยข้อมูลรูปภาพพืชมากกว่า 50,000 รูปภาพ จากพืชจำนวน 14 ชนิด ได้แก่ แอปเปิ้ล บลูเบอร์รี่ เชอร์รี่ ข้าวโพด องุ่น ส้ม พืช พริกหยวก มันฝรั่ง ราสเบอร์รี่ ถั่วเหลือง สควอช สตอเบอร์รี่ และมะเขือเทศ แต่ละรูปภาพได้รับการระบุชื่อโรคของใบพืชโดยผู้เชี่ยวชาญ งานวิจัยนี้มีวัตถุประสงค์

เพื่อค้นหาค่าที่เหมาะสมสำหรับแบบจำลองวิธีเรียนรู้เชิงลึก ResNet50 สำหรับการจำแนกรูปภาพโรคพืชที่แสดงลักษณะทางใบ ผู้วิจัยเลือกรูปภาพใบพืชจากชุดข้อมูล PlantVillage มาใช้ในการทดลอง จำนวน 20,638 รูปภาพ ประกอบด้วยรูปภาพใบพืชที่สมบูรณ์ และรูปภาพที่มีอาการโรคพืชจำนวน 15 คลาส จากพืช 3 ชนิด ประกอบด้วย พริกหยวก มันฝรั่ง และมะเขือเทศ ซึ่งคล้ายกับงานของ Kalvakolanu และ Anjaneya Teja Sarma [18] ดังแสดงในรูปภาพที่ 3 แบ่งข้อมูลเป็น 2 ชุด คือ ชุดข้อมูลเรียนรู้ (Training set) 75% และ ชุดข้อมูลทดสอบ (Test set) 25%

4.2 การเตรียมชุดข้อมูล

ไฟล์รูปภาพที่ใช้มีนามสกุล JPG และมีขนาดที่แตกต่างกันสำหรับงานวิจัยนี้ดำเนินการแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบพิกเซล และปรับขนาดรูปให้มีค่าเท่ากับ 100x 100 พิกเซล กำหนดคลาสของโรคใบพืชให้กับรูปภาพทุกรูป คือ ข้อมูลคลาส (class) ประกอบด้วย 15 คลาส ดังนี้



รูปที่ 3. ตัวอย่างรูปภาพที่ใช้ในการทดลอง

ตารางที่ 1 รายละเอียดข้อมูลรูปภาพที่นำมาใช้ในการทดลอง PlantVillage

ลำดับที่	ชื่อพืช	ชื่อโรค	ชื่อคลาส	จำนวนรูปภาพ
1	พริกหยวก	สมบูรณ์	Pepper_bell_healthy	1,478
2	พริกหยวก	โรคใบจุดแบคทีเรีย (Bacterial leaf spot)	Pepper_bell_Bacterial_spot	997
3	มะเขือเทศ	สมบูรณ์	Tomato_healthy	1,591
4	มะเขือเทศ	โรครากไหม้หยี (Leaf mold)	Tomato_Leaf_Mold	952

ลำดับที่	ชื่อพืช	ชื่อโรค	ชื่อคลาส	จำนวนรูปภาพ
4	มะเขือเทศ	โรคราก้ำมะหี (Leaf mold)	Tomato_Leaf_Mold	952
5	มะเขือเทศ	โรคใบหงิกเหลือง (Yellow leaf curl virus)	Tomato_Tomato_Yellow_Leaf_Cur l_Virus	3,208
6	มะเขือเทศ	โรคใบจุดแบคทีเรีย (Bacterial leaf spot)	Tomato_Bacterial_spot	2,127
7	มะเขือเทศ	โรคใบจุดวงกลม (Septoria leaf spot)	Tomato_Septoria_leaf_spot	1,771
8	มะเขือเทศ	โรคไรสองจุด (Two spotted spider mite)	Tomato_Spider_mites_Two_spotted_s pider_mite	1,676
9	มะเขือเทศ	โรคใบจุดวง (Early blight)	Tomato_Early_blight	1,000
10	มะเขือเทศ	โรคใบจุดเป้ากระสุน (Target spot)	Tomato_Target_Spot	1,404
11	มะเขือเทศ	โรคใบไหม้ (Late blight)	Tomato_Late_blight	1,909
12	มะเขือเทศ	โรคใบด่างลิบ (Cucumber mosaic virus)	Tomato_Tomato_mosaic_virus	373
13	มันฝรั่ง	สมบูรณ์	Potato_Healthy	152
14	มันฝรั่ง	โรคใบไหม้ (Late blight)	Potato_Late_blight	1,000
15	มันฝรั่ง	โรคใบจุดวง (Early blight)	Potato_Early_blight	1,000

