การจำแนกความเสียหายรถยนต์โดยการเรียนรู้เชิงลึก

ธนัช เบญจอนุอาชา[1*], วราภรณ์ วิยานนท์[2]

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาการจำแนกรูปภาพรถยนต์ที่มีความเสียหายและไม่มีความเสียหาย โดยใช้เทคนิคการเรียนรู้เชิงลึก เพื่อพัฒนาแบบจำลองการจำแนกความเสียหายรถยนต์โดยใช้โครงข่ายประสาทเทียมแบบสังวัฒนาการ หรือ Convolutional Neural Network (CNN) ได้แก่ VGG16, ResNet50 และ InceptionV3 ร่วมกับการใช้เทคนิคการเรียนรู้แบบถ่ายโอนโดยจำแนกออกเป็น 2 ประเภท ได้แก่ มีความเสียหาย และ ไม่มีความเสียหาย โดยได้ใช้ชุดข้อมูลจากเว็บไซต์ Kaggle [5] เปรียบเทียบประสิทธิภาพโมเดลซึ่งผู้วิจัยได้ทำการปรับพารามิเตอร์ต่างๆเพื่อให้เหมาะสมก็ชุดข้อมูลมีประสิทธิภาพดีที่สุด ซึ่งผลลัพธ์ที่ได้คือโมเดล VGG16 มีประสิทธิภาพมากที่สุด วัดค่าความแม่นยำ (accuracy) เท่ากับ 0.83 ตามมาด้วยโมเดล InceptionV3 เท่ากับ 0.81 และ ResNet50 เท่ากับ 0.68

คำสำคัญ : การเรียนรู้เชิงลึก, โครงสร้างประสาทเทียมแบบคอนโวลูชัน, การเรียนรู้แบบถ่ายโอน, การจำแนกรูปภาพ

---

[1] หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาการข้อมูล คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ กรุงเทพฯ 10110

[2] คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ กรุงเทพฯ 10110

* Corresponding author: Tel.: 062-4912826 E-mail address: thanus.ben@g.swu.ac.th

Car Damage Classification using Deep Learning

Thanus Benjaanuarcha*, Waraporn Viyanon

Abstract

The purpose of this research was to study the classification of damaged and undamaged car images using deep learning techniques to develop a car damage classification model using a convolutional neural network (CNN). The CNN used to develop the vehicle damage classification model were VGG16, ResNet50, and InceptionV3 using transfer learning techniques. The classifications fall into two categories: damaged and non-damaged vehicles. Datasets from Kaggle were used to develop and optimize models. This model is optimized for various parameters. for best performance The results showed that the accuracy of the VGG16, InceptionV3, and ResNet50 were 0.83, 0.81 and 0.68, respectively.

Keywords : Deep Learning, Convolution Neural Network, Transfer Learning, Image Classification

---

[1] Data Science, Faculty of Science, Srinakharinwirot University, Bangkok, 10110, Thailand

[2] Faculty of Science, Srinakharinwirot University, Bangkok, 10110, Thailand

* Corresponding author:Tel.: 062-4912826 E-mail address: thanus.ben@g.swu.ac.th

บทนำ

ปัจจุบันในประเทศไทยอุตสาหกรรมประกันภัยเป็นอุตสาหกรรมที่มีการแข่งขันกันสูง โดยปี พ.ศ. 2563 อัตราการเติบโตของเบี้ยประกันภัยอยู่ที่ 3.5% โดยมีเบี้ยรับรวมทั้งสิ้นอยู่ที่ 252,618,165,000 บาท สัดส่วนของประกันภัยรถยนต์ (Motor insurance) อยู่ที่ 57.80% หรือมีเบี้ยรับประมาณ 146,017,083,000 บาท [1] ซึ่งจะเห็นได้ว่าประกันภัยรถยนต์มีความสำคัญอย่างยิ่งในธุรกิจประกันภัย

เนื่องจากประกันภัยรถยนต์มีเบี้ยรับที่มากนั้นบอกถึงจำนวนที่บริษัทประกันภัยต้องรับผิดชอบในกรณีมีเคลมประกันภัยเกิดขึ้นก็มีโอกาสมากขึ้นไปด้วย บริษัทประกันภัยจะสูญเสียเงินในส่วนของเคลมที่มีการจ่ายเคลมเกินกว่ายอดการเคลมที่แท้จริง (Claim Leak) ในปริมาณมากมายมหาศาลและในการแจ้งเคลมในปัจจุบันส่วนใหญ่ลูกค้าต้องแจ้งเคลมประกันภัยโดยโทรศัพท์แจ้งบริษัทประกัน บริษัทประกันภัยจะส่งเจ้าหน้าที่ลงพื้นที่เก็บหลักฐานถ่ายรูปความเสียหายรถยนต์ ดำเนินการเอกสารต่างๆและใช้เวลาในการพิจารณาเคลมซึ่งทำให้เสียเวลาเป็นอย่างมาก ดังนั้น หากมีเครื่องมือมาช่วยในการจำแนกความเสียหายก็จะช่วยลดภาระงานของเจ้าหน้าที่ลงได้ในการลงพื้นที่ งานวิจัยนี้มุ่งเน้นการจำแนกความเสียหายจากรูปภาพเพื่อลดขั้นตอนการการลงพื้นที่และลดกระบวนการอื่นๆที่เกี่ยวข้อง เพื่อให้มีความแม่นยำในการวิเคราะห์จำแนกความเสียหาย บริษัทประกันภัยจึงมองหาการแก้ปัญหาด้วยนำเทคโนโลยี AI จากการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกสามารถช่วยแก้ปัญหาดังกล่าว

จากการศึกษางานวิจัยทีเกี่ยวข้อง มีงานวิจัยเกี่ยวกับรูปภาพรถยนต์ที่มีความเสียหายและน่าสนใจนำมาศึกษา

Patil, Kulkarni, Sriraman, และ Karande, 2017 [2] ในบทความนี้ใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning) ในการจำแนกประเภทความเสียหายจากรูปภาพรถยนต์ (Classifications of car damages) โดยใช้กระบวนการ Convolutional Neural Network (CNN) [4] ด้วยข้อมูลที่มีปริมาณน้อยโดยมีจำนวนรูปภาพจำแนกตามประเภทดังนี้ กันชนบุบ 186 รูป ประตูบุบ 155 รูป กระจกแตก 215 รูป ไฟหน้าแตก 197 รูป ไฟท้ายแตก 79 รูป รอยขีดขวน 186 รูป นรอยชน 182 รูป และไม่มีความเสียหาย 1,271 รูป ทำให้ต้องมีการทำการดัดแปลงข้อมูลต้นฉบับ (data augmentation) เป็นการเพิ่มข้อมูลด้วยการย่อ ขยาย หมุนซ้าย/ขวา Flip ซ้าย/ขวา/บน/ล่าง Crop มุมปรับสีเข้มขึ้น/เข้มลง ปรับแสง สว่าง/มืด ปรับ Contrast ปรับ Perspective เพิ่ม/ลด Noise เบลอภาพ เพื่อเพิ่มปริมาณรูปภาพและเปรียบเทียบประสิทธิภาพได้ว่าการทำการดัดแปลงข้อมูลต้นฉบับ (data augmentation) ได้ประสิทธิภาพที่ดีขึ้นกว่าไม่ได้ทำการดัดแปลงข้อมูลต้นฉบับ (data augmentation) ในการทดสอบกระบวนการ Convolutional Neural Network (CNN) ใช้โครงสร้างต่างๆเช่น Car, Inception, Alexnet, VGG19, VGG16 ,Resnet ผลลัพธ์ของการทดลองวิธีที่ดีที่สุดคือ Resnet ประสิทธิภาพความแม่นยำที่ 88.24% โดยไม่ได้ทำการ data augmentation และได้ประสิทธิภาพความแม่นยำที่ดีที่ 89.53% ด้วยการใช้การรวมวิธีการการถ่ายโอน (transfer) และการเรียนรูปแบบรวม (ensemble learning)

