โครงข่ายประสาทเทียมแบบคอนโวลูชันเชิงลึก สำหรับการจำแนกป้ายจราจรจำกัดความเร็ว

DEEP CONVOLUTIONAL NEURAL NETWORK FOR SPEED LIMIT SIGNS CLASSIFICATION

ธวัลหทัย จันทรเกษ

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2567

โครงข่ายประสาทเทียมแบบคอนโวลูชันเชิงลึก
สำหรับการจำแนกป้ายจราจรจำกัดความเร็ว

ธวัลหทัย จันทรเกษ

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2567

DEEP CONVOLUTIONAL NEURAL NETWORK FOR SPEED LIMIT SIGNS CLASSIFICATION

TAWANHATAI JANTARAGATE

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

2024

สารนิพนธ์

เรื่อง

โครงข่ายประสาทเทียมแบบคอนโวลูชันเชิงลึก

สำหรับการจำแนกป้ายจราจรจำกัดความเร็ว

ของ

ธวัลหทัย จันทรเกษ

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร.วีรยุทธ เจริญเรืองกิจ)

..... ประธาน

(ผู้ช่วยศาสตราจารย์ ดร.มาลีรัตน์ มะลิแย้ม)

..... กรรมการ

(อาจารย์ ดร.โสภณ มงคลลักษมี)

ชื่อเรื่อง	โครงข่ายประสาทเทียมแบบคอนโวลูชันเชิงลึก สำหรับการจำแนกป้ายจราจรจำกัดความเร็ว
ผู้วิจัย	ธวัลหทัย จันทรเกษ
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2567
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. วีรยุทธ เจริญเรืองกิจ

งานวิจัยนี้มีวัตถุประสงค์ เพื่อศึกษาโครงสร้างของแบบจำลองและเปรียบเทียบพารามิเตอร์ที่ส่งผลกับการจำแนกป้ายจราจรประเภทป้ายจำกัดความเร็ว ด้วยเทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน โดยใช้แบบจำลองทั้งหมด 5 ชนิด ได้แก่ LeNet-5 AlexNet VGG16 ShuffleNet และ MobileNet โดยใช้ภาพถ่ายจริงของป้ายจราจรจากชุดข้อมูลการจดจำป้ายจราจรเยอรมัน (German Traffic Sign Recognition Benchmark) จาก Kaggle โดยคัดเลือกเฉพาะที่ป้ายจราจรที่จำกัดความเร็วเท่านั้น จำนวนรูปภาพรวมทั้งสิ้น 12,780 ภาพ งานวิจัยนี้มีจุดประสงค์เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองกับกลุ่มภาพที่แตกต่างกันตามความละเอียด ได้แก่ กลุ่มภาพความละเอียดต่ำกลุ่มภาพความละเอียดสูง และกลุ่มภาพความละเอียดผสม ซึ่งแบ่งกลุ่มตามค่าเฉลี่ยของความละเอียดของรูปภาพ โดยใช้โปรแกรม KNIME เพื่อหากลุ่มภาพที่ทำให้แบบจำลองมีประสิทธิภาพในการคัดแยกมากที่สุด ซึ่งผลการทดลองพบว่าแบบจำลองที่มีประสิทธิภาพดีที่สุด คือ ShuffleNet มีความสามารถในการคัดแยกป้ายจราจรจำกัดความเร็วได้ดีที่สุด รองลงมาเป็น Lenet-5 และ AlexNet

คำสำคัญ : โครงข่ายประสาทเทียมแบบคอนโวลูชัน, การเรียนรู้เชิงลึก, การคัดแยกป้ายจราจร, การจำกัดความเร็ว

Title	DEEP CONVOLUTIONAL NEURAL NETWORK FOR SPEED LIMIT SIGNS CLASSIFICATION
Author	TAWANHATAI JANTARAGATE
Degree	MASTER OF SCIENCE
Academic Year	2024
Thesis Advisor	Assistant Professor Dr. Werayuth Charoenruengkit

The objective of this research is to study the structure of the model and compare the parameters that affect the classification of traffic signs with speed limit restrictions. Using convolutional neural network (CNN) techniques Using LeNet-5, AlexNet, VGG16, ShuffleNet, and MobileNet models, trained on traffic sign images dataset obtained from Kaggle named GTSRB - German Traffic Sign Recognition Benchmark, selecting images that are only speed limit signs. The total number of images was 12,780. The study seeks to compare model performance across groups of images with varying sizes or resolutions, divided into three groups: low resolution images, high resolution images, and mixed resolution images which are grouped according to the average resolution of the image using the KNIME Platform. The experimental findings ShuffleNet achieved the best performance with an Accuracy of 99% and LeNet-5 and AlexNet closely follow with Accuracy of 98%

Keyword : Convolutional Neural Network Deep Learning Traffic Sign Classification Speed Limit Enforcement

กิตติกรรมประกาศ

สารนิพนธ์นี้สำเร็จลุล่วงได้ด้วยความเมตตากรุณาช่วยเหลือ และความเอาใจใส่อย่างดียิ่ง ตลอดจนการให้คำแนะนำ และข้อคิดเห็นที่เป็นประโยชน์อย่างยิ่งสำหรับการปรับแก้ไขข้อบกพร่อง จากคณะกรรมการผู้ควบคุมสารนิพนธ์ ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างยิ่ง

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. วีรยุทธ เจริญเรืองกิจ ที่ได้ให้ความเมตตากรุณาเป็นที่ปรึกษาและให้ความช่วยเหลือชี้แนะแนวทางในสิ่งที่เป็นประโยชน์ต่อการศึกษาและการทำสารนิพนธ์นี้ด้วยความเอาใจใส่ตลอดมา ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณคณาจารย์และกรรมการบริหารหลักสูตรสาขาวิทยาการข้อมูล คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒทุกท่าน ที่ได้กรุณาประสิทธิ์ประสาทความรู้ต่าง ๆ ให้แก่ผู้วิจัย ตลอดจนให้ความช่วยเหลือในการทำวิจัยครั้งนี้

ขอขอบคุณพี่ ๆ และเพื่อน ๆ สาขาวิทยาการข้อมูล รวมถึงบุคคลอีกหลายท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ที่ได้ให้ความช่วยเหลือและเป็นกำลังใจให้กับผู้วิจัยมาโดยตลอด

สุดท้ายนี้ ผู้วิจัยขอน้อมรำลึกถึงคุณของบิดามารดาและครูอาจารย์ ที่อบรมสั่งสอนให้ความรู้เป็นกำลังใจและให้การสนับสนุนผู้วิจัยด้วยดีตลอดมา ขอกราบขอบพระคุณ

ธวัลหทัย จันทรเกษ

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ภูมิหลัง

ในปัจจุบัน การขับรถเร็วเป็นอุบัติเหตุที่เกิดขึ้นอย่างต่อเนื่องในประเทศไทย โดยในปี 2563-2564 มีอุบัติเหตุสูงถึงร้อยละ 77-79 ที่เกิดขึ้นจากการใช้ความเร็ว และสูงถึงร้อยละ 69-74 ที่เป็นสาเหตุส่งผลให้มีผู้เสียชีวิต (สถาบันเทคโนโลยีแห่งเอเชีย, 2565) ซึ่งนับว่าเป็นหนึ่งในสาเหตุหลักที่นำไปสู่อุบัติเหตุที่รุนแรงบนท้องถนน เสี่ยงต่อชีวิต ส่งผลกระทบทางสังคมและเศรษฐกิจ ตลอดจนการสูญเสียบุคคลสำคัญภายในครอบครัว แม้ว่าจะมีการนำเทคโนโลยีการตรวจจับความเร็วแบบอัตโนมัติเข้ามาใช้เพื่อการตรวจจับผู้ขับขี่ที่ใช้ความเร็ว ร่วมกับการกำหนดอัตราโทษทางกฎหมายที่สูงขึ้น แต่จำนวนของผู้กระทำความผิดก็ยังสูงขึ้น ซึ่งพิจารณาแล้ว พบว่าการแก้ปัญหาเรื่องความเร็วข้างต้นนั้น เป็นเพียงการแก้ไขปัญหาเพื่อลดอุบัติเหตุที่จะเกิดขึ้นเท่านั้น

ในขณะเดียวกัน การคิดค้นรถยนต์อัตโนมัติหรือไร้คนขับเป็นการทำงานร่วมกันของเทคโนโลยีด้านต่าง ๆ ที่เกี่ยวข้องกับคอมพิวเตอร์และนำเอาการเรียนรู้เชิงลึกมาช่วยในการตรวจจับวัตถุ (DEPA, 2019) ให้สามารถตัดสินใจได้ด้วยตัวเอง จากการประมวลผลและสกัดคุณลักษณะที่สำคัญของข้อมูลรูปภาพที่รับมา เพื่อระบุและจัดประเภทของวัตถุในภาพหรือวีดีโอได้ แม้ว่าจะมีเทคนิคการคัดแยกป้ายจราจรหลายวิธีที่ถูกพัฒนาขึ้นเพื่อตรวจจับและรับรู้ป้ายจราจร เช่น การคัดแยกตามสี ,ความเข้มของสี, การใช้รูปทรงเรขาคณิต แต่เมื่อพิจารณาแล้ว ความละเอียดของภาพมีผลต่อความชัดเจนของรายละเอียดบนภาพ โดยเมื่อความละเอียดสูงแบบจำลองจะสามารถตรวจจับข้อผิดพลาดหรือรายละเอียดเล็ก ๆ ที่อยู่บนภาพได้ดีขึ้น

ดังนั้น เพื่อทดสอบประสิทธิภาพการคัดแยกป้ายจราจรจำกัดความเร็วของแบบจำลองที่สร้างขึ้น กับรูปภาพที่มีลักษณะป้ายเหมือนกัน แต่แตกต่างกันด้วยรูปตัวเลขที่บ่งบอกถึงความเร็วที่จำกัดในแต่ละช่วงและความละเอียดของภาพที่มีผลต่อการตรวจจับและระบุป้ายจราจรได้โดยตรง เพื่อเปรียบเทียบแบบจำลองที่เหมาะสมในการคัดแยกป้ายจราจร เมื่อได้ผลลัพธ์แล้ว ส่วนงานที่เกี่ยวข้องสามารถนำไปใช้ในการจำแนกป้ายจราจร หรือ การตรวจสอบประเภทของป้ายจราจรดังกล่าวได้ เพื่อประโยชน์และลดปัญหาในการกระทำความผิดทางจราจรในอนาคตต่อไป

1.2 ความมุ่งหมายของงานวิจัย

1. ศึกษาและเปรียบเทียบประสิทธิภาพของโครงข่ายประสาทเทียมชนิดคอนโวลูชัน สำหรับการจำแนกป้ายจราจรจำกัดความเร็ว เพื่อให้ทราบถึงภาพรวมของประสิทธิภาพ และการนำไปประยุกต์ใช้งาน

2. ศึกษาและวิเคราะห์กลุ่มความละเอียดของรูปภาพ เพื่อหาความสัมพันธ์ที่สำคัญระหว่างความละเอียดและประสิทธิภาพแบบจำลองในการคัดแยกป้ายจราจรที่จำกัดความเร็ว

3. เพื่อสร้างและปรับปรุง รวมทั้งศึกษาตัวแปรต่าง ๆ ที่ส่งผลกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน สำหรับการจำแนกป้ายจราจรชนิดป้ายจำกัดความเร็ว

1.3 ความสำคัญของการวิจัย

งานวิจัยนี้มุ่งเน้นการศึกษาและพัฒนาแบบจำลอง รวมถึงตัวแปรที่ส่งผลกับการจำแนกป้ายจราจรประเภทป้ายจำกัดความเร็วที่มีความละเอียดแตกต่างกัน ซึ่งการจดจำและจำแนกป้ายจราจรมีความสำคัญอย่างมากในการพัฒนาระบบตรวจจับข้อมูลทางจราจร ซึ่งจะสามารถช่วยลดอุบัติเหตุและเพิ่มความปลอดภัยแก่ผู้ขับขี่ยานพาหนะ นอกจากนั้นยังสามารถนำเอาเทคโนโลยีนี้ไปใช้กับงานในอนาคตได้อีกด้วย ซึ่งผลลัพธ์จากการวิจัยที่ได้ จะถูกแปรผล เพื่อนำไปวิเคราะห์และปรับใช้งานในด้านการจราจรต่อไป

1.4 ขอบเขตของการวิจัย

ในการวิจัยครั้งนี้ผู้วิจัยได้กำหนดขอบเขตของการวิจัยไว้ดังนี้

1. ชุดข้อมูลสำหรับงานวิจัย เป็นชุดข้อมูลภาพถ่ายป้ายจราจร GTSRB - German Traffic Sign Recognition Benchmark (INI, 2013) มีข้อมูลภาพถ่ายป้ายจราจรจากประเทศเยอรมนี เพื่อทดสอบการจำแนกแบบหลายคลาส โดยผู้วิจัยได้คัดเลือกรูปภาพเฉพาะป้ายจราจรจำกัดความเร็ว ซึ่งมีจำนวนคลาส 8 คลาส และมีจำนวนภาพ 12,780 ภาพ จากทั้งหมด 50,000 ภาพ 42 คลาส

2. ศึกษาและสร้างแบบจำลองโครงข่ายประสาทเทียมชนิดคอนโวลูชัน 5 แบบจำลอง ได้แก่ แบบจำลอง LeNet-5, AlexNet, VGG16, ShuffleNet และ MobileNet

3. วิธีการประเมินประสิทธิภาพของแบบจำลองที่สร้างขึ้น โดยใช้ค่าความถูกต้อง (Accuracy) และค่าอื่น ๆ เช่น ค่าความแม่นยำ (Precision) ค่า Recall และ F-Measure

1.5 กรอบแนวคิดงานวิจัย

งานวิจัยนี้เป็นการศึกษาและเปรียบเทียบโครงสร้างของแบบจำลอง 5 ชนิด ได้แก่ Lenet-5, AlexNet, VGG16, ShuffleNet, MobileNet รวมถึงตัวแปรและปัจจัยแวดล้อมอื่นๆ ที่มีผลต่อการจำแนกป้ายจราจรชนิดป้ายจำกัดความเร็ว เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด ทั้งในเรื่องของประสิทธิภาพ ความถูกต้องและเวลา สำหรับการจำแนกป้ายจราจรตามประเภทป้ายจำกัดความเร็ว

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อต่อไปนี้

1. เอกสารและหลักการทฤษฎีที่เกี่ยวข้อง
    1.1 การใช้งานโปรแกรม Konstanz Information Miner Analytics Platform
    1.2 ข้อมูลที่เกี่ยวกับเครื่องหมายจราจรหรือป้ายจราจร (Traffic Sign)
    1.3 ทฤษฎีที่เกี่ยวข้องกับโครงข่ายประสาทเทียม (Neural Network) และโครงข่ายประสาทเทียมชนิดคอนโวลูชัน (Convolution Neural Network)
    1.4 ทฤษฎีที่เกี่ยวข้องกับการทำการขยายข้อมูลชุดฝึก (Data Augmentation)
2. งานวิจัยที่เกี่ยวข้อง

2.1 เอกสารและหลักการทฤษฎีที่เกี่ยวข้อง

2.1.1 การใช้งานโปรแกรม Konstanz Information Miner Analytics Platform

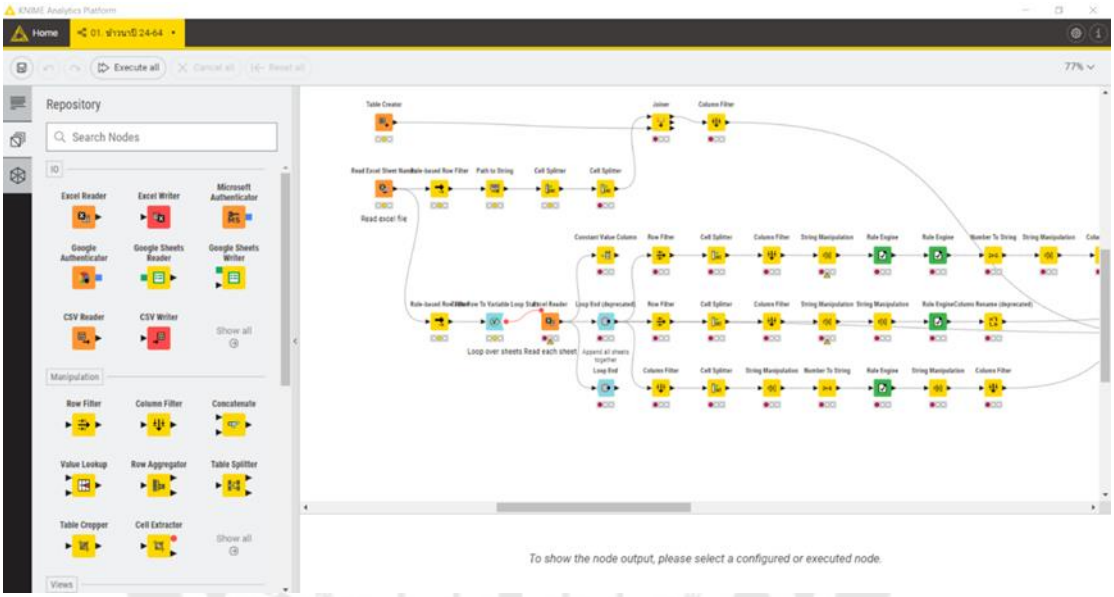
Konstanz Information Miner (KNIME) (Berthold และคนอื่น ๆ, 2009) เป็นซอฟต์แวร์โอเพนซอร์สที่ใช้ในงานด้านวิทยาศาสตร์ข้อมูล โดยสามารถใช้งานเพื่อบริหารจัดการข้อมูลที่อยู่ในรูปแบบต่าง ๆ ได้ เช่น ฐานข้อมูล SQL หรือไฟล์ CSV เป็นต้น ซึ่งลักษณะการทำงานจะเป็นกระบวนการเชื่อมต่อกัน (Workflow) ดังภาพประกอบ 1 โดยแสดงขั้นตอนการวิเคราะห์แบบเป็นลำดับตามการเชื่อมต่อของโหนด (Node) ซึ่งโหนดแต่ละตัวจะมีหน้าที่แตกต่างกัน เช่น โหนดการอ่านข้อมูล (Reader Node) โหนดการกรองข้อมูล (Filter Node) หรือโหนดการแสดงผลรูปแบบต่าง ๆ (Visualizations Node) โดยมีกระบวนการทำงานหลัก ๆ (KNIME, 2023) ดังต่อไปนี้

2.1.1.1 การรวบรวมข้อมูล (Access and Blend) เป็นขั้นตอนการรวมข้อมูลที่นำเข้ามาจากแหล่งต่าง ๆ เช่น ไฟล์ CSV, ฐานข้อมูล หรือการสกัดข้อมูลจากเว็บไซต์ (Web Scrapping)

2.1.1.2 การแปลงข้อมูล (Transform and Analyze) เป็นกระบวนการแปลงข้อมูลเพื่อทำให้ข้อมูลพร้อมสำหรับการวิเคราะห์ เช่น การทำความสะอาดข้อมูล การกรอง หรือการปรับโครงสร้างข้อมูลให้เหมาะสม

2.1.1.3 การแสดงภาพและสำรวจข้อมูล (Visualize and Explore) เป็นการแสดงข้อมูลในรูปแบบต่าง ๆ เพื่อให้เข้าใจข้อมูลและสามารถหาความสัมพันธ์ของข้อมูลที่เกี่ยวข้องกันได้

2.1.1.4 การบันทึกและนำไปใช้ใหม่ (Save & Reuse) เป็นกระบวนการบันทึกผลลัพธ์จากกระบวนการเชื่อมต่อที่สร้างขึ้น เพื่อการใช้งานในอนาคตและการแบ่งปันให้ผู้อื่นใช้งาน



ภาพประกอบ 1 ตัวอย่างหน้าจอโปรแกรม KNIME

2.1.2 ข้อมูลที่เกี่ยวกับเครื่องหมายจราจรหรือป้ายจราจร (Traffic Sign)

เครื่องหมายจราจร หรือ ป้ายจราจร เป็นสัญลักษณ์แสงหรือป้ายที่มีวัตถุประสงค์เพื่อจัดการและบังคับการเคลื่อนตัวของการจราจร การจอด หรืออาจจะเป็นการเตือน หรือการแนะนำ ในทางจราจร ให้มีความปลอดภัย สำหรับผู้ขับขี่พาหนะทุกประเภท โดยแต่ละป้ายจราจรมีความสำคัญและแบ่งตามวัตถุประสงค์ของการใช้งาน (กระทรวงคมนาคม, 2561) ดังนี้

2.1.2.1 ป้ายจราจรประเภทป้ายบังคับ เป็นป้ายที่ไว้บังคับใช้ผู้ขับขี่ยานพาหนะรวมถึงคนเดินทางเท้าให้ปฏิบัติตามข้อบังคับตามป้ายจราจรนี้อย่างเคร่งครัด โดยแบ่งออกเป็น 4 ประเภท ได้แก่ ป้ายบังคับประเภทกำหนดสิทธิ์ (Priority Regulating Signs) ป้ายบังคับประเภทห้ามหรือจำกัดสิทธิ์ (Prohibitory or Restrictive Signs) ป้ายบังคับประเภทคำสั่ง (Mandatory Signs) และป้ายบังคับประเภทป้ายอื่น ๆ

2.1.2.2 ป้ายจราจรประเภทป้ายเตือน เป็นป้ายที่แจ้งเตือนให้ผู้ขับขี่ยานพาหนะระมัดระวังหรือต้องเพิ่มความระวังในทางข้างหน้า เพื่อให้ผู้ขับขี่เพิ่มความระวังมากขึ้น ได้แก่ ป้าย

เตือนตามรูปแบบ และลักษณะที่กำหนด (ป้ายจราจรสีเหลือง-ดำ), ป้ายเตือนที่แสดงด้วยข้อความและ/หรือสัญลักษณ์ (ป้ายจราจรสีเหลือง-ดำ) และป้ายเตือนก่อสร้าง (ป้ายจราจรสีส้ม -ดำ)

2.1.2.3 ป้ายจราจรประเภทป้ายแนะนำ มีไว้เพื่อแนะนำข้อมูลการเดินทางการจราจรให้แก่ผู้ขับขี่ยานพาหนะ ไปยังจุดหมายได้อย่างปลอดภัย