4.3 ทดลองสร้างแบบจำลองจำแนกโรคพืช

หลังจากที่เตรียมชุดข้อมูลเรียบร้อยแล้วนำชุดข้อมูลที่ได้เข้าเรียนรู้ด้วยวิธี ResNet50 ที่ทำงานร่วมกับอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล ประกอบด้วย AdaDelta AdaGrad Adam RMSProp และ SGD พร้อมทั้งเปรียบเทียบการใช้ฟังก์ชันกระตุ้น ได้แก่ ReLU, Sigmoid และ Tanh เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสม กำหนดรอบการเรียนรู้จำนวน 10 รอบ ประกอบด้วยการทดลอง 15 รูปแบบ

4.4 การวัดและประเมินประสิทธิภาพ

การประเมินประสิทธิภาพการจำแนกประเภทโรคพืช เป็นการประเมินสัดส่วนของผลการทำนายที่ถูกต้องและไม่ถูกต้องของข้อมูลทดสอบ เป็นวิธีการที่ใช้กันอย่างแพร่หลายดังปรากฏในงานวิจัยลักษณะเดียวกันนี้ [10] แทนค่าข้อมูลในเมทริกซ์ความสับสน (Confusion matrix) ประกอบด้วย 4 ส่วน ได้แก่

True-positive (TP) หมายถึง ค่าที่แบบจำลองทำนายว่าจริง และข้อมูลนั้นเป็นจริง

False-positive (FP) หมายถึง ค่าที่แบบจำลองทำนายว่าเท็จ และข้อมูลนั้นเป็นเท็จ

True-negative (TN) หมายถึง ค่าที่แบบจำลองทำนายว่าจริง แต่ข้อมูลนั้นเป็นเท็จ

False-negative (FN) หมายถึง ค่าที่แบบจำลองทำนายว่าเท็จ แต่ข้อมูลนั้นเป็นจริง

ค่า TP FP TN และ FN ได้จากการตรวจสอบผลลัพธ์การทำนายของแบบจำลอง ตัวอย่างเช่น รูปภาพที่ทำนายคือรูปภาพใบมะเขือเทศสมบูรณ์ไม่มีโรค หากแบบจำลองทำนายว่าเป็นมะเขือเทศสมบูรณ์ไม่โรค จะนับเป็นค่า TP

นำค่าที่เหล่านี้ไปคำนวณด้วยเพื่อหาค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าอัตราการทำนายที่ถูกต้องเมื่อข้อมูลเป็นจริง (Recall) และค่า F-measure ดังสมการต่อไปนี้

4.4.1 ค่าความถูกต้อง

ค่าความถูกต้องเป็นการวัดผลรวมของการทำนายที่ถูกต้องทั้งหมด เทียบกับจำนวนการทำนายทั้งหมด

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (18)$$

4.4.2 ค่าความแม่นยำ

ค่าความแม่นยำวัดความถูกต้องของการที่แบบจำลองทำนายตรงกับค่าคำตอบจริง โดยพิจารณาจากผลการทำงานที่ตรงกับค่าคำตอบจริง เปรียบเทียบกับผลการทำงานทั้งหมด

$$Precision = \frac{TP}{TP+FP} \quad (19)$$

4.4.3 ค่าอัตราการจำแนกถูกต้องเมื่อข้อมูลเป็นจริง

เป็นเครื่องมือวัดความสามารถของแบบจำลองในการทำนายข้อมูลได้ถูกต้อง

$$Recall = \frac{TP}{TP+FN} \quad (20)$$

4.4.4 ค่า F-measure

เป็นค่าเฉลี่ยปรับถ่วงน้ำหนักของค่าความแม่นยำและค่าอัตราการจำแนกถูกต้องเมื่อข้อมูลเป็นจริง ซึ่งช่วยการหาสมดุลระหว่างค่าความแม่นยำและค่าอัตราการจำแนกถูกต้องเมื่อข้อมูลเป็นจริง