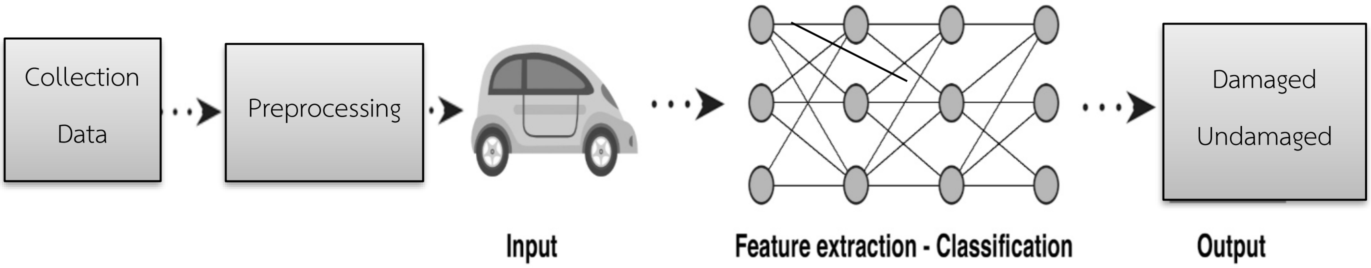
Kyu และ Woraratpanya, 2020 [3] ทำการศึกษาซึ่งในปัจจุบัน อุตสาหกรรมรถยนต์เติบโตซึ่งมีผลโดยตรงเกี่ยวข้องกับจำนวนอุบัติเหตุที่เพิ่มขึ้น ดังนั้นบริษัทประกันต่างๆเผชิญกับการเรียกร้องเคลมและการแก้ไขปัญหาเคลมที่เรียกร้องสูงเกิดความเป็นจริง จึงใช้ปัญญาประดิษฐ์ (AI) แมชชีนเลิร์นนิงและอัลกอริธึมการเรียนรู้เชิงลึกมามาช่วยแก้ปัญหาได้ปัญหาเหล่านี้สำหรับอุตสาหกรรมประกันภัย โดยบทความนี้จะใช้อัลกอริธึมการเรียนรู้เชิงลึก VGG16 และ VGG19 ในการตรวจจับและประเมินความเสียหายรถยนต์ ประเมินตำแหน่ง และความรุนแรง โดยได้พบว่า โมเดล CNN ได้ผ่านการอบรมด้วยชุดข้อมูล ImageNet [9] และตามด้วยการปรับ fine tuning เพื่อเพิ่มประสิทธิภาพ ผลลัพธ์ที่ได้ ความแม่นยำในการจำแนก damaged detection ของ VGG19 อยู่ที่ 95.22% และ VGG16 อยู่ที่ 94.56% ความแม่นยำในการจำแนกตำแหน่ง (ส่วนหน้า, ส่วนข้าง, ส่วนหลัง) VGG19 อยู่ที่

76.48% และVGG16 อยู่ที่ 74.39% และความแม่นยำในความรุนแรง (ความรุนแรงน้อย, ปานกลาง ,มาก) VGG19 อยู่ที่ 58.48% และ VGG16 อยู่ที่ 54.8% จากผลลัพธ์ประสิทธิภาพของ VGG19 นั้นดีกว่า VGG16

ดังนั้นในงานวิจัยนี้ ได้นำงานวิจัยที่เกี่ยวข้องกับรูปภาพมาศึกษาและนำวิธีการมาปรับใช้กับชุดข้อมูลที่ได้นำมาศึกษาเพื่อทดลองและปรับจูนให้โมเดลมีประสิทธิภาพที่ดีที่สุด

วิธีดำเนินการ

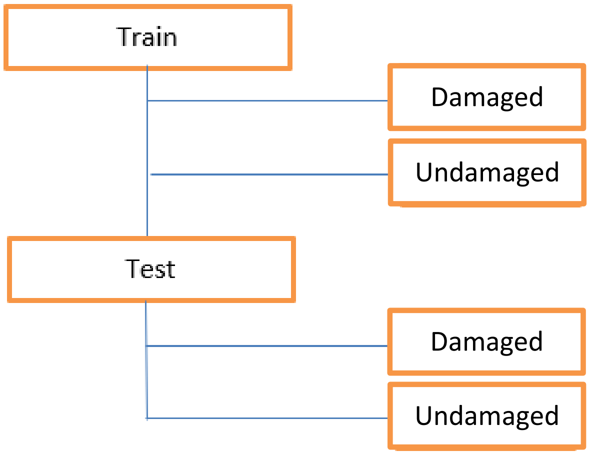
ขั้นตอนวิธีการดำเนินการประกอบไปด้วย Input รูปภาพ กระบวนการเตรียมข้อมูล (Preprocessing) สกัดคุณลักษณะ (feature extraction) จำแนกประเภทรถยนต์เสียหายและไม่เสียหาย (Classification) ตามภาพที่ 1 โดยมีขั้นตอนดังนี้



ภาพที่ 1 ขั้นตอนวิธีดำเนินการ

ขั้นตอนที่ 1 : การรวบรวมข้อมูล (Collection Data)

ผู้วิจัยเก็บรวบรวมข้อมูลจากเว็บไซต์ kaggle [5] เป็นหลัก โดยค้นหารูปภาพรถยนต์ทั้งรูปภาพที่มีความเสียหาย และไม่มีความเสียหาย ทำความสะอาดข้อมูลโดยการลบรูปภาพที่ไม่ชัดเจน ลบรูปภาพที่มีส่วนที่ไม่เกี่ยวข้องออก เช่น คน หรือ วัตถุต่างๆ จากนั้นนำมาสร้าง dataset เป็น train และ test อย่างละโฟลเดอร์ และ ภายในโฟลเดอร์ train จะมีโฟลเดอร์ undamaged ที่มีรูปภาพรถยนต์ที่ไม่มีความเสียหายอยู่จำนวน 920 รูปภาพ และ โฟลเดอร์ damaged รูปภาพรถยนต์ที่มีความเสียหายจำนวน 920 รูปภาพ และในส่วนของข้อมูล test ก็มี 2 โฟลเดอร์ undamaged มีรูปภาพจำนวน 230 รูปภาพ และ โฟลเดอร์ damaged มีรูปภาพจำนวน 230 รูปภาพ เช่นกัน แสดงโฟลเดอร์เก็บข้อมูล ภาพที่ 2 และสรุปจำนวนดังตารางที่ 1