ซึ่งป้ายจราจรจำกัดความเร็ว ถูกจัดอยู่ในประเภท ป้ายบังคับประเภทป้ายอื่น ๆ โดยมีการใช้พื้นหลังของป้ายเป็นสีขาว เส้นขอบของป้ายใช้สีแดง ตัวเลข ตัวอักษรและสัญลักษณ์ใช้สีดำ

2.1.3 ทฤษฎีที่เกี่ยวข้องกับโครงข่ายประสาทเทียม (Neural Network) และโครงข่ายประสาทเทียมชนิดคอนโวลูชัน (Convolution Neural Network)

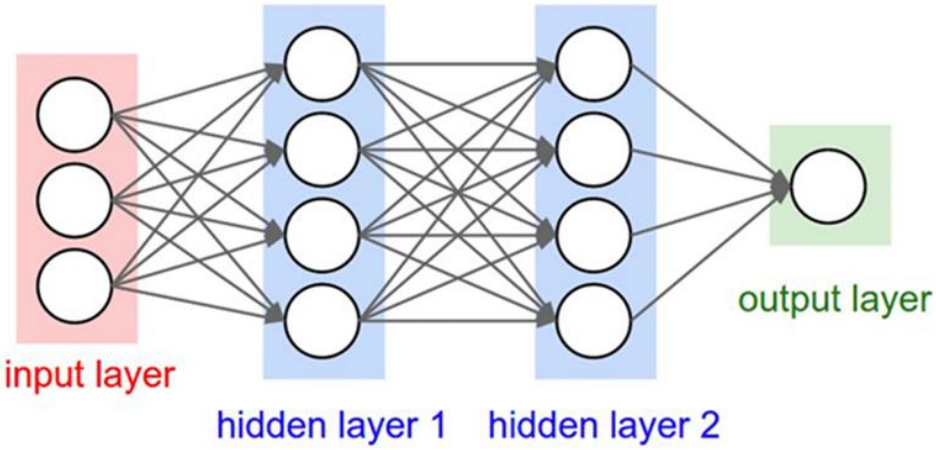
2.1.3.1 โครงข่ายประสาทเทียม (Neural Network)

เป็นวิธีการสร้างแบบจำลองข้อมูลชนิดหนึ่ง เหมือนการทำงานของสมองมนุษย์ โดยข้อมูลที่นำมาประมวลผลในแบบจำลองจะไหลเข้าไปในโครงข่ายที่เชื่อมต่อกัน ซ้อนทับกันหลายชั้น สามารถประมวลผล เรียนรู้ด้วยและปรับตัวเอง ตามความซับซ้อนของการวิเคราะห์เพื่อแก้ปัญหาและสถานการณ์ต่าง ๆ ได้ ซึ่งประกอบด้วย หน่วยประสาท (Neuron) ที่มีขนาดเล็กจำนวนมาก เชื่อมต่อกันเป็นชั้น (Layers) ซึ่งหน่วยเล็กที่สุด ถูกเรียกว่า เพอร์เซ็ปตรอน โดยมีการแบ่งชั้น ดังภาพประกอบ 2 ดังนี้

1. ชั้นข้อมูลรับเข้า (Input layer) เป็นชั้นแรกในโครงข่ายประสาทเทียมที่รับข้อมูลเข้าสู่โครงข่าย โดยไม่มีการคำนวณ หรือ ปรับค่าต่าง ๆ ซึ่งจำนวนของนิวรอนในชั้นนี้ จะเป็นตัวแทนของค่าข้อมูลที่ต้องการนำเข้ามาให้โครงข่ายประสาทเทียมเพื่อประมวลผล สามารถเป็นภาพ ข้อความ ข้อมูลตาราง หรือประเภทข้อมูลอื่น ๆ

2. ชั้นซ่อน (Hidden Layer) เป็นชั้นที่ซ่อนอยู่ตรงกลาง มีผลต่อประสิทธิภาพของแบบจำลอง โดยขึ้นอยู่กับจำนวนและขนาดของชั้นซ่อน รวมถึงจำนวนของหน่วยประมวลผลในแต่ละชั้นด้วย การเพิ่มหรือลดจำนวนของชั้นซ่อน หรือ จำนวนของหน่วยประมวลผลในแต่ละชั้นส่งผลต่อความซับซ้อนของโมเดล ซึ่งการเพิ่มความซับซ้อนและใช้เวลาคำนวณที่นานขึ้นนั้นก็ขึ้นอยู่กับจำนวนของชั้นซ่อนหรือจำนวนของหน่วยประมวลในแต่ละชั้นด้วย

3. ชั้นข้อมูลออก (output Layer) เป็นชั้นสุดท้ายในโครงข่ายประสาทเทียม ซึ่งรับข้อมูลที่ผ่านการวิเคราะห์ประมวลผลจากชั้นซ่อนก่อนหน้า ออกมาเป็นผลลัพธ์หรือการทำนาย โดยในแต่ละการเชื่อมโยงของนิวรอนที่มีการประมวลผลส่งต่อกันเป็นเครือข่าย จะมีค่าน้ำหนัก (Weight) เป็นค่าพื้นฐานของการเรียนรู้ จึงทำให้ในแต่ละเซลล์มีการเรียนรู้ที่แตกต่างกันส่งผลให้มีการปรับค่าน้ำหนักแต่ละครั้งด้วย

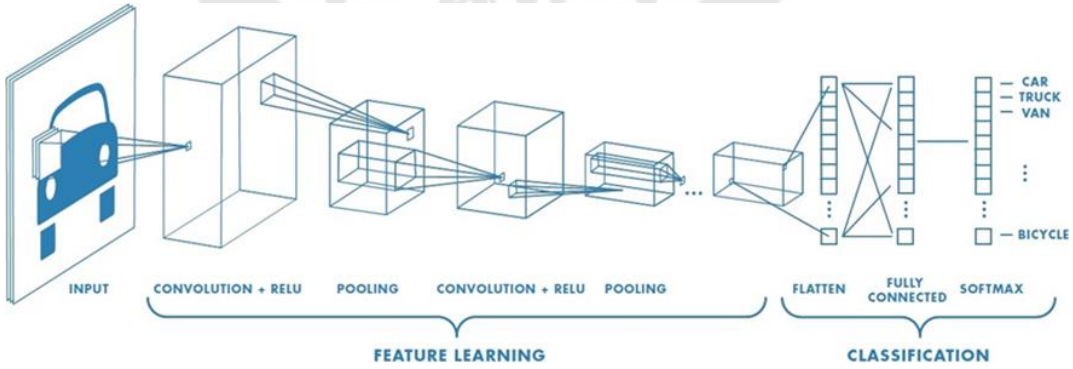


ภาพประกอบ 2 แสดงโครงสร้างโครงข่ายประสาทเทียม (Neural Network)

ที่มา: https:// www.i2tutorials.com/hidden-layers-in-neural-networks

2.1.3.2 โครงข่ายประสาทเทียมชนิดคอนโวลูชัน (Convolutional Neural Network)

โครงข่ายประสาทเทียมชนิดคอนโวลูชัน เป็นโครงข่ายประสาทเทียมรูปแบบหนึ่งที่ทำการประมวลทางสัญญาณและทางคอมพิวเตอร์วิชัน โดยสามารถรับข้อมูลประเภทเมทริกซ์ที่ถูกแปลงมาจากข้อมูลชนิดรูปภาพ และสกัดคุณลักษณะพิเศษ (Feature Extraction) ซึ่งมีวิธีการดำเนินการและส่วนประกอบที่สำคัญ ดังภาพประกอบ 3



ภาพประกอบ 3 โครงข่ายประสาทเทียมชนิดคอนโวลูชัน (Convolutional Neural Network)

ที่มา: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f14

1. ข้อมูลรับเข้า (Input) เป็นข้อมูลที่รับเข้าสู่โครงข่ายประสาทเทียมชนิดคอนโวลูชัน ซึ่งส่วนมากจะเป็นข้อมูลชนิดรูปภาพ ที่มีขนาดเท่ากับความกว้าง (Width) x ความสูง (Height) x ความลึก (Depth) โดยที่ความลึกหมายถึงจำนวนช่องทาง (Channel) ของสี เช่น RGB จะมีความลึกเป็น 3 ช่องทาง หรือหากเป็นภาพในโรงพิมพ์จะแทนด้วย 4 Channel คือ CMYK

2. การเรียนรู้คุณลักษณะ (Feature Learning) เป็นกระบวนการที่แบบจำลองเรียนรู้หรือสกัดคุณลักษณะที่สำคัญของข้อมูลรับเข้า ซึ่งประกอบด้วย

2.1 ชั้นคอนโวลูชัน (Convolutional layer) เป็นชั้นที่มีการดำเนินการทางคอนโวลูชันในโครงข่าย โดยประกอบด้วยข้อมูลนำเข้าและเมทริกซ์ถ่วงน้ำหนัก ที่เรียกว่าเคอร์เนลหรือฟิลเตอร์ (Kernel/Filter) ซึ่งกระบวนการทางคอนโวลูชัน เริ่มจากการนำเคอร์เนลหรือฟิลเตอร์ไปวางทับที่ภาพนำเข้า จากนั้นค่าในแต่ละช่องที่เมทริกซ์ทับกันจะถูกนำมาคูณกัน และผลลัพธ์จากการคูณนี้จะถูกรวมกันเพื่อให้ได้ผลลัพธ์สุดท้ายของช่องนั้น ๆ ของภาพ และเมื่อความสำคัญของข้อมูลที่อยู่บริเวณขอบของภาพไม่ถูกนำมาใช้งานให้เพียงพอ การใช้วิธีการแพดดิ้ง (Padding) จะถูกนำมาใช้เพื่อเพิ่มพิกเซลบริเวณขอบของภาพ โดยยังคงรักษาขนาดของภาพเหมือนเดิม ผลลัพธ์จากการคำนวณนี้จะเป็นเมทริกซ์ใหม่ที่เรียกว่า Feature Map หรือ Activation Map

2.2 ชั้นพลูลิ่ง (Pooling Layer) เป็นส่วนหนึ่งของโครงข่ายที่ใช้ในการลดขนาด (Down sampling) ของ Feature Map ที่ได้รับจากชั้นคอนโวลูชัน ชั้นนี้มักถูกนำมาใช้เพื่อลดตัวแปรที่ใช้คำนวนภายในโครงข่าย โดยใช้ฟังก์ชันทางคณิตศาสตร์ หรือวิธีการอื่น ๆ เช่น การใช้ค่าเฉลี่ย (Average Pooling) การใช้ค่าสูงที่สุด (Max Pooling) หรือ การใช้ค่าต่ำสุด (Min Pooling) เพื่อลดขนาดของ Feature Map

3. การจำแนกประเภท (Classification) เป็นขั้นตอนกระบวนการแบ่งข้อมูลเป็นประเภทหรือกลุ่มต่าง ๆ ตามคุณลักษณะหรือคุณสมบัติของข้อมูล ซึ่งประกอบด้วย

3.1 แฟลทเทิน (Flatten) เป็นการแปลงข้อมูลในรูปแบบเวกเตอร์ให้เหมาะสมสำหรับการใช้งานในชั้นเชื่อมโยงสมบูรณ์ (Fully-Connected Layer) หรือชั้นสุดท้ายของโครงข่ายประสาทเทียม ซึ่งในชั้นถัดไป ข้อมูลจะต้องอยู่ในรูปแบบเวกเตอร์ 1 มิติ

3.2 ชั้นเชื่อมโยงสมบูรณ์ (Fully-Connected Layer) เป็นชั้นที่เชื่อมโยงระหว่าง Feature Map และข้อมูลส่งออก (output) ของแบบจำลอง โดยส่วนของ Feature Map จะถูกแปลงให้เป็นลักษณะเวกเตอร์ (vector) แล้วเชื่อมต่อกับชั้นเชื่อมโยงสมบูรณ์ โดยแต่ละการ

เชื่อมต่อจะมีน้ำหนัก (weight) และไบแอส (bias) ที่ต้องเรียนรู้ในระหว่างการฝึกแบบจำลอง ซึ่งในชั้นนี้จะเป็นตัวแปรหลักของการได้ผลลัพธ์ที่สมบูรณ์ที่สุด

3.3 ฟังก์ชันซอฟต์แม็กซ์ (Softmax Function) จะอยู่ชั้นสุดท้าย มีไว้สำหรับการแปลงผลลัพธ์เป็นค่าความน่าจะเป็นของแต่ละเภท ซึ่งประเภทที่มีความน่าจะเป็นสูงสุดจะเป็นผลของการทำนายประเภทของข้อมูลนั้น ๆ

นอกจากนี้ พารามิเตอร์อื่น ๆ ที่สำคัญอีก เช่น

1. ตัวกรอง (Filters) เป็นตัวกรองที่ใช้ในการสกัดคุณลักษณะพิเศษจากข้อมูล โดยสามารถกำหนดขนาดและจำนวนของตัวกรองได้ตามต้องการ เช่น ตัวกรองขนาด 3 x 3, 5 x 5 พิกเซล ซึ่งแต่ละตัวกรองที่กำหนดก็จะให้ผลแตกต่างกัน

2. จำนวนตัวกรอง (Number of filters) ใช้ในการสกัดคุณลักษณะพิเศษจากข้อมูล ซึ่งมีได้มากกว่า 1 ตัว และน้ำหนักของแต่ละตัวจะแตกต่างกันก็ได้

3. การเลื่อน (Strides) เป็นการเลื่อนตำแหน่งของตัวกรองในการสกัดคุณลักษณะ เช่น Stride = 2 หมายความว่าจะเลื่อนครั้งละ 2 พิกเซล

4. การทำแพดดิ้ง (Padding) เป็นวิธีการนำข้อมูลบริเวณขอบรูปหรือบริเวณฟีเจอร์ที่สำคัญที่ซ่อนอยู่ในภาพ ให้ถูกนำมาใช้งาน เพื่อให้ข้อมูลที่ผ่านการคอนโวลูชันมีขนาดเท่าเดิมหลังจากกระบวนการทางคอนโวลูชัน เพื่อป้องกันการสูญเสียข้อมูลบริเวณขอบของภาพ ตัวอย่างการทำแพดดิ้ง เช่น Valid Padding Half/Same Padding ข้อมูล หรือ Fully Padding

5. ฟังก์ชันกระตุ้น (Activation Function) เป็นฟังก์ชันสมการที่รับผลรวมจากการประมวลผลทั้งหมด ในทุกชั้น (Dense) และทำการคำนวนเพื่อส่งต่อไปเป็น Output เช่น Activation Function ReLU, Sigmoid หรือ Tanh เป็นต้น

6. อัลกอริทึมเพื่อการหาค่าที่ดีที่สุด (Optimization Algorithms) ใช้ในการปรับค่าตัวแปรของเพื่อลดค่าความคาดเคลื่อนและปรับปรุงประสิทธิภาพของโมเดล เช่น AdaDelta Optimizer, Adam Optimizer, RMSProp Optimizer, Momentum Optimizer เป็นต้น

7. ฟังก์ชันสูญเสีย (Loss Function/Cost Function) ใช้คำนวณค่าความผิดพลาดระหว่างผลลัพธ์ของโมเดลกับค่าเป้าหมายที่แท้จริง เพื่อหาค่าถ่วงน้ำหนักที่ดีที่สุด ได้แก่ ครอส-เอนโทรปี ล็อซ (Cross-entropy loss)

2.1.3.3 โครงข่ายประสาทเทียมคอนโวลูชัน LeNet-5

ในบทวิจัย Gradient-Based Learning Applied to Document Recognition (Lecun, Bottou, Bengio, และ Haffner, 1998) Lenet-5 ถูกนำเสนอโดย Yann LeCun ในปี 1998 เป็นการเริ่มต้นใช้โครงข่ายประสาทเทียมในงานจดจำและจำแนกประเภทตัวเลขและตัวอักษรที่เขียนด้วยลายมือหรือที่พิมพ์ด้วยเครื่อง โครงสร้างของ LeNet-5 ประกอบด้วยชั้นคอนโวลูชัน (Convolutional Layers) ทั้งหมด 3 ชั้น มี Feature map ทั้งหมด 6, 16 และ120 ตามลำดับ โดยใช้ตัวกรอง (filters) ขนาด 5X5 และมีการเลื่อน (strides) ครั้งละ 1 พิกเซล ซึ่ง Feature map ในทุกชั้นจะถูกประมวลผลด้วยฟังก์ชันกระตุ้นชนิด hyperbolic tangent (tanh) ก่อนถูกส่งไปยังชั้นถัดไป หลังจากคอนโวลูชันแต่ละชั้นจะมีการทำ average pooling ขนาด 2x2 และเลื่อนครั้งละ 2 พิกเซล เพื่อลดขนาดของ feature maps และลดความซับซ้อน ซึ่งหลังจากการทำ poolingเสร็จสิ้น ข้อมูลจะถูกนำเข้าไปยังชั้นเชื่อมโยงสมบูรณ์ (fully connected layer) ที่มี 84 โหนด และสุดท้าย ชั้นออก (output layer) ประกอบด้วย 10 โหนด ซึ่งแทนเลข 0-9 เพื่อจำแนกประเภทของตัวเลข

2.1.3.4 โครงข่ายประสาทเทียมคอนโวลูชัน AlexNet

ในบทวิจัย ImageNet Classification with Deep Convolutional Neural Networks (Krizhevsky, Sutskever, และ Hinton, 2017) AlexNet ถูกเสนอโดย Alex Krizhevsky ซึ่งทำให้ CNN เป็นที่นิยมในงานที่เกี่ยวกับคอมพิวเตอร์วิทัศน์ ซึ่งใช้หลักการของคอนโวลูชันแบบ top-down ซึ่งโครงสร้างของ AlexNet ที่ประกอบด้วย ชั้นคอนโวลูชัน (Convolutional Layers) ทั้งหมด 5 ชั้นที่ โดยมีการใช้ฟังก์ชันกระตุ้น ReLU เพื่อเพิ่ม non-linearity เข้าไปในโครงข่าย มีการใช้พูลลิ่งค่าสูงสุด (Max Pooling) เพื่อลดขนาดของ feature maps ลดการคำนวณและเพิ่มความเป็นเสถียรให้กับแบบจำลอง มีชั้น Fully Connected Layers มีทั้งหมด 3 ชั้น ซึ่งมีจำนวนโหนดสูงสุดถึง 4096 โหนด มีการใช้ Dropout เพื่อป้องกัน overfitting โดยการสุ่มปิดบางโหนดในชั้นต่ำกว่า 1 เพื่อลดความเชื่อมโยงที่เกินมาในชั้นนี้ และชั้นสุดท้าย Output Layer มีจำนวนโหนดเท่ากับจำนวนคลาสทั้งหมดที่ต้องการจำแนก นอกจากนั้น AlexNet ยังมีคุณลักษณะอื่น ๆ เพื่อป้องกัน Overfitting เช่น การใช้เทคนิค Data Augmentation เช่น การทำซ้ำภาพต้นฉบับโดยหมุนภาพเล็กน้อย, สลับภาพซ้าย-ขวา, และการย่อ-ขยายภาพ เป็นต้น

2.1.3.5 โครงข่ายประสาทเทียมคอนโวลูชัน VGG16

ในบทวิจัย Very Deep Convolutional Networks For Large Scale Image Recognition (Simonyan และ Zisserman, 2014) VGG16 ถูกพัฒนาโดย Visual Geometry Group ทีมวิจัยจากมหาวิทยาลัย Oxford ในปี 2014 ซึ่งโครงสร้างของแบบจำลอง VGG16

ประกอบด้วย ชั้นคอนโวลูชัน 13 ชั้น ซึ่งใช้ filter size ขนาด 3x3 มีฟังก์ชันกระตุ้นเป็น ReLu มีชั้น Pooling จำนวน 5 ชั้น เพื่อลดขนาดของ Feature maps และมีชั้นเชื่อมโยงสมบูรณ์ ทั้งหมด 3 ชั้น โดยมีโหนดสูงสุด 4096 โหนด มีการใช้ Dropout เพื่อลด Overfitting และชั้นสุดท้าย Output Layer มีจำนวนโหนดเท่ากับจำนวนคลาสทั้งหมดที่ต้องการจำแนก ซึ่งเป็นแบบจำลองที่เหมาะสำหรับงานที่ต้องการจำแนกภาพหรือวัตถุที่มีความซับซ้อนของรายละเอียดมาก

2.1.3.6 โครงข่ายประสาทเทียมคอนโวลูชัน ShuffleNet

ในบทวิจัยเรื่อง ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices (Zhang, Zhou, Lin, และ Sun, 2018) ที่ ShuffleNet ถูกนำเสนอ ซึ่งแบบจำลองนี้ถูกออกแบบมาเพื่อลดการคำนวณที่ซับซ้อนและใช้ทรัพยากรอย่างมีประสิทธิภาพ โดยใช้ 2 เทคนิค คือ Pointwise Group Convolution และ Channel Shuffle Operation

2.1.3.7 โครงข่ายประสาทเทียมคอนโวลูชัน MobileNet

ในบทวิจัย เรื่อง MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (Howard และคนอื่น ๆ, 2017) MobileNet ถูกนำเสนอ ในปี ค.ศ. 2017 โดย Google Research โดยใช้โครงสร้างที่เรียบง่าย โดยการใช้คอนโวลูชันแบบ depthwise separable เพื่อสร้างเครือข่ายประสิทธิภาพสูงที่มีน้ำหนักเบา มีจุดประสงค์ออกแบบมาเพื่อการทำงานโดยไม่ใช้ทรัพยากรมากถูกออกแบบมาสำหรับใช้งานบนมือถือและอุปกรณ์ฝังตัว

2.1.4 ทฤษฎีที่เกี่ยวข้องกับการขยายข้อมูลชุดฝึก (Image/Data Augmentation)

การขยายข้อมูลชุดฝึก เป็นการปรับเปลี่ยน หรือ เปลี่ยนแปลงชุดข้อมูลต้นฉบับให้เป็นชุดข้อมูลใหม่ที่คล้ายคลึงกับข้อมูลเดิมแตกต่างจากของเดิมไปเล็กน้อย แต่ยังคงความหมายเหมือนเดิม เพื่อเพิ่มประสิทธิภาพการฝึกฝนแบบจำลองให้สามารถทำนายข้อมูลที่ไม่เคยเห็นมาก่อนได้ โดยมีหลายวิธีและเทคนิคที่ใช้ในกระบวนการนี้ เช่น

