การพัฒนาเว็บแอปพลิเคชันเพื่อจำแนกยางที่ดีและยางที่ไม่ดีโดยใช้เทคนิคการเรียนรู้ของเครื่อง

The Web Application Development for Classification Good Tire and Defective Tire Using Machine Learning Techniques

กฤตพงษ์ ชมเนตร, ปวีร์ อินทุลักษณ์, ปุณนที ปิ่นวิเศษ, พัฒนพิศิษฐ์ ขำปากพลี

สาขาวิชา เทคโนโลยีสารสนเทศ คณะ เทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
Emails: s6506021620067@email.kmutnb.ac.th, p.indulakshana@gmail.com, s6506021630038@email.kmutnb.ac.th, s6506021620091@email.kmutnb.ac.th

บทคัดย่อ

งานวิจัยนี้นำเสนอการประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่องในการจำแนกยางที่ดีและยางที่ไม่ดี โดยมีจุดประสงค์เพื่อช่วยคัดกรองคุณภาพของยางด้วยการทดลองกับ 2 อัลกอริทึม เพื่อนำมาเปรียบเทียบประสิทธิภาพ คือ อัลกอริทึมการเรียนรู้เชิงลึกด้วยโครงข่ายประสาทคอนโวลูชันและอัลกอริทึมการเรียนรู้แบบถ่ายโอน ข้อมูลแบ่งเป็น 2 คลาส คือ ยางที่ดีและยางที่ไม่ดี คลาสละ 1,000 ภาพ รวม 2,000 ภาพ เพื่อเปรียบเทียบประสิทธิภาพการจำแนกระหว่างการเรียนรู้เชิงลึกแบบโครงข่ายประสาทคอนโวลูชัน และ อัลกอริทึมการเรียนรู้แบบถ่ายโอน อัลกอริทึมเรียนรู้จากการแบ่งกลุ่มข้อมูลเป็น 2 ส่วน คือ กลุ่มข้อมูลฝึกร้อยละ 70 และกลุ่มข้อมูลทดสอบร้อยละ 30 ซึ่งผลการทดลองแสดงให้เห็นว่า ประสิทธิภาพในการจำแนกด้วยอัลกอริทึมโครงข่ายประสาทคอนโวลูชันมีค่าความถูกต้องเท่ากับ 0.56 ซึ่งน้อยกว่าอัลกอริทึมการเรียนรู้แบบถ่ายโอนด้วย MobileNetV2 ที่มีค่าความถูกต้องเท่ากับ 0.89

คำสำคัญ **–** การจำแนกยางที่ดีและยางที่ไม่ดี, โครงข่ายประสาทคอนโวลูชัน, การเรียนรู้แบบถ่ายโอน

ABSTRACT

This research presents the application of machine learning techniques in classifying good and defective tires, aiming to support tire quality screening. The study experiments with two algorithms to compare their performance: a deep learning algorithm using Convolutional Neural Networks (CNN) and a transfer learning algorithm. The dataset consists of 2,000 images, with 1,000 images for each class (good and defective tires). The data is split into a training set (70%) and a testing set (30%) to evaluate the classification performance of both CNN-based deep learning and transfer learning algorithms. The experimental results indicate that the CNN algorithm achieved an accuracy of 0.56, which is lower than the transfer learning algorithm's accuracy of 0.89.

Keywords **-** Classifying Good Tire and Defective Tire, Convolutional Neural Networks, Deep Learning

1. บทนำ

ในปัจจุบัน การควบคุมคุณภาพถือเป็นสิ่งสำคัญในอุตสาหกรรมการผลิต โดยเฉพาะอย่างยิ่งในการผลิตชิ้นส่วนยานยนต์ที่ต้องการมาตรฐานสูงเพื่อความปลอดภัยและประสิทธิภาพของการใช้งาน ยางรถยนต์ซึ่งเป็นหนึ่งในชิ้นส่วนหลักของยานพาหนะมีบทบาทสำคัญในการรับประกันความปลอดภัยและความมีประสิทธิภาพในการขับขี่ [1] หากเกิดข้อบกพร่องในยางรถยนต์ อาจส่งผลให้เกิดปัญหาด้านความปลอดภัย ความสูญเสียทางเศรษฐกิจ และอาจทำลายชื่อเสียงของแบรนด์

ดังนั้น การพัฒนาวิธีการตรวจสอบคุณภาพยางที่มีประสิทธิภาพและแม่นยำจึงเป็นสิ่งจำเป็นอย่างยิ่ง การตรวจสอบคุณภาพยางในปัจจุบันมักต้องอาศัยการตรวจสอบด้วยตนเอง ซึ่งใช้เวลานาน มีความซับซ้อน และเสี่ยงต่อความผิดพลาดเนื่องจากความเห็นส่วนบุคคล อย่างไรก็ตาม ด้วยความก้าวหน้าของเทคโนโลยีปัญญาประดิษฐ์ โดยเฉพาะการเรียนรู้ของเครื่อง ทำให้

สามารถนำระบบอัตโนมัติมาช่วยในการวิเคราะห์ข้อมูลภาพในปริมาณมากและจำแนกสินค้าระหว่างสินค้าที่ดีและสินค้าที่มีข้อบกพร่องได้อย่างแม่นยำ