ภาพที่ 2 แสดงโฟลเดอร์เก็บข้อมูล Train และ Test

ตารางที่ 1 จำนวนภาพการฝึก (train) และทดสอบ (test) จำแนกตามคลาส

Class	Training dataset (จำนวนรูป)	Test dataset (จำนวนรูป)
Undamaged	920	230
Damaged	920	230
Total	1,840	460

ขั้นตอนที่ 2 : การเตรียมข้อมูล (Preprocessing)

ขั้นตอนการเตรียมข้อมูลจากที่ได้เก็บข้อมูลและแยกโฟลเดอร์ train และ test แล้วทำการเตรียมข้อมูลโดยโปรแกรมไพทอน ใช้คำสั่ง ImageDataGenerator ในการปรับขนาดรูปภาพ (target size) ให้เป็น 128x128 ในขั้นตอนนี้จะทดสอบเปรียบเทียบประสิทธิภาพการแปลงข้อมูลต้นฉบับ (data augmentation) และไม่แปลงข้อมูลต้นฉบับ (without data augmentation) โดยสำหรับข้อมูลที่ไม่มีการแปลงข้อมูลต้นฉบับหลังจากขั้นตอนปรับขนาดรูปภาพให้เป็นพิกเซลแล้วข้ามขั้นตอนการแปลงข้อมูลต้นฉบับ จะนำข้อมูลไปสกัดคุณลักษณะ (feature extraction) ต่อไป

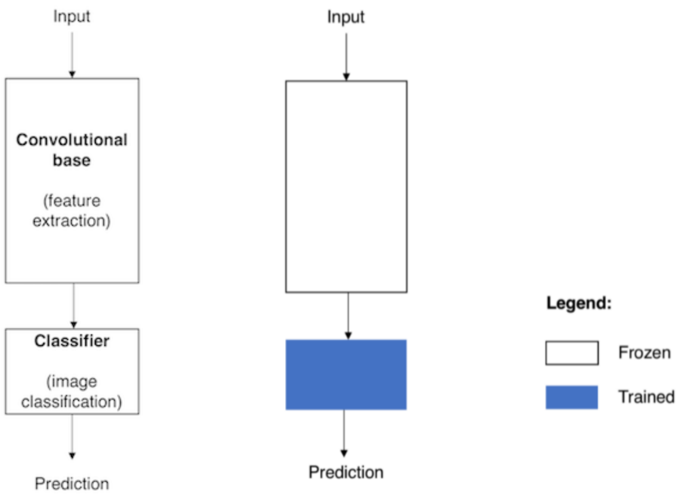
ขั้นตอนที่ 3 : การสกัดคุณลักษณะ (Feature extraction)

การสกัดคุณลักษณะ (Feature extraction) จะใช้การคำนวณคณิตศาสตร์เพื่อสกัดเอาคุณลักษณะจากรูปภาพออกมา เรียกว่า เคอร์เนล (Kernel) หรือ ตัวกรอง (Filter) โดยตัวกรอง 1 ตัวจะดึงคุณลักษณะที่สนใจออกมาได้ 1 อย่าง ดังนั้นจึงต้องมีตัวกรองหลายตัว เพื่อสกัดหาคุณลักษณะทางพื้นที่มาประกอบกัน โดยตัวกรองจะถูกทาบลงในพิกเซลภาพแรกมีลักษณะเป็นตาราง 2 มิติที่มีขนาดตามพื้นที่ย่อยๆที่พิจารณา ตัวกรองจะเลื่อนไปจนครบทุกพิกเซลในภาพ และจะได้ผังคุณลักษณะ (Feature map) ออกมา[4] ดังภาพที่ 3 โดยการสกัดคุณลักษณะจะใช้ข้อมูลปริมาณมากและเวลานาน ดังนั้นจึงใช้การเรียนรู้แบบถ่ายโอน (Transfer Learning) เป็นเทคนิคการเรียนรู้ของเครื่อง จะใช้การเรียนรู้ด้วยการนำโครงสร้างของโมเดลที่ฝึกเรียบร้อยแล้วนำข้อมูลของใหม่มาฝึก จะช่วยเรื่องในการประหยัดเวลาอย่างมากเนื่องจากระยะเวลาในการฝึกตั้งแต่ต้นจนจบกระบวนการใช้เวลานานและมีความซับซ้อน อีกทั้งยังต้องใช้ Data set ที่มีขนาดใหญ่และใช้เวลาในการประมวลผลหลายวันจนถึงหลายสัปดาห์

ผู้วิจัยได้ใช้การเรียนรู้แบบถ่ายโอน (Transfer Learning) ทั้ง 3 โครงสร้าง ซึ่งเป็นโครงสร้างคอนโวลูชัน (Convolution neural network)[4] ได้แก่ VGG16 [6], ResNet50 [7], InceptionV3 [8] ซึ่งใช้ weight จากฐานข้อมูล ImageNet [9]

ขั้นตอนที่ 4 : การจำแนกประเภท (Classification)

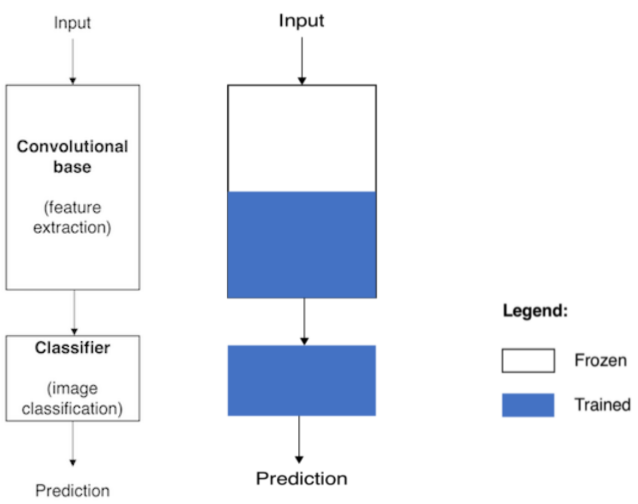
การจำแนกประเภท (Classification) เป็นขั้นตอนสุดท้ายที่รับข้อมูลจากการสกัดคุณลักษณะ (Feature extraction) ซึ่งเป็นโครงสร้างคอนโวลูชัน (Convolution neural network) ส่วนนี้จะยังไม่มีการฝึก ยังใช้ weight ข้อมูลจาก ImageNet [9] และใช้การเรียนรู้แบบถ่ายโอน (Transfer Learning) ส่งคุณลักษณะ (Feature extraction) มายังส่วนการจำแนกประเภท (Classification) ในส่วนนี้ได้ดำเนิน custom head ตามรูปภาพที่ 4 ในส่วนสีฟ้า โดยการเพิ่มชั้น pooling และ outputs dense = 1 ในการจำแนกประเภท (Classification) โดยใช้ activation sigmoid ในส่วนนี้จะเทรนด้วยชุดข้อมูล (Data Set) เพื่อให้โมเดลเรียนรู้และสามารถทำนายชุดข้อมูลที่เตรียมสำหรับวิจัยได้



ภาพที่ 4 แสดงการจำแนกประเภท (Classification)