2.1.4.1 การเลื่อนภาพขึ้นลง/ซ้ายขวา (Shift) เป็นเลื่อนภาพไปทางแนวนอนหรือแนวตั้งเพื่อสร้างภาพใหม่ที่มีตำแหน่งของวัตถุหรือวัตถุบนภาพเลื่อนไปตามทิศทางที่กำหนด

2.1.4.2 การบิดภาพ (Shear) เป็นการเอียงภาพไปในทิศทางที่กำหนดเพื่อสร้างภาพใหม่ที่มีการเอียงหรือเบี่ยงของวัตถุ

2.1.4.3 การขยายภาพ (Zoom) เป็นการขยายขนาดของภาพเพื่อสร้างภาพใหม่ที่มีข้อมูลในภาพมากขึ้นหรือน้อยลง

2.1.4.4 การพลิกภาพ (Flip) เป็นการสลับแกนของภาพ เช่น การสลับซ้าย-ขวา หรือ บน-ล่าง เพื่อสร้างภาพใหม่ที่มีลักษณะเดียวกันแต่กลับหงาย

2.1.4.5 การหมุนภาพ (Rotate) เป็นการหมุนภาพโดยระบุมุมที่ต้องการหมุน เพื่อสร้างภาพใหม่ที่มีการเอียงหรือหมุน

2.1.4.6 การเติมสีต่าง ๆ เป็นการเพิ่มหรือลดความเข้มของสีในภาพ เพื่อสร้างภาพใหม่ที่มีความสว่างหรือความเข้มของสีต่างกัน

2.1.4.7 การเติมแสงให้ภาพ เป็นการเพิ่มหรือลดการแสงในภาพ เพื่อสร้างภาพใหม่ที่มีระดับแสงต่างกัน

2.1.4.8 การทำภาพเบลอ (Blur) เป็น การทำให้ภาพเบลอ เพื่อสร้างภาพใหม่ที่มีความเบลอหรือความคมชัดต่างกัน

ภาพประกอบ 4 ตัวอย่างการทำการปรับเปลี่ยนรูปภาพ (Data Augmentation)

ที่มา: https://towardsdatascience.com/1000x-faster-data-\augmentation

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 บทความวิจัยเรื่อง Malaysia Traffic Sign Recognition with Convolutional Neural Network (Lau, Lim, และ Gopalai, 2015) เป็นบทความที่ทำการเปรียบเทียบ ระหว่าง shallow architecture neural network (RBFNN) และ deep architecture neural network (CNN) จากการเปรียบเทียบ พบว่า การทำ Convolution neural network โดยใช้การทดสอบแบบเพิ่มขึ้น ทำให้ผลลัพธ์นั้น 99% ซึ่งพบว่า deep architecture neural network (CNN) มีประสิทธิภาพที่ดีมาก

2.2.2 บทความวิจัยเรื่อง Traffic Sign Classification Using Convolutional Neural Network (Veli**č**kovi**ć**, Stojkovi**ć**, Dimi**ć**, Vasiljevi**ć**, และ Nagamalai, 2018) เป็นการนำเสนอแบบจำลองสำหรับการจำแนกป้ายจราจรที่ผู้เขียนออกแบบขึ้น โดยใช้ชุดข้อมูล ที่มีจำนวนประเภท 43 ประเภท ขนาด 30 x 30 พิกเซล ทั้งหมด 1,200 รูป ที่เป็นภาพสี RGB จากนั้นทำการแยกชุดฝึกฝน และ ชุดทดสอบ เป็น 80% กับ 20% ตามลำดับ ซึ่งโครงสร้างของโครงข่ายที่ผู้วิจัยออกแบบนั้น คือ กำหนดขนาดข้อมูลรับเข้าขนาด 30 x 30 x 3 มีคอนโวลูชัน 2 ชั้น โดยกำหนดชั้นแรกให้มีจำนวนฟิลเตอร์เท่ากับ 32 มี kernel size 3x3 การเลื่อน (Stride) เป็น 1 และในชั้นที่ 2 กำหนดจำนวน filter เท่ากับ 64 มี kernel size ขนาด 5 x 5 มีการทำ max pooling ขนาด 2 x 2 ระหว่างชั้น จากนั้นผ่านเข้าชั้น Fully connected layer อีก 3 ชั้น โดยไม่ทำ padding และใช้ Activation Function เป็น ReLu ผลปรากฏว่าค่าความแม่นยำ (Accuracy) เฉลี่ยอยู่ที่ 80%

2.2.3 บทความวิจัยเรื่อง CNN based Traffic Sign Classification using Adam Optimizer (Mehta, Paunwala, และ Vaidya, 2019) เป็นการเปรียบเทียบ Optimizer ในการปรับ Parameter ของโครงข่ายประสาทแบบคอนโวลูชัน โดยใช้แบบจำลองเดียวที่ผู้เขียนออกแบบ ใช้ชุดข้อมูล จาก Belgium traffic sign dataset (BTSD) ที่มีจำนวนรูปภาพทั้งหมด 7000 กว่ารูปภาพในการทำการทดลอง แบบจำลองที่ผู้วิจัยออกแบบ เป็นคอนวูโลชั่น 3 ชั้น ถูกแทรกโดยการทำ Max pooling ระหว่างชั้น จากนั้นมีการปรับ parameter เพื่อเปรียบเทียบ มีการเติม dropout ในอัตราที่ต่างกัน เพื่อทำการเปรียบเทียบกันระหว่างการเติม dropout และไม่เติม drop out มีการเปรียบเทียบระหว่าง Optimizer คือ SGD และ Adam และมีการปรับเปลี่ยน Activation Function เปรียบเทียบระหว่าง Sigmoid กับ Softmax ผลลัพธ์จากการวิจัยพบว่า Adam Optimizer มีความรวดเร็วกว่า เมื่อเทียบกับ Optimizers อื่น ๆ เช่น SGD เป็นต้น

2.2.4 บทความวิจัยเรื่องTraffic Sign Classification Using Deep Neural Network (Vincent, V. K, และ Mathew, 2020) ได้นำเสนอวิธีการทำ image Augmentation และออกแบบแบบจำลองเพื่อการคัดแยกป้ายจราจร โดยใช้ชุดข้อมูล GTSRB ที่มีจำนวนรูปภาพทั้งหมด 34,799 รูปภาพ แยกเป็นหมวดหมู่ทั้งหมด 43 ประเภท แต่เนื่องจากข้อมูล GTSRB มีความไม่เท่าเทียมกัน (Unbalanced) ของแต่ละประเภท ทำให้ผู้วิจัยเลือกทำ Image Augmentation เพื่อสร้างภาพให้มีจำนวนเพิ่มขึ้นแต่ไม่เปลี่ยนแปลงภาพเดิม จากนั้นทำการออกแบบแบบจำลองที่ชื่อว่า TS CNN ที่ประกอบด้วยชั้นคอนโวลูชัน 4 ชั้น , ชั้นพลูลิ่ง 2 ชั้นและชั้นการเชื่อมต่อแบบสมบูรณ์อีก 4 ชั้น กำหนดจำนวนการเทรนแบบจำลอง (Epoch) 50 รอบ, optimizer เป็น Adam Algorithm ใช้ Activation Function เป็น ReLu กับ Softmax และใช้ เทคนิคการ Drop out ในอัตราอยู่ที่ 20%

และ 50% ผลปรากฏว่า ค่าความถูกต้องอยู่ที่ 98.44% จากนั้นได้ทำการเปรียบเทียบเพิ่มเติมกับแบบจำลองที่เป็น baseline ซึ่ง VGG16 มีค่าอยู่ที่ 74.5% , CVOG + other features มีค่าอยู่ที่ 94.73% ,Capsule Network มีค่าอยู่ที่ 97.6% GoogleNet based CNN มีค่าอยู่ที่ 98% และ Hough transform with CNN มีค่าอยู่ที่ 98.2% ซึ่งแบบจำลองที่ผู้วิจัยออกแบบนั้นมีค่าความถูกต้องมากกว่าแบบจำลอง baseline อื่น ๆ ที่นำมาเปรียบเทียบ

2.2.5 บทความวิจัยเรื่อง A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network (Zaibi, Ladgham, และ Sakly, 2021) เป็นการเปรียบเทียบผลของแบบจำลองที่มีการปรับปรุงมาจากโครงข่ายประสาทเทียมแบบคอนโวลูชัน ชนิด LeNet-5 เพื่อให้ได้แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชันที่มีลักษณะ Lightweight Model และนำใช้งานได้ง่ายสำหรับแอปพลิเคชันฝังตัว (embedded application) โดยมีการปรับปรุงจากแบบจำลอง LeNet-5 มีการปรับเปลี่ยนพารามิเตอร์ เช่น จำนวนฟิลเตอร์ในชั้นคอนโวลูชัน ปรับฟังก์ชันกระตุ้น มีการเติม dropout เพื่อไม่ให้แบบจำลองเกิดการ overfitting เป็นต้น จากนั้นเปรียบเทียบระหว่าง Classic LeNet-5 กับแบบจำลองที่ผู้เขียนมีการปรับเปลี่ยน โดยใช้ชุดข้อมูลที่แตกต่างกัน ได้แก่ German Traffic Sign Recognition Benchmark (GTSRB) database และ Belgian Traffic Sign Data Set (BTSD) ผลลัพธ์ของการทดลอง มีจำนวนพารามิเตอร์ที่ลดลงเท่ากับ 0.38 ล้าน และค่าความแม่นยำคือ 99.84% สำหรับ GTSRB และ 98.37% สำหรับ BTSD

2.2.6 บทความวิจัยเรื่อง Traffic Sign Detection and Recognition Using Deep Learning Approach (Rahman และ Maruf, 2023) เป็นการเปรียบเทียบ 3 แบบจำลอง ได้แก่ CNN, InceptionV3, และ AlexNet โดยใช้ชุดข้อมูลที่ผู้วิจัยรวบรวมเองซึ่งเป็นป้ายจราจรจากท้องถนนในเขตเมืองดากา ของประเทศบังคลาเทศ จำนวนกว่า 7000 รูปภาพและมีจำนวนประเภท 70 ประเภท ซึ่งภาพทั้งหมดจะถูกเปลี่ยนขนาดให้เป็นขนาด 128 × 128 พิกเซล มีการปรับเปลี่ยนรูปภาพ เช่น การซูม การเอียง การหมุน และปัจจัยอื่น ๆ เพิ่มเติม เพื่อไปใช้ในการทำการปรับเปลี่ยนรูปภาพ ด้วยโมเดล CNN, InceptionV3, และ AlexNet พบว่า มีความแม่นยำสูงสุดที่ประมาณ 99.71%

บทที่ 3
วิธีดำเนินการวิจัย

ในการวิจัยครั้งนี้ ผู้วิจัยได้ดำเนินการตามขั้นตอนดังนี้

1. การเตรียมข้อมูลสำหรับการวิจัย (Data Acquisition & Data Filtering)
2. การเตรียมข้อมูล (Data Preprocessing)
3. การสร้างแบบจำลอง (Modeling)
4. การทดลอง (Classification)
5. การประเมินผล (Evaluation)

ภาพประกอบ 5 แสดงกระบวนการดำเนินงานวิจัย

จากภาพประกอบ 5 แสดงให้เห็นถึงกระบวนการดำเนินงานวิจัยทั้งหมด โดยเริ่มจากการเตรียมข้อมูลสำหรับงานวิจัย รวมถึงการเลือกชุดข้อมูลและการคัดกรองข้อมูล (Data Acquisition & Data Filtering) กระบวนการเตรียมข้อมูลก่อนเข้าแบบจำลอง (Preprocessing) การสร้างแบบจำลอง (Modeling) การปรับพารามิเตอร์ และการประเมินผล โดยมีจุดประสงค์เพื่อศึกษาและนำวิธีการมาปรับใช้กับชุดข้อมูล ตลอดจนการทดสอบและประเมินประสิทธิภาพ มีขั้นตอนดังนี้

3.1 การเตรียมข้อมูลสำหรับการวิจัย (Data Acquisition & Data Filtering)

3.1.1 ชุดข้อมูลที่ใช้ในงานวิจัย

ชุดข้อมูลสำหรับงานวิจัย เป็นชุดข้อมูลภาพถ่ายป้ายจราจร GTSRB - German Traffic Sign Recognition Benchmark ที่มีข้อมูลภาพถ่ายป้ายจราจรจากประเทศเยอรมนี เพื่อทดสอบการจำแนกแบบหลายคลาส โดยผู้วิจัยได้คัดเลือกรูปภาพเฉพาะป้ายจราจรจำกัดความเร็ว ซึ่งมีจำนวนคลาส 8 คลาส และมีจำนวนภาพ 12,780 ภาพ เพื่อทำการวิจัยในครั้งนี้ ดังตาราง 1

ตาราง 1 แสดงจำนวนภาพในชุดข้อมูลแยกตามประเภทของป้ายจำกัดความเร็ว

Class	ประเภทป้ายจราจร	จำนวนภาพ	ตัวอย่างรูปภาพ
0	'Speed limit (20km/h)',	210	
1	'Speed limit (30km/h)',	2,220	
2	'Speed limit (50km/h)',	2,250	
3	'Speed limit (60km/h)',	1,410	
4	'Speed limit (70km/h)',	1,980	
5	'Speed limit (80km/h)',	1,860	
6	'Speed limit (100km/h)',	1,440	
7	'Speed limit (120km/h)',	1,410	
	รวมทั้งสิ้น	12,780 ภาพ	

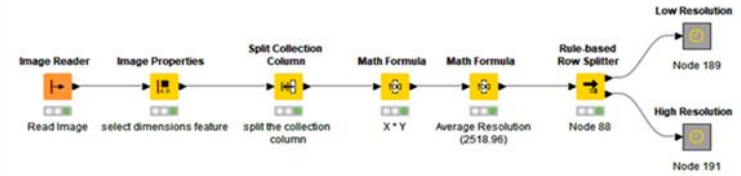
3.1.2 การเลือกชุดข้อมูล

เมื่อทำการคัดเลือกชุดข้อมูล ตามข้อ 3.1.1 แล้ว จากนั้นทำการวิเคราะห์และแบ่งแยกภาพตามคุณสมบัติมิติของภาพ (Dimensions Feature) โดยใช้โปรแกรม KNIME Analytics Platform เพื่อวางกระบวนการเชื่อมต่อกัน (Workflow) ดังภาพประกอบ 6 เริ่มต้นด้วยการอ่านข้อมูลภาพจากชุดข้อมูล โดยใช้โหนด Image Reader จากนั้นใช้โหนด Image Properties เพื่อเลือกคุณสมบัติมิติของภาพ เช่น ความกว้าง (Width), ความยาว (Height), และความลึก (Depth) หลักจากที่ได้คุณสมบัติมิติของภาพแล้ว ใช้ Math Formula เพื่อคำนวณความละเอียดของภาพ โดยใช้สูตร ความกว้าง x ความยาว เพื่อหาความละเอียดของภาพแต่ละภาพ นอกจากนี้ ยังใช้ Math Formula อีกครั้งเพื่อหาค่าเฉลี่ยของความละเอียดของรูปภาพทั้งหมด เพื่อให้ทราบถึงความละเอียดเฉลี่ยของชุดข้อมูลภาพทั้งหมด โดยค่าเฉลี่ยของความละเอียดอยู่ที่ 2,518.96 เมื่อได้ค่าเฉลี่ยความละเอียดของภาพทั้งหมดแล้ว ใช้ Rule-based Row Splitter เพื่อแยกภาพตามเงื่อนไขที่กำหนดไว้ โดยการแบ่งเป็น 3 กลุ่ม ดังนี้

1. กลุ่มภาพความละเอียดสูง (High Resolution) จะเป็นภาพที่ทำการคำนวณความละเอียดของภาพแล้วสูงกว่าหรือเท่ากับค่าเฉลี่ย

2. กลุ่มภาพความละเอียดต่ำ (Low Resolution) จะเป็นภาพที่ทำการคำนวณความละเอียดของภาพแล้วต่ำกว่าค่าเฉลี่ย

3. กลุ่มภาพความละเอียดผสม คือ กลุ่มภาพเดิมที่ได้จากชุดข้อมูล ซึ่งเป็นข้อมูลของกลุ่มภาพความละเอียดต่ำผสมกับกลุ่มภาพความละเอียดสูง

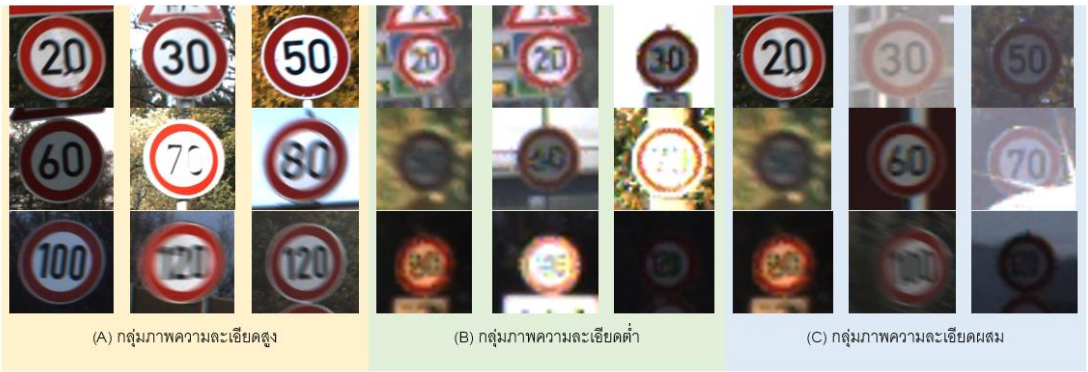


ภาพประกอบ 6 แสดง Workflow จากโปรแกรม KNIME Analytics Platform ที่ใช้จัดกลุ่มของภาพตามความละเอียด

โดยมีจำนวนภาพแต่ละกลุ่มดังตาราง 2 โดยมีจุดประสงค์ของการทดลอง คือ ต้องการทราบว่า กลุ่มภาพชนิดใดที่จะทำให้ประสิทธิภาพของแบบจำลองที่สร้างขึ้นมีประสิทธิภาพในการจำแนกมากที่สุด

ตาราง 2 แสดงจำนวนภาพในแต่ละประเภท แยกตามกลุ่มความละเอียดของภาพ

Class	ประเภทป้ายจราจร	กลุ่มภาพความละเอียดสูง	กลุ่มภาพความละเอียดต่ำ	กลุ่มภาพความละเอียดผสม
0	'Speed limit (20km/h)',	92	118	210
1	'Speed limit (30km/h)',	970	1,250	2,220
2	'Speed limit (50km/h)',	640	1,610	2,250
3	'Speed limit (60km/h)',	364	1,046	1,410
4	'Speed limit (70km/h)',	551	1,429	1,980
5	'Speed limit (80km/h)',	385	1,475	1,860
6	'Speed limit (100km/h)',	348	1,092	1,440
7	'Speed limit (120km/h)',	327	1,083	1,410
	รวมทั้งสิ้น	3,677	9,103	12,780



ภาพประกอบ 7 แสดงตัวอย่างรูปภาพของภาพป้ายจราจรจำกัดความเร็วทั้ง 3 ชุด ตามความละเอียดของภาพที่ใช้ในการทดลอง

3.1.3 การแบ่งชุดข้อมูล

จากนั้นในภาพประกอบ 8 มีการแบ่งข้อมูลทั้งหมด 3 ส่วน ได้แก่ ชุดข้อมูลสำหรับฝึกฝน (Training Set) ชุดข้อมูลที่ใช้ตรวจสอบ (Validation Set) และชุดข้อมูลสำหรับทดสอบ (Test Set) โดยวิธีการ train_test_split จากไลบรารี sklearn ออกเป็น 80:10:10 ซึ่งสามารถอธิบายได้ดังนี้

ส่วนแรกเป็น ชุดข้อมูลสำหรับฝึกฝน (Training Data) คิดเป็น 80% ของข้อมูลทั้งหมด และส่วนที่สองเป็น ข้อมูลชั่วคราว (Temporary Data) คิดเป็น 20% ของข้อมูลทั้งหมด จากนั้น แบ่งข้อมูลชั่วคราวที่ได้จากส่วนแรก เป็นข้อมูลตรวจสอบ (Validation Data) คิดเป็น 50% ของข้อมูลชั่วคราว และข้อมูลทดสอบ (Testing Data) คิดเป็น 50% ของข้อมูลชั่วคราว ดังแสดงในตาราง 3 โดยอัตราส่วนของ ชุดข้อมูลการตรวจสอบและทดสอบไม่เท่ากัน เนื่องจากวิธีการแบ่งข้อมูล ลักษณะของการสุ่ม ซึ่งมีการกำหนด Random_state เพื่อให้การสุ่มคงที่เมื่อรันข้อมูลซ้ำหลายรอบ โดยในภาพรวมมีความสมดุล แม้จะแตกต่างกันเล็กน้อยในแต่ละคลาส

```
# Split data into training and temporary data (80:20)
X_train, X_temp, y_train, y_temp = train_test_split(images, labels, test_size=0.2, random_state=42)

# Split temporary data into validation and testing sets (50:50)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
```

ภาพประกอบ 8 วิธีการแบ่งข้อมูลโดยใช้ Python

ตาราง 3 การแบ่งชุดข้อมูลจำแนกตามกลุ่มภาพและตามคลาส

Class	กลุ่มภาพความละเอียดสูง			กลุ่มภาพความละเอียดต่ำ			กลุ่มภาพความละเอียดผสม		
	Train Data	Validate Data	Test Data	Train Data	Validate Data	Test Data	Train Data	Validate Data	Test Data
	80%	10%	10%	80%	10%	10%	80%	10%	10%
0	69	14	9	83	16	19	163	19	28
1	760	98	112	1005	138	107	1770	232	218
2	517	66	57	1291	153	166	1783	235	232
3	295	34	35	834	102	110	1121	151	138
4	450	50	51	1130	159	140	1594	182	204
5	312	32	41	1184	134	157	1511	171	178
6	279	39	30	895	95	102	1152	144	144
7	259	35	33	860	113	110	1130	144	136
รวม	2941	368	368	7282	910	911	10224	1278	1278