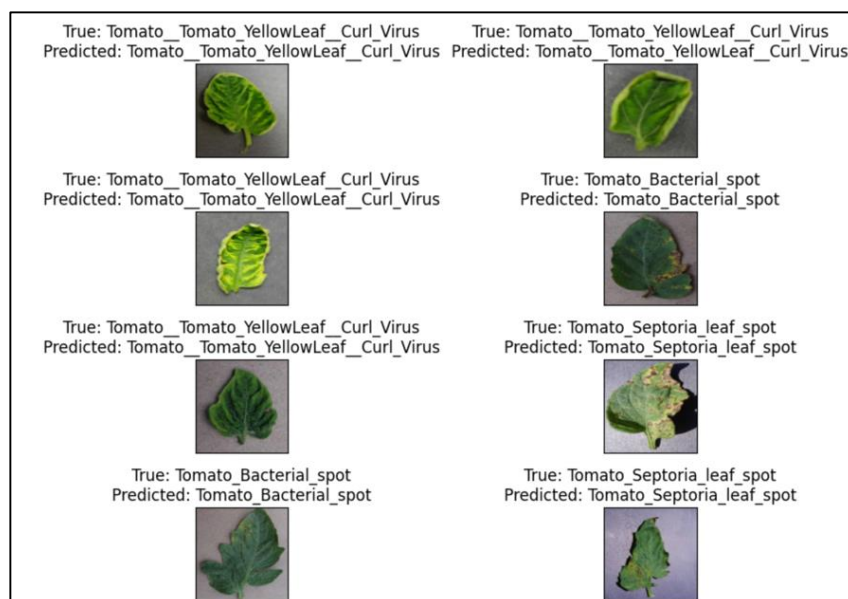
$$F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (21)$$

5. ผลการวิจัย

ผลการทดลองจำแนกรูปภาพโรคพืชด้วยวิธี ResNet50 ที่ทำงานร่วมกับอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล และฟังก์ชันกระตุ้น ตารางที่ 2 แสดงข้อมูลผลการทดลอง ปรากฏว่าผลการทดลองพบว่า วิธี ResNet50 ได้รับการกำหนดค่าพารามิเตอร์ที่เหมาะสม โดยใช้อัลกอริทึมเพิ่มประสิทธิภาพ RMSprop และฟังก์ชันกระตุ้น Sigmoid ให้ค่าความถูกต้อง (Accuracy) สูงสุดคือ 0.94 ค่าความแม่นยำ (Precision) 0.94 และค่าอัตราการจำแนกถูกต้องเมื่อข้อมูลเป็นจริง (Recall) 0.93 และค่า F-measure เท่ากับ 0.93 และรูปภาพที่ 4 แสดงการเปรียบเทียบข้อมูลการทำนายจากแบบจำลองและข้อมูลจริง

6. สรุปผลการวิจัยและข้อเสนอแนะ

การจำแนกรูปภาพด้วยวิธีการเรียนรู้เชิงลึกเป็นงานที่เป็นประโยชน์ต่องานต่างๆ มากมาย การพิจารณาเลือกใช้อัลกอริทึมเพิ่มประสิทธิภาพ และฟังก์ชันกระตุ้น ที่เหมาะสมกับลักษณะข้อมูลเป็นส่วนหนึ่งที่จะช่วยเพิ่มประสิทธิภาพของแบบจำลองงานวิจัยนี้ทำการศึกษาเปรียบเทียบประสิทธิภาพการจำแนกรูปภาพโรคพืชด้วยวิธี ResNet50 ในกระบวนการทดลองได้สร้างแบบจำลองทำงานร่วมกับอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล และฟังก์ชันกระตุ้น เพื่อค้นหาค่าที่เหมาะสมสำหรับวิธี ResNet50 สำหรับการจำแนกรูปภาพโรคพืช กำหนดชุดข้อมูล



รูปที่ 4. แสดงตัวอย่างผลการทำนายโรคพืชจากลักษณะใบพืช

ตารางที่ 2 ผลการทดลองและประเมินประสิทธิภาพการจำแนกโรคพืชโดยใช้ชุดข้อมูล PlantVillage