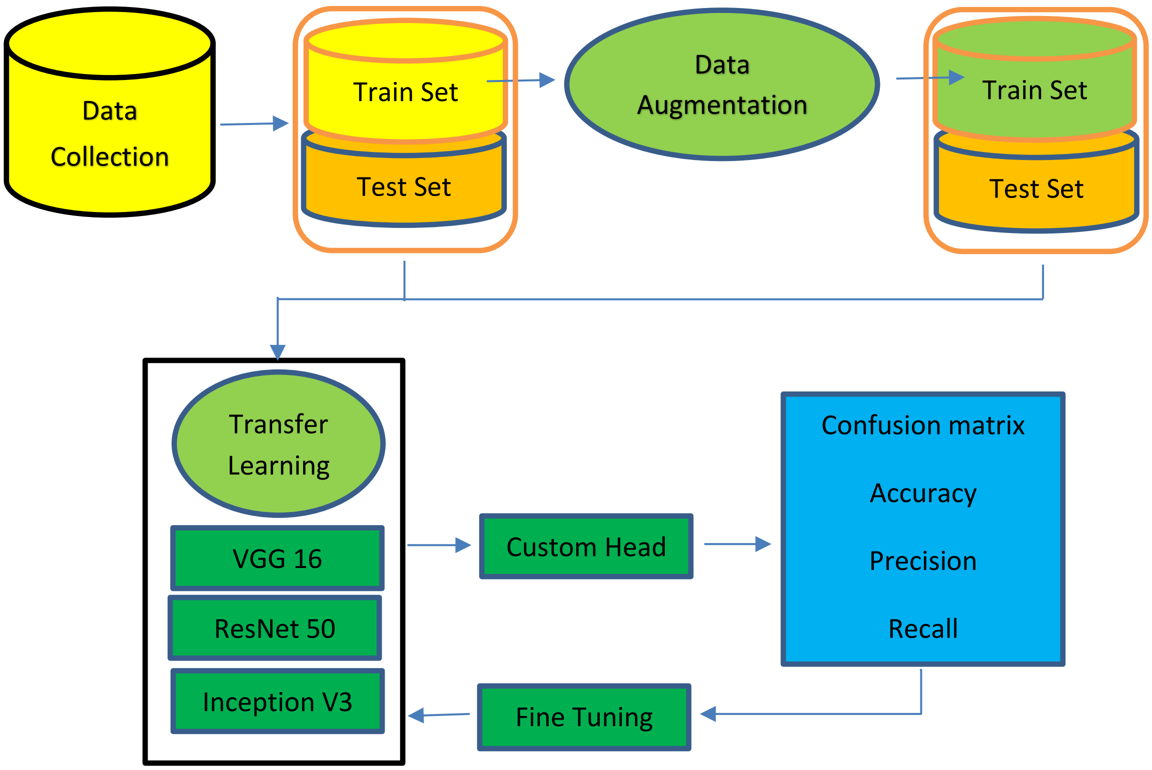
ขั้นตอนที่ 5 : การทดลอง fine tune

ในขั้นตอนนี้หลังจากได้ผลลัพธ์ทุกโมเดลแล้วจะดำเนินการทดสอบ fine tune เพิ่มเพื่อทดลองปรับประสิทธิภาพโมเดล โดยจะทำการนำชุดข้อมูล (data set) เข้าไปเทรนในชั้น (layer) ท้ายๆของโครงสร้างคอนโวลูชัน (Convolution network) ทั้ง 3 โครงสร้าง ส่วนสีฟ้าตามภาพที่ 5



ภาพที่ 5 แสดงการ fine tune บางชั้นของโครงสร้างคอนโวลูชัน (Convolution network)

โดยภาพรวมขั้นตอนการวิจัยสรุปดังภาพที่ 6 ซึ่งหลังจากขั้นตอนที่ 4 การจำแนกประเภท (Classification) แล้ว วัดผลประสิทธิภาพแต่ละโมเดลหลังจากนั้นดำเนินการ Fine Tuning ในขั้นตอนที่ 5 เพื่อปรับปรุงประสิทธิภาพของโมเดล



ภาพที่ 6 ภาพรวมขั้นตอนการทดลอง

การทดลอง

ในขั้นตอนได้ทำการทดลองเพื่อปรับจูนพารามิเตอร์ต่างๆได้แก่ max pooling, average pooling, batch size, learning rate ของการ optimizer เพื่อปรับให้โมเดลมีประสิทธิภาพที่ดีที่สุด โดยจะเปรียบเทียบผลพารามิเตอร์ต่างๆดังนี้

1.การเปรียบเทียบระหว่าง max pooling และ average pooling

Pooling layer นี้อยู่ในส่วน Classification ตามที่ได้ทำการเพิ่มเข้าไปในขั้นตอนที่ 4 โดยกำหนดพารามิเตอร์ต่างๆ ดังนี้ Batch size กำหนดที่ 256 ค่า Learning rate เป็น RMSprop เท่ากับ 0.0001 และกำหนด กำหนดรอบการเรียนรู้ที่ 500 รอบ (epoch) และใช้โมเดล VGG16 ในการทดสอบซึ่งผลที่ได้จากการทดลองเปรียบเทียบประสิทธิภาพพบว่า max pooling layer นั้นทำให้โมเดลมีประสิทธิภาพดีกว่า ดังตารางที่ 1

ตารางที่ 1 เปรียบเทียบผลลัพธ์ประสิทฺธิภาพของ Max pooling และ Average pooling

pooling	Max pooling	Average pooling
accuracy	0.81	0.80

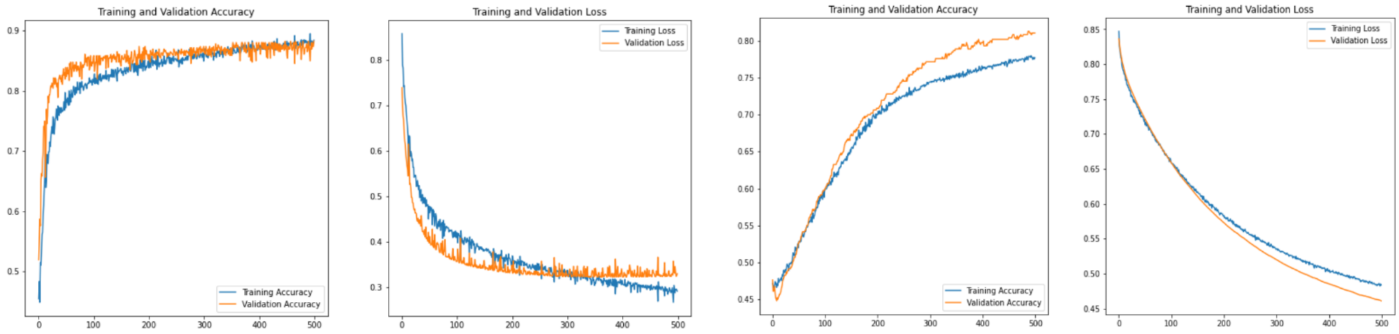
2.การเปรียบเทียบระหว่าง learning rate 0.0001 และ 0.001