3.2 **การเตรียมข้อมูล** (Data Preprocessing)

โดยในขั้นตอนนี้จะใช้โปรแกรมไพทอน ไลบรารี TensorFlow เพื่อโหลดข้อมูลภาพจากโฟเดอร์ที่แบ่งชุดข้อมูลไว้ โดย Lenet-5 มี input shape = 32, 32 , 3 ซึ่งไม่ได้ปรับรูปภาพให้เป็นภาพขาวดำตามแบบจำลองเดิม และแบบจำลองอื่นๆ มี input shape = 224, 224, 3 จากนั้นปรับค่าพิกเซลให้อยู่ในช่วง 0-1 หรือ -1 ถึง 1 โดยใช้การหารค่าของพิกเซลด้วย 255 ต่อมา ทำการแปลงข้อมูลเป็นรูปแบบที่เหมาะสม ซึ่งในขั้นตอนนี้จะทำการทดสอบประสิทธิภาพกับข้อมูลภาพ 2 ชุด ได้แก่ ข้อมูลรูปภาพที่ไม่ได้ขยายข้อมูลชุดฝึก (without Data Augmentation) และข้อมูลรูปภาพที่ขยายข้อมูลชุดฝึก (with Data Augmentation) ดังตาราง 5 โดยกำหนดให้มีการขยายข้อมูลรูปภาพ เพื่อเพิ่มข้อมูลรูปภาพให้ชุดฝึกฝน 20% โดยใช้ Image DataGenerator ใน TensorFlow/Keras เพื่อสร้างข้อมูลรูปภาพเพิ่มเติมให้ชุดฝึก ดังตาราง 4

ตาราง 4 แสดงวิธีการที่กำหนดเพื่อขยายข้อมูลชุดฝึก (Data Augmentation)

วิธีการ	**คำสั่งใน** ImageDataGenerator
การหมุนภาพแบบสุ่มในหน่วยองศา	rotation_range=20
การเคลื่อนที่แนวนอน	width_shift_range=0.1
การเคลื่อนที่แนวตั้ง	height_shift_range=0.1
การเอียง	shear_range=0.2
การซูมภาพแบบสุ่ม	zoom_range=0.2
การพลิกภาพแนวนอน	horizontal_flip=False
การพลิกภาพแนวตั้ง	vertical_flip=False
การเติมพิกเซลที่สร้างขึ้นใหม่	fill_mode='nearest'

ตาราง 5 แสดงจำนวนภาพที่ขยายข้อมูลชุดฝึก (with Data Augmentation) จำแนกตามกลุ่มภาพและตามคลาส

Class	**กลุ่มภาพ ความละเอียดสูง**			**กลุ่มภาพ ความละเอียดต่ำ**			**กลุ่มภาพ ความละเอียดผสม**		
	Train Data	Validate Data	Test Data	Train Data	Validate Data	Test Data	Train Data	Validate Data	Test Data
0	79	14	9	97	16	19	193	19	28
1	916	98	112	1216	138	107	2117	232	218
2	627	66	57	1562	153	166	2128	235	232
3	346	34	35	994	102	110	1359	151	138
4	539	50	51	1363	159	140	1929	182	204
5	382	32	41	1422	134	157	1801	171	178
6	328	39	30	1051	95	102	1409	144	144
7	312	35	33	1033	113	110	1332	144	136
รวม	3529	368	368	8738	910	911	12268	1278	1278

3.3 การสร้างแบบจำลอง (Modeling)

ในงานวิจัยนี้ได้นำเสนอการใช้เทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) ซึ่งใช้แบบจำลอง LeNet-5 ,AlexNet, VGG16, ShuffleNet และ MobileNet โดยใช้ภาษาไพทอน และไลบรารี Skit-learn, Numpy และ Tensorflow เพื่อสร้างแบบจำลองในงานวิจัยครั้งนี้ ทั้ง 5 แบบจำลอง ซึ่งเป็นโมเดลที่ได้รับความนิยมในช่วงเริ่มต้นของการศึกษางานวิจัยด้าน Deep learning และยังเป็นแนวทางสำคัญในการพัฒนาแบบจำลองอื่น ๆ ในภายหลังด้วย ดังนี้

3.3.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน LeNet-5

ลักษณะโครงสร้างของ Lenet-5 เข้าใจง่ายและไม่ซับซ้อน ดังตาราง 6 โดยประกอบด้วย ชั้นคอนโวลูชัน จำนวน 3 ชั้น ชั้นพลูลิ่ง จำนวน 2 ชั้น โดยใช้ค่าเฉลี่ยในการทำพูลลิ่ง (Average pooling) ชั้นเชื่อมโยงสมบูรณ์จำนวน 1 ชั้น และชั้นข้อมูลออก ที่ในแบบจำลองเดิมมี 10 นิวรอน แต่ในการทำแบบจำลองเพื่อจำแนกป้ายจำกัดความเร็ว ที่มีจำนวนประเภท 8 ประเภท จึงเปลี่ยนเป็น 8 ตามประเภทที่จำแนก

ตาราง 6 โครงสร้างแบบจำลองของ Lenet-5

โครงสร้าง	พารามิเตอร์
ชั้นรับเข้า (Input Layer)	32 x 32
ชั้นคอนโวลูชันแรก (First convolution layer)	filters = 6 , kernel_size = 5 Activation function = Tanh/ReLu
ชั้นพูลลิ่ง (Pooling Layer)	Average pooling Size = 2x2 Stride = 2
ชั้นคอนโวลูชันที่ 2 (Second convolution layer)	filters = 16 kernel_size = 5 Activation function = Tanh/ReLu
ชั้นพูลลิ่ง (Pooling Layer)	Average pooling Size = 2x2 Stride = 2
ชั้นคอนโวลูชันที่ 3 (Third convolution layer)	filters = 120 kernel_size = 5 Activation function = Tanh/ReLu
ชั้นเชื่อมโยงสมบูรณ์ (Fully Connected Layer)	Neurons = 84 unit Activation function = Softmax
ชั้นข้อมูลออก (Output)	Neurons = 8 unit Activation function = Softmax

3.3.2 โครงข่ายประสาทเทียมแบบคอนโวลูชัน AlexNet

ลักษณะโครงสร้างของ AlexNet ยังคงเข้าใจง่ายแต่มีความซับซ้อนมากกว่า Lenet-5 ดังตาราง 7 โดยประกอบด้วย ชั้นคอนโวลูชัน จำนวน 5 ชั้น ชั้นพลูลิ่ง จำนวน 3 ชั้น โดยใช้ค่ามากในการทำพูลลิ่ง (Max pooling) และชั้นเชื่อมโยงสมบูรณ์จำนวน 2 ชั้น และชั้นข้อมูลออก ที่ในแบบจำลองเดิม มี1000 นิวรอน แต่ในการทำแบบจำลองเพื่อจำแนกป้ายจำกัดความเร็วมีจำนวนประเภท 8 ประเภท จึงเปลี่ยนเป็น 8 ตามประเภทที่จำแนก นอกจากนั้นยังมีการทำ Drop out เพื่อลดการ overfitting ให้กับแบบจำลอง และมีการใช้ฟังก์ชั่นกระตุ้น (Activation Function) เป็น Relu อีกด้วย

ตาราง 7 โครงสร้างแบบจำลองของ AlexNet

โครงสร้าง	พารามิเตอร์
ชั้นรับเข้า (Input Layer)	224 x 224
ชั้นคอนโวลูชันแรก (First convolution layer)	filters = 96 filters size = 11x11 Activation function = ReLu Stride = 4
ชั้นพูลลิ่ง (Pooling Layer)	Max pooling filters size = 3x3 Stride = 2
ชั้นคอนโวลูชันที่ 2 (Second convolution layer)	filters = 256 filters size = 5x5 Activation function = ReLu Stride = 1 Padding = 2
ชั้นพูลลิ่ง (Pooling Layer)	Max pooling filters size = 3x3 Stride = 2
ชั้นคอนโวลูชันที่ 3 (Third convolution layer)	filters = 384 filters size = 3x3 Activation function = ReLu Stride = 1, Padding = 1
ชั้นคอนโวลูชันที่ 4 (fourth convolution layer)	filters = 384 filters size = 3x3 Activation function = ReLu Stride = 1 Padding = 1
ชั้นคอนโวลูชันที่ 5 (fifth convolution layer)	filters = 256 filters size = 3x3 Activation function = ReLu Stride = 1 Padding = 1
ชั้นพูลลิ่ง (Pooling Layer)	Max pooling filters size = 3x3 Stride = 2

ตาราง 7 (ต่อ)

โครงสร้าง	พารามิเตอร์
Drop out	Rate = 0.5
ชั้นเชื่อมโยงสมบูรณ์แรก (First Fully Connected Layer)	Neurons = 4,096 unit Activation function = ReLu
Drop out	Rate = 0.5
ชั้นเชื่อมโยงสมบูรณ์ที่ 2 (Second Fully Connected Layer)	Neurons = 4,096 unit Activation function = ReLu
ชั้นข้อมูลออก (Output)	Neurons = 8 unit Activation function = Softmax

3.3.3 **โครงข่ายประสาทเทียมแบบคอนโวลูชัน** VGG16

ลักษณะโครงสร้างของ VGG16 ประกอบด้วย ชั้นคอนโวลูชัน 13 ชั้น ดังตาราง 8 ซึ่งใช้ filter size ขนาด 3x3 มีฟังก์ชันกระตุ้นเป็น ReLu มีชั้น Pooling จำนวน 5 ชั้น เพื่อลดขนาดของ Feature maps และมีชั้นเชื่อมโยงสมบูรณ์ ทั้งหมด 3 ชั้น โดยมีโหนดสูงสุด 4096 โหนด และชั้นสุดท้าย Output Layer มีจำนวนโหนดเท่ากับจำนวนคลาสทั้งหมดที่ต้องการจำแนก ซึ่งเป็นแบบจำลองที่เหมาะสำหรับงานที่ต้องการการจำแนกภาพที่ซับซ้อน การจำแนกวัตถุหรือภาพที่มีรายละเอียดสูง

ตาราง 8 โครงสร้างแบบจำลองของ VGG16

โครงสร้าง	พารามิเตอร์
ชั้นรับเข้า (Input Layer)	224 x 224
บล็อกคอนโวลูชันแรก (First Convolution Block)	2x (Filters: 64, Kernel Size: 3x3, Activation Function: ReLU)
ชั้นพูลลิ่ง (Pooling Layer)	Max Pooling, Size: 2x2
บล็อกคอนโวลูชันที่ 2 (Second Convolution Block)	2x (Filters: 128, Kernel Size: 3x3, Activation Function: ReLU)
บล็อกคอนโวลูชันที่ 3 (Third Convolution Block)	3x (Filters: 256, Kernel Size: 3x3, Activation Function: ReLU)
ชั้นพูลลิ่ง (Pooling Layer)	Type: Max Pooling, Size: 2x2

ตาราง 8 (ต่อ)

โครงสร้าง	พารามิเตอร์
บล็อกคอนโวลูชันที่ 4 (Fourth Convolution Block)	3x (Filters: 512, Kernel Size: 3x3, Activation Function: ReLU)
ชั้นพูลลิ่ง (Pooling Layer)	Type: Max Pooling, Size: 2x2
บล็อกคอนโวลูชันที่ 5 (Fifth Convolution Block)	3x (Filters: 512, Kernel Size: 3x3, Activation Function: ReLU)
ชั้นพูลลิ่ง (Pooling Layer)	Type: Max Pooling, Size: 2x2
ชั้นเชื่อมโยงสมบูรณ์ (Fully Connected Layer)	Neurons: 4096, Activation Function: ReLU
ชั้นเชื่อมโยงสมบูรณ์ (Fully Connected Layer)	Neurons: 4096, Activation Function: ReLU
ชั้นข้อมูลออก (Output)	Neurons = 8 unit Activation Function: Softmax

3.3.4 โครงข่ายประสาทเทียมแบบคอนโวลูชัน ShuffleNet

ลักษณะโครงสร้างของ ShuffleNet ดังตาราง 9 ซึ่งถูกออกแบบมาเพื่อลดการคำนวณที่ซับซ้อนและใช้ทรัพยากรอย่างมีประสิทธิภาพ โดยการใช้วิธีการดำเนินการ 2 วิธี คือ Pointwise Group Convolution และ Channel Shuffle Operation

ตาราง 9 โครงสร้างแบบจำลองของ ShuffleNet

โครงสร้าง	พารามิเตอร์
Input Layer	224 x 224
Conv2D	Filters: 24, Kernel Size: 3x3, Strides: 2, Padding: 'same', Use Bias: False
BatchNormalization	-
ReLU	-
MaxPooling2D	Pool Size: 3x3, Strides: 2, Padding: 'same'
ShuffleNetV2 Block 1 (x3)	Out Channels: 116, Strides: 2 (ครั้งแรก) (ครั้งที่ 2-3), Groups: 2

ตาราง 9 (ต่อ)

โครงสร้าง	พารามิเตอร์
ShuffleNetV2 Block 2 (x7)	Out Channels: 232, Strides: 2 (ครั้งแรก), 1 (ครั้งที่ 2-7), Groups: 2
ShuffleNetV2 Block 3 (x3)	Out Channels: 464, Strides: 2 (ครั้งแรก), 1 (ครั้งที่ 2-3), Groups: 2
Conv2D	Filters: 1024, Kernel Size: 1x1, Strides: 1, Padding: 'same', Use Bias: False
BatchNormalization	-
ReLU	-

3.3.5 โครงข่ายประสาทเทียมแบบคอนโวลูชัน MobileNet

ลักษณะโครงสร้างของ MobileNet ดังตาราง 10 เรียบง่าย โดยการใช้คอนโวลูชันแบบ depthwise separable เพื่อสร้างเครือข่ายประสิทธิภาพสูงที่มีน้ำหนักเบา มีจุดประสงค์ออกแบบมาเพื่อการทำงานโดยไม่ใช้ทรัพยากรมากถูกออกแบบมาสำหรับใช้งานบนมือถือและอุปกรณ์ฝังตัว

ตาราง 10 โครงสร้างแบบจำลองโครงข่ายประสาทเทียมคอนโวลูชัน MobileNet

โครงสร้าง	พารามิเตอร์
Input Layer	224x224
Conv2D	Filters: 32, Kernel Size: 3x3, Strides: 2, Padding: 'same', Activation: ReLU
Depthwise Separable Conv Block 1	Depthwise Filters: 32, Pointwise Filters: 64, Strides: 1, Activation: ReLU
Depthwise Separable Conv Block 2	Depthwise Filters: 64, Pointwise Filters: 128, Strides: 2, Activation: ReLU

ตาราง 10 (ต่อ)

โครงสร้าง	พารามิเตอร์
Depthwise Separable Conv Block 3	Depthwise Filters: 128, Pointwise Filters: 128, Strides: 1, Activation: ReLU
Depthwise Separable Conv Block 4	Depthwise Filters: 128, Pointwise Filters: 256, Strides: 2, Activation: ReLU
Depthwise Separable Conv Block 5	Depthwise Filters: 256, Pointwise Filters: 256, Strides: 1, Activation: ReLU
Depthwise Separable Conv Block 6	Depthwise Filters: 256, Pointwise Filters: 512, Strides: 2, Activation: ReLU
Depthwise Separable Conv Block 7-12	Depthwise Filters: 512, Pointwise Filters: 512, Strides: 1, Activation: ReLU
Depthwise Separable Conv Block 13	Depthwise Filters: 512, Pointwise Filters: 1024, Strides: 2, Activation: ReLU
Depthwise Separable Conv Block 14	Depthwise Filters: 1024, Pointwise Filters: 1024, Strides: 1, Activation: ReLU
Global Average Pooling	-
Dense	Neurons: 8, Activation Function: 'softmax'

3.4 การทดลอง

ผู้วิจัยได้เปรียบเทียบแบบจำลองทั้งสอง โดยแบ่งเป็น 2 แบบ ดังนี้

3.4.1 เปรียบเทียบประสิทธิภาพของโครงข่ายประสาทเทียมชนิดคอนโวลูชัน โดยใช้แบบจำลองทั้ง 5 ชนิดกับกลุ่มรูปภาพทั้ง 3 กลุ่มที่ไม่ได้ขยายข้อมูลชุดฝึก (without Data Augmentation)

3.4.2 เปรียบเทียบประสิทธิภาพของโครงข่ายประสาทเทียมแบบคอนโวลูชัน โดยใช้แบบจำลองทั้ง 5 ชนิดกับกลุ่มภาพทั้ง 3 กลุ่มที่ขยายข้อมูลชุดฝึก (with Data Augmentation)

โดยแบบจำลองทั้งสอง มีการทดลองกับฟังก์ชันกระตุ้นของแบบจำลอง Lenet-5 จำนวน 2 ฟังก์ชันคือ Hyperbolic Tangent Activation Function (Tanh) และ rectified linear unit (ReLU) และมีการปรับรอบการฝึกฝน เป็น 10 และ 20 รอบ และได้กำหนด Batch Size เป็น 32

3.5 การประเมินผล (Evaluation)

ในการวัดประสิทธิภาพของแบบจำลองที่สร้างขึ้นด้วย โครงข่ายประสาทเทียมแบบคอนโวลูชัน ด้วยตัวแปรที่แตกต่างกันนั้น เพื่อดูประสิทธิภาพในการจำแนก รวมถึงเวลาในการฝึกฝนและทดสอบแบบจำลอง ผู้วิจัยได้กำหนดหลักเกณฑ์ในการประเมินประสิทธิภาพ ดังนี้

3.5.1 ค่าความถูกต้อง (Accuracy) เป็นการประเมินผลแบบจำลองในเรื่องของความถูกต้องในการทำนายคลาสทั้งหมด เพื่อบอกว่าแบบจำลองทำนายได้ถูกมากหรือน้อยเมื่อเทียบกับข้อมูลทั้งหมด ดังสมการ 1

$$Accuracy = \frac{(TP + TN}{(TP + TN + FP + FN)} \quad (1)$$

3.5.2 ค่าความแม่นยำ (Precision) เป็นการประเมินผลแบบจำลองในเรื่องความแม่นยำของการทำนายในคลาสบวก (Positive Class) ซึ่งเป็นจำนวนจริงที่ทำนายถูกต้องเมื่อเทียบกับการทำนายทั้งหมดว่าเป็นคลาสนั้น ดังสมการ 2

$$Precision = \frac{(TP)}{(TP + FP)} \quad (2)$$

3.5.3 ค่า Recall เป็นการวัดจำนวนของจริงที่ทำนายถูก เมื่อเทียบกับจำนวนทั้งหมดของคลาสจริงดังสมการ 3

$$Recall = \frac{(TP)}{(TP + FN)} \quad (3)$$

3.5.4 ค่า F-Measure ค่าเฉลี่ยคู่ของ Precision และ Recall เพื่อใช้วัดประสิทธิภาพของการทำนายทั้งสองตัวพร้อมกัน ดังสมการ 4

$$F_1 = 2\,x\ \frac{precision\ x\ recall}{precision\ +recall} \qquad 4)$$

โดยกำหนดให้

TP (True Positive) เป็นจำนวนตัวอย่างที่ถูกต้องที่แบบจำลองสามารถจำแนกได้ว่าเป็นคลาสที่เราสนใจ

FP (False Positive) เป็นจำนวนตัวอย่างที่แบบจำลองจำแนกได้ว่าเป็นคลาสที่เราสนใจ แต่ในความจริงแล้วไม่ใช่

TN (True Negative) เป็นจำนวนตัวอย่างที่แบบจำลองจำแนกได้ว่าไม่ใช่คลาสที่เราสนใจ และในความจริงก็ไม่ใช่

FN (False Negative) เป็นจำนวนตัวอย่างที่แบบจำลองจำแนกได้ว่าไม่ใช่คลาสที่เราสนใจ แต่ในความจริงแล้วเป็นคลาสที่เราสนใจ

บทที่ 4
ผลการดำเนินงานวิจัย

จากการศึกษาโครงสร้างของแบบจำลอง 5 ชนิด ได้แก่ Lenet-5, Alexnet, VGG16, ShuffleNet, MobileNet ซึ่งมีจุดประสงค์ในการเปรียบเทียบแบบจำลองดังกล่าวกับกลุ่มภาพที่มีความละเอียดแตกต่างกัน และเปรียบเทียบตัวแปรต่าง ๆ ที่ส่งผลกับการจำแนกป้ายจราจรประเภทป้ายจำกัดความเร็ว โดยจะทำการแบ่งผลการทดลอง ดังนี้

1. ผลลัพธ์การเปรียบเทียบประสิทธิภาพของแบบจำลองกับกลุ่มภาพที่ไม่ได้ขยายข้อมูลชุดฝึก (without Data Augmentation)

1.1 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม

1.2 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ

1.3 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดสูง

2. ผลลัพธ์การเปรียบเทียบประสิทธิภาพของแบบจำลองกับกลุ่มภาพที่ขยายข้อมูลชุดฝึก (Data Augmentation)

2.1 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม

2.2 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ

2.3 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดสูง

4.1 ผลลัพธ์การเปรียบเทียบประสิทธิภาพของแบบจำลองกับกลุ่มภาพที่ไม่ได้ขยายข้อมูลชุดฝึก (without Data Augmentation)