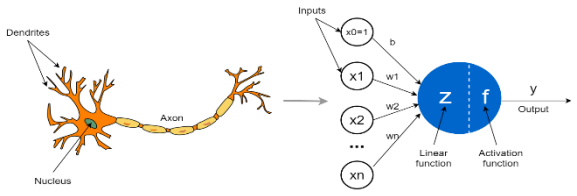
งานวิจัยนี้มุ่งเน้นไปที่การประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่องในการจำแนกคุณภาพยางโดยใช้ข้อมูลภาพ โดยเฉพาะอย่างยิ่ง มุ่งเปรียบเทียบประสิทธิภาพของโครงข่ายประสาทคอนโวลูชัน (Convolutional Neural Network หรือ CNN) และอัลกอริทึมการเรียนรู้แบบถ่ายโอน (Transfer Learning) ซึ่ง CNN เป็นที่นิยมใช้ในการจำแนกภาพเนื่องจากความสามารถในการจับลำดับชั้นของลักษณะเฉพาะในภาพ ขณะที่การเรียนรู้แบบถ่ายโอนช่วยนำความรู้จากโมเดลที่ผ่านการฝึกฝนมาแล้วจากข้อมูลจำนวนมากมาใช้เพื่อเพิ่มความแม่นยำ

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 โครงข่ายประสาทเทียม (Artificial neural networks)

โครงข่ายประสาทเทียม หรือ Artificial Neural Networks (ANN) [2] เป็นโครงสร้างการคำนวณในสาขาของปัญญาประดิษฐ์ (Artificial Intelligence: AI) ที่ได้รับแรงบันดาลใจจากการทำงานของเซลล์ประสาทในสมองมนุษย์ โดย ANN ใช้หลักการเชื่อมโยงและส่งต่อสัญญาณระหว่างหน่วยคำนวณหลาย ๆ หน่วย (เรียกว่า นิวรอน) เพื่อสร้างแบบจำลองที่สามารถประมวลผลข้อมูลที่ซับซ้อนและเรียนรู้จากข้อมูลได้

การทำงานของโครงข่ายประสาทเทียมเริ่มจากการส่งข้อมูลผ่านนิวรอนจากชั้นข้อมูลขาเข้า ข้อมูลจะถูกประมวลผลผ่านชั้นซ่อน โดยนิวรอนแต่ละตัวจะคำนวณผลลัพธ์จากการคูณระหว่างข้อมูลและน้ำหนัก จากนั้นผ่านกระบวนการแปลง (Activation Function) ซึ่งช่วยให้ระบบสามารถเรียนรู้ความสัมพันธ์ที่ซับซ้อนขึ้นได้ จากนั้นผลลัพธ์สุดท้ายจะถูกส่งไปยังชั้นข้อมูลขาออกเพื่อแสดงคำตอบ

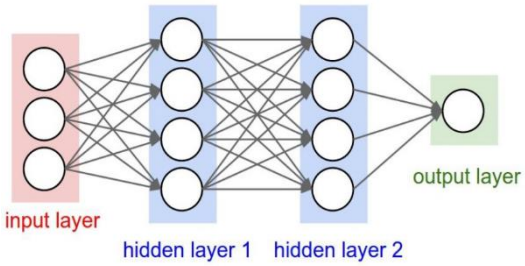


ภาพ 1 โครงสร้าง โครงข่ายประสาทเทียม [3]

2.2 การเรียนรู้เชิงลึก (Deep Learning)

Deep Learning หรือการเรียนรู้เชิงลึก [4] เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence: AI) และเป็นรูปแบบย่อยของการเรียนรู้ของเครื่อง (Machine Learning: ML) ที่มีความสามารถในการเรียนรู้จากข้อมูลจำนวนมาก โดยการทำงานของ Deep Learning มีพื้นฐานอยู่บนโครงข่ายประสาทเทียมหลายชั้น (Artificial Neural Networks) ซึ่งถูกออกแบบมาให้เลียนแบบการทำงานของสมองมนุษย์ โดยเฉพาะการรับรู้และประมวลผลข้อมูลที่ซับซ้อน เช่น การรู้จำภาพและเสียง การประมวลผลภาษา การทำนาย และการวิเคราะห์ข้อมูล

หลักการทำงานของ Deep Learning คือการสร้างโครงข่ายประสาทเทียมหลายชั้น (Deep Neural Network) โดยแต่ละชั้นจะทำหน้าที่ดึงคุณลักษณะเฉพาะจากข้อมูลที่ซับซ้อน โดยทั่วไปแล้ว โครงข่ายจะประกอบด้วยชั้นข้อมูลขาเข้า (Input Layer) ชั้นซ่อนหลายชั้น (Hidden Layers) และชั้นข้อมูลขาออก (Output Layer) ซึ่งช่วยให้ระบบสามารถเรียนรู้จากลักษณะเฉพาะที่ซับซ้อนมากขึ้นเรื่อย ๆ เมื่อผ่านแต่ละชั้น ทำให้สามารถสร้างโมเดลที่มีความแม่นยำสูงได้ในหลากหลายแอปพลิเคชัน เช่น การแปลภาษา การรู้จำใบหน้า และระบบแนะนำข้อมูล



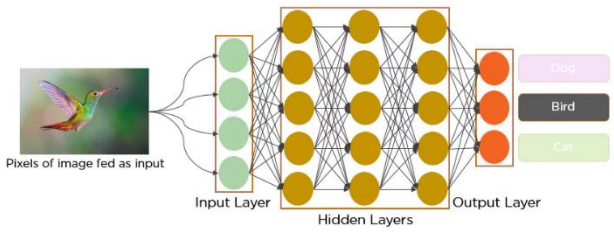
ภาพ 2 โครงสร้าง Deep Learning [5]

2.3 โครงข่ายประสาทคอนโวลูชัน (Convolutional neural networks algorithm : CNN)