ในการทดสอบเปรียบเทียบประสิทธิภาพของพารามิเตอร์ learning rate RMSprop โดยได้ทดลองสอบการเปรียบเทียบค่าที่ 0.0001 และ 0.001 พบว่า accuracy เท่ากับ 0.81 และ 0.80 ตามลำดับ ตารางที่ 2 โมเดลมีประสิทธิภาพที่ใกล้เคียงกัน และเมื่อมาพิจารณากราฟของ learning rate RMSprop เท่ากับ 0.001 พบว่าเกิด overfit จึงได้ผลสรุปว่า learning rate RMSprop เท่ากับ 0.0001 นั้นมีประสิทธิภาพที่ดีกว่า เนื่องจากโมเดลค่อยๆเปลี่ยนแปลงที่ละน้อยๆในในการเรียนรู้แต่ละรอบทำให้กราฟมีความนิ่งกว่า ดังภาพที่ 7

ตารางที่ 2 เปรียบเทียบผลลัพธ์ประสิทฺธิภาพของ learning rate 0.001 และ learning rate 0.0001

Optimizer RMSprop	Learning rate = 0.001	Learning rate = 0.0001
accuracy	0.80	0.81

learning rate RMSprop เท่ากับ 0.001 learning rate RMSprop เท่ากับ 0.0001



ภาพที่ 7 กราฟแสดง Accuracy และ Loss เปรียบเทียบกันระหว่าง learning rate RMSprop เท่ากับ 0.001 และ 0.0001

ปรับพารามิเตอร์เปรียบเทียบกันระหว่าง Batch size ขนาดต่างๆ

จากการทดสอบปรับพารามิเตอร์ก่อนหน้านี้เราพบว่า พารามิเตอร์ pooling และ optimizer ที่ดีที่สุดสำหรับชุดข้อมูลนี้ (data set) คือ max pooling และ optimizer RMSprop learning rate 0.0001 การทดสอบ batch จึงตั้งค่าพารามิเตอร์เหมือนกันดังนี้ กำหนด pooling เป็น max pooling และ optimizer คือ RMSprop learning rate 0.0001

ตารางที่ 3 เปรียบเทียบผลลัพธ์ประสิทธิภาพของ Batch size ขนาดต่างๆ

Batch size	**Without Augmentation**			**With Augmentation**		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
16	0.87	0.87	0.87	0.80	0.81	0.80
128	0.83	0.83	0.83	0.80	0.81	0.80
256	0.81	0.81	0.81	0.76	0.77	0.76

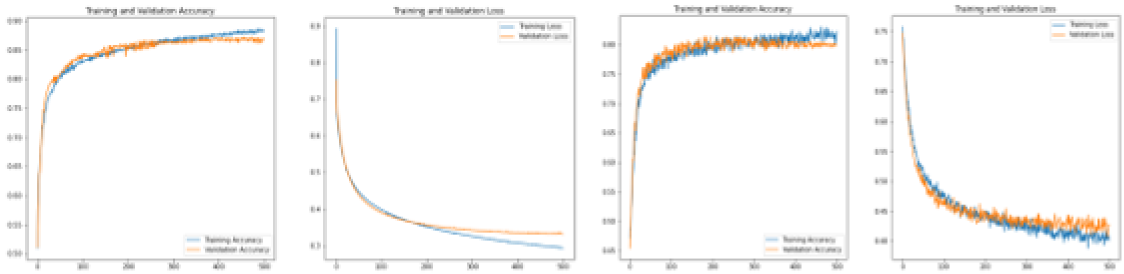
**สำหรับ Batch size 512 เนื่องจากข้อจำกัดของทรัพยากรคอมพิวเตอร์ (Kernel dead) ทำให้ไม่สามารถทดสอบ batch size 512 ได้

จากผลการทดลองตารางที่ 3 เมื่อเปรียบเทียบค่า Batch size ขนาดต่างๆ พบว่า accuracy batch size 16, 128, 256 มีค่า 0.87, 0.84, 0.81 ตามลำดับ และ 0.80, 0.80, 0.76 สำหรับการดัดแปลงข้อมูลต้นฉบับ (data augmentation) และเมื่อพิจารณาจากกราฟความสัมพันธ์ระหว่าง accuracy และ loss พบว่า Batch ขนาด 16 เกิด over fitting สำหรับ batch ที่ 128 และ 256 กราฟมีการเรียนรู้ที่ดี ภาพที่ 8 เมื่อพิจารณาค่าความแม่นยำ (Accuracy) ดังนั้นจึงสรุปได้สว่า batch size เหมาะสมกับ data set นี้คือ ขนาด 128

VGG16

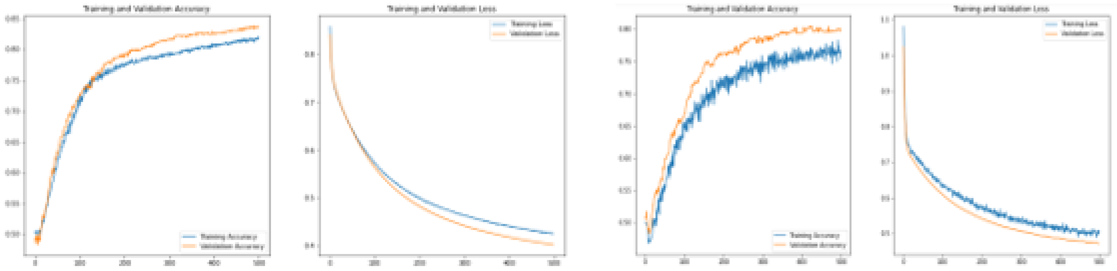
Batch 16

without data augmentation with data augmentation



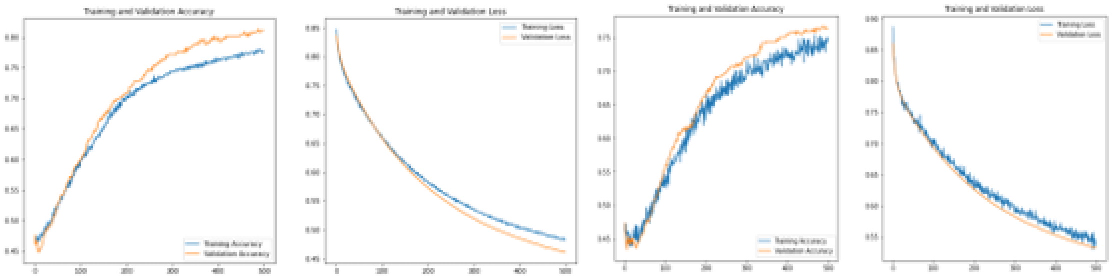
Batch 128

without data augmentation with data augmentation



Batch 256

without data augmentation with data augmentation



ภาพที่ 8 กราฟแสดงความสัมพันธ์ระหว่าง accuracy และ loss ของ Batch size ขนาดต่างๆ

จากการทดสอบปรับเพื่อทดสอบพารามิเตอร์เบื้องต้น ได้พารามิเตอร์ที่ทำให้โมเดลมีประสิทธิภาพมากที่สุด ดังนี้ max pooling RMSprop Learning เท่ากับ 0.0001 และ Batch size เท่ากับ 128 ดังนั้นผู้วิจัยดำเนินการตั้งค่าพารามิเตอร์ดังกล่าวในแต่ละโมเดล ผลลัพธ์ที่ได้นั้นจะกล่าวในหัวข้อผลการวิจัยและอภิปรายผลการวิจัย