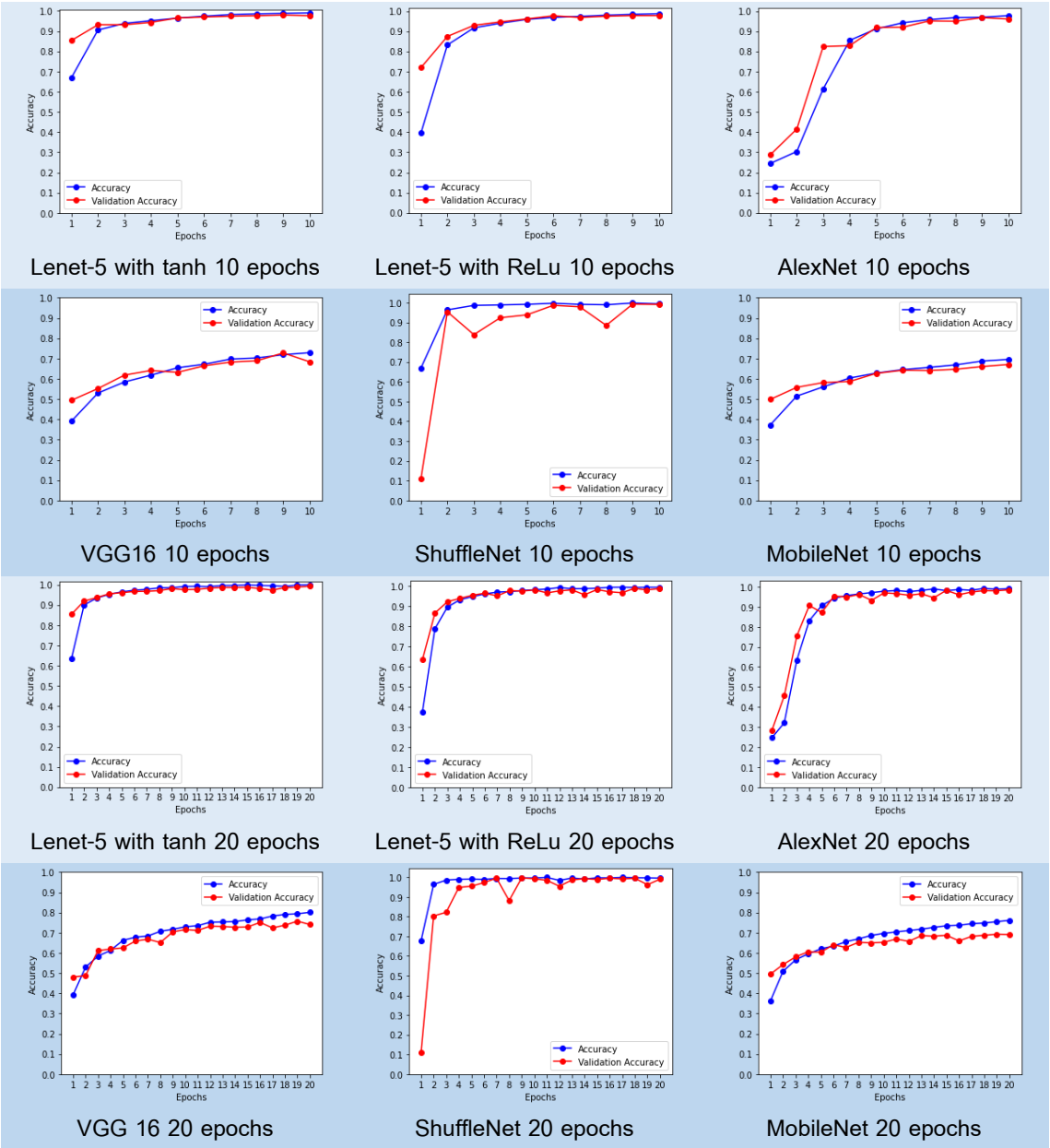
4.1.1 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม

การเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 5 ชนิดกับกลุ่มภาพความละเอียดผสม โดยเป็นกลุ่มภาพดั้งเดิมจากชุดข้อมูล ที่ผสมระหว่างความละเอียดสูงและความละเอียดต่ำ ผู้วิจัยจึงเลือกกลุ่มภาพความละเอียดผสม เป็นการทดลองตั้งต้น ผลการทดลองเป็นดังตาราง 11

ตาราง 11 แสดงผลลัพธ์ประสิทธิภาพของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม

แบบจำลอง	Activation Function	Epochs	accuracy	precision	recall	f1-score
LeNet-5	Tanh	10	0.98	0.98	0.98	0.98
		20	0.99	0.99	0.99	0.99
LeNet-5	ReLu	10	0.98	0.98	0.98	0.98
		20	0.99	0.99	0.99	0.99
AlexNet	ReLu	10	0.96	0.97	0.96	0.96
		20	0.98	0.98	0.98	0.98
VGG16	ReLu	10	0.69	0.77	0.69	0.69
		20	0.79	0.80	0.79	0.79
ShuffleNet		10	0.99	0.99	0.99	0.99
		20	0.99	0.99	0.99	0.99
MobileNet		10	0.68	0.69	0.68	0.68
		20	0.70	0.71	0.70	0.69

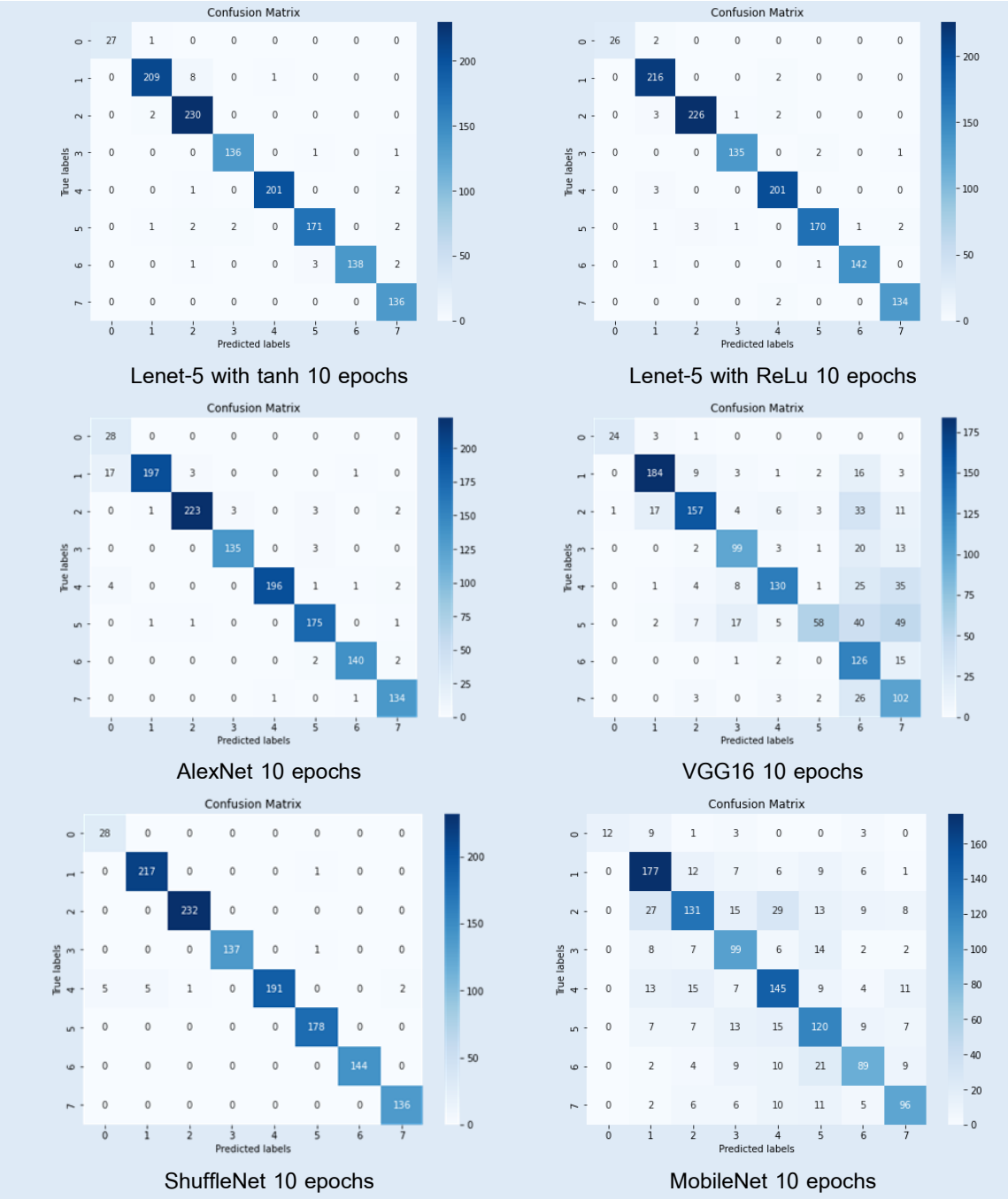
จากตาราง 11 และภาพประกอบ 9 แสดงให้เห็นว่า ในกลุ่มภาพที่มีความละเอียดผสม แบบจำลอง LeNet-5 ที่ใช้ Activation Function เป็น Tanh และ ReLu ซึ่งมี Epochs เท่ากับ 20 และแบบจำลอง ShuffleNet ที่ Epochs 10 และ 20 ได้ค่าความถูกต้องสูงสุดที่ 0.99 ซึ่งเป็นแบบจำลองที่มีประสิทธิภาพดีที่สุดในกลุ่มนี้ รองลงมา เป็นแบบจำลอง Lenet-5 ที่ใช้ Activation Function เป็น ReLu ซึ่งมี Epochs เท่ากับ 10 และแบบจำลอง Alexnet ที่มี Epochs เท่ากับ 20 ซึ่งมีค่าความถูกต้อง อยู่ที่ 0.98



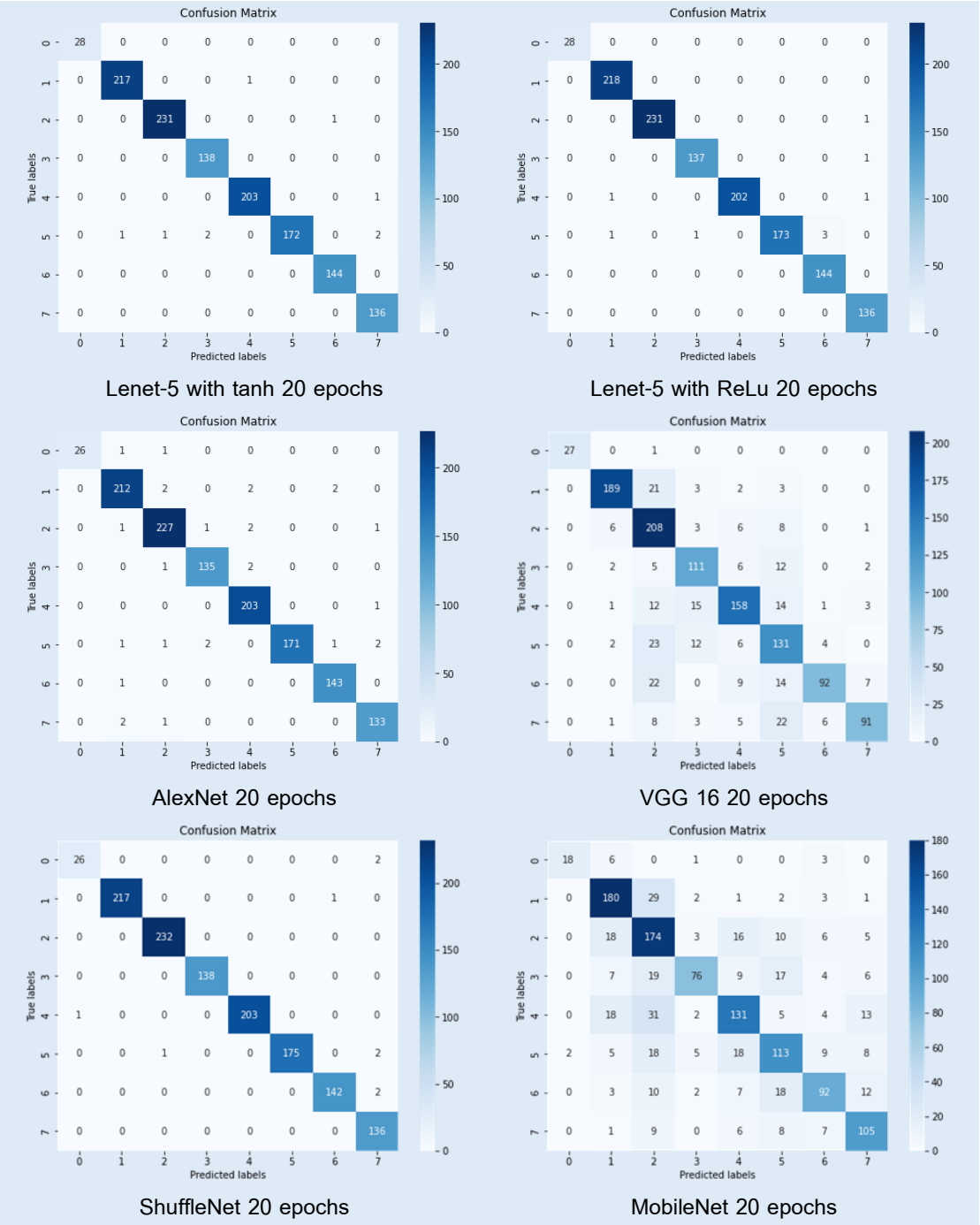
ภาพประกอบ 9 แสดงกราฟเส้นแสดงค่าความถูกต้องของแบบจำลอง
กับกลุ่มภาพความละเอียดผสมเทียบกับจำนวนรอบการฝึก

ซึ่งผลลัพธ์ของ Confusion Matrix ดังภาพประกอบ 10 และภาพประกอบ 11 แสดงให้เห็นว่า ในกลุ่มภาพที่มีความละเอียดผสม จะเห็นว่า แบบจำลอง Lenet-5, AlexNet และ ShuffleNet มีการทายได้อย่างถูกต้องมาก มีเพียงเล็กน้อยเท่านั้นที่ทายผิด แต่แบบจำลอง VGG16 และ MobileNet มีการทำนายที่ผิดค่อนข้างมาก ตัวอย่างเช่น แบบจำลอง VGG16

ที่มี Epoch เท่ากับ 10 นั้น ทายคลาส 5 เป็นคลาส 7 จำนวน 49 ภาพจากทั้งหมด 178 ภาพ และในแบบจำลอง MobileNet ที่มี Epoch เท่ากับ 20 นั้น ทายคลาส 4 เป็นคลาส 2 จำนวน 31 ภาพ จาก 204 ภาพ



ภาพประกอบ 10 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม รอบฝึก 10 รอบ



ภาพประกอบ 11 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม รอบฝึก 20 รอบ

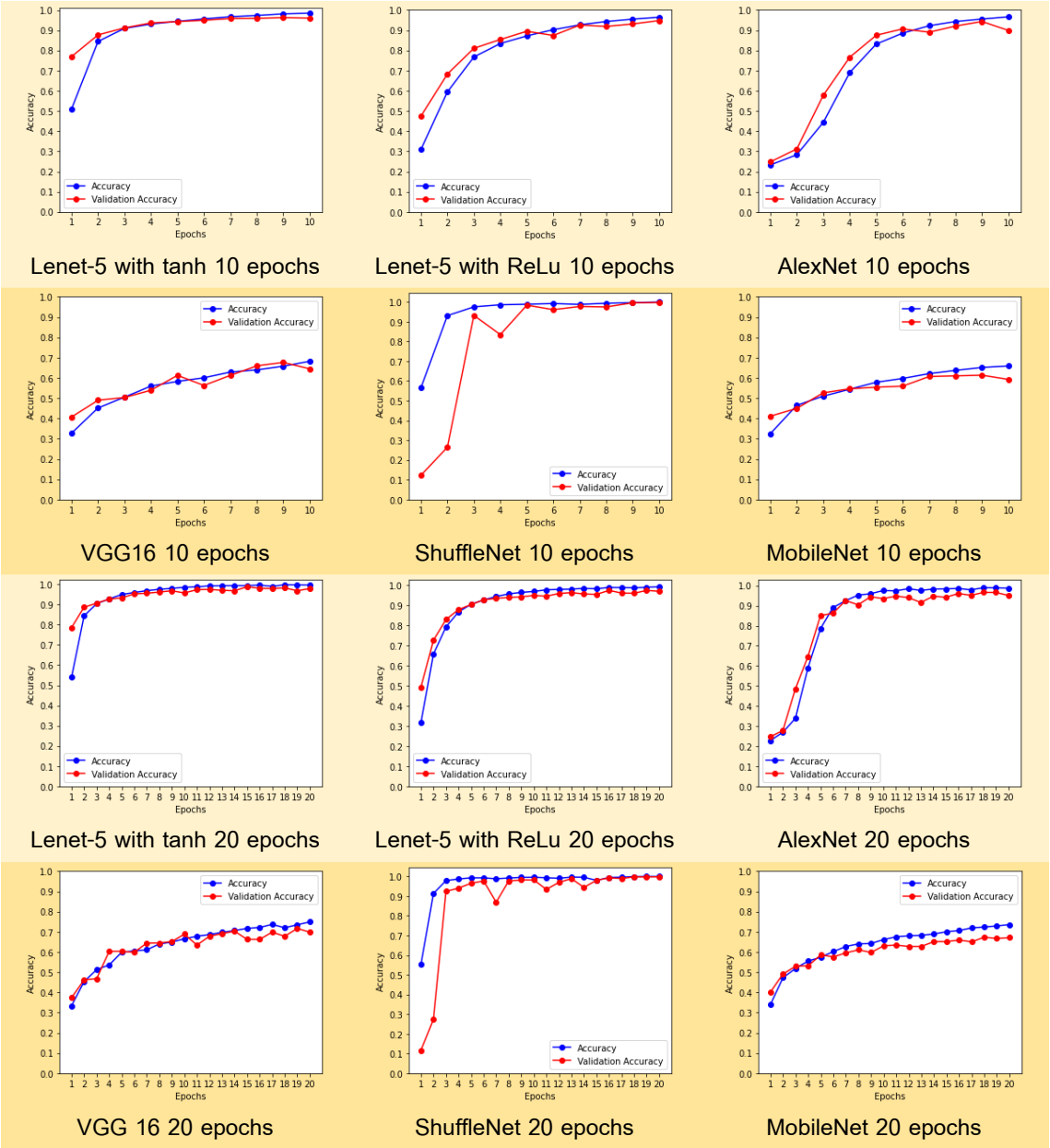
ดังนั้น เราจึงทำการแยกกลุ่มภาพออกมาเป็นอีก 2 กลุ่ม เพื่อศึกษาว่าเฉพาะกลุ่มภาพความละเอียดแบบใดที่ทำให้แบบจำลองมีประสิทธิภาพและส่งผลต่อการคัดแยกมากกว่ากัน

4.1.2 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ

ตาราง 12 แสดงผลลัพธ์ประสิทธิภาพของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ

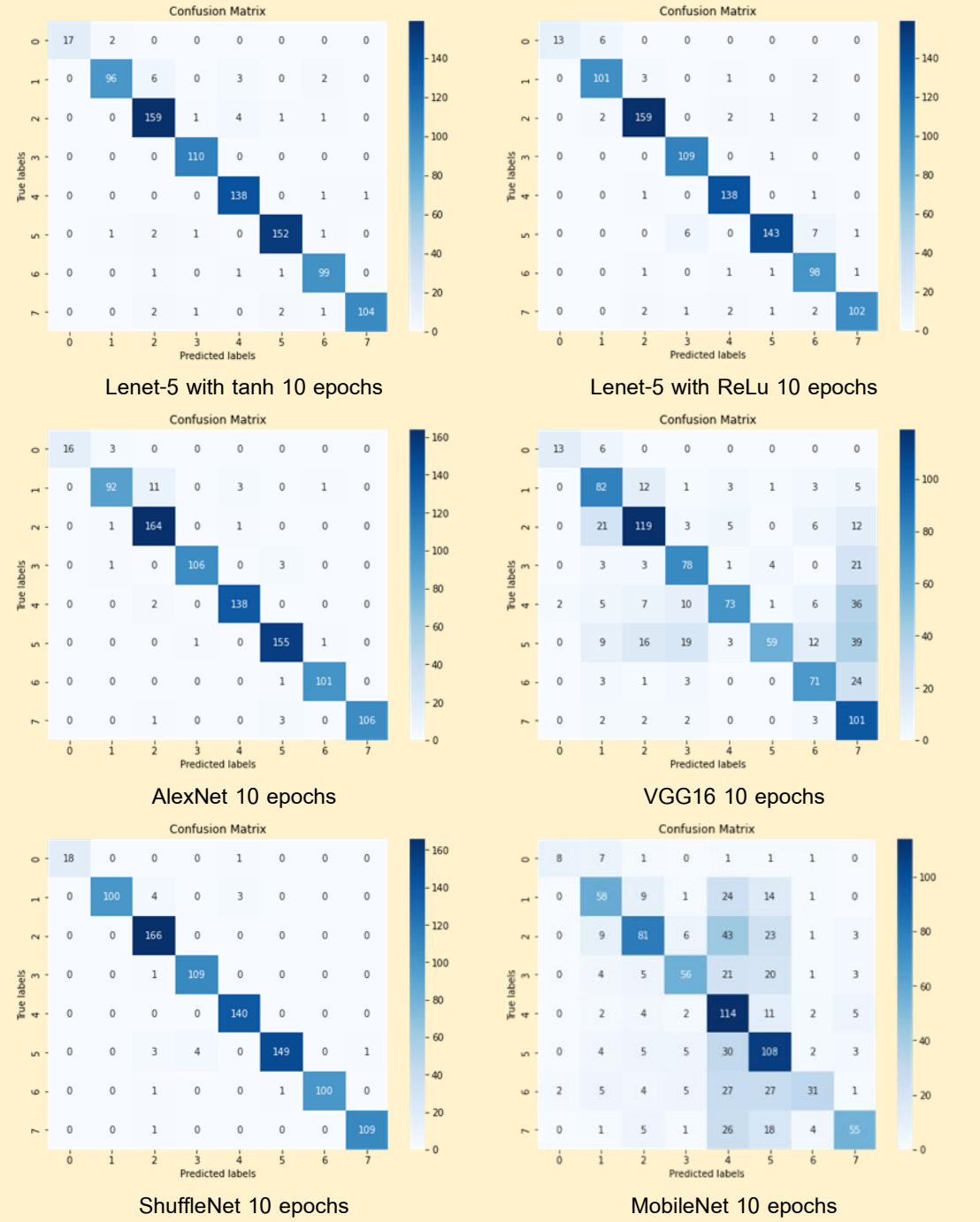
แบบจำลอง	Activation Function	Epochs	accuracy	precision	recall	f1-score
LeNet-5	Tanh	10	0.96	0.96	0.96	0.96
		20	0.98	0.98	0.98	0.98
	ReLu	10	0.95	0.95	0.95	0.95
		20	0.97	0.97	0.97	0.97
AlexNet	ReLu	10	0.96	0.96	0.96	0.96
		20	0.96	0.96	0.96	0.96
VGG16	ReLu	10	0.65	0.73	0.65	0.65
		20	0.72	0.76	0.72	0.72
ShuffleNet		10	0.98	0.98	0.98	0.98
		20	1.00	1.00	1.00	1.00
MobileNet		10	0.56	0.63	0.56	0.56
		20	0.64	0.65	0.64	0.64