แบบจำลอง/ วิธีการ	อัลกอริทึมเพิ่ม ประสิทธิภาพ	ฟังก์ชันกระตุ้น	Test Loss:	Accuracy	Precision	Recall	F-measure
ResNet50	Adam	ReLU	0.32545	0.92	0.91	0.89	0.90
ResNet50	SGD	ReLU	0.20043	0.93	0.93	0.91	0.92
ResNet50	RMSprop	ReLU	0.37554	0.93	0.92	0.92	0.92
ResNet50	Adadelta	ReLU	1.90761	0.43	0.35	0.31	0.29
ResNet50	Adagrad	ReLU	0.38899	0.88	0.87	0.84	0.85
ResNet50	Adam	Sigmoid	0.24063	0.92	0.92	0.92	0.92
ResNet50	SGD	Sigmoid	0.77080	0.79	0.68	0.67	0.65
ResNet50	RMSprop	Sigmoid	0.18339	0.94	0.94	0.93	0.93
ResNet50	Adadelta	Sigmoid	2.50328	0.20	0.09	0.10	0.05
ResNet50	Adagrad	Sigmoid	1.83950	0.57	0.42	0.41	0.38
ResNet50	Adam	Tanh	0.30924	0.90	0.91	0.89	0.89
ResNet50	SGD	Tanh	0.21608	0.93	0.93	0.91	0.92
ResNet50	RMSprop	Tanh	0.22305	0.93	0.92	0.92	0.92
ResNet50	Adadelta	Tanh	1.79982	0.52	0.43	0.40	0.37
ResNet50	Adagrad	Tanh	0.44325	0.88	0.87	0.83	0.85

กำหนดรอบการเรียนรู้ (Epoch) จำนวน 10 รอบ PlantVillage จำนวน 20,638 ภาพ ประกอบด้วยรูปภาพใบพืชที่สมบูรณ์ และรูปภาพที่มีอาการโรคพืช 15 คลาส ดังแสดงในตารางที่ 1 แบ่งชุดข้อมูลเป็น 2 คือ ชุดข้อมูลเรียนรู้ 75% และ ชุดข้อมูลทดสอบ 25% ผลการทดลองพบว่า วิธี ResNet50 ได้รับการกำหนดค่าพารามิเตอร์ที่เหมาะสม โดยใช้ Optimizer RMSprop และฟังก์ชันกระตุ้น Sigmoid ให้ค่าความถูกต้อง (Accuracy) สูงสุดคือ 0.94 ค่าความแม่นยำ (Precision) 0.94 และค่าอัตราการจำแนกถูกต้องเมื่อข้อมูลเป็นจริง (Recall) 0.93 และค่า F-measure เท่ากับ 0.93 ดังนั้นในการดำเนินการศึกษาการจำแนกรูปภาพใบพืชปัจจัยที่ส่งผลต่อความถูกต้องคือ การเลือกวิธีการเรียนรู้ และการปรับค่าอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล และฟังก์ชันกระตุ้น งานวิจัยของนศพชาณัน ชินปัญชรนะ [13] เป็นงานที่มีลักษณะคล้ายกันคือจำแนกข้อมูลโรคพืชในมะเขือเทศ ผู้วิจัยได้แสดงผลการทดลองแบบจำลองวิธีคอนโวลูชัน โดยใช้ฟังก์ชันกระตุ้น ReLU มีความถูกต้อง 87.96% ซึ่งสามารถปรับปรุงประสิทธิภาพได้โดยการปรับค่าอัลกอริทึมในการเพิ่มประสิทธิภาพของโมเดล และฟังก์ชันกระตุ้น

แต่อย่างไรก็ตาม งานวิจัยนี้ นำข้อมูลเพียงบางส่วนจากชุดข้อมูล Plant Village ซึ่งประกอบด้วยข้อมูลพืชเพียง 3 ชนิด ซึ่งคล้ายกับงานของ [18] จึงเป็นข้อจำกัดของงานวิจัยครั้งนี้ และด้วยข้อจำกัดของข้อมูลรูปภาพที่เป็นรูปภาพตัวอย่างที่เตรียมข้อมูลสำหรับการจำแนกข้อมูลโดยเฉพาะในสถานการณ์จริงรูปภาพที่รวบรวมได้อาจประกอบด้วยองค์ประกอบที่ส่งผลต่อประสิทธิภาพการจำแนก ดังนั้นงานวิจัยในอนาคตจึงจำเป็นต้องเสนอแบบจำลองสำหรับการจำแนกข้อมูลโดยใช้ชุดข้อมูลขนาดใหญ่ขึ้นและหลากหลายขึ้น รวมทั้งใช้ข้อมูลรูปภาพที่ถ่ายจากสถานที่จริง ที่มีองค์ประกอบครบถ้วนเพื่อให้แบบจำลองสามารถประยุกต์ใช้ได้จริง