การทดสอบ fine tune

ในขั้นตอนนี้หลังจากได้ผลลัพธ์ทุกโมเดลแล้วจะดำเนินการทดสอบ fine tune เพิ่มเพื่อทดลองปรับประสิทธิภาพโมเดล โดยจะทำการนำ data set เข้าไปเทรนใน layer ท้ายๆของโครงสร้างทั้ง 3 โครงสร้าง ตามที่ได้กล่าวในวิธีดำเนินการขั้นตอนที่ 5 โดยดำเนินการดังนี้

1.โมเดล VGG16

จากโมเดล VGG16 มีจำนวนชั้น (layer) ของโมเดลมีทั้งหมด 19 ชั้น (layer) รวม max pooling layer และ outputs dense ซึ่งรวมทั้งหมด 21 ชั้น ทำการปรับแต่งชั้นเพื่อเทรนโมเดลตั้งแต่ชั้นที่ 16 ถึงชั้นที่ 21 รวมทั้งหมด 6 ชั้น

2.โมเดล ResNet50

จากโมเดล ResNet50 มีจำนวนชั้น (layer) ของโมเดลมีทั้งหมด 175 ชั้น (layer) รวม max pooling layer และ outputs dense ซึ่งรวมทั้งหมด 177 ชั้น ทำการปรับแต่งชั้นเพื่อเทรนโมเดลตั้งแต่ชั้นที่ 144 ถึงชั้นที่ 177 รวมทั้งหมด 34 ชั้น

3.โมเดล InceptionV3

จากโมเดล InceptionV3 มีจำนวนชั้น (layer) ของโมเดลมีทั้งหมด 311 ชั้น (layer) รวม max pooling layer และ outputs dense ซึ่งรวมทั้งหมด 313 ชั้น ทำการปรับแต่งชั้นเพื่อเทรนโมเดลตั้งแต่ชั้นที่ 281 ถึงชั้นที่ 313 รวมทั้งหมด 33 ชั้น

โดยทุกโมเดลได้การทำการ complie โมเดล optimizer RMSprop learning late 0.0001 ทำการฝึก 500 epoch ตามเดิม และกำหนดให้สิ้นสุดการฝึกโดยใช้ early stopping กำหนดค่า loss ใน test set หากไม่ลดลง 3 รอบสุดท้ายของการฝึก (patience = 3) ผลลัพธ์ที่ได้นั้นจะกล่าวในหัวข้อผลการวิจัยและอภิปรายผลการวิจัยหัวข้อถัดไป

ผลการวิจัยและอภิปรายผลการวิจัย

จากการทดลองเทรนโมเดลทั้งสิ้น 6 โมเดล และทำการเปรียบเทียบผลลัพธ์ประสิทธิภาพโมเดล ได้ผลการทดลองดังต่อไปนี้

1 เปรียบเทียบผลลัพธ์ประสิทธิภาพของแต่ละโมเดล

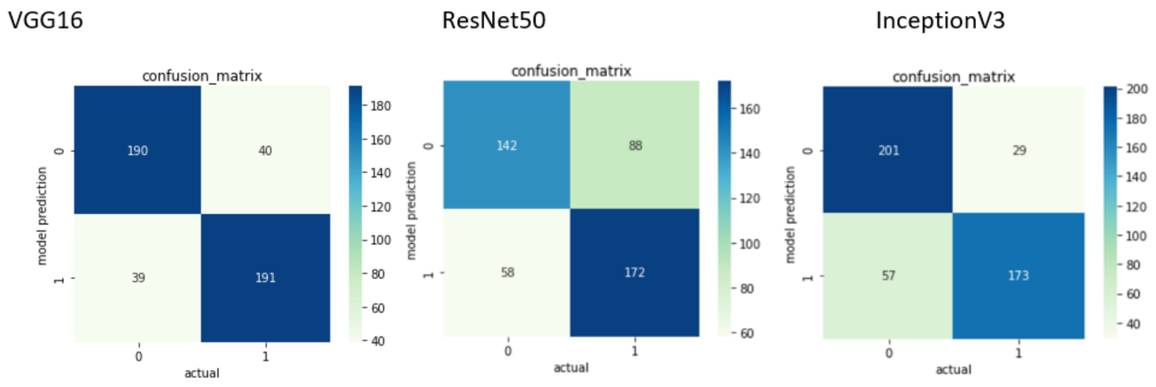
จากตารางที่ 4 นั้นได้แสดงให้เห็นว่า โมเดล VGG16 มีประสิทธิภาพดีที่สุดสำหรับข้อมูลชุดนี้ และรองลงมาคือ InceptionV3 และ Resnet50 ตามลำดับ โดยมี accuracy 0.83 0.81 และ 0.68 ตามลำดับ และสำหรับ preprocessing ที่ทำการดัดแปลงข้อมูลต้นฉบับ (data augmentation) โมเดลที่มีประสิทธิภาพดีที่สุดคือ VGG16 InceptionV3 และ ResNet50 โดยมี accuracy 0.80 0.77 และ0.64 ตามลำดับ โดยทุกโมเดลกำหนดรอบการเรียนรู้ที่ 500 รอบ (epoch)

ตารางที่ 4 เปรียบเทียบผลลัพธ์ประสิทธิภาพของแต่ละโมเดล

Model	Without Augmentation				With Augmentation			
	Accuracy	Precision	Recall	Epoch	Accuracy	Precision	Recall	Epoch
VGG 16	**0.83**	0.83	0.83	500	**0.80**	0.81	0.80	500
Resnet 50	0.68	0.69	0.68	500	0.64	0.64	0.64	500
InceptionV3	0.81	0.82	0.81	500	0.77	0.78	0.77	500

ผลลัพธ์ confusion matrix ในแต่ละโมเดลดังนี้

ไม่ทำการดัดแปลงข้อมูลต้นฉบับ (data without augmentation)



ภาพที่ 9 confusion matrix ของโมเดลเดล VGG16 ResNet50 และ InceptionV3

จากผลการทดลองสรุปได้จาก data set ที่ใช้เป็นข้อมูลทดสอบสรุปได้ดังนี้ (ภาพที่ 9)