โครงข่ายประสาทเทียมแบบคอนโวลูชัน [6] คือสถาปัตยกรรมการเรียนรู้เชิงลึก (Deep Learning) ที่ออกแบบมาเพื่อประมวลผลข้อมูลเชิงพื้นที่ เช่น ภาพและวิดีโอ CNN เป็นเทคนิคหนึ่งในสาขาปัญญาประดิษฐ์ที่มีประสิทธิภาพสูงในงานจำแนกประเภทข้อมูล การตรวจจับวัตถุ และการประมวลผล

ภาพ เนื่องจากสามารถดึงลักษณะเฉพาะของภาพที่มีความซับซ้อนออกมาได้อย่างแม่นยำ

สำหรับโครงข่ายประสาทคอนโวลูชัน มีโครงสร้างหลักที่ประกอบด้วยหลายชั้น ซึ่งแต่ละชั้นจะทำหน้าที่เฉพาะเพื่อสกัดคุณลักษณะและวิเคราะห์ข้อมูลที่ซับซ้อน 1) Convolution Layer เป็นชั้นที่มีการใช้การคอนโวลูชันเพื่อสกัดคุณลักษณะสำคัญของข้อมูลภาพ เช่น รูปร่าง ขอบ และลวดลาย โดยใช้ตัวกรอง (Filter หรือ Kernel) เลื่อนผ่านข้อมูลและคำนวณค่าเชิงพื้นที่ออกมา การคอนโวลูชันนี้ช่วยให้โมเดลสามารถระบุคุณลักษณะสำคัญที่เกี่ยวข้องกับการจำแนกประเภทได้อย่างมีประสิทธิภาพ 2) Pooling Layer เป็นชั้นที่ทำการลดขนาดของข้อมูลภาพโดยการเลือกค่าที่มีความสำคัญที่สุดจากกลุ่มข้อมูลย่อยในภาพ เช่น การทำ Max Pooling ซึ่งจะเลือกค่าแม็กซิมัมในพื้นที่ย่อยของภาพ ช่วยลดจำนวนพารามิเตอร์ ลดการใช้พลังงานการประมวลผล และเพิ่มความทนทานต่อการเปลี่ยนแปลงของภาพบางประการ 3) Fully Connected Layer เป็นชั้นที่เชื่อมต่อโหนดทุกโหนดในชั้นก่อนหน้าเข้ากับโหนดทั้งหมดในชั้นถัดไป ซึ่งคล้ายกับโครงสร้างของโครงข่ายประสาทเทียมแบบดั้งเดิม (Traditional Neural Networks) ชั้นนี้มักใช้ในการทำนายหรือจัดประเภทข้อมูลตามคุณลักษณะที่สกัดได้จากชั้นก่อนหน้า



ภาพ 3 โครงสร้าง โครงข่ายประสาทคอนโวลูชัน [7]

ผู้ใช้ควรมีพื้นฐานเกี่ยวกับการเรียนรู้เชิงลึก (Deep Learning) และเข้าใจหลักการทำงานของโครงข่ายประสาทเทียม (Neural Networks) โดยเฉพาะ Convolutional Neural Network ว่าทำงานอย่างไรในด้านการประมวลผลภาพและการสกัดคุณลักษณะ เพื่อให้สามารถปรับโครงสร้างโมเดลให้เหมาะสมกับงานที่ต้องการได้

2.4 การเรียนรู้แบบถ่ายโอน (Transfer Learning : TL)

Transfer Learning หรือการเรียนรู้แบบถ่ายโอน [8] เป็นเทคนิคหนึ่งในสาขาของ Machine Learning และ Deep Learning ที่มีบทบาทสำคัญในงานวิจัยด้านปัญญาประดิษฐ์ โดยการเรียนรู้แบบถ่ายโอนมุ่งเน้นการนำความรู้จากงานหนึ่งที่มีโครงสร้างหรือข้อมูลคล้ายคลึงกันไปประยุกต์ใช้กับอีกงานหนึ่ง ซึ่งช่วยลดเวลาและทรัพยากรในการฝึกฝนโมเดลใหม่ได้อย่างมีประสิทธิภาพ แนวคิดนี้มีความสำคัญอย่างยิ่งในงานวิจัยที่ข้อมูลมีจำนวนจำกัดหรือหาได้ยาก เช่น การวิเคราะห์ข้อมูลการแพทย์และการจำแนกภาพในอุตสาหกรรมเฉพาะทาง

หลักการของ Transfer Learning เริ่มจากการสร้างโมเดลพื้นฐาน (Base Model) ที่ได้รับการฝึกฝนบนชุดข้อมูลขนาดใหญ่ในงานประเภทหนึ่ง จากนั้นนำโมเดลที่ผ่านการฝึกฝนนี้ไปใช้กับงานใหม่ โดยทั่วไปแล้ว นักวิจัยจะปรับเฉพาะชั้นท้าย ๆ ของโมเดลหรือฝึกซ้ำบางชั้นเท่านั้นเพื่อให้โมเดลสามารถปรับตัวให้เหมาะสมกับข้อมูลเฉพาะที่ใช้งานในงานใหม่ เช่น การวิเคราะห์ภาพทางการแพทย์ การรู้จำวัตถุที่เฉพาะเจาะจง หรือการวิเคราะห์ข้อมูลทางการเงิน วิธีการนี้ช่วยลดความซับซ้อนและค่าใช้จ่ายในการฝึกโมเดลลงได้อย่างมีประสิทธิภาพ เนื่องจากโมเดลได้นำความรู้ที่มีอยู่แล้วจากงานหนึ่งมาปรับใช้ในงานใหม่

2.5 เทนเซอร์โฟร์ (Tensorflow)