จากตาราง 12 และภาพประกอบ 12 แสดงให้เห็นว่า กลุ่มภาพที่มีความละเอียดต่ำ แบบจำลอง ShuffleNet ที่ Epoch เท่ากับ 20 ได้ค่า Accuracy และ Precision ที่ 1.00 ซึ่งเป็นแบบจำลองที่มีประสิทธิภาพดีที่สุดสำหรับในกลุ่มภาพที่มีความละเอียดต่ำ รองลงมาเป็น Lenet-5 ที่มี Activation Function เป็น Tanh และมี Epochs เท่ากับ 20 ได้ค่า Accuracy สูงสุดที่ 0.98

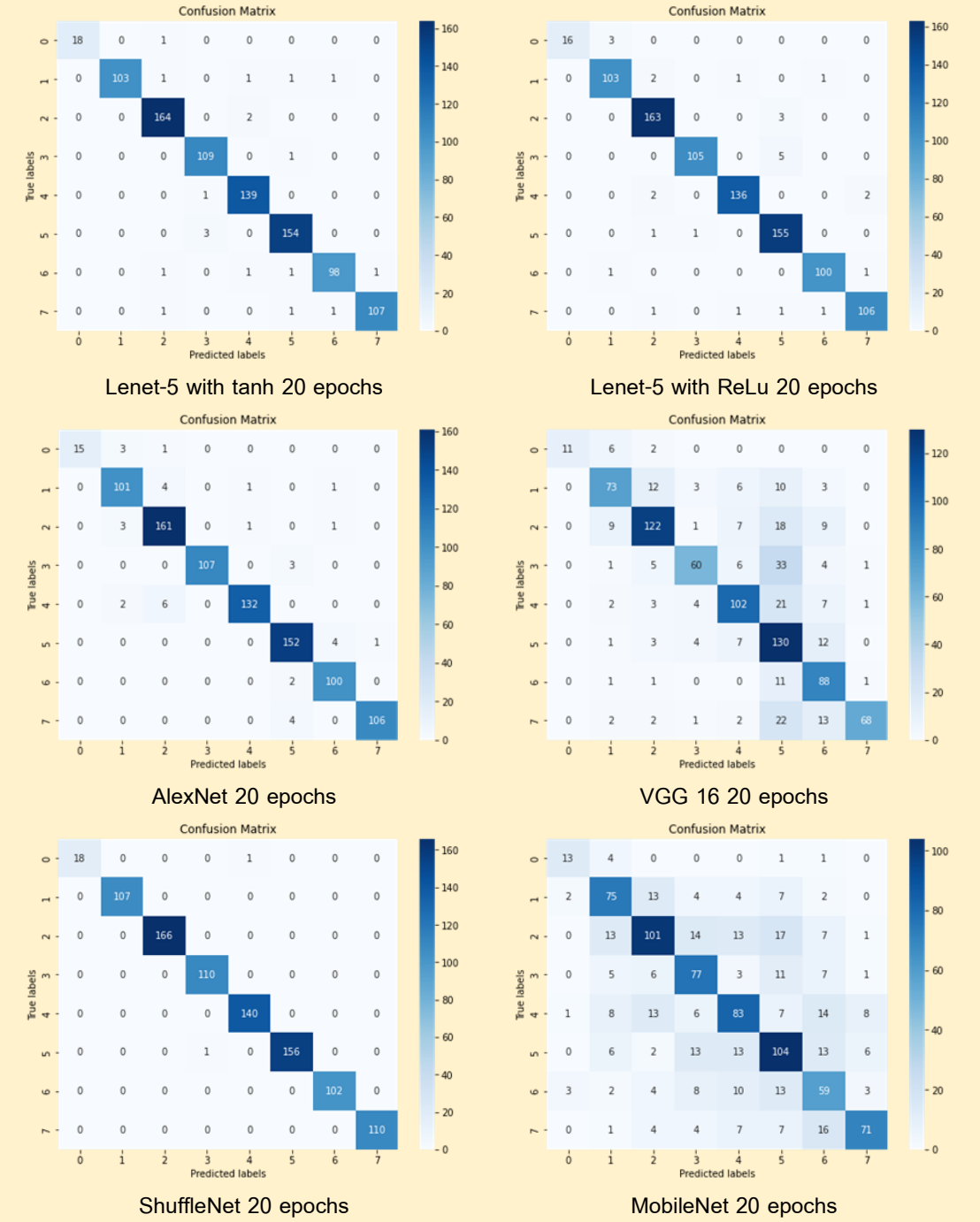


ภาพประกอบ 12 แสดงกราฟเส้นแสดงค่าความถูกต้องของแบบจำลอง
กับกลุ่มภาพความละเอียดต่ำเทียบกับจำนวนรอบการฝึก

ซึ่งจากผลลัพธ์ของ Confusion Martrix ดังภาพประกอบ 13 และภาพประกอบ 14 แสดงให้เห็นว่า ในกลุ่มภาพความละเอียดต่ำ แบบจำลอง VGG16 และ MobileNet ยังคงทายผิด ซึ่งเมื่อเปรียบเทียบกับ ShuffleNet แล้วมีจำนวนคลาสที่ทายผิดค่อนข้างมาก เช่น แบบจำลอง VGG16 ทำนายคลาส 5 เป็น คลาส 7 ถึง 39 ภาพ และแบบจำลอง MobileNet ทำนายคลาส 2 เป็นคลาส 4 ถึง 43 ภาพ



ภาพประกอบ 13 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ รอบฝึก 10 รอบ



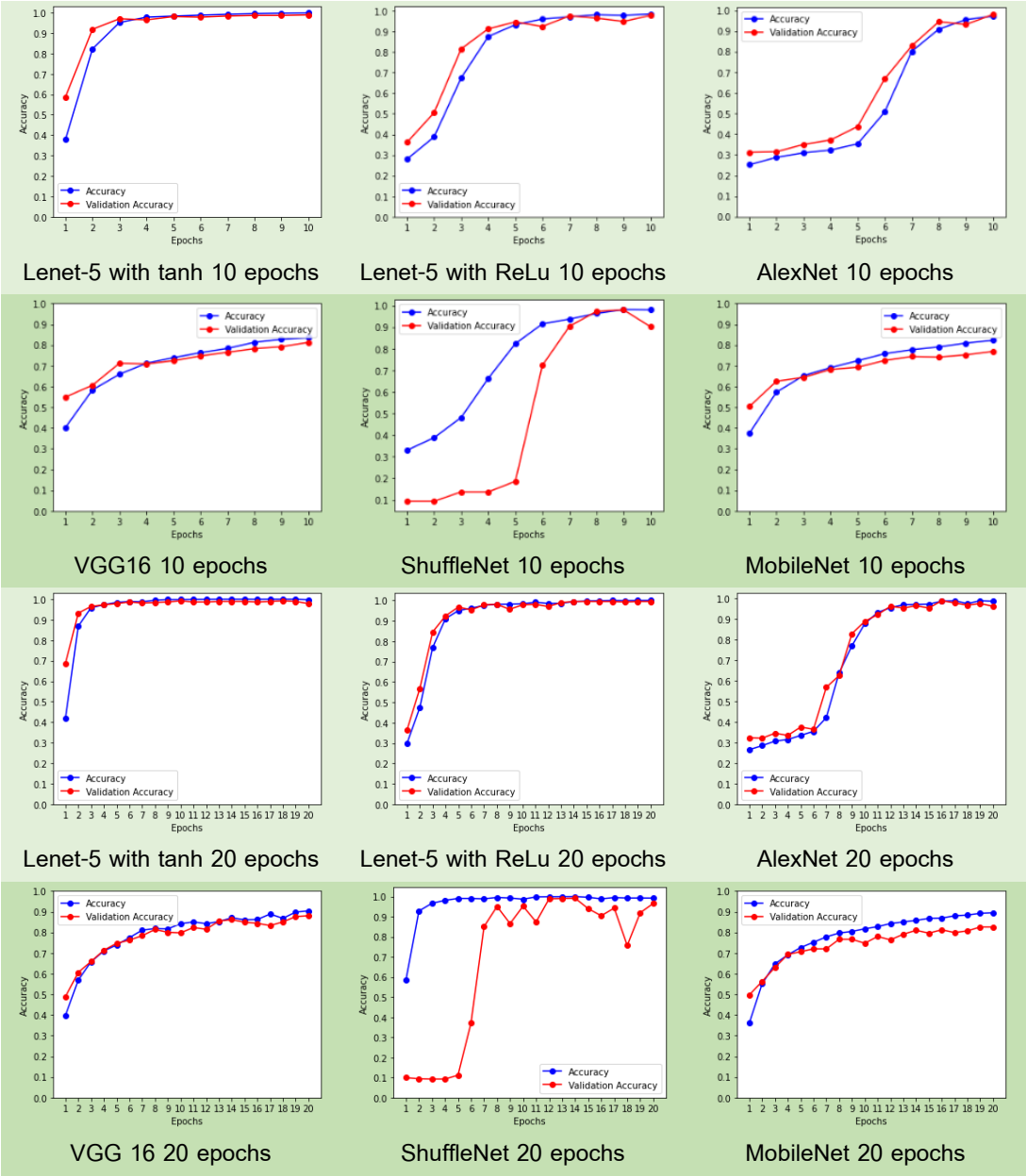
ภาพประกอบ 14 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ รอบฝึก 20 รอบ

4.1.3 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดสูง

ตาราง 13 แสดงผลลัพธ์ประสิทธิภาพของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดสูง

แบบจำลอง	Activation Function	Epochs	accuracy	precision	recall	f1-score
LeNet-5	Tanh	10	0.99	0.99	0.99	0.99
		20	0.98	0.98	0.98	0.98
	ReLu	10	0.98	0.98	0.98	0.98
		20	0.99	0.99	0.99	0.99
AlexNet	ReLu	10	0.78	0.81	0.78	0.78
		20	1.00	1.00	1.00	1.00
VGG16	ReLu	10	0.83	0.85	0.83	0.83
		20	0.88	0.87	0.87	0.87
ShuffleNet		10	0.97	0.97	0.97	0.97
		20	0.97	0.97	0.97	0.97
MobileNet		10	0.77	0.77	0.77	0.77
		20	0.82	0.83	0.82	0.82

จากตาราง 13 และภาพประกอบ 15 แสดงให้เห็นว่า ในกลุ่มภาพที่มีความละเอียดสูง แบบจำลองที่มีประสิทธิภาพที่สูงสุดคือ AlexNet และมีจำนวน Epochs เท่ากับ 20 ซึ่งมีความแม่นยำที่สูงกว่า แบบจำลองที่มีประสิทธิภาพรองลงมา นั้นคือ LeNet-5 ที่ใช้ Activation Function เป็น Tanh และมีจำนวน Epochs เท่ากับ 10 ซึ่งมีค่าความแม่นยำที่ 0.99 และ LeNet-5 ที่ใช้ Activation Function เป็น ReLU และมีจำนวน Epochs เท่ากับ 20 ซึ่งมีความแม่นยำอยู่ที่ 0.99 โดยทั้งสองแบบจำลองนี้มีประสิทธิภาพใกล้เคียงกัน ส่วน Shufflenet ที่ค่อนข้างดีในการทดลองที่ 1 และ 2 นั้น อาจเห็นได้ว่า ภาพกลุ่มความละเอียดสูงมีผลต่อการประเมิน ซึ่งค่าความถูกต้องต่อรอบการฝึกฝนในช่วงแรกไม่ค่อยดีมากนัก ผิดกับแบบจำลอง Lenet-5 และ AlexNet ที่ช่วงค่าความถูกต้องกับรอบการฝึกนั้น ดีตั้งแต่ช่วงแรก ๆ



ภาพประกอบ 15 แสดงกราฟเส้นแสดงค่าความถูกต้องของแบบจำลอง
กับกลุ่มภาพความละเอียดสูงเทียบกับจำนวนรอบการฝึก

จากการทดสอบนี้ จะเห็นได้ว่า ภาพความละเอียดสูง ทำให้มีค่าความถูกต้องมากขึ้น ในขณะเดียวกันภาพความละเอียดต่ำ จะทำให้แบบจำลองมีความยากในการทำนายและเกิดข้อผิดพลาดได้ง่ายกว่า ซึ่งแสดงให้เห็นว่า ความละเอียดของภาพมีผลต่อประสิทธิภาพของแบบจำลองที่สร้างขึ้น

จากนั้น จึงทำการเพิ่มขนาดของข้อมูล (Data Augmentation) เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลอง ระหว่างข้อมูลที่ไม่ได้ทำ Data Augmentation กับ ข้อมูลที่มีการทำ Data Augmentation แบบจำลองชนิดได้ จะสามารถทนต่อการฝึกข้อมูลที่ต่างไปจากเดิมได้มากกว่ากัน

4.2 ผลลัพธ์การเปรียบเทียบประสิทธิภาพของแบบจำลองกับกลุ่มภาพที่ขยายข้อมูลชุดฝึก (Data Augmentation)

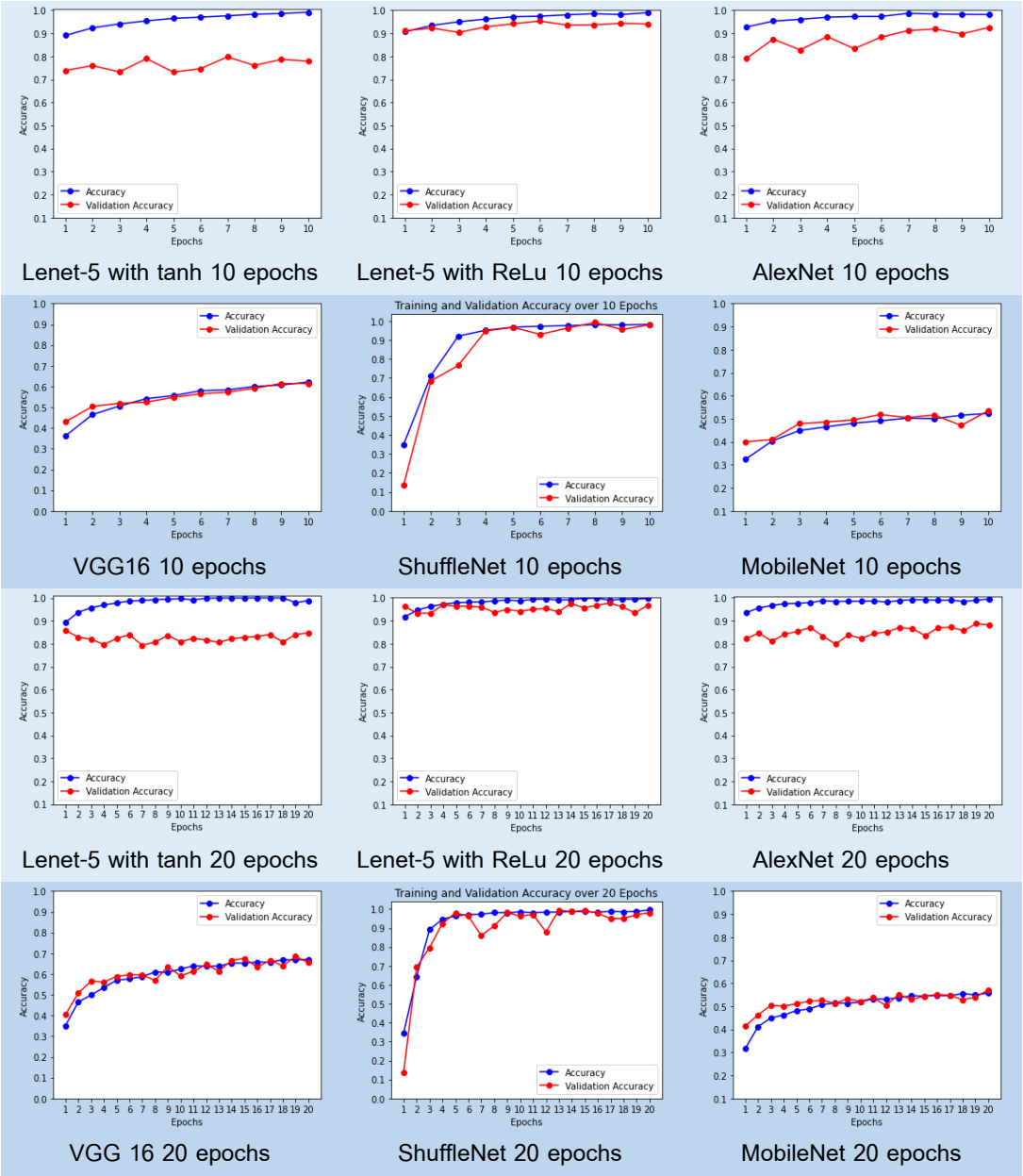
4.2.1 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม

ตาราง 14 แสดงผลลัพธ์ประสิทธิภาพของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสมที่มีการขยายข้อมูลชุดฝึก

แบบจำลอง	Activation Function	Epochs	accuracy	precision	recall	f1-score
LeNet-5	Tanh	10	0.98	0.98	0.98	0.98
		20	0.99	0.99	0.99	0.99
	ReLu	10	0.99	0.99	0.99	0.99
		20	0.99	0.99	0.99	0.99
AlexNet	ReLu	10	0.98	0.98	0.98	0.98
		20	0.98	0.98	0.98	0.98
VGG16	ReLu	10	0.61	0.74	0.61	0.61
		20	0.68	0.74	0.68	0.68
ShuffleNet		10	0.98	0.98	0.98	0.98
		20	0.99	0.99	0.99	0.99
MobileNet		10	0.54	0.57	0.54	0.54
		20	0.59	0.62	0.59	0.58

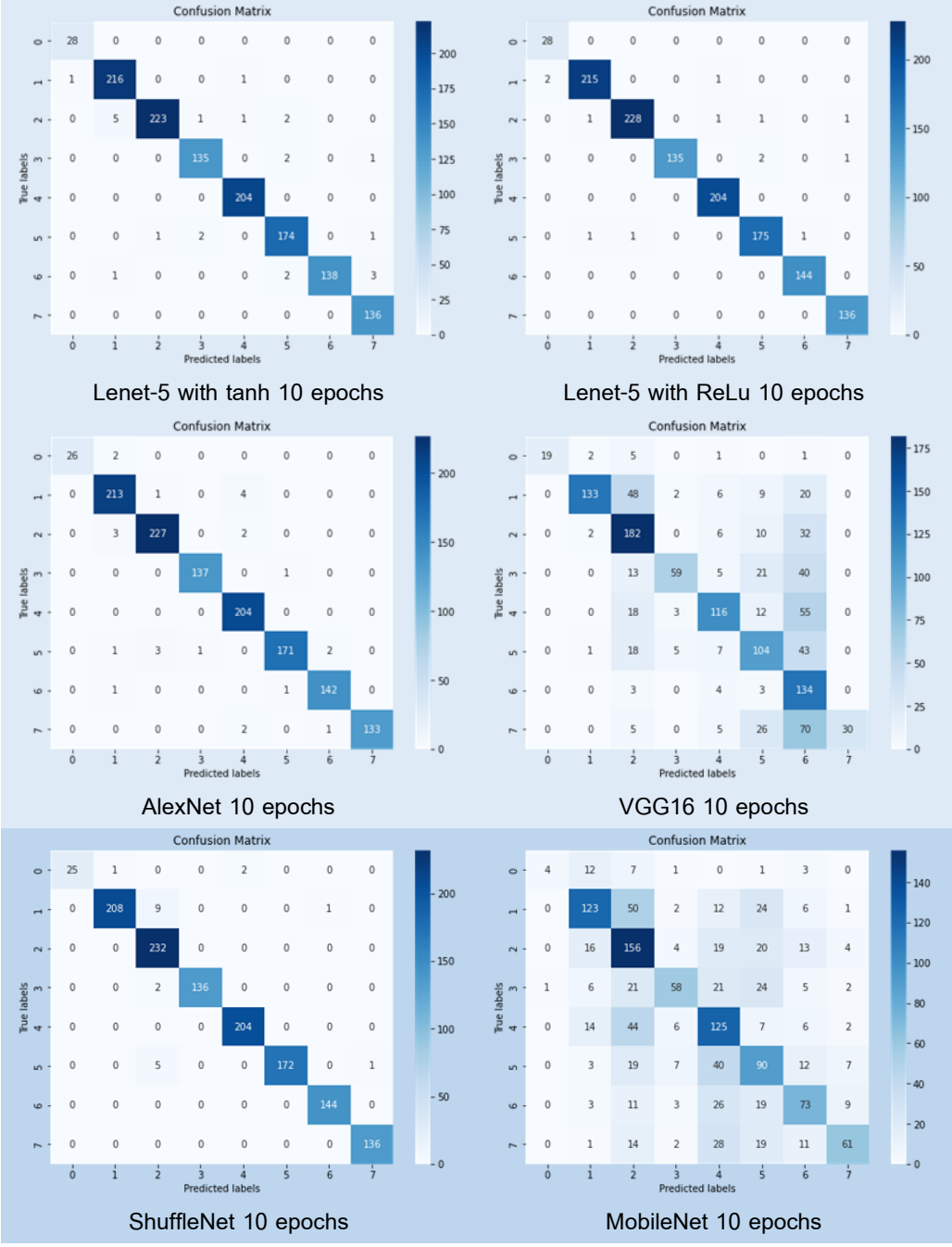
จากตาราง 14 แสดงให้เห็น แบบจำลอง ShuffleNet ที่มี epoch = 20 มีค่า accuracy = 0.99 และจากภาพประกอบ 16 เห็นได้ว่า แบบจำลอง Lenet-5 และ AlexNet เมื่อมีการทำการเพิ่มรูปภาพแล้ว กราฟจะแสดงค่า Accuracy และ Validation Accuracy ซึ่งการที่ค่า Accuracy ในข้อมูลฝึกสูงกว่า ค่า Validation Accuracy แสดงให้เห็นว่าแบบจำลองอาจเกิดปัญหาเรียนรู้ที่เกินไปกับข้อมูลฝึก ซึ่งประสิทธิภาพในการทำนายข้อมูลฝึกมากกว่า โดยมีการ

ปรับตัวเพื่อตรงกับข้อมูลฝึกจนเกินไป ทำให้ไม่สามารถทำนายข้อมูลที่ไม่เคยเห็นมาก่อน (validation data) ได้ดีเท่าที่ควร ซึ่งอาจทำให้ผลลัพธ์จริงไม่แม่นยำเท่ากับชุดข้อมูลที่ไม่ได้ทำ Data Augmentation

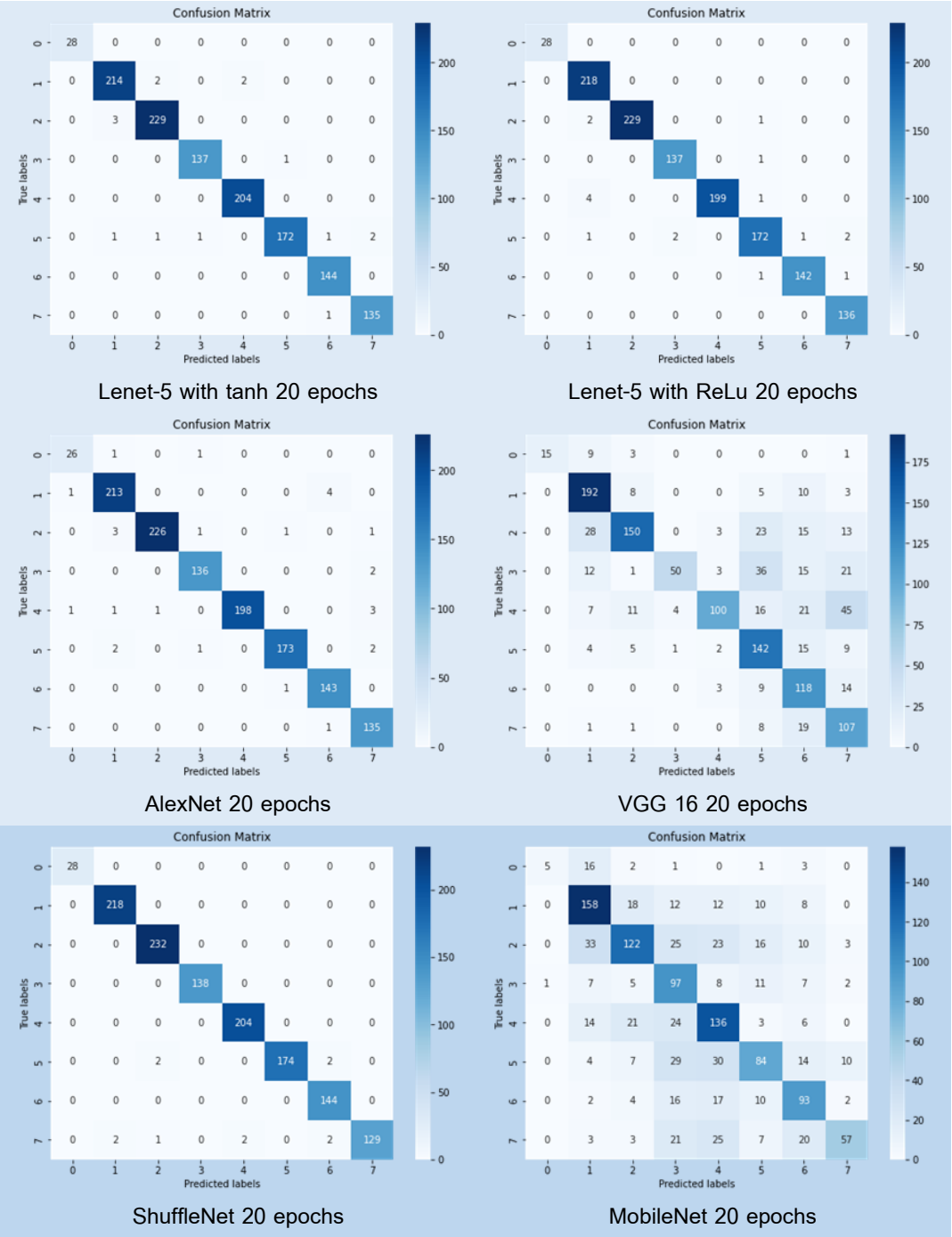


ภาพประกอบ 16 แสดงกราฟเส้นแสดงค่าความถูกต้องของแบบจำลอง
กับกลุ่มภาพความละเอียดผสมที่มีการทำ Data Augmentation เทียบกับจำนวนรอบการฝึก

จากภาพประกอบ 17 และภาพประกอบ 18 เป็น Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสมที่มีการทำ Data Augmentation โดยแบบจำลอง Lenet-5 AlexNet และ ShuffleNet มีการทำนายที่ถูกต้องค่อนข้างมาก เมื่อเทียบกับแบบจำลอง VGG16 และ MobileNet ที่ยังมีส่วนที่ทำนายผิดค่อนข้างเยอะ



ภาพประกอบ 17 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม จำนวนรอบฝึก 10 รอบ



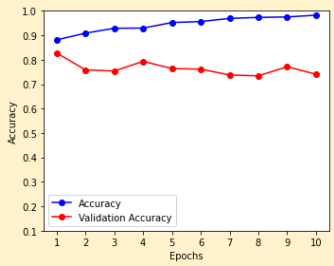
ภาพประกอบ 18 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดผสม จำนวนรอบฝึก 10 รอบ

4.2.2 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ

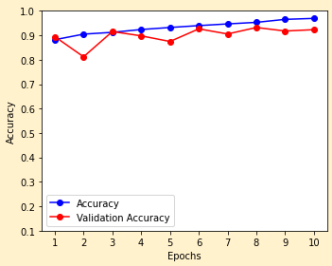
ตาราง 15 แสดงผลลัพธ์ประสิทธิภาพของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำที่มีการขยายข้อมูลชุดฝึก (Data Augmentation)

แบบจำลอง	Activation Function	Epochs	accuracy	precision	recall	f1-score
LeNet-5	Tanh	10	0.98	0.98	0.98	0.98
		20	0.98	0.98	0.98	0.98
	ReLu	10	0.97	0.97	0.97	0.97
		20	0.98	0.98	0.98	0.98
AlexNet	ReLu	10	0.97	0.98	0.97	0.97
		20	0.98	0.98	0.98	0.98
VGG16	ReLu	10	0.57	0.62	0.57	0.57
		20	0.62	0.70	0.62	0.61
ShuffleNet		10	0.97	0.98	0.97	0.97
		20	1.00	1.00	1.00	1.00
MobileNet		10	0.42	0.49	0.42	0.40
		20	0.50	0.52	0.50	0.49

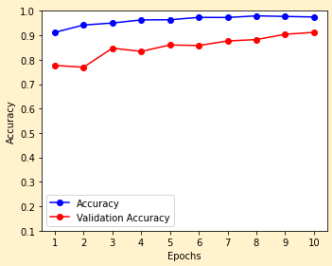
จากตาราง 15 แสดงให้เห็นถึง แบบจำลอง ShuffleNet ที่ epoch = 20 มีค่า Accuracy อยู่ที่ 1.00 รองลงมาจะเป็น แบบจำลอง LeNet-5 ที่ใช้ Activation Function เป็น Tanh และมี Epochs เท่ากับ 10 และ 20 ได้ค่าความแม่นยำ 0.98 และแบบจำลอง LeNet-5 ที่ใช้ Activation Function เป็น ReLU และมี Epochs เท่ากับ 20 ได้ค่าความแม่นยำเช่นเดียวกัน นอกจากนี้ยังมีแบบจำลอง AlexNet ที่มี Epochs เท่ากับ 20 และได้ค่าความแม่นยำเท่ากับ 0.98 เช่นเดียวกัน แต่เมื่อพิจารณาจากภาพประกอบ 19 จะเห็นว่า LeNet-5 ที่ใช้ Activation Function เป็น Tanh มีค่า Accuracy ในข้อมูลฝึกสูงกว่า ค่า Validation Accuracy ค่อนข้างห่าง จึงมีโอกาสที่แบบจำลองเกิดปัญหาเรียนรู้ที่เกินไปกับข้อมูลฝึก ซึ่งประสิทธิภาพในการทำนายข้อมูลฝึกมากกว่า



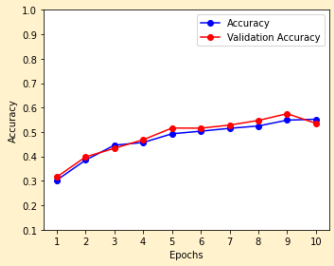
Lenet-5 with tanh 10 epochs



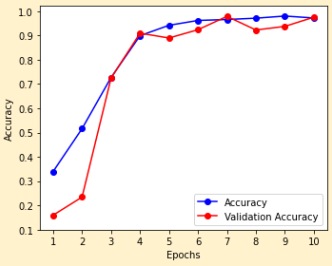
Lenet-5 with ReLu 10 epochs



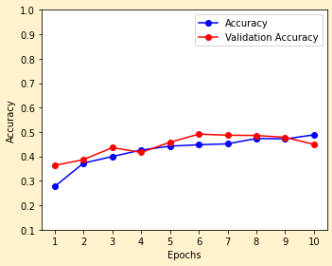
AlexNet 10 epochs



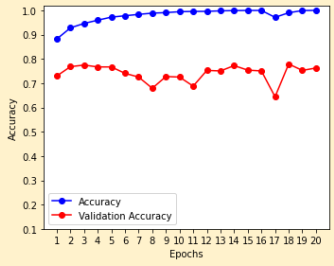
VGG16 10 epochs



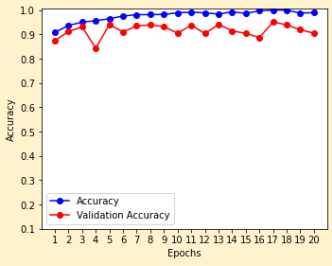
ShuffleNet 10 epochs



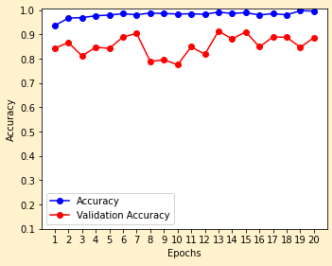
MobileNet 10 epochs



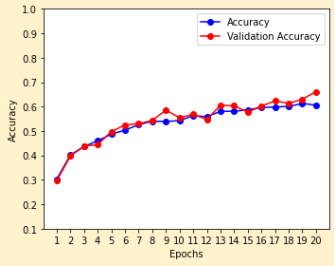
Lenet-5 with tanh 20 epochs



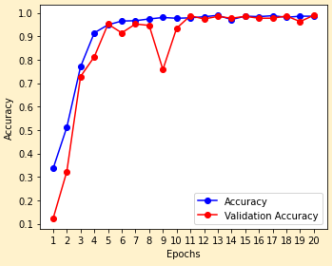
Lenet-5 with ReLu 20 epochs



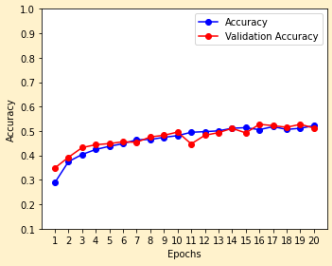
AlexNet 20 epochs



VGG 16 20 epochs



ShuffleNet 20 epochs

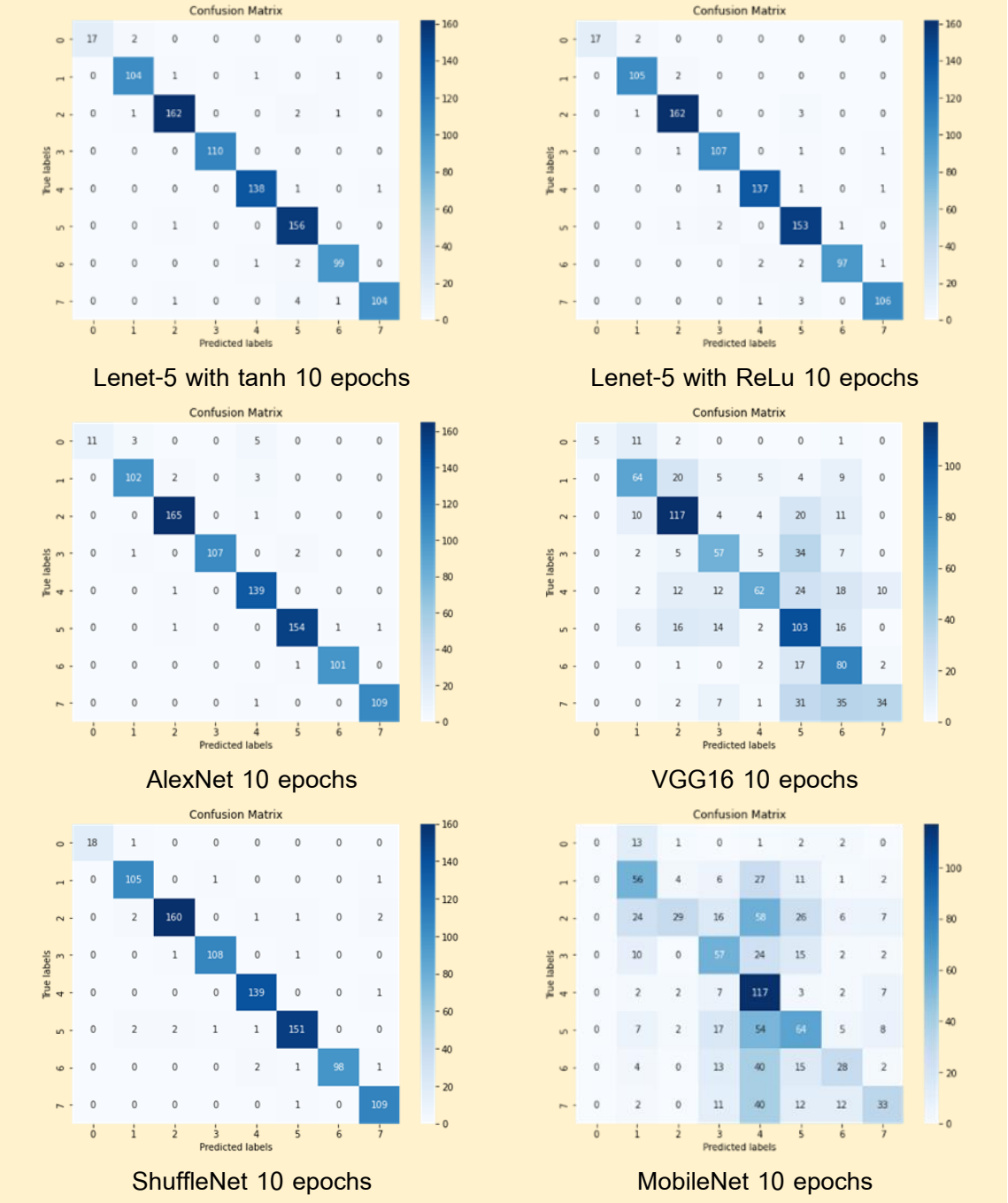


MobileNet 20 epochs

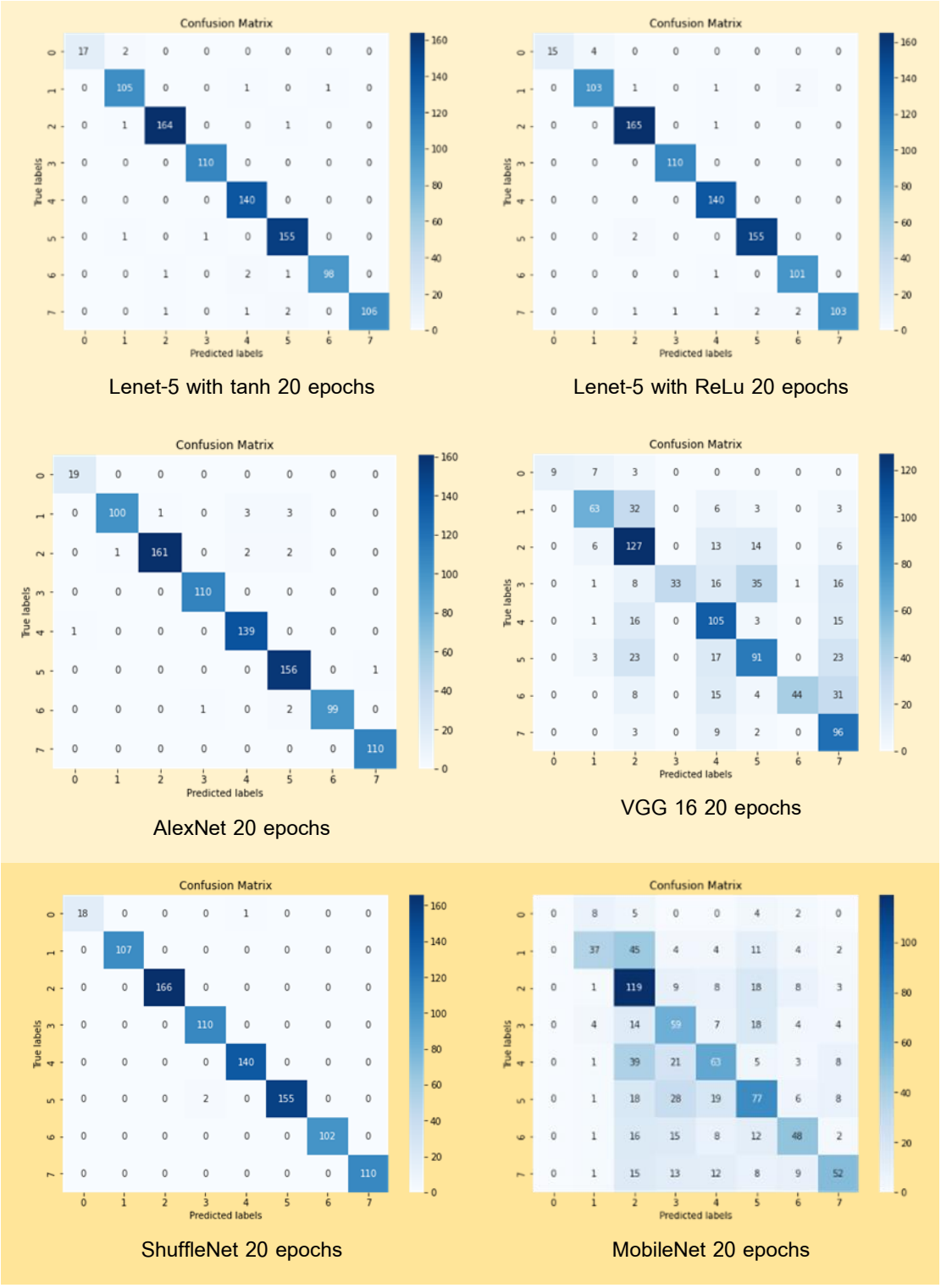
ภาพประกอบ 19 แสดงกราฟเส้นแสดงค่าความถูกต้องของแบบจำลอง
กับกลุ่มภาพความละเอียดต่ำที่มีการทำ Data Augmentation เทียบกับจำนวนรอบการฝึก

จากภาพประกอบ 20 และภาพประกอบ 21 เป็น Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำที่มีการทำ Data Augmentation จะเห็นว่า แบบจำลอง VGG16 และแบบจำลอง MobileNet เมื่อทำ Data Augmentation แล้ว ยังคงมีการทำนาย

ที่ผิดพลาด และมีประสิทธิภาพลดลงทำนาย เมื่อเทียบกับชุดข้อมูลภาพที่ไม่ได้ทำ Data Augmentation



ภาพประกอบ 20 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ จำนวนรอบฝึก 10 รอบ



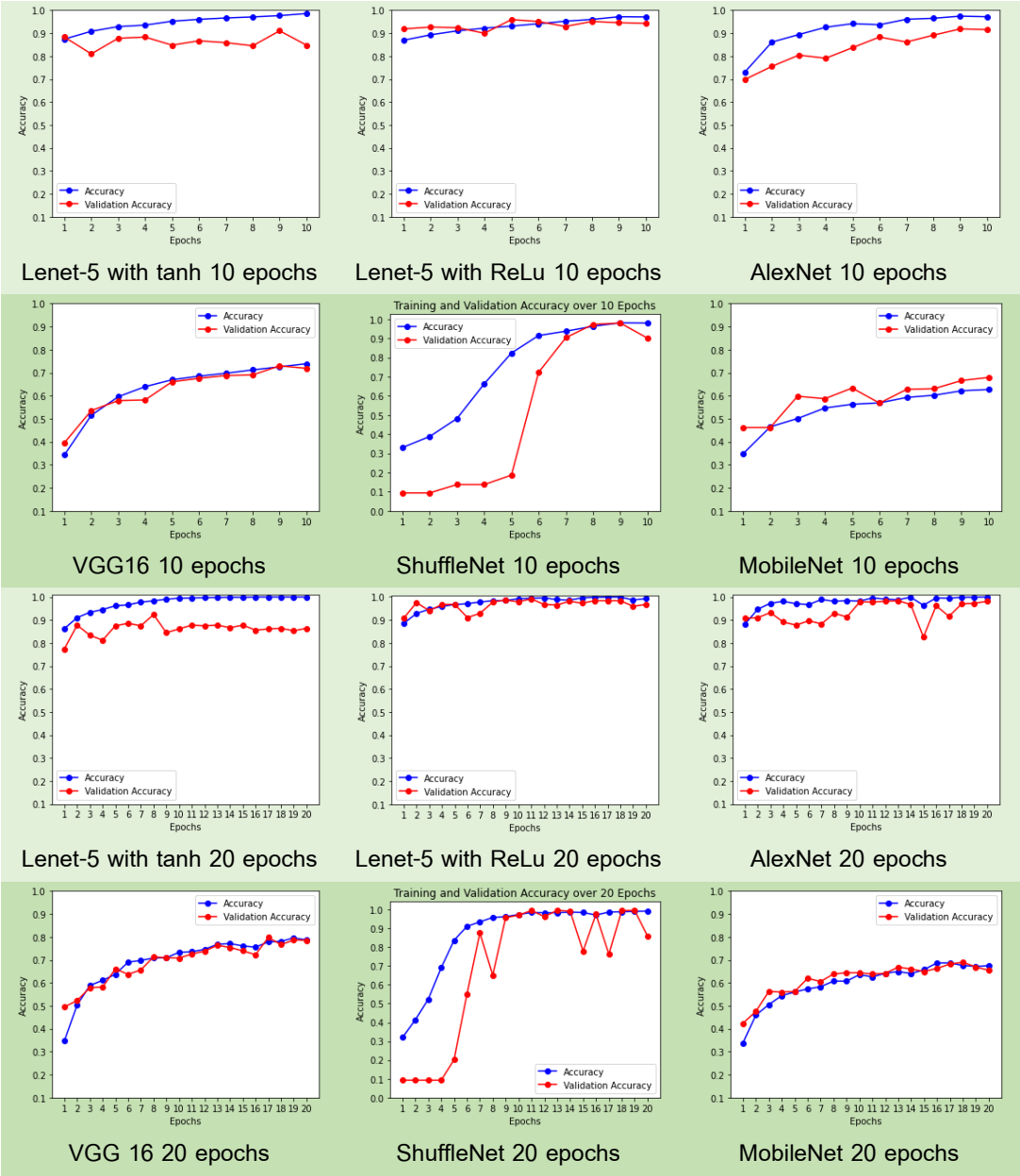
ภาพประกอบ 21 Confusion Matrix ของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดต่ำ จำนวนรอบฝึก 20 รอบ

4.2.3 ผลลัพธ์ประสิทธิภาพแต่ละแบบจำลองกับกลุ่มภาพความละเอียดสูง

ตาราง 16 แสดงผลลัพธ์ประสิทธิภาพของแต่ละแบบจำลองกับกลุ่มภาพความละเอียดสูงที่มีการขยายข้อมูลชุดฝึก (Data Augmentation)

แบบจำลอง	Activation Function	Epochs	accuracy	precision	recall	f1-score
LeNet-5	Tanh	10	0.99	0.99	0.99	0.99
		20	0.99	0.99	0.99	0.99
	ReLu	10	0.99	0.99	0.98	0.99
		20	0.97	0.97	0.97	0.97
AlexNet	ReLu	10	0.98	0.99	0.99	0.99
		20	0.99	0.99	0.99	0.99
VGG16	ReLu	10	0.71	0.73	0.71	0.70
		20	0.74	0.77	0.74	0.74
ShuffleNet		10	0.96	0.96	0.96	0.96
		20	0.99	0.99	0.99	0.99
MobileNet		10	0.65	0.66	0.65	0.64
		20	0.67	0.70	0.67	0.66

จากตาราง 16 แสดงให้เห็นถึง แบบจำลองที่มีประสิทธิภาพที่ดีที่สุดในกลุ่มนี้ ได้แก่ แบบจำลอง LeNet-5 ที่ใช้ Activation Function เป็น Tanh ที่มี Epochs เท่ากับ 10 และ 20 ได้ค่าความแม่นยำ 0.99 แบบจำลอง LeNet-5 ที่ใช้ Activation Function เป็น ReLu ที่มี Epochs เท่ากับ 10 ได้ค่าความแม่นยำ 0.99 แบบจำลอง AlexNet ที่มี Ephocs เท่ากับ 20 ค่า Accuracy เท่ากับ 0.99 และแบบจำลอง ShuffleNet ที่มี Epochs เท่ากับ 20 มีค่า Accuracy เท่ากับ 0.99 แต่เมื่อพิจารณาจากภาพประกอบ 18 จะเห็นว่า LeNet-5 และ AlexNet ค่า Accuracy ในข้อมูลฝึกสูงกว่า ค่า Validation Accuracy ค่อนข้างห่าง จึงมีโอกาสที่แบบจำลองเกิดปัญหาเรียนรู้ที่เกินไปกับข้อมูลฝึก ซึ่งประสิทธิภาพในการทำนายข้อมูลฝึกมากกว่า แต่ก็ยังห่างน้อยกว่ากลุ่มภาพที่มีความละเอียดต่ำ



ภาพประกอบ 22 แสดงกราฟเส้นแสดงค่าความถูกต้องของแบบจำลอง
กับกลุ่มภาพความละเอียดสูงที่มีการทำ Data Augmentation เทียบกับจำนวนรอบการฝึก

ซึ่งหลังจากการทำ Data Augmentation เป็นสิ่งที่ดี เนื่องจากมีการลดปัญหาการทำนายที่ไม่แม่นยำและเพิ่มประสิทธิภาพในการจำแนกภาพ อย่างไรก็ตาม เมื่อมีการเพิ่มความซับซ้อนในแบบจำลองด้วย Data Augmentation อาจทำให้เกิดปัญหาแบบจำลองเกิดปัญหาเรียนรู้ที่เกินไปกับข้อมูลฝึก ซึ่งเป็นปัญหาที่มักจะเกิดขึ้นเมื่อโมเดลมีความซับซ้อนมากเกินไป ทำให้โมเดลทำนายข้อมูลในชุดฝึกได้ดีกว่าที่ควรและไม่สามารถทำนายข้อมูลในชุดทดสอบได้ดี

เท่ากับในชุดฝึก โดยจากการทดลองสามารถชี้ให้เห็นถึงความแตกต่างของแบบจำลองทั้ง 5 แบบ ได้ดังตาราง 17

ตาราง 17 แสดงความแตกต่างของแบบจำลองทั้ง 5 แบบ

คุณลักษณะ	Lenet-5	AlexNet	VGG16	ShuffleNet	MobileNet
โครงสร้างแบบจำลอง	เรียงลำดับการทำงานเป็นชั้น	เรียงลำดับการทำงานเป็นชั้น	เรียงลำดับการทำงานเป็นชั้น	Shuffle Unit	เรียงลำดับการทำงานเป็นชั้น
ขนาดข้อมูลรับเข้า	32x32	224x224	224x224	224x224	224x224
Channels	1 Chanel (Grayscale)	3 Chanel (RGB)	3 Chanel (RGB)	3 Chanel (RGB)	3 Chanel (RGB)
ขนาดตัวกรอง (Filter Size)	Small	Various	3x3	Grouped	3x3
การทำพูลลิ่ง (Pooling)	Average/Max	Max	Max	Average	Average
Activation Function	Tanh/ReLU	ReLU	ReLU	ReLU	ReLU
Dropout	ไม่มี	มี	มี	ไม่มี	มี
พารามิเตอร์	60,000	60 ล้าน	138 ล้าน	1.3 ล้าน	4.2 ล้าน
Training Average Time*	50 วินาที	93 นาที	96 นาที	8 นาที	26 นาที
Test Average Time*	2 วินาที	8 วินาที	5 นาที	23 วินาที	15 วินาที
ข้อดี	- โครงสร้างไม่ซับซ้อน - เข้าใจง่าย - ใช้พารามิเตอร์น้อย	- โครงสร้างซับซ้อนกว่า Lenet-5 - มีการทำ Dropout	- โครงสร้างซับซ้อนแต่เข้าใจง่าย	- ใช้เทคนิคการแบ่งกลุ่มและสลับตำแหน่งเพื่อลดความซับซ้อนของแบบจำลอง	- ใช้โครงสร้างไม่ซับซ้อนทำให้เหมาะสมกับอุปกรณ์ที่มีทรัพยากรจำกัด

ตาราง 17 (ต่อ)

คุณลักษณะ	Lenet-5	AlexNet	VGG16	ShuffleNet	MobileNet
ข้อเสีย	- แบบจำลองขนาดเล็ก - ไม่เหมาะกับข้อมูลซับซ้อน	- ต้องใช้ทรัพยากรมาก - มีพารามิเตอร์มาก	- ต้องใช้ทรัพยากรมาก - ใช้เวลาฝึกค่อนข้างนาน	- มีความซับซ้อนในการออกแบบโครงสร้าง	- มีข้อจำกัดในการใช้งาน - ไม่เหมาะกับงานที่ต้องการความละเอียดสูง
เหมาะกับ	งานด้านตัวเลขแบบพิมพ์และข้อมูลที่ไม่ซับซ้อน	ข้อมูลขนาดใหญ่ ด้านงานภาพที่ต้องการความละเอียดสูง	ด้านการจำแนกภาพทั่วไปและงานที่ต้องการความแม่นยำสูง	งานที่ต้องการประสิทธิภาพสูงแต่ใช้พลังงานในการประมวลผลน้อย	ระบบที่ต้องการความรวดเร็วและอุปกรณ์เคลื่อนที่

* ซึ่งเวลา Training Average Time และ Test Average Time ในตาราง 17 เป็นเวลาที่มาจากการเฉลี่ยของเวลาระหว่างแบบจำลองกับกลุ่มภาพความละเอียดต่าง ๆ

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

ในการวิจัยเรื่องโครงข่ายประสาทเทียมแบบคอนโวลูชันเชิงลึกสำหรับการจำแนกป้ายจราจรจำกัดความเร็ว ผู้วิจัยได้ทำการประเมินประสิทธิภาพของแบบจำลอง ซึ่งหลังจากดำเนินงานแล้ว สามารถสรุปผลการดำเนินงาน โดยแบ่งเป็นหัวข้อดังต่อไปนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การประเมินผลการเปรียบเทียบประสิทธิภาพของแบบจำลอง ได้แก่ Lenet-5, Alexnet, VGG16, ShuffleNet, MobileNet ในการจำแนกป้ายจราจรที่มีความเร็วที่มีจำนวน 8 คลาส ได้แก่ 20, 30, 50, 60, 70, 80, 100, และ 120 กิโลเมตรต่อชั่วโมง โดยการทดลองจะแบ่งชุดข้อมูลออกเป็น 3 กลุ่มตามความละเอียดของภาพ โดยใช้แพลตฟอร์ม Knime เข้ามาจัดการข้อมูลส่วนนี้ เพื่อนำไปวิเคราะห์ผลของการจำแนกป้ายจราจรในแต่ละกลุ่ม หาแบบจำลองที่เหมาะสมที่สุดสำหรับงานการจำแนกป้ายจราจรจำกัดความเร็ว โดยใช้ชุดข้อมูลจริงจากภาพถ่ายป้ายจราจร ที่มีความคล้ายคลึงกับประเทศไทย ซึ่งทำการแบ่งชุดข้อมูลออกเป็น ชุดฝึกฝน, ชุดการตรวจสอบ, และชุดทดสอบในอัตราส่วน 80:10:10 และใช้รอบการฝึกฝนที่ต่างกันนั้นคือ epochs 10 รอบ และ 20 นอกจากนี้ในแบบจำลอง Lenet-5 ยังมีการเลือกใช้ฟังก์ชันกระตุ้นระหว่าง Tanh และ ReLu เพื่อทำให้เกิดความเข้าใจเกี่ยวกับประสิทธิภาพของแต่ละโมเดลในระหว่างการฝึก

ผลการทดลองพบว่า เมื่อพิจารณาภาพรวมของกลุ่มความละเอียดทั้ง 3 กลุ่ม แบบจำลอง ShuffleNet มีความสามารถในการคัดแยกป้ายจราจรจำกัดความเร็วได้ดีที่สุด รองลงมาเป็น Lenet-5 และ AlexNet แต่เมื่อหากพิจารณากลุ่มความละเอียดแยกกันแล้ว แบบจำลอง ShuffleNet ในรอบต้นๆ ของการฝึกฝน อาจจะได้ผลลัพธ์ไม่ดีเท่ากลุ่มภาพความละเอียดผสม กับ ความละเอียดต่ำ

5.2 อภิปรายผลการวิจัย

เนื่องจากงานวิจัยนี้ได้ประเมินผลการเปรียบเทียบประสิทธิภาพของแบบจำลองโครงข่ายประสาทเทียมคอนโวลูชัน จึงทำให้เห็นถึงปัญหาที่เกิดขึ้นในการทดลอง ที่ส่งผลต่อความน่าเชื่อถือของผลลัพธ์ ดังต่อไปนี้

5.2.1 ชุดข้อมูล

จากการทดลองพบว่า ชุดข้อมูลที่เลือกใช้มีความหลากหลายของจำนวนประเภท แต่ผู้วิจัยคัดเลือกเฉพาะป้ายจราจรจำกัดความเร็วเท่านั้น จึงอาจจะส่งผลต่อการฝึกฝนของแบบจำลอง ในจำนวนของภาพถ่ายในแต่ละชุดที่เลือกมา มีมากกว่าบางประเภทหลายเท่า นอกจากนั้น สาเหตุที่อาจสงผลต่อความถูกต้องและน่าเชื่อถือ ก็ยังมีเรื่องของลักษณะตัวเลขที่อยู่บนป้ายมีความคล้ายคลึงกัน ซึ่งทำให้เกิดความสับสนในการจำแนกของแบบจำลอง ที่ทำนายข้อมูลแต่ละคลาสในภาพไม่ต่างกันมากพอที่จะจำแนกได้อย่างถูกต้อง ซึ่งอาจจะเป็นผลจากความละเอียดของภาพ ขนาดของภาพ มุมมอง แสง และสภาวะแวดล้อมอื่น ๆ

5.2.2 การเลือกแบบจำลองและประสิทธิภาพของแบบจำลอง

จากการทดลองพบว่า เนื่องจาก Lenet-5 เป็นแบบจำลองที่เก่าแก่และเรียบงาน ซึ่งมีข้อดีคือเหมาะสำหรับการเรียนรู้และใช้พารามิเตอร์ค่อนข้างน้อย ส่วน AlexNet เป็นแบบจำลองที่มีการใช้ฟังก์ชันกระตุ้น ReLu มีการทำเทคนิค Dropout แต่เป็นแบบจำลองที่ใหญ่และใช้การประมวลผลสูงซึ่งอาจจะไม่เหมาะกับอุปกรณ์ที่มีทรัพยากรจำกัด ส่วน VGG16 มีโครงสร้างของแบบจำลองที่เป็นระบบ แต่มีจำนวนพารามิเตอร์ที่ค่อนข้างสูง ซึ่งใช้ทรัพยากรในการประมวลผลสูงและใช้เวลาค่อนข้างนาน ซึ่ง ShuffleNet เป็นแบบจำลองที่มีการใช้เทคนิคเพื่อลดการใช้ทรัพยากร มีความเร็วในการประมวลผลสูง และ MobileNet เป็นแบบจำลองขนาดเล็กที่พัฒนาเพื่อใช้งานในอุปกรณ์พกพา โดยเมื่อเปรียบเทียบด้านประสิทธิภาพของแบบจำลองแล้วนั้น พบว่า VGG16 มีค่าความถูกต้องดี แต่ใช้เวลาประมวลผลนาน ซึ่ง ShuffleNet ใช้เวลาประมวลผลที่น้อยกว่า ส่วน MobileNet มีประสิทธิภาพดีเมื่อเทียบกับการใช้ทรัพยากรที่ต่ำ ส่วน Lenet-5 เนื่องจากเป็นแบบจำลองที่เรียบง่าย จึงมีค่าความถูกต้องไม่สูงมาก เมื่อเปรียบเทียบกับแบบจำลองที่ซับซ้อนกว่า ส่วน AlexNet นั้น มีความถูกต้องสูงกว่า LeNet-5 เนื่องจากโครงสร้างที่ซับซ้อนกว่า Lenet-5

5.2.3 ความละเอียดของภาพแต่ละกลุ่ม

จากการทดลองพบว่า กลุ่มของภาพมีผลต่อประสิทธิภาพของแบบจำลอง เช่น กลุ่มภาพที่มีความละเอียดสูง แบบจำลองเก็บรายละเอียดของภาพได้มาก จึงทำให้แบบจำลองมีการแยกแยะคุณลักษณะได้ดีขึ้น เช่น ภาพชัดสีสว่าง ก็จะทำให้แบบจำลองมีข้อมูลที่ใช้เรียนรู้ได้แม่นยำกว่า ภาพเล็กมีเงาดำมืด เมื่อภาพความละเอียดสูงก็จำเป็นต้องเพิ่มจำนวนชั้นของแบบจำลองเพื่อให้สามารถจับคุณลักษณะของภาพได้ครบถ้วน ซึ่งทำให้แบบจำลองมีความซับซ้อน ต้องใช้พื้นที่ในการเก็บข้อมูลและพลังงานในการประมวลผลมากขึ้น ซึ่งอาจเป็นปัญหาในการใช้งานบนอุปกรณ์ที่มีทรัพยากรจำกัด เช่น โทรศัพท์มือถือหรือระบบฝังตัว แล้วภาพที่มีความละเอียดสูงอาจทำให้แบบจำลองมีแนวโน้มที่จะเกิด Overfitting เนื่องจากมีข้อมูลมากขึ้นที่จะทำให้แบบจำลองเรียนรู้จากรายละเอียดเล็ก ๆ น้อย ๆ ของภาพมากเกินไป ซึ่งในบางครั้งการปรับขนาดของภาพให้เหมาะสมกับแบบจำลอง อาจทำให้ข้อมูลบางส่วนหายไปหรือเกิดการเบลอ ซึ่งส่งผลต่อการจำแนกของแบบจำลองได้

5.3 ข้อเสนอแนะ

5.3.1 ข้อเสนอแนะด้านการปรับโครงสร้างของแบบจำลอง ซึ่งในงานวิจัยนี้มีการปรับแต่งเพียงแค่ฟังก์ชันกระตุ้นของแบบจำลอง Lenet-5 และรอบในการฝึกฝนเท่านั้น ยังไม่ได้ปรับปรุงพารามิเตอร์อื่น ๆ เพิ่มเติม

5.3.2 ข้อเสนอแนะด้านการใช้ชุดข้อมูลทดสอบ เนื่องจากงานวิจัยนี้มุ่งเน้นเพียงป้ายจราจรประเภทจำกัดความเร็วเท่านั้น โดยทั่วไปแล้ว ป้ายจราจรยังมีประเภทอื่น ๆ อีก ซึ่งสามารถเพิ่มความหลากหลายในชุดข้อมูลทดสอบโดยการรวมป้ายจราจรประเภทอื่น ๆ เช่น ป้ายจราจรที่แสดงเป็นเครื่องหมายทางเส้นทาง ป้ายจราจรที่เกี่ยวข้องกับการจำกัดความสูง หรือป้ายจราจรที่แสดงสัญญาณเตือนทางจราจร ที่สามารถเพิ่มความท้าทายในการทดสอบและปรับปรุงประสิทธิภาพของโมเดล

5.3.3 ข้อเสนอแนะด้านการเลือกความละเอียดของภาพ การใช้ภาพที่มีความละเอียดสูงในการวิเคราะห์ทางการแพทย์หรือการตรวจจับวัตถุในระบบอัตโนมัติที่ต้องการความแม่นยำสูง ในขณะที่ภาพที่มีความละเอียดต่ำ อาจเหมาะสำหรับการประมวลผลแบบเรียลไทม์ในอุปกรณ์พกพาหรือระบบฝังตัว

5.3.4 ข้อเสนอแนะด้านการทดลองกับโครงสร้างแบบจำลองอื่น ๆ เพื่อเปรียบเทียบประสิทธิภาพในการจำแนกป้ายจราจร โดยใช้ชุดข้อมูลที่หลากหลาย เพื่อหาโมเดลที่มีประสิทธิภาพสูงสุดและนำไปปรับใช้ในงานวิจัยต่อไป

บรรณานุกรม

Berthold, M., Cebron, N., Dill, F., Gabriel, T., Kötter, T., Meinl, T., . . . Wiswedel, B. (2009). KNIME: the Konstanz information miner. *First publ. in: Data Analysis, Machine Learning and Applications: Proceedings of the* 31*st Annual Conference of the Gesellschaft für Klassifikation e.V., Albert-Ludwigs-Universität Freiburg, March* 7□9*,* 2007. *New York: Springer,* 2008*, V.*

DEPA. (2019). เทคโนโลยีที่สำคัญในยุคดิจิทัล: เทคโนโลยีรถยนต์ไฟฟ้าและไร้คนขับ Tech Series: Electric and Autonomous Cars. https://www.depa.or.th/th/article-view/tech-series-electric-and-autonomous-cars

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

INI. (2013). GTSRB - German Traffic Sign Recognition Benchmark. https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

KNIME. (2023). KNIME software overview. https://www.knime.com/software-overview

Krizhevsky, A., Sutskever, I., และ Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM,* 60(6), 84–90.

Lau, M. M., Lim, K. H., และ Gopalai, A. A. (2015, 21-24 July 2015). *Malaysia traffic sign recognition with convolutional neural network.* Paper presented at the 2015 IEEE International Conference on Digital Signal Processing (DSP).

Lecun, Y., Bottou, L., Bengio, Y., และ Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE,* 86(11), 2278-2324.

Mehta, S., Paunwala, C., และ Vaidya, B. (2019, 15-17 May 2019). *CNN based Traffic Sign Classification using Adam Optimizer.* Paper presented at the 2019 International Conference on Intelligent Computing and Control Systems (ICCS).

Rahman, U. S., และ Maruf. (2023). Traffic Sign Detection and Recognition Using Deep Learning Approach (331-343).

Simonyan, K., และ Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 1409.1556.

VeliČkoviĆ, N., StojkoviĆ, Z., DimiĆ, G., VasiljeviĆ, J., และ Nagamalai, D. (2018). Traffic Sign Classification Using Convolutional Neural Network. *Computer Science & Information Technology*, 177-186.

Vincent, M. A., V. K, R., และ Mathew, S. P. (2020, 3-5 Dec. 2020). *Traffic Sign Classification Using Deep Neural Network.* Paper presented at the 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS).

Zaibi, A., Ladgham, A., และ Sakly, A. (2021). A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network. *Journal of Sensors,* 2021, 8870529.

Zhang, X., Zhou, X., Lin, M., และ Sun, J. (2018). *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices*.

กระทรวงคมนาคม. (2561). คู่มือ มาตรฐานป้ายจราจร. 1.

สถาบันเทคโนโลยีแห่งเอเชีย, ม. ศ. (2565). รายงานสถานการณ์ อุบัติเหตุทางถนนของประเทศไทย พ.ศ. 2561-2564. กรุงเทพฯ.

ประวัติผู้เขียน