โมเดล VGG16

ทายรถยนต์ที่ไม่มีความเสียหายถูกต้อง 190 คัน ทายผิด 40 คัน

ทายรถยนต์ที่มีความเสียหายถูกต้อง 191 คัน ทายผิด 39 คัน

รวมทายถูกต้องทั้งหมด 381 คัน ทายผิด 79 คัน

โมเดล ResNet50

ทายรถยนต์ที่ไม่มีความเสียหายถูกต้อง 142 คัน ทายผิด 88 คัน

ทายรถยนต์ที่มีความเสียหายถูกต้อง 172 คัน ทายผิด 58 คัน

รวมทายถูกต้องทั้งหมด 314 คัน ทายผิด 146 คัน

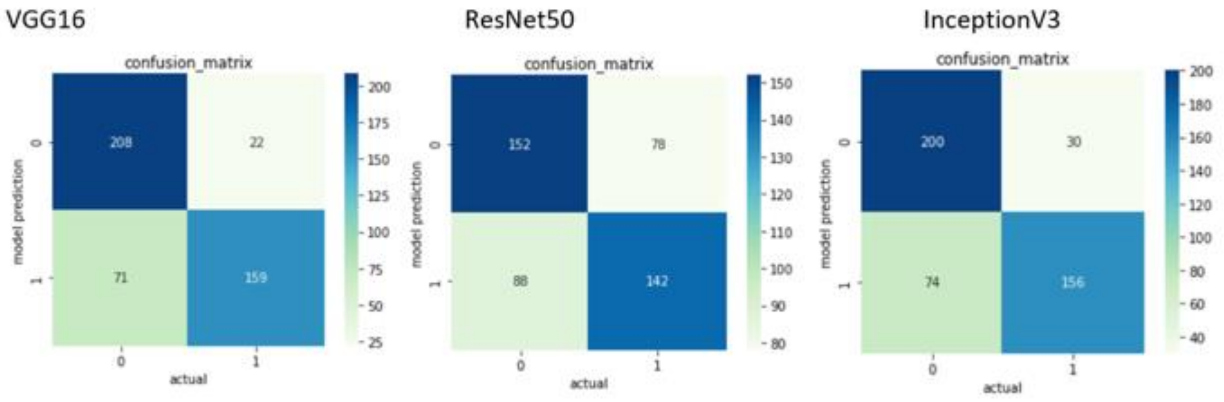
โมเดล InceptionV3

ทายรถยนต์ที่ไม่มีความเสียหายถูกต้อง 201 คัน ทายผิด 29 คัน

ทายรถยนต์ที่มีความเสียหายถูกต้อง 173 คัน ทายผิด 57 คัน

รวมทายถูกต้องทั้งหมด 374 คัน ทายผิด 86 คัน

ทำการดัดแปลงข้อมูลต้นฉบับ (data augmentation)



ภาพที่ 10 confusion matrix ของโมเดลเดล VGG16 ResNet50 และ InceptionV3 with data augmentation

จากผลการทดลองเพิ่มการดัดแปลงข้อมูลต้นฉบับ (data augmentation) ในข้อมูลการเทรนสรุปได้จาก data set ที่ใช้เป็นข้อมูลทดสอบสรุปได้ดังนี้ ภาพที่ 10

โมเดล VGG16

ทายรถยนต์ที่ไม่มีความเสียหายถูกต้อง 208 คัน ทายผิด 22 คัน

ทายรถยนต์ที่มีความเสียหายถูกต้อง 159 คัน ทายผิด 71 คัน

รวมทายถูกต้องทั้งหมด 367 คัน ทายผิด 93 คัน

โมเดล ResNet50

ทายรถยนต์ที่ไม่มีความเสียหายถูกต้อง 152 คัน ทายผิด 78 คัน

ทายรถยนต์ที่มีความเสียหายถูกต้อง 142 คัน ทายผิด 88 คัน

รวมทายถูกต้องทั้งหมด 294 คัน ทายผิด 166 คัน

โมเดล InceptionV3

ทายรถยนต์ที่ไม่มีความเสียหายถูกต้อง 200 คัน ทายผิด 30 คัน

ทายรถยนต์ที่มีความเสียหายถูกต้อง 126 คัน ทายผิด 74 คัน

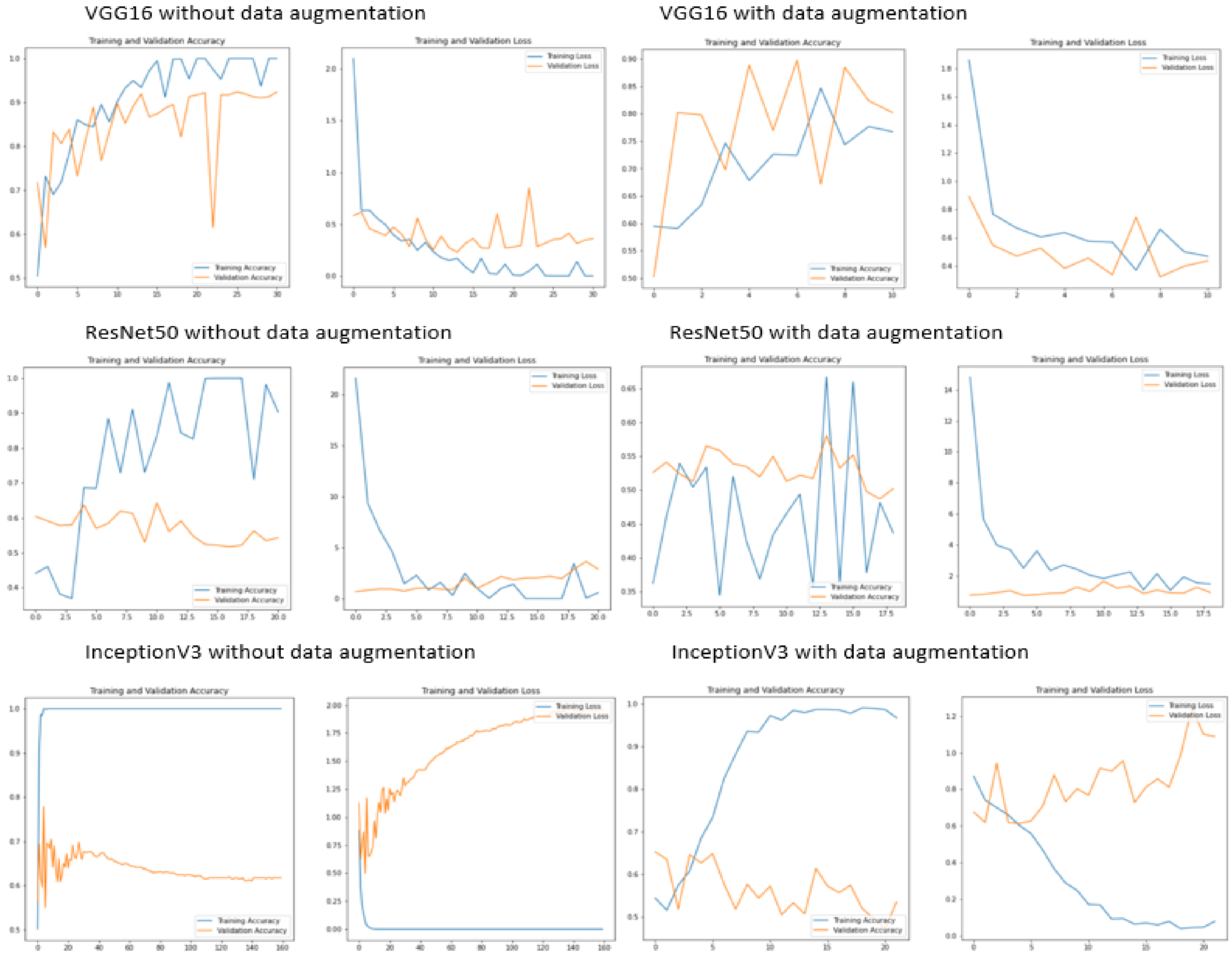
รวมทายถูกต้องทั้งหมด 326 คัน ทายผิด 104 คัน