TensorFlow [9] เป็น Framework แบบโอเพ่นซอร์สที่พัฒนาโดย Google เพื่อใช้ในการสร้างและฝึกโมเดลปัญญาประดิษฐ์ (AI) โดยเฉพาะในสาขา Machine Learning และ Deep Learning ซึ่งช่วยให้นักวิจัยและนักพัฒนาเขียนโปรแกรมสร้างโมเดลทางคณิตศาสตร์และโครงข่ายประสาทเทียมได้อย่างมีประสิทธิภาพ TensorFlow ถูกออกแบบมาให้สามารถจัดการข้อมูลและคำนวณอย่างรวดเร็วด้วยการใช้หน่วยประมวลผลกลาง (CPU) และหน่วยประมวลผลกราฟิก (GPU) ซึ่งช่วยเพิ่มความเร็วในการฝึกฝนโมเดลได้เป็นอย่างดี รวมถึงยังรองรับการทำงานบนอุปกรณ์หลากหลาย ตั้งแต่คอมพิวเตอร์ส่วนบุคคล เซิร์ฟเวอร์ ไปจนถึงอุปกรณ์พกพา นอกจากนี้ TensorFlow ยังสามารถทำงานในสภาพแวดล้อมแบบกระจาย (Distributed Environment) ซึ่งช่วยให้สามารถจัดการข้อมูลขนาดใหญ่และฝึกโมเดลที่ซับซ้อนได้

2.6 Google Colab

Colab หรือ Google Colab (Google Colaboratory) [10] คือเครื่องมือออนไลน์จาก Google ที่ให้บริการโน้ตบุ๊ค Jupyter แบบคลาวด์ ซึ่งเราสามารถเขียนและรันโค้ด Python ได้ โดยเฉพาะการใช้งานในด้านการวิเคราะห์ข้อมูลและการพัฒนาโมเดล Machine Learning และ Deep Learning

2.7 MobileNetV2

MobileNetV2 [11] เป็นสถาปัตยกรรมของโครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network) ที่ออกแบบมาโดย Google เพื่อลดความซับซ้อนของการคำนวณและใช้ทรัพยากรที่น้อยลงในการประมวลผล แต่ยังคงรักษาความแม่นยำในการจำแนกรูปภาพให้สูง โดยมีการพัฒนาเพิ่มเติมจาก MobileNet รุ่นแรกให้มีประสิทธิภาพมากขึ้นในงานการจำแนกภาพ (Image Classification) และงานอื่น ๆ ที่เกี่ยวข้องกับวิชันคอมพิวเตอร์ (Computer Vision)

MobileNetV2 ใช้เทคนิคสำคัญที่เรียกว่า Inverted Residuals และ Linear Bottlenecks ซึ่งช่วยให้โมเดลสามารถเรียนรู้คุณลักษณะที่มีความละเอียดสูงในเลเยอร์ที่ลึกได้โดยไม่เพิ่มจำนวนพารามิเตอร์มากเกินไป โดยโครงสร้างของโมเดลมีการใช้ convolution แบบพิเศษที่เรียกว่า depthwise separable convolution เพื่อลดความซับซ้อนของการคำนวณลงเมื่อเทียบกับ convolution แบบทั่วไป

2.8 งานวิจัยที่เกี่ยวข้อง

เทคนิคการเรียนรู้เชิงลึกได้นำไปประยุกต์ใช้กับงานหลายๆ งาน เช่น งานวิจัยในหัวข้อเรื่อง Tyre Quality Classification เขียนโดย mateuszk ได้พัฒนาระบบการแยกยางที่ดีกับยางที่เสีย ด้วย Transfer Learning ตัวโมเดล EfficientNetB4 โดยใช้รูปภาพทั้งหมด 1,856 รูปและแบ่งเป็น Train set 1,624 รูป และ Test set 232 รูป จำนวนคลาส 2คลาส ซึ่งให้ผลค่าความถูกต้องแม่นยำอยู่ในระดับดีมาก คิดเป็นร้อยละ 95.69% [12]

งานวิจัยในหัวข้อเรื่อง Tyre Defect Classification เขียนโดย satyaprakashshukl ได้พัฒนาระบบการแยกยางที่ดีกับยางที่เสีย ด้วย CNN โดยใช้รูปภาพทั้งหมด 1,856 รูปและแบ่งเป็น Train set 1,486 รูป และ Test set 370 รูป จำนวนคลาส 2 คลาส ซึ่งให้ผลค่าความถูกต้องแม่นยำอยู่ในระดับกลาง คิดเป็นร้อยละ 69.73% [13]

งานวิจัยในหัวข้อเรื่อง Tyre quality classification เขียนโดย stepan konecy ได้พัฒนาระบบการแยกยางที่ดีกับยางที่เสียด้วย Transfer Learning ตัวโมเดล Resnet50 โดยใช้รูปภาพทั้งหมด 1,856 รูปและแบ่งเป็น Train set 1,485 รูป และ Test set 371 รูป จำนวนคลาส 2คลาส ซึ่งให้ผลค่าความถูกต้องแม่นยำอยู่ในระดับกลาง คิดเป็นร้อยละ 88.67% [14]

3. วิธีดำเนินการศึกษา

ในการพัฒนาโมเดลเพื่อจำแนกรูปภาพยางที่ดีกับยางที่ไม่ดีด้วยอัลกอริทึมการเรียนรู้แบบถ่ายโอนและโครงข่ายประสาทคอนโวลูชันมีการดำเนินงาน 4 ขั้นตอนด้วยกัน ได้แก่ 1) การรวบรวมข้อมูล 2) การสร้างโมเดลจำแนกภาพ 3) การวัดประสิทธิภาพโมเดล 4) การพัฒนาเว็บไซต์

3.1 การรวบรวมและเตรียมข้อมูล

จะใช้รูปภาพยางที่ดีและยางที่ไม่ดี โดยนำรูปภาพมาจากเว็บ Kaggle และที่ Roboflow Universe อย่างละ 1,000 ภาพ รวมทั้งหมดจำนวน 2,000 ภาพ แบ่งข้อมลภาพออกเป็น 2 คลาส ได้แก่ ยางที่ดีและยางที่ไม่ดี คลาสละ 1,000 รูปภาพ และโดยกำหนดขนาดรูปภาพเท่ากับ 180x180 แบ่งข้อมูลภาพออกเป็น 70:30 จำนวน 1,400 เป็นชุดสำหรับการฝึกสอน (Traning Set) และจำนวน 600 เป็นภาพชุดทดสอบ (Test set) แสดงดังตารางที่ 1

ตาราง 1 จำนวนรูปภาพที่นำมาใช้ในการฝึกสอนและทดสอบ

ชนิดของภาพ	จำนวนภาพ	ชุดข้อมูลฝึกสอน	ชุดข้อมูลทดสอบ
	1,000	700	300
	1,000	700	300
รวม	2,000	1,400	600

3.2 การสร้างโมเดลจำแนกภาพ

การจำแนกภาพโดยการสร้างโมเดลคือการนำข้อมูลรูปภาพที่ได้มาทำการฝึกและสร้างโมเดลเพื่อทำนายผลด้วยภาษา Python และไลบรารี่ที่ชื่อ TensorFlow ซึ่งเป็นไลบรารี่ที่ใช้สำหรับฝึกสอนและสร้างโมเดลโดยใช้ CNN และ TL โดยใช้ตัว MobileNetV2 เป็น Base Model โดยโครงสร้างของ CNN จะมี

conv2d: ชั้นคอนโวลูชัน 32 ฟิลเตอร์

max_pooling2d: ลดขนาดข้อมูล

conv2d_1: ชั้นคอนโวลูชัน 64 ฟิลเตอร์

max_pooling2d_1: ลดขนาดข้อมูล

flatten: แปลงข้อมูลเป็นเวกเตอร์ 1 มิติ

dense: Fully Connected 256 หน่วย

dropout: ลดการ Overfitting

dense_1: ชั้นสุดท้าย ให้ผลลัพธ์ 1 หน่วย

และโครงสร้างของ TL จะมีดังนี้

input_layer_2: รับข้อมูลอินพุตขนาด 180x180x3 (ภาพสี RGB)

sequential และ true_divide: ปรับข้อมูลให้อยู่ในช่วงที่เหมาะสม

subtract: ดึงค่าพิกเซลจากการทำงานของลำดับชั้นก่อนหน้า

mobilenetv2_1.00_224: โมเดล MobileNetV2 ช่วยสกัดฟีเจอร์จากภาพขนาดเอาต์พุต 6x6x1280

global_average_pooling2d: ลดขนาดข้อมูลเหลือเวกเตอร์ 1280 หน่วย

dropout: ลดการ Overfitting

dense: ชั้นสุดท้าย ให้ผลลัพธ์เป็น 1 หน่วย

โดยทำการฝึกจำนวน 20 รอบดังภาพที่ 1 กับ CNN โดยทำการฝึกจำนวน 20 รอบเท่ากัน ดังภาพที่ 2 และเมื่อฝึกสอนเสร็จจะได้ผลการสร้างโมเดลคือ tire1s.weights.h5 และ model_tune.json

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 178, 178, 32)	896
max_pooling2d (MaxPooling2D)	(None, 89, 89, 32)	0
conv2d_1 (Conv2D)	(None, 87, 87, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 43, 43, 64)	0
flatten (Flatten)	(None, 118336)	0
dense (Dense)	(None, 256)	30,294,272
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

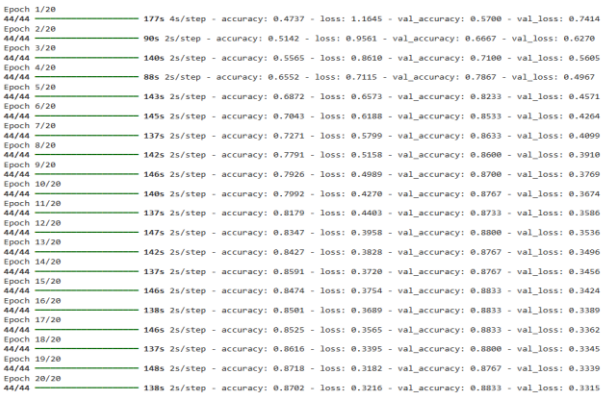
Total params: 30,313,921 (115.64 MB)
Trainable params: 30,313,921 (115.64 MB)
Non-trainable params: 0 (0.00 B)

ภาพที่ 1 โครงสร้างของ CNN

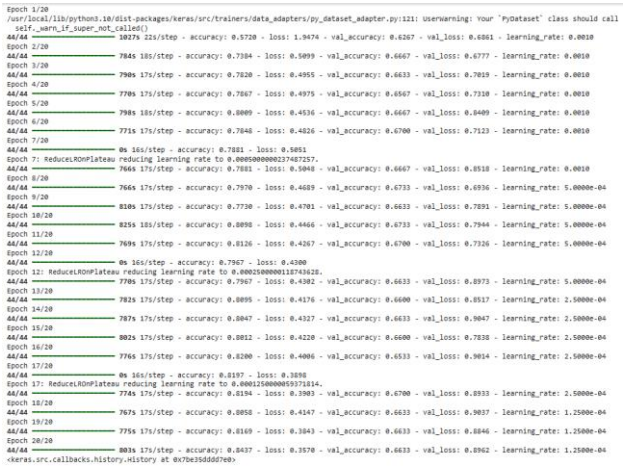
Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 180, 180, 3)	0
sequential (Sequential)	(None, 180, 180, 3)	0
true_divide (TrueDivide)	(None, 180, 180, 3)	0
subtract (Subtract)	(None, 180, 180, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 6, 6, 1280)	2,257,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 1)	1,281

Total params: 2,259,265 (8.62 MB)
Trainable params: 1,281 (5.00 KB)
Non-trainable params: 2,257,984 (8.61 MB)

ภาพที่ 2 โครงสร้างของ TL



ภาพที่ 3 ตัวอย่างค่าประสิทธิภาพในแต่ละรอบของการฝึกสอนโมเดลจำนวน 20 epoch ของ Transfer Learning



ภาพที่ 4 ตัวอย่างค่าประสิทธิภาพในแต่ละรอบของการฝึกสอนโมเดลจำนวน 20 epoch ของ CNN

3.3 การวัดประสิทธิภาพโมเดล

ในงานวิจัยนี้ใช้มาตรวัดประสิทธิภาพทั้งหมด 4 ตัวคือ ค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่าความถ่วงดุล (F1-score) ซึ่งแต่ละค่ามี

วิธีการคำนวณดังสมการที่ (1) ถึงสมการที่ (4) ตามลำดับ

$$precision = \frac{TPs}{TPs + FPs} \quad (1)$$

$$recall = \frac{TPs}{TPs + FNs} \quad (2)$$

$$accuracy = \frac{TPs + TNs}{TPs + TNs + FPs + FNs} \quad (3)$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

โดยค่า True Positive (TPs) คือ จำนวนภาพที่ทายถูกต้องและคำเฉลยเป็นภาพที่ถูกต้อง

True Negative (TNs) คือ จำนวนภาพที่ทายไม่ถูกต้องและคำเฉลยเป็นภาพที่ไม่ถูกต้อง

False Positive (FPs) คือ จำนวนภาพที่ทายถูกต้องและคำเฉลยเป็นภาพที่ไม่ถูกต้อง

False Negative (FNs) คือ จำนวนภาพที่ทายไม่ถูกต้องและคำเฉลยเป็นภาพที่ไม่ถูกต้อง

3.4 การพัฒนาเว็บไซต์

ในการศึกษาผู้วิจัยดำเนินการพัฒนาเว็บไซต์ด้วยภาษา Python เพื่อเรียกใช้โมเดลชื่อ ไฟล์ model-1.h5 และพัฒนาเว็บไซต์ด้วยเฟรมเวิร์ค Flask

การเรียกใช้งานจะเรียกใช้ในหน้า app.py เพื่อทำการรับรูปภาพที่ต้องการจำแนกในฟังก์ชัน get_output() ดังภาพที่ 2 และนำภาพที่อัปโหลดส่งไปใช้ฟังก์ชัน predict_label เพื่อทำการเปลี่ยนขนาดรูปภาพที่อัปโหลดเข้ามาให้เท่ากับ 120x120 พิกเซล จากนั้นทำการ prediction รูปภาพแล้วส่งข้อมูลกลับมาแสดงผล ดังภาพที่ 3

```
def get_output():
  if request.method == 'POST':
    img = request.files['my_image']
    img_path = "static/" + img.filename
    img.save(img_path)
    p1 = predict_label(img_path)
  return render_template("index.html", prediction = p1[0], TEXT = p1[1], img_path = img_path)
```

ภาพที่ 3 ตัวอย่างโปรแกรมการรับค่าภาพที่ต้องการจำแนก

```
# Prediction function
def predict_label(img_path):
    test_image = load_img(img_path, target_size=(180, 180))  # Adjust if base model requires a different size
    test_image = img_to_array(test_image)
    test_image = preprocess_input(test_image)  # Preprocess for transfer learning
    test_image = np.expand_dims(test_image, axis=0)

    result_value = model.predict(test_image)
    confidence = round(result_value[0][0] * 100, 2) if result_value[0][0] > 0.5 else round((1 - result_value[0][0]) * 100, 2)
    prediction = 'Good' if result_value[0][0] > 0.5 else 'Defective'

    return confidence, prediction
```

ภาพที่ 4 ตัวอย่างโปรแกรมที่นำภาพมาจำแนกหาคำตอบ

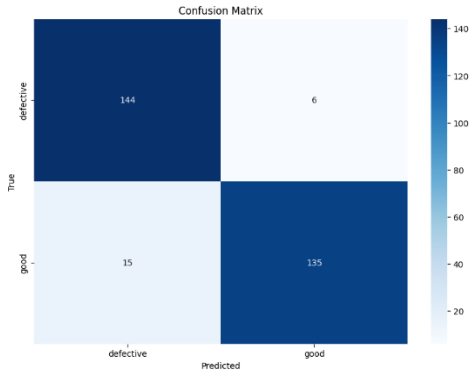
4. ผลการศึกษาและการอภิปรายผล

ในส่วนของผลการศึกษา ผู้วิจัยแบ่งผลการศึกษาเป็น 2 ส่วน คือ ส่วนของการเปรียบเทียบประสิทธิภาพของทั้งสองอัลกอริทึม และส่วนของการนำโมเดลไปพัฒนาเว็บแอพพลิเคชัน ซึ่งมีรายละเอียดดังนี้

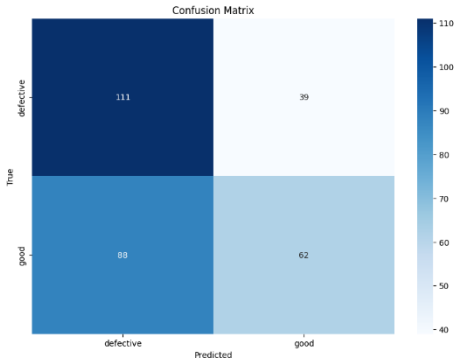
4.1 ผลการวัดประสิทธิภาพโมเดล

การศึกษานี้ได้ทดลองเปรียบเทียบการสร้างโมเดลการเรียนรู้จดจำรูปภาพยางดีและยางที่ไม่ดีด้วยเทคนิคการเรียนรู้ของเครื่อง ซึ่งผู้วิจัยได้ทำการทดลองกับอัลกอริทึมจำนวน 2 อัลกอริทึม ประกอบไปด้วย อัลกอริทึมโครงข่ายประสาทคอนโวลูชัน (CNN) และการเรียนรู้แบบถ่ายโอน (TL) เพื่อจำแนกคำตอบเป็น 2 คลาส คือ Good กับ Defective จำแนกรูปภาพจำนวนทั้งสิ้น 2,000 ภาพ โดยผู้วิจัยได้แบ่งชุดข้อมูลออกเป็น 2 ชุด คือ ชุดข้อมูลฝึก จำนวนร้อยละ 70 และชุดข้อมูลทดสอบจำนวนร้อยละ 30 จากข้อมูลทั้งหมด และวัดประสิทธิภาพด้วยมาตรวัดจำนวน 4 มาตรวัด หลังจากนั้นจะนำผลของการจำแนกของทั้งสองอัลกอริทึมมาเปรียบเทียบประสิทธิภาพ และคัดเลือกอัลกอริทึมที่ให้ค่าประสิทธิภาพที่สูงที่สุดนำไปพัฒนาแอพพลิเคชันต่อไป ผลการเปรียบเทียบแสดงในตารางที่ 2

จากตารางที่ 2 ผลการทดลองแสดงให้เห็นว่าอัลกอริทึม TL สามารถจำแนกได้แม่นยำกว่าอัลกอริทึม CNN ในทุกมาตรวัดประสิทธิภาพ โดยให้ค่าความแม่นยำ (Precision) เท่ากับ 0.91 ค่าความระลึก (Recall) ค่าความถ่วงดุล (F1-score) และค่าความถูกต้อง (Accuracy) เท่ากับ 0.89 ดังนั้นอัลกอริทึม CNN จึงถูกนำมาใช้ในการพัฒนาต่อยอดสำหรับการจำแนกบนเว็บแอพพลิเคชันต่อไป



ภาพที่ 5 แสดง Confusion Matrix ของ TL



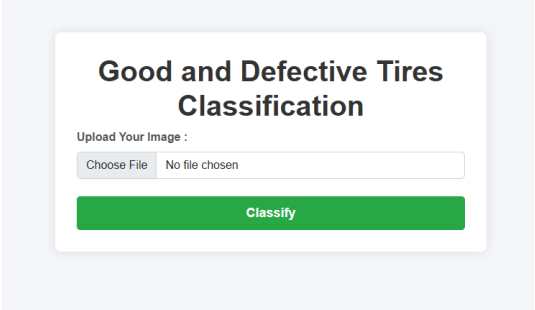
ภาพที่ 6 แสดง Confusion Matrix ของ CNN

ตาราง 2 ประสิทธิภาพของอัลกอริทึม TL และ CNN

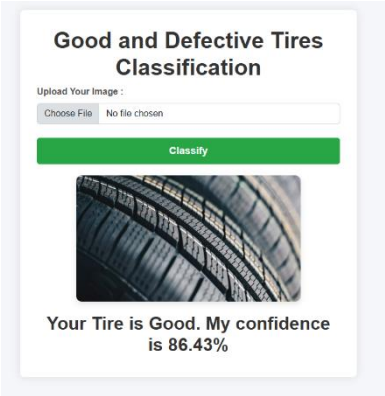
Algorithms	TL	CNN
Precision	**0.90**	0.59
Recall	**0.88**	0.58
F1-Score	**0.88**	0.58
Accuracy	**0.88**	0.56

4.2 ผลการพัฒนาเว็บไซต์

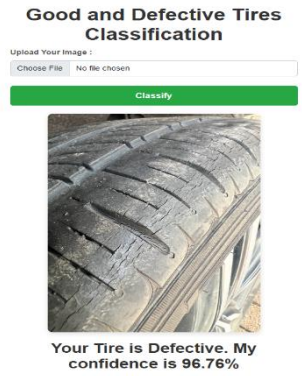
ผลลัพธ์ของการพัฒนาเว็บไซต์จำแนกคนสวมแมสและคนไม่สวมแมส มีรายละเอียดดังภาพ 4 ซึ่งแสดงหน้าเว็บไซต์ ผู้ใช้งานสามารถเลือกอัปโหลดรูปภาพคนที่สวมแมสและไม่สวมแมสเพื่อให้โมเดลทำการจำแนกหาคำตอบ โดยหากตัวโมเดลเรียนรู้แล้วจำแนกออกมาเป็นคนสวมแมส จะแสดงตัวอย่างดังภาพที่ 5 และหากไม่สวมแมสจะแสดงตัวอย่างดังภาพที่ 6



ภาพที่ 7 ตัวอย่างหน้าเว็บไซต์สำหรับการอัพโหลดรูปภาพ



ภาพที่ 8 ตัวอย่างหน้าเว็บไซต์แสดงผลการจำแนกจากโมเดลเป็นยางดี



ภาพที่ 9 ตัวอย่างหน้าเว็บไซต์แสดงผลการจำแนกจากโมเดลเป็นยางไม่ดี

5. บทสรุป

งานวิจัยนี้นำเสนอการจำแนกรูปภาพยางที่ดีกับยางที่ไม่ดีด้วยเทคนิคการเรียนรู้ของเครื่อง การดำเนินงานเริ่มจากการรวบรวมชุดข้อมูลหรือรูปภาพยางที่ดีและยางที่ไม่ดีจำนวน 2,000

จากภาพจากเว็บไซต์ Kaggle และแหล่งอื่นๆ รวมทั้งหมดจำนวน 2,000 ภาพ แบ่งข้อมลภาพออกเป็น 2 คลาส ได้แก่ ยางที่ดีและยางที่ไม่ดี คลาสละ 1,000 รูปภาพ การทดลองกับอัลกอริทึมจำนวน 2 อัลกอริทึม ประกอบไปด้วย อัลกอริทึมโครงข่ายประสาทคอนโวลูชัน (CNN) และการเรียนรู้แบบถ่ายโอน (TL) เพื่อจำแนกคำตอบเป็น 2 คลาส คือ Good กับ Defective หลังจากนั้นนำโมเดลให้ค่าประสิทธิภาพที่ดีที่สุดไปพัฒนาต่อเป็นเว็บแอพพลิเคชันที่พัฒนาด้วยภาษา python

ผลการทดสอบประสิทธิภาพการจำแนกยางที่ดีและยางที่ไม่ดี พบว่าการเรียนรู้แบบถ่ายโอนสามารถจำแนกภาพได้ดีกว่าอัลกอริทึมโครงข่ายประสาทคอนโวลูชั่นโดยดูได้จากค่าประสิทธิภาพทั้ง 4 ค่า ซึ่งค่าความแม่นยำ (Precision) ของ TL มีค่าเท่ากับ 0.90 ในส่วนของค่าความระลึก (Recall) ค่าความถ่วงดุล (F1-score) และค่าความถูกต้อง (Accuracy) ของ TL มีค่าเท่ากับ 0.88 เท่ากันทั้ง 3 ค่า สาเหตุอาจเนื่องมาจากการจัดเตรียมชุดข้อมูลรูปภาพสำหรับการฝึกสอนอาจจะยังไม่ครอบคลุมเพราะในบางกรณีภาพที่เราเตรียมบางภาพก็ถ่ายใกล้หรือไกลมากเกินไปจึงทำให้เกิดความคลาดเคลื่อนได้

ซึ่งเว็บแอพพลิเคชันนี้สามารถนำไปใช้ในการจำแนกยางที่ดีและยางที่ไม่ดีได้ในระดับหนึ่ง และถ้าหากได้รับการพัฒนาและปรับปรุงคาดว่าภายในอนาคตจะทำงานได้อย่างมีประสิทธิภาพมากขึ้น สำหรับข้อเสนอแนะเพิ่มเติมในการพัฒนาต่อคือการเพิ่มอัลกอริทึมในการจำแนกเพื่อให้ค่าประสิทธิภาพที่ดีขึ้น และนำโมเดลที่ได้ไปพัฒนาเพื่อใช้กับภาพวิดีโอเพื่อให้สามารถใช้งานได้จริงในอนาคต

เอกสารอ้างอิง

[1] The Importance of Quality Control in the Automotive Industry, [ออนไลน์] 2563. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://manufacturingdigital.com/smart-manufacturing/importance-quality-control-automotive-industry

[2] Neural network (machine learning), [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://en.wikipedia.org/wiki/Neural_network_(machine_learning)

[3] The Concept of Artificial Neurons (Perceptrons) in Neural Networks, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc

[4] What is Deep Learning?, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://aws.amazon.com/th/what-is/deep-learning/

[5] Biological motivation and connections, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://cs231n.github.io/neural-networks-1/

[6] Convolutional neural network, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://en.wikipedia.org/wiki/Convolutional_neural_network

[7] Introduction to Convolutional Neural Networks (CNN), [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

[8] Transfer Learning, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://en.wikipedia.org/wiki/Transfer_learning

[9] TensorFlow คืออะไร?, [ออนไลน์] 2566. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://thaiconfig.com/artificial-intelligence-ai/what-is-tensorflow/

[10] Colab คืออะไร เริ่มต้นเรียนรู้ เขียนโปรแกรม AI, Machine Learning โดยไม่ต้องลงโปรแกรม สอนวิธีเปิด Jupyter Notebook ที่อยู่ใน GitHub บน Google Colab – Colab ep.1, [ออนไลน์] 2562. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://www.bualabs.com/archives/1687/what-is-colab-open-jupyter-notebook-in-github-on-google-colab-create-open-in-colab-button-colab-ep-1/

[11] What is MobileNetV2? Features, Architecture, Application and More, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก

https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2/
[12] Tyre Quality - 90% ACC with Custom SE-ResNet, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://www.kaggle.com/code/mateuszk013/tyre-quality-90-acc-with-custom-se-resnet/notebook#-4-%7C-Transfer-Learning-with-EfficientNetB4-%E2%86%91
[13] Tyre Defect Classification, [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://www.kaggle.com/code/satyaprakashshukl/tyre-defect-classification
[14] Tyre quality classification [ออนไลน์] 2567. สืบค้นเมื่อวันที่ 30 ตุลาคม 2567. จาก https://www.kaggle.com/code/dolenv0/tyre-quality-classification#ResNet50