เอกสารอ้างอิง

- [1] W. H. Organization, The State of Food Security and Nutrition in the World 2021: Transforming food systems for food security, improved nutrition and affordable healthy diets for all. Food & Agriculture Org., 2021.
- [2] R. Thyagaraj, T. Y. Satheesha, and S. Bhairannawar, "Plant Leaf Disease Classification Using Modified SVM

- With Post Processing Techniques," in 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), 16-17 June 2023 2023, pp. 1-4, doi: 10.1109/ICAISC58445.2023.10201001.
- [3] Y. Kaya and E. Gürsoy, "A novel multi-head CNN design to identify plant diseases using the fusion of RGB images," *Ecological Informatics*, vol. 75, p. 101998, 2023/07/01/ 2023, doi: <https://doi.org/10.1016/j.ecoinf.2023.101998>.
- [4] Y. Chen et al., "Plant image recognition with deep learning: A review," *Computers and Electronics in Agriculture*, vol. 212, p. 108072, 2023/09/01/ 2023, doi: <https://doi.org/10.1016/j.compag.2023.108072>.
- [5] S. B. Jha and R. F. Babiceanu, "Deep CNN-based visual defect detection: Survey of current literature," *Computers in Industry*, vol. 148, p. 103911, 2023/06/01/ 2023, doi: <https://doi.org/10.1016/j.compind.2023.103911>.
- [6] P. Sharma, D. R. Nayak, B. K. Balabantaray, M. Tanveer, and R. Nayak, "A survey on cancer detection via convolutional neural networks: Current challenges and future directions," *Neural Networks*, vol. 169, pp. 637-659, 2024/01/01/ 2024, doi: <https://doi.org/10.1016/j.neunet.2023.11.006>.
- [7] M. A. K. Raiaan et al., "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," *Decision Analytics Journal*, vol. 11, p. 100470, 2024/06/01/ 2024, doi: <https://doi.org/10.1016/j.dajour.2024.100470>.
- [8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533-536, 1986.
- [9] H. H. Xiu, "Research on Activation Function in Deep Convolutional Neural Network," presented at the Proceedings of the 2020 Conference on Artificial Intelligence and Healthcare, Taiyuan, China, 2020. [Online]. Available: <https://doi.org/10.1145/3433996.3434001>.
- [10] V. Balafas, E. Karantoumanis, M. Louta, and N. Ploskas, "Machine Learning and Deep Learning for Plant Disease Classification and Detection," *IEEE Access*, vol. 11, pp. 114352-114377, 2023, doi: 10.1109/ACCESS.2023.3324722.
- [11] M. Belmir, W. Difallah, and A. Ghazli, "Plant Leaf Disease Prediction and Classification Using Deep Learning," in 2023 International Conference on Decision Aid Sciences and Applications (DASA), 16-17 Sept. 2023 2023, pp. 536-540, doi: 10.1109/DASA59624.2023.10286672.
- [12] พิมพ์ ชีวาประกอบกิจ, "การปรับปรุงประสิทธิภาพในการจำแนกภาพด้วยโครงข่ายประสาทแบบคอนโวลูชัน โดยใช้เทคนิคการเพิ่มภาพ," *Journal of Engineering and Digital Technology (JEDT)*, vol. 7, no. 1, pp. 59-64, 2562.
- [13] นัศพ์ชาณัน ชินปัญช์ธนะ, "การจำแนกภาพโรคใบมะเขือเทศด้วยเทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน," *JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY*, vol. 13, no. 2, pp. 40-49, 2566.
- [14] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021/03/01/ 2021, doi: <https://doi.org/10.1016/j.ecoinf.2020.101182>.
- [15] มธุรส ผ่านเมือง and ชนนิกันต์ รอดมรรณี, "การจำแนกภาพใบสมุนไพรรด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชันเชิงลึก," *วารสารวิชาการวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครสวรรค์*, vol. 15, no. 22, pp. 51-65, 2566.
- [16] R. Ding et al., "Improved ResNet Based Apple Leaf Diseases Identification," *IFAC-PapersOnLine*, vol. 55, no. 32, pp. 78-82, 2022/01/01/ 2022, doi: <https://doi.org/10.1016/j.ifacol.2022.11.118>.
- [17] D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.
- [18] A. T. S. Kalvakolanu, "Plant disease detection from images," *arXiv preprint arXiv:2003.05379*, 2020.