2 เปรียบเทียบผลลัพธ์ประสิทธิภาพของแต่ละโมเดลหลังจากการทำลอง fine tune

ผลการทำลอง fine tune โดยใช้คำสั่ง Early Stopping ให้สิ้นสุดการฝึก โดยกำหนด monitor ที่ค่า loss หาก loss ไม่เปลี่ยนแปลง 3 epoch โมเดลจะหยุดการฝึกลง จากตารางที่ 4 นั้นได้แสดงให้เห็นว่า โมเดล VGG16 จำนวน 31 epoch มีประสิทธิภาพดีที่สุดสำหรับข้อมูลชุดนี้ และรองลงมาคือ InceptionV3 จำนวน 160 epoch และ Resnet50 จำนวน 21 epoch โดยมี accuracy 0.92 0.69 และ 0.55 ตามลำดับ และสำหรับ preprocessing ที่ทำการดัดแปลงข้อมูลต้นฉบับ (data augmentation) โมเดลที่มีประสิทธิภาพดีที่สุดคือ VGG16 จำนวน 11 epoch InceptionV3 จำนวน 22 epoch และ ResNet50 จำนวน 19 epoch โดยมี accuracy 0.85 0.60 และ0.50 ตามลำดับ ตามตารางที่ 5

ตารางที่ 5 เปรียบเทียบผลลัพธ์ประสิทธิภาพของแต่ละโมเดลหลังจากการทดลอง fine tune

Model	Without Augmentation				With Augmentation			
	Accuracy	Precision	Recall	Epoch	Accuracy	Precision	Recall	Epoch
VGG 16	**0.92**	0.92	0.92	31	**0.85**	0.80	0.80	11
Resnet 50	0.55	0.54	052	21	0.50	0.50	0.50	19
InceptionV3	0.69	0.62	0.58	160	0.60	0.53	0.44	22



ภาพที่ 11 กราฟแสดงความสัมพันธ์ระหว่าง accuracy และ loss กับ epochs ของโมเดล fine tune ทั้ง 6 โมเดล

ทั้งนี้เมื่อพิจารณากราฟแสดงความสัมพันธ์ระหว่าง accuracy และ loss กับ epochs ของโมเดล ทั้ง 6 โมเดล (ภาพที่ 11) จะเห็นได้ว่าโมเดลมีกราฟเรียนรู้ที่ไม่นิ่งและเกิด overfitting เนื่องจากปริมาณรูปภาพในการฝึก (Train) มีปริมาณไม่เยอะเมื่อเทียบกับชุดข้อมูล ImageNet ที่เป็นฐานข้อมูลสำหรับการฝึกโมเดล VGG16, InceptionV3 และ ResNet50 ซึ่งเป็นโมเดลที่ถูกฝึก (Pre-train) มาเป็นอย่างดีแล้วทั้งสิ้น

สรุปผลการวิจัย

งานวิจัยนี้เป็นการศึกษาวิธีการจำแนกประเภทรูปภาพรถยนต์ที่มีความเสียหาย และไม่มีความเสียหายโดยใช้การเรียนรู้เชิงลึก (Deep Learning) ใช้วิธีการเรียนรู้แบบถ่ายโอน (transfer learning) ได้นำโครงสร้างสถาปัตยกรรม 3 โครงสร้าง ได้แก่ VGG16, ResNet50, InceptionV3 ได้ผลลัพธ์ที่ดีที่สุดคือโมเดล VGG16 เป็นโมเดลที่มีประสิทธิภาพสูงที่สุดในการเปรียบเทียบทั้งหมด 6 โมเดล สำหรับ data set ชุดนี้ สามารถทำนายจำแนกรูปภาพรถยนต์ที่มีความเสียหาย และ รูปที่ไม่มีความเสียหายได้ดี ซึ่งสามารถวัดประสิทธิภาพมีผลลัพธ์ดังนี้ Accuracy = 0.83 Precision = 0.83 Recall = 0.83 และ F1 = 0.83 อยู่ในเกณฑ์ที่ดี สามารถจำแนกรถยนต์ที่ไม่มีความเสียหายถูกต้อง 192 คัน ทายผิด 38 คัน และรถยนต์ที่มีความเสียหายถูกต้อง 193 คัน ทายผิด 37 คัน รวมทายถูกต้องทั้งหมด 385 คัน ทายผิดทั้งหมด 75 คัน จากข้อมูลทดสอบ 460 รูปภาพ

งานวิจัยในอนาคต ผู้วิจัยจะทำการการทดลองปรับจูนพารามิเตอร์ต่างๆ fine tune โมเดลเพื่อเพิ่มประสิทธิภาพของโมเดลและศึกษาโครงสร้างสถาปัตยกรรมแบบอื่นๆเพิ่มเติมมาทดสอบวัดเปรียบเทียบประสิทธิภาพ โดยเฉพาะโมเดลใหม่ๆที่มีประสิทธิภาพที่ดีเช่น EfficientNet และใช้คอมพิวเตอร์ที่มีการ์ดจอประมวลผลคุณภาพสูงเพื่อประสิทธิภาพในการทำการวิจัย

กิตติกรรมประกาศ

การจัดทำวิจัยฉบับบนี้สำเร็จลุล่วงไปได้ด้วยการให้ความช่วยเหลือแนะนำของ ผศ.ดร.วราภรณ์ วิยานนท์ ซึ่งเป็นอาจารย์ที่ปรึกษาที่ได้กรุณาให้คำแนะนำข้อคิดเห็น ตรวจสอบ และแก้ไขร่างสารนิพนธ์มาโดยตลอด จึงขอขอบคุณมา ณ ที่นี้

ขอขอบคุณ คณาจารย์ทุกท่านในภาควิชาวิทยาการข้อมูล มหาวิทยาลัยศรีนครินทรวิโรฒ ที่แนะนำ ความรู้และแนวทางการทำสารนิพนธ์นี้

เอกสารอ้างอิง

[1] สำนักงานคณะกรรมการกำกับและส่งเสริมการประกอบธุรกิจประกันภัย และสำนักงานสภาพัฒนาการเศรษฐกิจและสังคมแห่งชาติ

[2] Kalpesh Patil, Mandar Kulkarni, Anand Sriraman, Shirish Karande. Deep Learning Based Car Damage Classification. 2017 16th IEEE International Conference on Machine Learning and Apllications (ICMLA):50-54

[3] Kyu, P. M. and K. Woraratpanya (2020). Car Damage Detection and Classification. Proceedings of the 11th International Conference on Advances in Information Technology: 1-6.

[4] Keiron O'Shea, Ryan Nash. An introductional Convolution Neural Networks. arXiv:1511.08458v2 [cs.NE] 2 Dec 2015

[5] https://www.kaggle.com/anujms/car-damage-detection

[6] Simonyan, K. and A. Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv 1409.1556.

[7] He, K., et al. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[8] Szegedy, C., et al. (2016). Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[9] Alex Krizhevssky, Ilya Sutskever, Geoffrey E.Hinton. ImageNet Classification with Deep Convolution Neural Networks. 2012 the Proceeding of the 25th International Conference on Neural Information Processing Systems: 1097-1105.

[10] Singh, R., et al. (2019). Automating Car Insurance Claims Using Deep Learning Techniques. 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM).