

การตรวจหาและจำแนกรอยโรคของภาพเอกซเรย์ปอดอักเสบ COVID-19
โดยใช้เทคนิคการเรียนรู้เชิงลึก

DETECTION AND CLASSIFICATION OF COVID-19 CHEST X-RAY
BY DEEP LEARNING TECHNIQUE



วรรณิกา ศรีอารา

WANNIKA SONARRA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DETECTION AND CLASSIFICATION OF COVID-19 CHEST X-RAY
BY DEEP LEARNING TECHNIQUE

The seal of King Mongkut's Institute of Technology Ladkrabang is a circular emblem. It features a central sunburst with rays emanating from a central point. Below the sunburst are three tiered, ornate structures resembling traditional Thai stupas or pagodas, each supported by a decorative base. The entire emblem is surrounded by a circular border containing Thai text. The text on the left side reads 'เทคโนโลยีบัณฑิต' (Technology Graduate) and the text on the right side reads 'ลาดกระบัง' (Ladkrabang).

WANNIKA SONARRA

THE THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN BIOMEDICAL ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2023

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



KMITL-2023-EN-D- 317-038

COPYRIGHT 2023

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การตรวจหาและจำแนกรอยโรคของภาพเอกซเรย์ปอดอักเสบ COVID-19 โดยใช้เทคนิคการเรียนรู้เชิงลึก
นักศึกษา	นางสาววรรณิกา ศรอารา
รหัสประจำตัว	63601061
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมชีวการแพทย์
ปี	พ.ศ. 2566
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.สรินพร วิสิฐสุทธาทพงศ์

บทคัดย่อ

ปัจจุบันการแพร่ระบาดของเชื้อปอดอักเสบโคโรนาไวรัสหรือโรคโควิด COVID-19 ยังเป็นปัญหารุนแรงที่เกิดขึ้นในประเทศไทยและยังเป็นปัญหาวิกฤตในระดับโลก เนื่องจากจำนวนผู้ติดเชื้อที่เพิ่มมากขึ้นทุก ๆ วัน ซึ่งส่งผลทำให้โรงพยาบาลตรวจคัดกรองผู้ติดเชื้อเป็นไปอย่างล่าช้า ต้องรอผลตรวจยาวนานและการขาดแคลนบุคลากรกับสถานที่ ๆ จำกัดในการดูแลรักษา เนื่องจากข้อจำกัดดังกล่าว ดังนั้นเราจึงได้วิจัยทางคลินิกด้วยระบบอัลกอริทึมการทำนายจึงเกิดขึ้น อัลกอริทึมการทำนายผลอาจจะช่วยลดความเครียดในระบบการดูแลสุขภาพด้วยการระบุโรค ซึ่งการจำแนกภาพเป็นวิธีหนึ่งที่น่าสนใจในการประมวลผลภาพ โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network; CNN) เป็นอัลกอริทึมที่ถูกนำมาใช้อย่างแพร่หลายสำหรับการจำแนกภาพ การแยกภาพเอกซเรย์ปอดจากผู้ป่วยที่เป็นโควิด (COVID-19) ออกจากผู้ป่วยติดเชื้อปอดอักเสบทั่วไปและผู้ป่วยปอดปกติ เพื่อประเมินประสิทธิภาพการคาดการณ์ของแบบจำลอง ค่าความแม่นยำ คะแนน F1-score การเรียกคืน AUC และคะแนนความถูกต้อง จะสังเกตเห็นได้ว่าแบบจำลองการทำนายที่ได้รับ การฝึกฝนเกี่ยวกับผลการตรวจสามารถนำมาใช้ในการทำนายการติดเชื้อโควิด (COVID-19) มีประโยชน์สำหรับแพทย์และทีมงานทางการแพทย์ด้านหน้าในการจัดลำดับความสำคัญของทรัพยากรอย่างถูกต้องและลดการทำซ้ำ การอ่านภาพที่เกิดขึ้นหลาย ๆ รอบ เพื่อหารอยโรคในปอดในผู้ป่วยที่มีจำนวนมากและมีความเปลี่ยนแปลงอย่างรวดเร็วของรอยโรค ดังนั้นในงานวิจัยนี้ได้พัฒนาโมเดลขึ้นเพื่อช่วยแพทย์และทีมงานทางการแพทย์ในการตรวจสอบผลการตรวจเบื้องต้นและยังสามารถนำมาใช้สำหรับการศึกษาการทำนายทางคลินิกได้อีกด้วย

คำสำคัญ : การจำแนกภาพ, โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN), ภาพเอกซเรย์ปอดโควิด (CXR) โรคปอดติดเชื้อเฉียบพลัน COVID-19

Thesis	Detection and Classification of COVID-19 Chest X-ray by Deep Learning Technique
Student	Miss Wannika Sonarra
Student ID	63601061
Degree	Master of Engineering
Program	Biomedical Engineering
Year	2023
Thesis Advisor	Assoc.Prof.Dr. Sarinporn Visitsattapongse

Abstract

The Coronavirus disease (COVID-19) infection has become a pandemic, and this is the most critical problem that has occurred in Thailand and also expanded all over the world. As such, it is not astonishing to know that this virus has had a direct effect on hospitals with the delayed screening of patients because of the increasing number of daily cases and the shortage of medical personnel and restricted treatment space. Due to such restrictions, in this study, we used a clinical decision-making system with predictive algorithms. Predictive algorithms could potentially ease the strain on healthcare systems by identifying the diseases. Moreover, image classification is one interesting aspect of image processing. Convolutional neural network (CNN) is a widely used algorithm for image classification by separating the images of the COVID-19 disease, images with a lung infection, and normal images. To evaluate the predictive performance of our models, precision, F1-score, recall, receiver operating characteristic (ROC) curve (area under the ROC curve), and accuracy scores were used. It was observed that the predictive models trained on the laboratory findings could be used to predict the COVID-19 infection as well and could be helpful for medical experts to appropriately prioritize the resources. This could be employed to assist medical experts in validating their initial laboratory findings and could also be used for clinical prediction studies.

Keywords: Convolutional neural network (CNN), image of COVID-19, coronavirus,
deep learning

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ข้าพเจ้าต้องขอขอบคุณเป็นอย่างยิ่งด้วยความอนุเคราะห์
รองศาสตราจารย์ ดร.สรินพร วิสิษฐสัทธาพงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้กรุณาสละเวลาให้
คำปรึกษา ตรวจสอบ และชี้แนะตลอดจนแก้ไขข้อบกพร่องวิทยานิพนธ์
รองศาสตราจารย์ ดร.ชูชาติ ปินทวิรุจน์ กรรมการสอบวิทยานิพนธ์ ที่ให้ข้อเสนอแนะที่เป็นประโยชน์
ทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น
พี่ และเพื่อนๆ ร่วมหลักสูตรทุกท่านที่ให้ความช่วยเหลือ และให้กำลังใจมาตลอดการทำวิทยานิพนธ์
ขอขอบคุณทุกท่านที่มีส่วนสนับสนุนการทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วง โดยไม่ได้กล่าวนามไว้ ณ ที่นี้
สุดท้ายนี้ขอขอบคุณ คุณพ่อ คุณแม่ สำหรับการดูแลด้วยความรัก การสนับสนุนในทุกด้านอย่างดีที่สุด
คอยเป็นกำลังใจที่ดีเสมอมา และที่สำคัญต้องขอขอบคุณตัวเองมาก ๆ ที่อดทน พยายาม จนมาถึงวันนี้
ผู้วิจัยหวังเป็นอย่างยิ่งว่าคุณค่าและประโยชน์ที่เกิดจากวิทยานิพนธ์ฉบับนี้ ขอมอบอุทิศแด่ผู้มีพระคุณ
ทุกท่าน บุคลากรทางการศึกษาและผู้ที่เกี่ยวข้อง ตลอดจนเป็นประโยชน์ในการสร้างสรรค์ ต่อยอดหรือพัฒนาเป็น
นวัตกรรมด้านสุขภาพใหม่ ๆ ต่อไปในภายภาคหน้า

วรรณิกา ศรีอารา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูปและอักษรย่อ	X
บทที่1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	2
1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการทำวิจัย	2
1.4 สมมุติฐานของการศึกษา	2
1.5 ขอบเขตการศึกษา	3
1.6 แผนการดำเนินงาน	3
1.7 ขั้นตอนและเวลาในการศึกษา	4
1.8 โครงสร้างวิทยานิพนธ์	5
บทที่2 หลักการและทฤษฎีที่เกี่ยวข้อง	6
2.1 ภาพเอกซเรย์ปอด (Chest X-ray imaging)	6
2.2 รูปแบบของข้อมูลที่ใช้ในการวิจัยและรอยโรคที่เกี่ยวข้อง	7
2.2.1 โรคปอดอักเสบจากเชื้อไวรัสโคโรนา (COVID-19)	7
2.2.2 อาการความแตกต่างระหว่างการติดเชื้อปอดอักเสบทั่วไปกับการติดเชื้อ โคโรนา COVID-19	8
2.2.3 โรคปอดอักเสบจากเชื้อไวรัสโคโรนา (COVID-19)	9
2.2.4 สัญญาณเตือนของอาการติดเชื้อ	9
2.2.5 กลุ่มเสี่ยงปอดอักเสบรุนแรงจากเชื้อไวรัสโคโรนา (COVID-19)	9
2.2.6 แนวทางการรักษาปอดอักเสบในผู้ป่วยโควิด (COVID-19)	10
2.3 หลักการทำงานและส่วนประกอบเครื่องเอกซเรย์ปอดทั่วไป	10
2.3.1 ส่วนประกอบพื้นฐานของเครื่องเอกซเรย์ทั่วไป	11

IV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ฮาร์ดแวร์ (Hardware)	12
2.4.1 CPU: (Central Processing Unit)	12
2.4.2 GPU: (Graphics Processing Unit)	12
2.5 ซอฟต์แวร์ (Software)	13
2.5.1 ภาษา Python	13
2.5.2 Pandas	13
2.5.3 NumPy	13
2.5.4 OpenCV-Python	13
2.5.5 Keras	13
2.6 ชนิดของภาพ	14
2.6.1 DICOM	14
2.6.2 JPEG	14
2.6.3 PNG	14
2.7 โครงข่ายประสาทเทียม (ANN)	16
2.8 เพอร์เซปตรอนหรือนิวรอน	18
2.8.1 ค่าถ่วงน้ำหนัก	18
2.8.2 ค่าเบี่ยงเบน	19
2.8.3 ฟังก์ชันกระตุ้น	19
2.9 การทำคอนโวลูชัน Convolution Layer/Filter (CONV)	20
2.10 การขจัดความเป็นเชิงเส้น (ReLU)	21
2.11 SoftMax	22
2.12 การทำพูลลิ่ง (POOL)	22
2.13 Dropout	23
2.14 Global Average Pooling	23
2.15 การเชื่อมต่อกันของแต่ละเลเยอร์โดยสมบูรณ์ (Full Connection (FC))	24
2.16 การเรียนรู้เชิงลึก (Deep learning)	25
2.17 อัลกอริทึมปรับปรุงประสิทธิภาพ	25
2.18 ทฤษฎีการ Transfer Learning	25
2.19 สถาปัตยกรรมแบบจำลองที่นำมาใช้เพื่อทำนายในงานวิจัย	27
2.19.1 โมเดล Inception v3	27

	2.19.2 โมเดล ResNet50	29
	2.19.3 โมเดล VGG16	30
	2.19.4 โมเดลVGG19	32
	2.19.5 โมเดล MobileNet V2	33
	2.19.6 โมเดล InceptionResNet V2	34
	2.20 ทฤษฎีการปรับแต่ง	35
	2.21 การทดสอบประสิทธิภาพและการประเมินผล	36
	2.21.1 การทดสอบการทำงานของโมเดล	36
	2.21.2 ตัวชี้วัดการประเมินผล	38
	2.21.2.1 ค่าความถูกต้อง (Accuracy)	38
	2.21.2.2 ค่าความแม่นยำ (Precision Metric)	39
	2.21.2.3 ค่าการเรียกคืน (Recall Metric)	39
	2.21.2.4 ค่าตัวชี้วัด F1 (F1-Score Metric)	39
	2.21.3.5 ค่าความจำเพาะ (Specificity)	39
	2.22 ศัพท์เฉพาะและคำอนุพันธ์	40
	2.23 พื้นที่ใต้เส้นโค้ง ROC (AUC - ROC)	40
	2.24 งานวิจัยที่เกี่ยวข้อง	41
บทที่ 3	ขั้นตอนและวิธีดำเนินการวิจัย	42
	3.1 ลักษณะความผิดปกติของภาพเอกซเรย์ปอดอักเสบโควิด (COVID-19)	42
	3.2 การเตรียมชุดข้อมูล	43
	3.3 เครื่องมือ+สเปกคอมพิวเตอร์ที่ใช้ในการวิจัย	44
	3.4 การออกแบบอัลกอริทึมของโปรแกรม	45
	3.4.1 เทคนิคที่เราได้ทำการสอนโดยมีตัวแปรในการสอนโมเดล คร่าว ๆ ดังนี้	45
	3.4.2 โมเดลที่ใช้ในงานวิจัย	45
	3.5 ขั้นตอนและกระบวนการเตรียมงานวิจัย	46
	3.6 ขั้นตอนกระบวนการฝึกสอน	47
	3.6.1 ศึกษาการทำงานของแบบจำลองกับการจำแนกภาพปอดทั้ง 3 ชนิด	47
	3.6.2 ออกแบบโครงสร้างเครือข่ายประสาทเทียม CNNs ปรับปรุงแบบมา	47
	จากแบบจำลองทั้ง 6 แบบ	
บทที่ 4	การทดลองและผลการทดลอง	49

4.1 ระบบปฏิบัติการงานวิจัยและภาษาที่ใช้ในการทดลองในงานวิจัย	49
4.2 ชุดข้อมูลที่ใช้ในการทดลองและวิธีแบ่งข้อมูลเพื่อสอนโมเดล	49
4.3 การเตรียมชุดข้อมูลและโมเดลที่ใช้ในการทดลอง	51
4.4 วิธีการดำเนินการทดลอง	52
4.5 ขั้นตอนการโอนถ่ายข้อมูล	53
4.6 การเพิ่มข้อมูล (Data Augmentation)	54
4.7 ขั้นตอนการ Training โมเดลและ Code	55
4.8 Custom Head	56
4.9 ผลการทำนายที่ได้จากการทดลอง	68
4.10 การวิเคราะห์ความสัมพันธ์จากข้อมูลหลังทำการฝึกสอนโมเดลรุ่นที่ 1	73
4.11 การวิเคราะห์ความสัมพันธ์จากข้อมูลหลังทำการฝึกสอนโมเดลรุ่นที่ 2	79
4.12 การประเมินประสิทธิภาพบนฐานข้อมูล	86
บทที่ 5 บทสรุปและข้อเสนอแนะจากงานวิจัย	87
5.1 อภิปรายผลและสรุป	87
5.2 บทวิจารณ์และข้อเสนอแนะ	88
บรรณานุกรม	
ประวัติผู้วิจัย	
ภาคผนวก ก.	
ภาคผนวก ข.	

สารบัญตาราง

ตารางที่		หน้า
1.1	ขั้นตอนและระยะเวลาในการดำเนินโครงการของภาคเรียนที่ 1/2563	4
1.2	ขั้นตอนและระยะเวลาในการดำเนินโครงการของภาคเรียนที่ 2/2563	4
2.1	แสดงอาการระหว่างการติดเชื้อปอดอักเสบโควิด (COVID-19) และการติดเชื้อปอดอักเสบทั่วไป (Pneumonia)	8
2.2	แสดงตัวอย่างของชนิดและขนาดของภาพดิจิทัล	16
2.3	แสดงส่วนประกอบของโมเดล VGG16, VGG19 Architecture	33
2.4	แสดงรายละเอียดของแต่ละของโมเดล	36
2.5	นิยาม Confusion matrix 3x3 ใช้พิจารณาในงานวิจัยและแสดงตัวอย่างการหาค่า Confusion Matrix แบบละเอียด	37
4.1	แสดงขั้นตอนการแบ่งภาพและชนิดของภาพ ชุดที่ 1	50
4.2	แสดงขั้นตอนการแบ่งภาพและชนิดของภาพ ชุดที่ 2	50
4.3	แสดงรายละเอียดของภาพเอกซเรย์ทรวงอกแยกตามฐานข้อมูล	50
4.4	แสดงตัวอย่างภาพเอกซเรย์ปอดปกติ ปอดอักเสบทั่วไปและปอดอักเสบโควิด COVID-19	51
4.5	แสดง Cropping การขยับภาพ การย่อ-ขยายภาพ การพลิกกลับซ้าย-ขวา การปรับเปลี่ยนสี	55
4.6	แสดงชนิดของโมเดลและค่า Training จากการฝึกสอน รุ่นที่ 1	68
4.7	แสดงชนิดของโมเดลและค่า Training จากการฝึกสอน รุ่นที่ 2	68
4.8	แสดงชนิดของโมเดลและค่า Validation จากการฝึกสอนรุ่นที่ 1	69
4.9	แสดงชนิดของโมเดลและค่า Validation จากการฝึกสอนรุ่นที่ 2	69
4.10	แสดงชนิดของโมเดลและค่า Testing Result จากการสอนรุ่นที่ 1	70
4.11	แสดงชนิดของโมเดลและค่า Testing Result จากการสอนรุ่นที่ 2	70
4.12	แสดงชนิดของโมเดลและค่า Testing Result และค่า AUC จากการสอนโมเดลรุ่นที่ 1	71
4.13	แสดงชนิดของโมเดลและค่า Testing Result และค่า AUC จากการสอนโมเดลรุ่นที่ 2	71
4.14	แสดงเปรียบเทียบประสิทธิภาพ Transfer learning 6 models/ รุ่นที่ 1	72
4.15	แสดงเปรียบเทียบประสิทธิภาพ Transfer learning 6 models/ รุ่นที่ 2	72

VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่		หน้า
2.1	แสดงตัวอย่างรูปภาพเอกซเรย์ปอด CXR จากฐานข้อมูล	7
2.2	แสดงลักษณะที่ลำแสงเอกซ์ผ่านตัวผู้ป่วยไปตกกระทบบนแผ่นรับภาพ (Detector)	11
2.3	แสดงส่วนประกอบโดยทั่วไปของหลอดเอกซเรย์ (X-ray Tube)	12
2.4	แสดงโครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)	17
2.5	แสดงโครงข่ายประสาทเทียมแบบคอนโวลูชัน CNN	17
2.6	แสดงการทำงานของเซลล์ประสาทของสัตว์กับ ANN System (perceptron)	18
2.7	แสดงการทำงานของ Convolutional layer	21
2.8	แสดง ReLU Activation Function	22
2.9	แสดงการทำของพูลลิ่ง (Pooling layer)	22
2.10	แสดงการไม่มี Dropout และมี Dropout	23
2.11	Global average pooling layer for object localization	23
2.12	การทำคอนโวลูชัน (Convolutional Neural Network: CNN)	22
2.13	การเรียนรู้เชิงลึก (Deep Learning architecture)	23
2.14	แสดงโครงสร้างโมเดล Inception V3 Architecture	25
2.15	แสดงโครงสร้างโมเดล Inception V3 ประกอบด้วยชั้น Conv และ Maxpooling	27
2.16	การทำ Feature map ของตัวอย่าง RGB โดยใช้ 1x1 ชั้น Conv	28
2.17	แสดงโครงสร้างโมเดล ResNet50 Architecture	29
2.18	แสดงโมเดล VGG16 Architecture	30
2.19	แสดงโมเดล VGG19 Architecture	32
2.20	แสดงโมเดล MobileNet v2 Architecture	33
2.21	แสดงโมเดล InceptionResNet V2 Architecture	35
2.22	แสดงโครงสร้าง networks ของโมเดล Inception V3 ที่มีการ custom head	36
3.1	แสดงภาพเอกซเรย์ปอดปกติ (Normal) และ Anatomy ของปอด (Lung)	42
3.2	แสดงภาพเอกซเรย์ปอดอักเสบโควิด COVID-19	43
3.3	แสดงภาพเอกซเรย์ปอดอักเสบทั่วไป (Pneumonia)	43
3.4	แสดง Flow Chart แสดงขั้นตอนการแบบข้อมูลภาพรุ่นที่ 1และ2	44
3.5	แสดง Flow Chart ขั้นตอนที่ใช้สอนโมเดลในงานวิจัย	46

3.6	แสดงกระบวนการนำข้อมูลภาพเข้าฝึกสอน	46
3.7	แสดงกระบวนการนำข้อมูลภาพเข้าฝึกสอนโดยใช้โมเดล CNNs network	46
4.1	แสดงกระบวนการฝึกสอนโมเดลรุ่นที่ 1 ไปยังรุ่นที่ 2	53
4.2	แสดงกระบวนการโอนถ่ายข้อมูลไปยังรุ่นที่ 2	54
4.3	แสดงโครงสร้างการทดลองร่วมกับกระบวนการต่าง ๆ ที่ใช้ในการวิจัย	57
4.4	แสดงการดึงโมเดลที่จำเป็นต้องใช้ออกมาในการฝึกสอนโมเดล	58
4.5	ชุดคำสั่งปรับค่าถ่วงน้ำหนัก 180 และการปรับขนาด Batch size 32	59
4.6	ชุดคำสั่งดึงภาพจากโฟลเดอร์เข้าโปรแกรมและกำหนดการแบ่งภาพเป็น 3 คลาสคือ ภาพปอดปกติ ภาพปอดอักเสบและปอดอักเสบโควิด COVID-19 โดยกำหนดขนาดของภาพ Input=150	59
4.7	ชุดคำสั่งการกำหนดชื่อของแต่ละคลาสของ Input นำเข้าฝึกสอน	59
4.8	ชุดคำสั่งการส่งข้อมูลภาพเข้าไปใน Folder	60
4.9	ชุดคำสั่งการอ่านไฟล์ภาพ การดึงข้อมูลภาพและการปรับภาพในรูปแบบต่าง ๆ	61
4.10	ชุดคำสั่งการดึงโมเดลจาก keras applications และปรับ Input ที่ 180,180,3	62
4.11	คำสั่งดึงโครงข่ายโมเดลมาใช้ในโปรแกรม เช่น โมเดล ResNet	62
4.12	ชุดคำสั่งการดึงอัลกอริทึมเข้ามาปรับใช้กับโมเดลโครงข่ายประสาทเทียมที่มีอยู่	62
4.13	ชุดคำสั่งการสอนโมเดลตามที่อัลกอริทึมกำหนด ฝึกสอนจำนวน 50 epochs	63
4.14	แสดงการทำนายผล Testing ของค่าความถูกต้อง แม่นยำและ F1-Score	64
4.15	ชุดคำสั่งสร้างกราฟแสดงความสัมพันธ์ระหว่าง accuracy กับ จำนวน epochs	65
4.16	ชุดคำสั่งสร้างกราฟแสดงความสัมพันธ์ระหว่าง loss กับ จำนวน epochs	65
4.17	นิยาม cm_plot Function	66
4.18	คำสั่งแสดง Plot Confusion Matrix False Positive (FP)	67
4.19	ทดลองนับจำนวนภาพของ Test Dataset ในแต่ละ Class	67
4.20	แสดงคำนวณค่า Confusion Matrix	67
4.21	คำสั่งแสดง Confusion matrix ของ AUC กราฟสัมพันธ์ระหว่าง True Positive (TP) กับ False Positive (FP)	67
4.22	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs	71
4.23	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs	72
4.24	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs	72
4.25	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs	74
4.26	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs	74

4.27	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs	74
4.28	เมทริกซ์ Validation แสดงความสัมพันธ์ระหว่างประเภทของรอยโรคการทำนายผลของทั้ง 6 โมเดลสำหรับการวิเคราะห์ด้วยชุดข้อมูลการตรวจสอบความถูกต้องของ: (a) Inceptionv3, (b) VGG19, (c) VGG16, (d) InceptionResnet V2, (e) MoblieNet V2 และ (f) Resnet 50 โดยอ้างอิงถึงภาพเอกซเรย์ของผู้ป่วย COVID-19 ปกติ ตามลำดับ	75
4.29	Confusion matrix ของ Inception V3 โมเดล	76
4.30	Confusion matrix ของ Vgg19 โมเดล	76
4.31	Confusion matrix ของ ResNet 50 โมเดล	77
4.32	Confusion matrix ของ Vgg16 โมเดล	77
4.33	Confusion matrix ของ MobileNet V2 โมเดล	78
4.34	Confusion matrix ของ InceptionResnet V2 โมเดล	78
4.35	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs/รุ่น 2	79
4.36	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs/รุ่น 2	79
4.37	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs/รุ่น 2	80
4.38	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs/รุ่น 2	80
4.39	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs/รุ่น 2	80
4.40	กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs/รุ่น 2	81
4.41	เมทริกซ์ Validation แสดงความสัมพันธ์ระหว่างประเภทของรอยโรคการทำนายผลของทั้ง 6 โมเดลสำหรับการวิเคราะห์ด้วยชุดข้อมูลการตรวจสอบความถูกต้องของ: (a) Inceptionv3, (b) VGG19, (c) VGG16, (d) InceptionResnet V2, (e) MoblieNet V2 และ (f) Resnet 50 โดยอ้างอิงถึงภาพเอกซเรย์ของผู้ป่วย COVID-19 ปกติ ตามลำดับ	82
4.42	Confusion matrix ของ InceptionResnet V2 โมเดล	83
4.43	Confusion matrix ของ Inception V3 โมเดล	83
4.44	Confusion matrix ของ Resnet50 โมเดล	84
4.45	Confusion matrix ของ VGG16 โมเดล	84
4.46	Confusion matrix ของ VGG19 โมเดล	85
4.47	Confusion matrix ของ MobileNet V2 โมเดล	85

XI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และอักษรย่อ

CNN	Convolutional Neural Networks
Conv	Convolution layer
CXR	Chest X-Ray
DICOM	Digital Imaging and Communications in Medicine
IncepV3	InceptionV3 Model
Max pool	Max pooling Layer
Ref	Reference
ReLU	Rectified Linear Activation Function
ResNet50	Deep Residual Networks Model
Softmax	Soft max activation Function Layer
VGG19	Visual Geometry Group 19 Model
RT-PCR	Rapid Antigen Test-Polymerase chain reaction
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus-2
kVp	Kilovoltage peak
mA	milliampere
HT	Hypertension
CKD	Chronic kidney disease
DM	Diabetes Mellitus
STEMI	ST-Elevated Myocardial Infarction
CA	Cancer
COVID-19	Coronavirus disease 19
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ECMO	Extracorporeal Membrane Oxygenation
OPD	Out Patient Department
RAM	Random Access Memory
ML	Machine Learning
CT Scan	Computerized Tomography

MRI	Magnetic Resonance Imaging
US	Ultrasound
ANN	Artificial neural networks
NumPy	Numeric Python
Pandas	Panel Data
CR/DR	Computed Radiography/ Digital Radiography
PA	Posteroanterior
AP	Anteroposterior
DL	Deep learning



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องด้วยการแพร่ระบาดและการติดเชื้อไวรัสสโปกดอ๊กเสบโคโรน่า 2019 หรือโรค COVID-19 ตั้งแต่ปี 2019 ยังคงเป็นปัญหาใหญ่ของคนไทยในประเทศไทยและต่างประเทศ ซึ่งปัญหานี้ได้แผ่ขยายไปทั่วโลกทุกมุมโลก[1, 2] ดังนั้นการเฝ้าระวัง ติดตามสถานการณ์การติดเชื้อเป็นไปอย่างใกล้ชิด เพื่อป้องกันการติดเชื้อเพิ่มขึ้นซึ่งนับเป็นสิ่งสำคัญมาก เนื่องจากเชื้อไวรัสชนิดนี้มีการกลายพันธุ์และการแพร่กระจายได้ง่ายและรวดเร็วโดยผ่านอากาศเป็นสื่อกลางในการกระจายของเชื้อ เช่น ไอ จาม การพูดคุย โดยไม่สวมหน้ากากอนามัยและการสัมผัสสิ่งคัดหลั่ง[3] ซึ่งสถานการณ์การแพร่ระบาดในครั้งนี้มีผลกระทบต่อสังคมไทยไปทุกมิติ อาทิเช่น ทั้งทางด้านเศรษฐกิจ สังคม และสุขภาพ เป็นต้น ในที่นี้จะกล่าวถึงทางด้านสุขภาพเมื่อมีผู้ติดเชื้อเพิ่มมากขึ้นเรื่อย ๆ ในทุก ๆ วัน[2] การติดเชื้อจากคนสู่คนที่เกิดขึ้นได้ง่าย การแพร่กระจายของเชื้อส่งผลให้เกิดอัตราการเสียชีวิตเพิ่มสูงในเวลาต่อมา สาเหตุจากการขาดความรู้ด้านการป้องกันโรคระบาด กำลังของบุคลากรและทรัพยากรตั้งนั้นส่งผลกระทบต่อตรงกับการด้านการแพทย์และสาธารณสุขก็ต้องมีมาตรการเพิ่มอัตรากำลังบุคลากรเพื่อดูแลผู้ติดเชื้อที่เข้ามารับรักษาในโรงพยาบาลอย่างแข็งขันและเร่งด่วน ถึงแม้ว่าปัจจุบันจะมีวิธีจำแนกผู้ป่วยด้วยการตรวจคัดกรองผู้ติดเชื้อออกจากผู้ป่วยทั่วไปก็ตาม โดยสามารถแบ่งการคัดแยกออกเป็นสองประเภทคือ การตรวจแบบเก็บตัวอย่างระดับโมเลกุลหรือสิ่งคัดหลั่งและการทดสอบด้วยซีรัมผ่านทางจมูกและปากของผู้ป่วยหรือ (RT-PCR), Swap จากตัวอย่างเหล่านี้จึงจะสามารถตรวจสอบหารหัสทางพันธุกรรมเฉพาะของไวรัสหรือโปรตีนที่ทำให้เกิดโรค COVID-19 (เช่น ไวรัสโคโรนา โรคระบบทางเดินหายใจเฉียบพลันรุนแรง เรียกว่า SARS-CoV-2) ได้ แต่ด้วยระยะเวลาการรอคอยผลที่ต้องใช้เวลาและการบริการทางการแพทย์ที่ยังทำได้ไม่ทั่วถึงนัก ปัญหาการขาดแคลนบุคลากรทางการแพทย์ผู้เชี่ยวชาญส่งผลกระทบต่อผู้ติดเชื้อบางส่วนขาดโอกาสในการเข้าถึงการรักษาได้ทันทั่วถึงและถูกต้องกับระยะเวลาอาการของโรคส่งผลทำให้ผู้ติดเชื้อบางส่วนต้องเสียชีวิต ในทางตรงกันข้ามวิธีการทดสอบด้วยซีรัมกับการตรวจตัวอย่างเลือดของผู้ที่อาจจะติดเชื้อเพื่อหาร่องรอยของแอนติบอดีจำเพาะ[4] มีข้อเสียของการตรวจด้วยเทคนิคนี้คือ ขั้นตอนการตรวจด้วยตัวเองที่ซับซ้อนและต้องใช้ระยะเวลาหลายชั่วโมงในห้องแล็บและระยะเวลาการรอผลค่อนข้างยาวนาน จึงจำเป็นต้องมีการฝึกอบรมบุคลากรทางการแพทย์ใหม่ ๆ ท่ามกลางการแพร่ระบาดของโรค ดังนั้นเทคนิคทางเลือกที่สามารถตรวจคัดกรองโรคโควิดได้เป็นอย่างดีและรวดเร็วอีกอย่างคือ การถ่ายภาพเอกซเรย์ปอด (CXR) ด้วยเครื่องฉายพลังงานรังสีเอกซ์หรือเครื่องเอกซเรย์ อย่างไรก็ตามการใช้รังสีเอกซ์ในการถ่ายภาพเอกซเรย์ปอด(CXR) ยังเป็นหนึ่งในขั้นตอน

การตรวจร่างกายพื้นฐานที่บ่งชี้ถึงการติดเชื้อได้ดีอีกด้วย[4,5] โดยในงานวิจัยนี้ได้นำเอาเทคนิคโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชันการเรียนรู้เชิงลึก (CNN: Deep learning)[3] ร่วมการจำแนกภาพเอกซเรย์ปอด (CXR) ข้อดีคือ เพื่อเพิ่มประสิทธิภาพในการตัดสินใจในการรักษาให้กับแพทย์และลดระยะเวลาในการแปลผลซ้ำในรอยโรคปอดที่สามารถเปลี่ยนแปลงได้อย่างรวดเร็ว โดยทั้งนี้ผู้วิจัยคาดหวังว่างานวิจัยนี้จะมีประโยชน์สำหรับแพทย์และบุคลากรหน้างานที่ต้องดูแลผู้ป่วยรวมถึงยังต้องจัดสรรทรัพยากรที่มีให้เกิดประโยชน์สูงสุดเพื่อช่วยลดอัตราการติดเชื้อและการเสียชีวิตผู้ป่วยได้เป็นอย่างดี

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาเรียนรู้และพัฒนาโมเดลการเรียนรู้เชิงลึก (CNN; Deep-learning) สำหรับการคัดแยกโรคของผู้ป่วยปอดปกติกับปอดอักเสบทั่วไปและปอดอักเสบที่เกิดจากสาเหตุการติดเชื้อทางเดินหายใจโควิด (COVID-19) โดยใช้สถาปัตยกรรมโครงข่ายโมเดลในรูปแบบต่าง ๆ

1.2.2 เพื่อประเมินวิเคราะห์และเปรียบเทียบประสิทธิภาพความถูกต้องกับเวลาการทำนายของโมเดลจากการเรียนรู้เชิงลึกสำหรับการวิเคราะห์ปัจจัยที่มีผลต่อประสิทธิภาพการทำงานของโมเดล

1.2.3 ในทางกลับกันยังช่วยเพิ่มคุณภาพชีวิตแบ่งเบาภาระให้กับบุคลากรทางการแพทย์หน้างานช่วยลดความตึงเครียด ประหยัดทรัพยากรและสามารถมีเวลาไปช่วยผู้ป่วยอื่น ๆ ได้มากยิ่งขึ้น

1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการทำวิจัย

1.3.1 จากสาเหตุในปัจจุบันที่เกิดโรคอุบัติใหม่โควิด (COVID-19) ตั้งแต่ปี 2019 ทำให้ผู้วิจัยได้ทำการศึกษาแนวคิดทางการคัดกรองรอยโรคของปอดปกติแยกออกจากปอดอักเสบทั่วไปและปอดอักเสบที่ติดเชื้อโควิด (COVID-19) โดยใช้ภาพข้อมูลภาพจากการเอกซเรย์ปอดทั่วไป (CXR) ร่วมกับเทคนิคโครงข่ายประสาทเทียม (CNN: Deep learning) โดยทดลองใช้สถาปัตยกรรมแบบ 6 โมเดลประกอบไปด้วย InceptionV3, ResNet50, VGG16, VGG19, MobileNetV2 และ InceptionResNet V2

1.4 สมมุติฐานของการศึกษา

เนื่องด้วยการศึกษารอยโรคที่เกี่ยวกับปอดติดเชื้อโรคระบาดของปอดโควิด(COVID-19) ยังคงเป็นเรื่องที่ต้องเฝ้าระวังและติดตามสถานการณ์แพร่ระบาด ดังนั้นการคัดแยกผู้ป่วยปอดปกติ(CXR) ออกจากโรคปอดอักเสบทั่วไปและปอดอักเสบโควิด(COVID-19) ให้รวดเร็วและถูกต้อง โดยการใช้ภาพ

เอกซเรย์(CXR) 2มิติ กับการทดสอบกับโมเดลโครงข่ายประสาทเทียมการเรียนรู้เชิงลึก (CNN) เพื่อการคัดแยกโรคแล้วนั้นยังสามารถทำนายผลของภาพได้ดีและใช้เวลาได้อย่างเหมาะสม

1.5 ขอบเขตการศึกษา

งานวิจัยของเราใช้สถาปัตยกรรมโมเดลการเรียนรู้เชิงลึก (CNN: Deep learning) 6 โมเดล ได้แก่ Inception V3, ResNet 50, VGG16, VGG19, Mobilenet V2และInceptionResNet V2กับชุดข้อมูลภาพเอกซเรย์ปอดสาธารณะใน Kaggle.com จำนวนกว่า 6,000 ภาพ เขียนโปรแกรมในไลบรารี TensorFlow keras และใช้การประมวลผลบน CPU สำหรับการตัดแยกภาพ 3 ประเภทคือ ปอดปกติ (Normal images) ภาพปอดอักเสบทั่วไป (Pneumonia images)และภาพปอดโควิด (COVID-19 images) เพื่อเปรียบเทียบประสิทธิภาพแต่ละโมเดลโดยเครื่องมือวิเคราะห์ผลลัพธ์ Accuracy, Precision, Recall, F1-Score, Confusion Matrix, AUC RUC curve

1.6 แผนการดำเนินงาน

1. ทบทวนวรรณกรรมกับงานวิจัยที่เกี่ยวข้องและศึกษาเพิ่มเติมเกี่ยวกับโปรแกรมต่าง ๆ
2. รวบรวมและจัดเก็บข้อมูลภาพเอกซเรย์ปอด (CXR) จากแหล่งข้อมูลสาธารณะ
3. เตรียมชุดข้อมูลภาพเอกซเรย์เพื่อการทดสอบและศึกษาเทคนิคการทำแบ่งส่วนภาพแบบ manual เทคนิค
4. เรียนรู้การเขียนโปรแกรมภาษา python ที่ใช้หาความรู้เพิ่มเติมและเทคนิคต่าง ๆ เพื่อพัฒนาอัลกอริทึมสำหรับการปรับสอนโปรแกรมและโมเดลต่าง ๆ
5. ทำการสอนโมเดล ทดลองโมเดลในรูปแบบต่าง ๆ ที่เลือกไว้ เก็บผลจากการทดลองมาวิเคราะห์ และประเมินผล สรุปผลการทดลอง
6. เรียบเรียงผลงานทางวิชาการที่ต้องตีพิมพ์
7. จัดทำวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ขั้นตอนและเวลาในการศึกษา

ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินโครงการของภาคเรียนที่ 1/2563

1/2563					
ขั้นตอนการดำเนินงาน	ระยะเวลา				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาหาแหล่งข้อมูล คัดแยกและโหลดภาพมาเก็บไว้					
2. ศึกษากระบวนการทำ ขั้นตอนการแบ่งส่วนข้อมูลภาพ					
3. หาความรู้เพิ่มเติมและทดลองนำภาพเข้ากระบวนการแบ่งส่วนแบ่ง Manual					
4. ศึกษารูปแบบการเรียนรู้เชิงลึก โค้ดภาษา Python หาความรู้เพื่อเติมการเขียนโปรแกรม ดูงานวีวีย่อยเพิ่ม					
5. ศึกษาออกแบบอัลกอริทึมที่จะใช้ในงานวิจัยและแก้ปัญหา Error ต่าง ๆ ที่เกิดขึ้นกับโค้ดที่เขียน					

ตารางที่ 1.2 ขั้นตอนและระยะเวลาในการดำเนินโครงการของภาคเรียนที่ 2/2563

2/2563					
ขั้นตอนการดำเนินงาน	ระยะเวลา				
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ใช้โปรแกรมที่ออกแบบไว้ใน การทดลองและปรับพารามิเตอร์และทดสอบกับโมเดลต่าง ๆ					
2. เก็บข้อมูลที่ได้จากการทดลองและอ่านรีวิวกเพิ่มจากงานวิจัยที่ใกล้เคียงเพื่อนำมาปรับใช้					
3. เก็บผลและแก้ไขอัลกอริทึม โค้ด ที่เหมาะสมกับโมเดลที่เลือกสอน					
4. ทำการทดลองและเก็บผลให้โปรแกรมมีความแม่นยำสูงที่สุด					
5. สรุปผล และเขียนงานวิจัยคร่าว ๆ					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.8 โครงสร้างวิทยานิพนธ์

จากผลการศึกษาและงานวิจัยได้นำมาเสนอในวิทยานิพนธ์ฉบับนี้โดยทำการแบ่งเนื้อหาตาม
บทต่าง ๆ ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายวัตถุประสงค์
ของการศึกษา ทฤษฎีหรือแนวความคิดที่ใช้ในการทำวิจัย สมมุติฐานของการศึกษา ขอบเขตการศึกษา
แผนการดำเนินงาน ขั้นตอนและเวลาในการศึกษา รวมถึงโครงสร้างวิทยานิพนธ์โดยรวม เป็นต้น

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

บทที่ 3 ขั้นตอนและวิธีดำเนินการวิจัย

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 บทสรุปผลการทดลองและข้อเสนอแนะจากงานวิจัย



บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ภาพเอกซเรย์ปอด (Chest X-ray Imaging)

การถ่ายภาพเอกซเรย์ปอด (CXR) เป็นการตรวจวินิจฉัยทางรังสีวิทยาในรูปแบบหนึ่ง หลักการคือ เครื่องเอกซเรย์ฉายลำแสงที่มีพลังงานความถี่สูง ความยาวคลื่นสั้น ที่เรียกว่ารังสีเอกซ์ (X-ray) ผ่านไปยัง ส่วนบริเวณหน้าอกและปอด (Lung) เทคนิคทั่วไปที่นิยมใช้ในโรงพยาบาลโรงเรียนแพทย์ปัจจุบัน คือ 110 kVp 200-250 mA ผ่านตัววัตถุหรือในที่นี้คือ ผู้ป่วย ตกกระทบบนแผ่นฟิล์มหรือแผ่นรับภาพแบบ ดิจิตอลที่มีคุณสมบัติในการรับแสง ด้วยคุณสมบัติที่สามารถทะลุทะลวงผ่านวัตถุต่าง ๆ ได้ของรังสีเอกซ์ นี้เองทำให้ภาพมีความเข้มของการทะลุทะลวงผ่านวัตถุตัวกลางและไปตกกระทบบนแผ่นรับที่ให้ความเข้ม เป็นระดับสีเทา ขาว เทา ดำ โดยที่ถ้ารังสีผ่านวัตถุที่มีความหนาแน่นต่ำ รังสีสามารถทะลุทะลวงได้ดี เช่น อากาศ จะมองเห็นเป็นสีดำ ในทางกลับกันถ้าเริ่มมีความหนาแน่นขึ้น รังสีทะลุผ่านได้ยากขึ้นแต่ยังผ่าน ได้ เช่น เนื้อเยื่อกล้ามเนื้อ น้ำ จะเห็นเป็นสีเทา หรือกระดูก จะเห็นเป็นสีขาว จากความหนาแน่นสูง รังสี ทะลุผ่านได้น้อย จากคุณสมบัติดังกล่าวทำให้เกิดเป็นรูปร่างขึ้นส่งผลให้รังสีแพทย์ มองเห็นและแปลผลจากภาพเอกซเรย์เป็นรูปร่างอวัยวะได้ ทำการประเมินวิเคราะห์ภาพที่ผิดปกติได้ ง่ายมากขึ้น โดยอาศัยหลักการนี้เองทำให้ภาพปอดปกติ (CXR) สามารถมองเห็นภาพเอกซเรย์บริเวณ เนื้อปอดเป็นสีดำ จากพยาธิสภาพของปอดที่มีคุณลักษณะโปร่งประกอบด้วยถุงลมเล็ก ๆ มีอากาศบรรจุ อยู่ นั่น จะเห็นว่ากรณีเมื่อผู้ป่วยมีโรค เกิดความผิดปกติบริเวณเนื้อปอดหรือมีอาการปอดติดเชื้อ เช่น เกิดปอดอักเสบติดเชื้อปอดอักเสบโควิด (COVID-19) สภาพของเนื้อปอดจะเปลี่ยนไปด้วยเมื่อมี การอักเสบเกิดขึ้น ทำให้เห็นบริเวณที่ติดเชื้อมีสีขาวกว่าภาพปอดปกติ อาจเกิดขึ้นบางส่วนหรือบริเวณ ทั่วทั้งปอดได้เช่นกัน และด้วยเหตุประการนี้เองเมื่อรูปแบบในบริเวณพื้นที่ปอดที่เปลี่ยนไปช่วยให้แพทย์ วินิจฉัยโรคได้ด้วยหลักการทางการแพทย์โดยการเปรียบเทียบระหว่างภาพปกติที่มีอยู่กับภาพเอกซเรย์ ปอดที่ติดเชื้อโควิด (COVID-19) ดังตัวอย่างรูปที่ 2.1 (ก) ภาพเอกซเรย์ปอดเชื้อโควิด (COVID-19) รูป (ข) ภาพเอกซเรย์ปอดเชื้อทั่วไป (Pneumonia) และกลุ่ม (Negative) คือ ภาพเอกซเรย์ปอดที่ไม่ติดเชื้อ ดังตัวอย่างรูป(ค) ดังนี้



(ก) ปอดติดเชื้อโควิด COVID-19 (ข) ปอดติดเชื้อทั่วไป (ค) ปอดปกติ

รูปที่ 2.1 แสดงตัวอย่างรูปภาพเอกซเรย์ปอด CXR จากฐานข้อมูล

2.2 รูปแบบของข้อมูลที่ใช้ในการวิจัยและรอยโรคที่เกี่ยวข้อง

จากการศึกษาข้อมูลของผู้วิจัยพบว่าส่วนใหญ่โรงพยาบาลมีขั้นตอนการตรวจหาโรคด้วยการใช้ตัวทดสอบด้วยน้ำยาผ่านทางจมูกและปาก (RT-PCR), Swap กับภาพเอกซเรย์ปอดทั่วไปเป็นขั้นตอนการคัดแยกผู้ป่วยเบื้องต้นเพื่อตรวจหารอยโรคปอดอักเสบโควิด (COVID-19) ข้อมูลภาพที่นำมาใช้แปรผลเป็นภาพ 2D สองมิติ ขนาด 400x400 pixel เทคนิคการถ่ายหรือพารามิเตอร์ที่นิยมใช้ในปัจจุบันในโรงเรียนแพทย์ด้วยเครื่องเอกซเรย์ดิจิทัลให้ค่าประมาณ 110 kVp 200-250 mA ข้อมูลภาพเอกซเรย์ที่เรานำมาวิเคราะห์จะประกอบด้วยชนิดของรอยโรค 2 ตัวอย่าง จากภาพทั้งหมด 3 ตัวอย่าง เทียบกับภาพเอกซเรย์ปอดปกติ ดังนี้

1. ภาพเอกซเรย์ปอดปกติ (Normal)
2. ปอดอักเสบทั่วไป (Pneumonia)
3. ปอดอักเสบจากการติดเชื้อโควิด (COVID-19)

2.2.1 โรคปอดอักเสบจากเชื้อไวรัสโควิด (COVID-19)

ปอดอักเสบจากการติดเชื้อไวรัสนั้น สาเหตุทั่วไปจากเมื่อเวลาที่เราหายใจเอาละอองอากาศที่มีเชื้อโรคเข้าไปในปอด ซึ่งในอากาศนั้นมีเชื้อลอยปะปนอยู่มากมายตั้งแต่ไวรัส แบคทีเรีย เชื้อรา เชื้อวัณโรคหรือเชื้อโรคอื่น ๆ ที่ยังมีชีวิตและสามารถลอยอยู่ในอากาศได้ระยะเวลาหนึ่ง ซึ่งปกติโดยทั่วไปร่างกายของมนุษย์จะมีกลไกการกำจัดสิ่งแปลกปลอมหรือเชื้อโรคออกจากร่างกายได้อยู่แล้วโดยธรรมชาติขึ้นอยู่กับภูมิคุ้มกันของแต่ละบุคคลที่แต่ต่างกัน ในกรณีถ้าภูมิคุ้มกันไม่ดี ภูมิคุ้มกันบกพร่องหรือผู้ที่มีโรคประจำตัวเกี่ยวกับปอด เช่น โรคถุงลมโป่งพอง (Pulmonary Emphysema Disease) มะเร็งปอด (Cancer of Lung) เป็นต้น การได้รับปริมาณเชื้อเข้าร่างกายจำนวนมากพอและเชื้อมีความแข็งแรงในภาวะที่ร่างกายอ่อนแอร่วมด้วยเชื้ออาจสร้างความเสียหายและเข้าไปทำร้ายถุงลมในปอดได้ ทำให้ร่างกายเกิดปฏิกิริยาต่อต้านกับเชื้อจึงทำให้เกิดการอักเสบของปอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 อาการความแตกต่างระหว่างการติดเชื้อปอดอักเสบทั่วไปกับการติดเชื้อCOVID-19

พยาธิสภาพของอาการปอดอักเสบจากการติดเชื้อทั่วไปกับการติดเชื้อไวรัสโคโรนา (COVID-19) นั้นอาจจะสังเกตอาการได้ค่อนข้างยากสักหน่อย ในบางรายอาการเริ่มแรกอาจจะไม่แตกต่างกันกับหวัดมากนักส่วนใหญ่มักเริ่มต้นจากการมีไข้สูงติดต่อกันเป็นเวลานาน ๆ ต่อมาจะเริ่มมีอาการไอแห้ง ๆ และมีเสมหะ ในบางรายเวลาหายใจลึก ๆ อาจจะมีอาการเจ็บหน้าอกร่วมด้วย ซึ่งในรายที่อาการหนักเชื้อไวรัสเกิดการทำลายเนื้อปอด ส่งผลให้ส่วนที่แลกเปลี่ยนก๊าซเสียหาย ทำให้มีผู้ป่วยมีอาการหอบเหนื่อยตามมาการทำกิจวัตรประจำวันที่เคยเดิมไม่รู้สึกเหนื่อย เช่น เดิน หรือพูด จะรู้สึกเหนื่อยขึ้น ซึ่งสามารถแบ่งความแตกต่างตามอาการของโรคได้คร่าว ๆ ได้ตามตาราง ดังนี้

ตารางที่ 2.1 แสดงอาการระหว่างการติดเชื้อปอดอักเสบโคโรนา (COVID-19) และการติดเชื้อปอดอักเสบทั่วไป (Pneumonia)

ติดเชื้อปอดอักเสบโคโรนา COVID-19	ติดเชื้อปอดอักเสบทั่วไป
- มีไข้สูงเกิน 37.5 องศา ติดต่อกัน 48 ชม.	- มีไข้สูง แต่อาการจะค่อยๆดีขึ้น
- ไอ มีเสมหะ เจ็บคอนานเกิน 4 วัน หรือเสมหะอาจมีเลือดติดมาด้วย	- ไอ จาม เล็กน้อย อาการจะค่อยๆดีขึ้น
- อาจมีอาการคลื่นไส้ อาเจียน ท้องเสีย หายใจลำบาก เหนื่อยมาก	- ไม่มีอาการท้องเสีย
- กรณีอาการรุนแรงจะเกิดปอดอักเสบหรือปอดบวม	- คัดจมูก มีน้ำมูก หายใจไม่สะดวก
- มีอาการปอดเมื่อยตัวมาก ทานข้าวไม่ค่อยได้หรือไม่อร่อยในบางรายอาจเสียชีวิต จมูกไม่ได้กลิ่น	- อ่อนเพลีย ปวดตามตัวเล็กน้อย

2.2.3 โรคปอดอักเสบจากเชื้อไวรัสโคโรนา (COVID-19)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปอดอักเสบที่เกิดจากการติดเชื้อไวรัสโคโรนา (COVID-19) ส่วนใหญ่มักเกิดจากการอักเสบภายในเนื้อปอด ส่งผลโดยตรงต่อการแลกเปลี่ยนก๊าซออกซิเจน โดยในช่วงแรกผู้ป่วยจะไม่แสดงอาการใด ๆ ตูมภายนอกเหมือนปกติไม่มีอาการใด ๆ แสดงให้เห็นเด่นชัดแม้จะเกิดการอักเสบอยู่ภายในปอดที่เรียกว่า “Silent Hypoxemia” อาการต่อมาผู้ป่วยจะมีอาการหอบเหนื่อยหายใจลำบาก ในบางรายอาจรุนแรงถึงขั้นหายใจล้มเหลวและจำเป็นต้องใช้เครื่องช่วยหายใจหรือการให้ออกซิเจน เมื่อร่างกายรับเชื้อโควิด (COVID-19) เข้าไปความสามารถในการต่อสู้กับเชื้อโรคของแต่ละคนจะแตกต่างกัน ส่งผลให้มีโอกาสเกิด การเกิดปอดอักเสบ ได้โดยการแสดงควมรุนแรงหลังจากการติดเชื้อแต่ละคนก็ไม่เท่ากันขึ้นอยู่กับภูมิคุ้มกันของแต่ละบุคคล

2.2.4 สัญญาณเตือนของอาการติดเชื้อ

1. มีไข้สูงติดต่อกัน 48 ชั่วโมง
2. ไอแห้ง ๆ และเจ็บคอมาก
3. มีเสมหะ
4. อาจมีอาการหอบเหนื่อย
5. รู้สึกหายใจต้องใช้แรงมาก
6. ทำกิจวัตรประจำวันปกติไม่เหนื่อย แต่ตอนนี้ทำอะไรนิดหน่อยเหนื่อยกว่าปกติที่เคยเป็น

2.2.5 กลุ่มเสี่ยงปอดอักเสบรุนแรงจากเชื้อไวรัสโคโรนา (COVID-19)

ภาวะกลุ่มผู้ที่ถ้าหากได้รับเชื้อโควิด (COVID-19) มีความเสี่ยงที่ปอดอักเสบรุนแรง เนื่องจากแต่เดิมปอดทำงานไม่มีประสิทธิภาพอยู่แล้ว และอาจส่งผลให้การเสียชีวิตได้ถ้าไม่ได้รับการรักษาที่ถูกต้อง ได้แก่

1. ผู้สูงอายุ เนื่องจากอายุที่มากขึ้นอวัยวะภายในร่างกายจะเริ่มเสื่อมสภาพลงตามไปด้วยทำให้ภูมิคุ้มกันไม่แข็งแรง
2. ผู้ที่มีประวัติเป็นโรคเกี่ยวกับปอด (Lung) เรื้อรัง ซึ่งมีอยู่ด้วยกันหลายโรค เช่น โรคถุงลมโป่งพอง (Pulmonary Emphysema Disease), หอบหืด (Asthma), หรือมะเร็งปอด (CA Lung) เป็นต้น

3. ผู้ที่มีภาวะโรคอ้วน หรือน้ำหนักเกินกว่ามาตรฐาน ค่าระดับดัชนีมวลกาย (Body mass index) หรือ BMI ตั้งแต่ 25 ขึ้นไป
4. ผู้ที่มีโรคประจำตัวเรื้อรัง ที่เพิ่มความเสี่ยงปอดอักเสบ เช่น โรคตับ (Cirrhosis), ความดันโลหิตสูง (HT), โรคไตวายเรื้อรัง (CKD), โรคเบาหวาน (DM), โรคหัวใจวาย (STEMI) และมะเร็ง (CA) เป็นต้น

2.2.6 แนวทางการรักษาปอดอักเสบในผู้ป่วยโควิด (COVID-19)

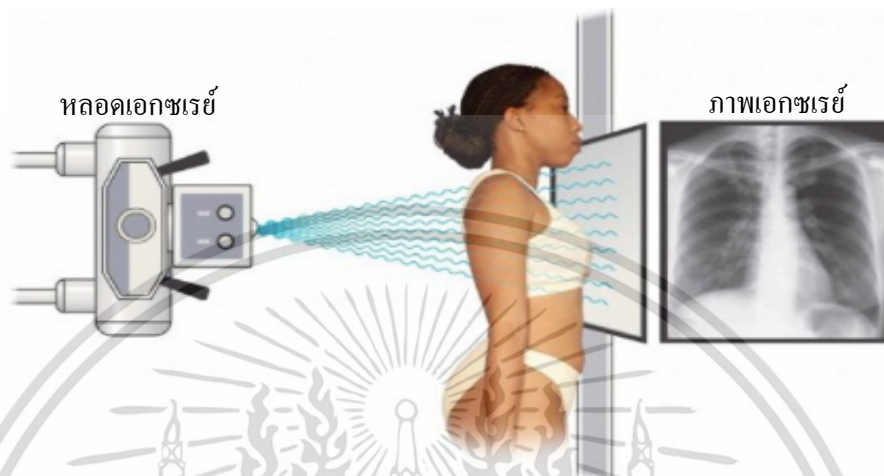
1. ภาวะปอดอักเสบไม่รุนแรง แนวทางการรักษาโดยแพทย์จะให้ยาต้านไวรัสและพิจารณาให้ยาต้านการอักเสบร่วมด้วย ในรายที่มีความอึดตัวของออกซิเจนในเลือดต่ำ
2. ภาวะปอดอักเสบรุนแรง แพทย์พิจารณาให้ยาต้านไวรัสและยาต้านการอักเสบโดยมีข้อบ่งชี้ในการใช้คือ มีภาวะอัตราไหลออกซิเจนต่ำอัตราไหลออกซิเจนสูง ใช้เครื่องช่วยหายใจ และในผู้ป่วยอาการหนักบางรายอาจจำเป็นต้องใช้เครื่องช่วยพยุงการทำงานของหัวใจและปอด (ECMO) แล้วแต่กรณี
3. แพทย์จะพิจารณา ทำกายภาพบำบัดเพื่อฟื้นฟูสมรรถภาพปอดเมื่ออาการปอดอักเสบดีขึ้นจะเปลี่ยนไปตรวจรักษาและติดตามเป็นแบบผู้ป่วยนอก (OPD) หากเริ่มแข็งแรงและสามารถดูแลตัวเองได้แพทย์ก็จะพิจารณาให้กลับไปดูแลต่อเองที่บ้าน

2.3 หลักการทำงานและส่วนประกอบเครื่องเอกซเรย์ปอดทั่วไป

ปัจจุบันเครื่องเอกซเรย์ทั่วไปใช้ประโยชน์จากคุณสมบัติของรังสีเอกซ์เป็นพื้นฐานสำหรับการทำงานทางด้านรังสีวิทยาทางการแพทย์ การรักษาด้วยรังสี การวิจัยและการพัฒนาต่าง ๆ โดยหลัก ๆ เพื่อวัตถุประสงค์ในการถ่ายภาพในด้านอื่น ๆ เช่น ใช้ทางด้านวิศวกรรมต่าง ๆ เครื่องเอกซเรย์เป็นหนึ่งในอุปกรณ์ทางการแพทย์ที่พบมากที่สุดที่ใช้ในโรงพยาบาลและห้องปฏิบัติการเพื่อวินิจฉัยการติดเชื้อ การติดตามอาการและค้นหาการแตกหักของอวัยวะของร่างกายได้เป็นอย่างดี ซึ่งนั่นเป็นผลดีที่ทำให้แพทย์สามารถวินิจฉัยโรคได้ง่าย แม่นยำ และรวดเร็วมากขึ้น

รังสีเอกซ์เป็นรังสีแม่เหล็กไฟฟ้าที่อยู่ในช่วงความถี่ 30 EHz-30 Phz ช่วงความยาวคลื่นของรังสีเอกซ์อยู่ระหว่าง 0.01 นาโนเมตรถึง 10 นาโนเมตร ในสเปกตรัมแม่เหล็กไฟฟ้ารังสีเอกซ์ตั้งอยู่ระหว่างแสงอัลตราไวโอเล็ตและรังสีแกมมา รังสีเอกซ์สามารถแบ่งออกเป็นสองประเภทได้คือ รังสีเอกซ์อ่อนและรังสีเอกซ์แข็ง ค่าของความยาวคลื่นของรังสีเอกซ์แข็งเท่ากับ 100 picometers ในขณะที่รังสีเอกซ์อ่อนมีความยาวคลื่นที่ค่อนข้างสั้นกว่าประมาณ 10 นาโนเมตร รังสีเอกซ์ถูกค้นพบในปี 1895 โดย

Wilhelm Conrad Röntgen ศาสตราจารย์ที่ Würzburg University ในประเทศเยอรมนี รังสีเอกซ์ จะผลิตภายในของหลอดเอกซเรย์การฉายรังสีผ่านไปยังตัวของผู้ป่วยไปตกกระทบที่แผ่นรับภาพ [10,11] ดังรูป 2.2

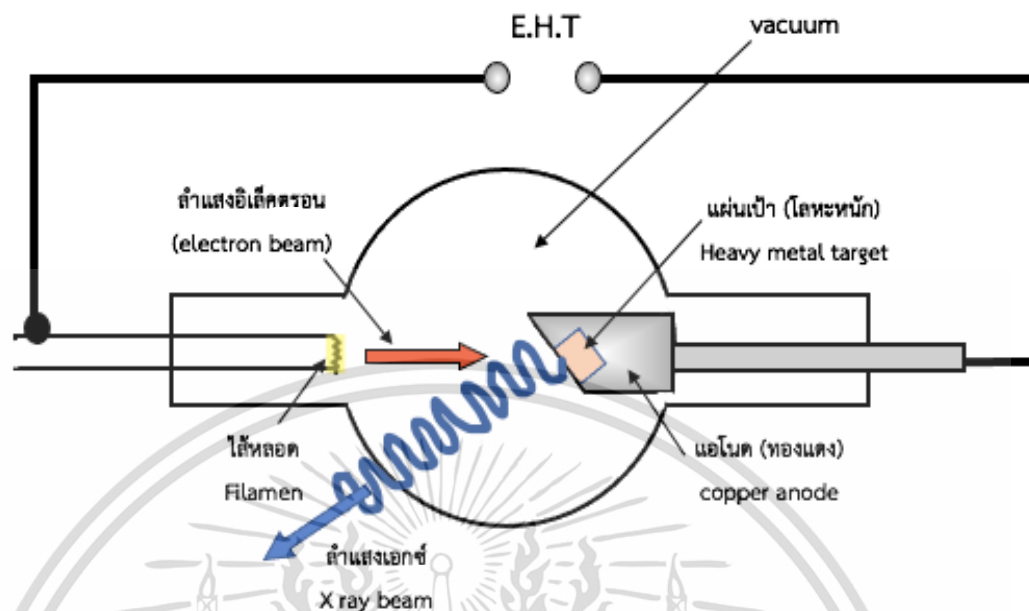


รูปที่ 2.2 แสดงลักษณะที่ลำแสงเอกซ์ผ่านตัวผู้ป่วยไปตกกระทบบนแผ่นรับภาพ (Detector) (ที่มา: <https://bacsi247.org/blog/di-chup-x-quang-co-phai-coi-quan-ao-khong>)

2.3.1 ส่วนประกอบพื้นฐานของเครื่องเอกซเรย์ทั่วไป

เครื่องกำเนิดรังสีเอกซ์เป็นอุปกรณ์ที่ใช้ในการสร้างลำแสงเอกซ์ระบบการถ่ายภาพเอกซเรย์ X-ray ประกอบด้วยแหล่งกำเนิดรังสีเอกซ์หรือเครื่องกำเนิดไฟฟ้า (หลอด X-ray) และระบบตรวจจับภาพท้อเอกซเรย์ (ไดโอดสูญญากาศสูง) ทำงานโดยการแปลงอิเล็กตรอนจากเส้นใยทังสเตนแคโทดที่ร้อนไปยังแผ่นดิสก์ขั้วบวกแรงดันไฟฟ้าสูงจุดที่อิเล็กตรอน (ลำแสง) โจมตีเป้าหมายเรียกว่า จุดโฟกัส ที่โฟตอน X-Ray โฟกัสจะถูกนำไปทุกทิศทาง รังสีเอกซ์เกิดขึ้นจากแผ่นดิสก์เป้าหมายที่มุมฉากและมุ่งเน้นไปที่ collimator สำหรับความคมชัดของการดูมากขึ้นเราใช้ Photomultipliers ภาพได้รับและดูบนแผ่นถ่ายภาพที่มีพื้นที่เบาและมีดบนฟิล์มแสดงถึงการเจาะเนื้อเยื่อสูงและต่ำ แผ่นผังพื้นฐานของหลอดเอกซเรย์แสดงอยู่ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงส่วนประกอบโดยทั่วไปของหลอดเอกซเรย์ (X-ray Tube)

2.4 ฮาร์ดแวร์ (Hardware)

2.4.1 CPU: (Central Processing Unit) [22] หน่วยประมวลผลกลาง เปรียบเสมือนสมองกลของเครื่องคอมพิวเตอร์ โดยจะรับชุดคำสั่งผ่านหน่วยความจำ RAM ที่มีการส่งข้อมูลชุดคำสั่งมาจาก Hard Disk Storage Device อุปกรณ์สำหรับจัดเก็บข้อมูล และ Peripheral Device อุปกรณ์ต่อพ่วงต่าง ๆ เช่น Mouse, Keyboard, Printer, USB Drive เป็นต้น

2.4.2 GPU: (Graphics Processing Unit) [22] หน่วยประมวลผลภาพกราฟิก นิยมเรียกกันว่าการ์ดจอเป็นชุดอุปกรณ์ที่มีความสำคัญสำหรับคอมพิวเตอร์ชนิดหนึ่ง ทำหน้าที่ประมวลผลส่งข้อมูลสัญญาณภาพให้กับหน้าจอ Monitor ในรูปแบบภาพนิ่ง และ ภาพเคลื่อนไหว ไฟล์วิดีโอต่าง ๆ

- CPU ใช้หลักการประมวลผลแบบ Linear ประมวลผลและแสดงผลออกมาตามลำดับของชุดคำสั่งที่เข้ามา
- GPU ใช้หลักการประมวลผลแบบ Parallel สามารถรับข้อมูลชุดคำสั่งมาได้ทั้งหมดและแสดงผลพร้อมกันทีเดียวได้เลย

2.5 ซอฟต์แวร์ (Software)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 ภาษา Python

ไลบรารีคือ ชุดของโค้ดที่ใช้บ่อยซึ่งนักพัฒนาสามารถใช้ในโปรแกรมภาษา Python เพื่อหลีกเลี่ยงการเขียนโค้ดขึ้นใหม่ทั้งหมดตามค่าเริ่มต้นแล้ว Python จะมาพร้อมกับไลบรารีมาตรฐานที่มีฟังก์ชันที่น่ากลับมาใช้ใหม่ได้มากมายนอกจากนี้ยังมีไลบรารี Python มากกว่า 137,000 รายการสำหรับการประยุกต์ใช้ต่าง ๆ รวมถึงการพัฒนาเว็บวิทยาศาสตร์ข้อมูลและ (ML)

2.5.2 Pandas

Pandas มาจากคำว่า Panel Data (ชุดข้อมูลหลายมิติ) มีจุดเด่นด้านการวิเคราะห์ข้อมูล (Data Analysis) มีโครงสร้างข้อมูลที่เหมาะสมและยืดหยุ่น คุณสามารถใช้เพื่อจัดการชุดข้อมูลเวลาและข้อมูลที่มีโครงสร้าง เช่น ตารางและอาร์เรย์ ตัวอย่างเช่น คุณสามารถใช้ Pandas เพื่ออ่านเขียน ผสานรวม กรอง และจัดกลุ่มข้อมูลได้ หลายคนจึงใช้สำหรับงานด้านวิทยาศาสตร์ข้อมูลการวิเคราะห์ข้อมูล และ ML

2.5.3 NumPy

NumPy(Numeric Python) เป็นโมดูลส่วนเสริมของ Python ที่มีฟังก์ชันเกี่ยวกับทางคณิตศาสตร์และการคำนวณต่าง ๆ มาใช้งานเป็นไลบรารียอดนิยมที่นักพัฒนาใช้เพื่อสร้างและจัดการอาร์เรย์ จัดการรูปร่างเชิง ตรรกะและดำเนินการคำนวณพีชคณิตเชิงเส้นได้อย่างง่ายดายโดย NumPy รองรับการทำงานร่วมกับภาษาต่าง ๆ มากมาย เช่น C และ C++

2.5.4 OpenCV-Python

OpenCV-Python เป็นไลบรารีที่นักพัฒนาใช้ในการประมวลผลรูปภาพสำหรับการประยุกต์ใช้เชิงคอมพิวเตอร์วิทัศน์ ที่มีฟังก์ชันมากมายสำหรับงานประมวลผลภาพ เช่น การอ่านและเขียน รูปภาพพร้อมกัน การสร้างสภาพแวดล้อม 3 มิติจาก 2 มิติ และการบันทึกภาพและวิเคราะห์ภาพจากวิดีโอ

2.5.5 Keras

Keras เป็นไลบรารี Neural Network เชิงลึกของ Python ที่รองรับการประมวลผลข้อมูล การแสดงภาพข้อมูลและอื่น ๆ อีกมากมายได้อย่างยอดเยี่ยม Keras รองรับ Neural Network มากมาย ซึ่งมีโครงสร้างแบบโมดูลที่ให้ความยืดหยุ่นในการเขียน Application ด้านนวัตกรรม

2.6 ชนิดของภาพ

2.6.1 DICOM file image [23] Digital Imaging and Communications in Medicine (DICOM) เป็นมาตรฐานที่กำหนดโดย National Electrical Manufacturers Association (NEMA) โดยมีจุดประสงค์เพื่อการเผยแพร่ ภาพทางการแพทย์ เช่น CT scans, MRIs, CR, หรือ US และ ภาพทางการแพทย์ อื่น ๆ

รูปภาพที่แสดงจะประกอบไปด้วยจุดจำนวนมากเรียกแต่ละจุดนี้ว่า “pixels” ซึ่งจำนวน pixels นี้ก็ขึ้นอยู่กับชนิดของ Digital medical image ที่เราใช้ เช่น ถ้าเป็นของ CT scan จะเป็น 512x512 pixels ขณะที่ของ MRI จะเป็น 256x256 pixels โดยพื้นที่ในการจัดเก็บและ Resolution ของ Digital image ต่าง ๆ เป็นไปตามตารางนี้

2.6.2 JPEG Joint Photographic Experts Group (JPEG) เป็นรูปภาพดิจิทัลที่นิยมใช้กันบ่อยที่สุดทางเลือกรูปหนึ่ง ในการถ่ายและจัดเก็บรูปภาพ ไฟล์ JPEG จะผ่านกระบวนการบีบอัด เพื่อลดขนาดของไฟล์รูปภาพลงอย่างมาก จึงง่ายต่อการจัดเก็บและโหลดบนหน้าเว็บ โดยรูปภาพ JPEG นั้นสามารถเก็บสีได้สูงสุดถึง 16 ล้านสี

2.6.3 PNG Portable Network Graphics (PNG) เป็นไฟล์ที่สามารถบีบอัดได้และรองรับสีได้ 16 ล้านสีเช่นเดียวกับ JPEG โดยไฟล์ประเภทนี้มักจะใช้สำหรับกราฟิกบนเว็บ โลโก้ แผนภูมิ และภาพประกอบมากกว่าที่จะใช้สำหรับภาพถ่ายคุณภาพสูง เนื่องจากไฟล์ประเภทนี้ใช้พื้นที่เก็บข้อมูลมากกว่า JPEG สิ่งหนึ่งที่ไฟล์ PNG มีแต่ JPEG ไม่มีก็คือความสามารถในการรองรับกราฟิกที่มีพื้นหลังโปร่งใส

ความแตกต่างระหว่างภาพ JPEG และ PNG

แม้จะมีลักษณะใกล้เคียงกันและเป็นที่ยอมรับเหมือนกันแต่ไฟล์ JPEG และ PNG ก็มีความแตกต่างกันอยู่มากมาย เนื่องจากทั้งคู่มีกระบวนการบีบอัดที่ไม่เหมือนกัน ไฟล์ JPEG จึงมีข้อมูลน้อยกว่า PNG ดังนั้น ไฟล์ JPEG จึงมีขนาดเล็กกว่า ทั้งนี้ ไฟล์ PNG นั้นรองรับพื้นหลังแบบโปร่งใส จึงเป็นที่นิยมสำหรับการออกแบบกราฟิก ไม่เหมือนกับไฟล์ JPEG การบีบอัดแบบสูญเสียข้อมูลเทียบกับแบบไม่สูญเสียข้อมูล เมื่อต้องพิจารณาว่าจะใช้ไฟล์ประเภท JPEG หรือ PNG ดี คุณต้องทำความเข้าใจเสียก่อนว่าไฟล์แต่ละประเภทใช้กระบวนการบีบอัดที่ต่างกันไฟล์ JPEG ได้รับการออกแบบมาเพื่อจัดเก็บภาพถ่ายดิจิทัลคุณภาพสูงที่เต็มไปด้วยรายละเอียดและสีกันอย่างมีประสิทธิภาพ ไฟล์ประเภทนี้จะบีบอัดรูปภาพขนาดใหญ่ให้เป็นไฟล์ขนาดเล็กกว่าเดิมมาก จึงสามารถแชร์และอัปโหลดไปยังโลกออนไลน์ได้อย่างง่ายดาย

ข้อเสียของไฟล์ JPEG

ไฟล์ JPEG ใช้กระบวนการบีบอัดแบบสูญเสียข้อมูล ซึ่งหมายความว่าข้อมูลบางอย่างของรูปภาพจะถูกกลบไปอย่างถาวรเมื่อทำให้มีขนาดเล็กลง ซึ่งอาจส่งผลกระทบต่อคุณภาพของไฟล์ในระยะยาว เนื่องจากทุกครั้งที่คุณแก้ไขและบันทึกไฟล์ คุณจะสูญเสียข้อมูลมากขึ้น ด้วยเหตุนี้ ช่างภาพมืออาชีพบางรายจึงนิยมใช้ไฟล์ RAW ที่ไม่ผ่านการบีบอัดมากกว่าในทางกลับกัน ไฟล์ PNG ใช้ประโยชน์จากการบีบอัดแบบไม่สูญเสียข้อมูล ดังนั้นไฟล์ PNG จะไม่สูญเสียข้อมูลใด ๆ เมื่อบีบอัดรูปภาพ คุณภาพจะยังคงเหมือนเดิม ไม่ว่าจะแก้ไขและบันทึกไฟล์กี่ครั้งก็ตาม รูปภาพดังกล่าวจะไม่เบลอหรือผิดเพี้ยน ไฟล์ PNG จึงเหมาะสำหรับโลโก้ที่คมชัดและแผนภูมิที่มีตัวเลขจำนวนมาก การบีบอัดแบบสูญเสียข้อมูลอาจส่งผลให้คุณภาพลดลง แต่ไฟล์ JPEG ก็สามารถย่อรูปภาพขนาดใหญ่ให้มีขนาดที่จัดการได้ง่ายขึ้น ซึ่งจะเป็นประโยชน์ในกรณีที่คุณไม่มีพื้นที่จัดเก็บในดิสก์ให้ใช้งานมากมาย ทั้งยังช่วยให้โหลดหน้าเว็บได้รวดเร็วขึ้นอีกด้วย

ข้อเสียของไฟล์ PNG

คือการบีบอัดแบบไม่สูญเสียข้อมูล ซึ่งจะทำให้ไฟล์ PNG มีขนาดใหญ่กว่า เนื่องจากไฟล์ประเภทนี้เก็บข้อมูลไว้อย่างละเอียด โดยปกติแล้วไฟล์ประเภทนี้จะมีขนาดใหญ่กว่า JPEG และ GIF ทั้งยังใช้พื้นที่เก็บข้อมูลมากกว่า และอาจทำให้หน้าเว็บตอบสนองช้าลงได้

ความโปร่งใส

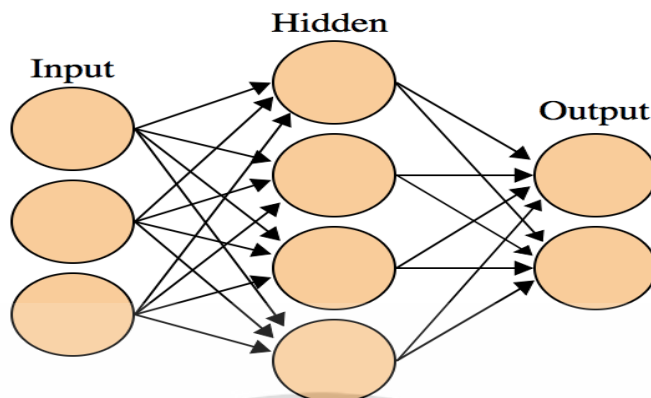
ระหว่างไฟล์ JPEG กับ PNG คือความสามารถในการรองรับความโปร่งใสในรูปภาพไฟล์ JPEG ไม่รองรับพื้นหลังแบบโปร่งใส โลโก้และกราฟิกที่ไม่ใช่สี่เหลี่ยมผืนผ้าและมีข้อความจำนวนมากนั้นจึงมักไม่เหมาะกับรูปแบบไฟล์นี้ นอกจากนี้ รูปภาพ JPEG ยังมีปัญหาในเรื่องของการทำให้กลมกลืนไปกับหน้าเว็บที่มีสีพื้นหลังหลากหลายสีอีกด้วยในทางกลับกัน ไฟล์ PNG นั้นรองรับความโปร่งใส อีกทั้งยังมีความโปร่งใสหลายระดับให้เลือกใช้อีกด้วย ดังนั้นรูปภาพ PNG จะผสานเข้ากับสีพื้นหลังต่าง ๆ ของหน้าเว็บได้ดีกว่าและข้อความก็จะอ่านได้ง่ายกว่า

ตารางที่ 2.2 แสดงตัวอย่างของชนิดและขนาดของภาพดิจิทัล

Sizes of image produced by digital imaging modalities			
Modalities	Pixel matrix	Total pixels	Size of images
Nuclear medicine	128 × 128	16,384	0.16 MP
MRI	256 × 256	65,536	0.06 MP
USG	512 × 512	262,144	0.25 MP
DSA	512 × 512	262,144	0.25 MP
CT scan	512 × 512	262,144	0.25 MP
CR/DR	2048 × 2048	4,194,304	4 MP
Digital mammography	4000 × 5000	20,000,000	81 MP
Digital cameras	User defined	User defined	3-20 MP

2.7 โครงข่ายประสาทเทียม (ANN)

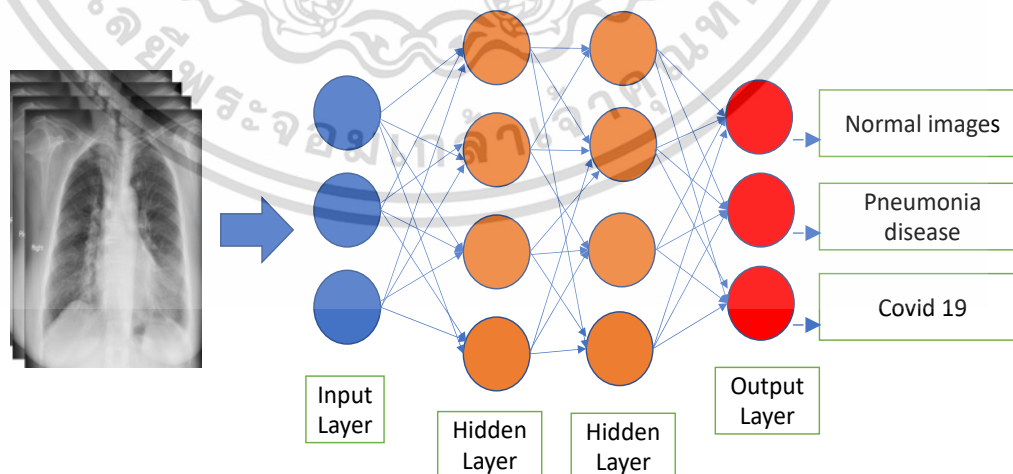
โครงข่ายประสาทเทียม (Artificial neural networks: ANN) หรือ ข่ายงานประสาทเทียม (Connectionist systems) คือระบบคอมพิวเตอร์จากโมเดลทางคณิตศาสตร์เพื่อจำลองการทำงานโครงข่ายประสาทชีวภาพที่อยู่ในสมองของสัตว์[26] หลักการคือ เมื่อผู้เขียนโปรแกรมป้อนเข้าข้อมูลเข้าในระบบคอมพิวเตอร์เรียนรู้ข้อมูลแบบอัตโนมัติแล้วทำการประมวลผลผลลัพธ์ออกมา โครงข่ายประสาทเทียมประกอบด้วยชั้นดำเนินการหลัก 3 ชั้น ได้แก่ ชั้นข้อมูลนำเข้า (input layer) ชั้นการประมวลผลซ่อน (hidden layer) และชั้นแสดงผล (output layer) โดยในชั้นข้อมูลนำเข้ามีโหนดรับค่าข้อมูลที่จะทำการประมวลผลชั้นการประมวลผลซ่อนนั้นประกอบไปด้วยการเรียงตัวของเพอร์เซปตรอนหลายเพอร์เซปตรอนโดยแต่ละตัวจะเชื่อมกับทุกโหนดในชั้นข้อมูลนำเข้าและเชื่อมกับทุกเพอร์เซปตรอนในชั้นการประมวลผลซ่อนชั้นถัดไปโดยในโครงข่ายประสาทเทียมจะมีชั้นการประมวลผลซ่อนหลายชั้นได้ขึ้นอยู่กับความต้องการประมวลผลชั้นผลลัพธ์จะเป็นชั้นสุดท้ายของโครงข่ายประสาทเทียม โดยมีจำนวนโหนดตามรูปแบบของปัญหาที่ต้องการประมวลผล โดยแต่ละโหนดจะเชื่อมกับทุกโหนดในชั้นการประมวลผลซ่อนสุดท้ายแสดงการเชื่อมต่อของโครงข่ายประสาทเทียมดังภาพที่ 2.4



รูปที่ 2.4 แสดงโครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)

(ที่มา: <https://th.wikipedia.org/wiki/โครงข่ายประสาทเทียม>)

เมื่อชั้นการประมวลผลซ่อน (Hidden Layer) อยู่มีหลาย ๆ ชั้นมากขึ้นเรียกว่า โครงข่ายประสาทเทียมคอนโวลูชันหรือ (Convolution Neural Networks) เป็นโครงข่ายประสาทเทียมชนิดหนึ่งที่มีการดำเนินการประมวลผลทางคณิตศาสตร์ที่เรียกว่า คอนโวลูชัน (Conv) กระบวนการจำแนกภาพเทคนิคการเรียนรู้เชิงลึก (CNN: Deep-learning) จะประยุกต์ใช้โครงข่ายประสาทเทียมหลาย ๆ ชั้น เพื่อค้นหาคุณลักษณะและการทำซ้ำหลาย ๆ รอบจนกระทั่งได้ลักษณะที่มีความถูกต้องที่สุดในการจำแนกจะแสดงค่าความถูกต้องเป็นค่าความน่าจะเป็น (Probability) เช่น เมื่อนำภาพเอกซเรย์ปอดเข้าสู่กระบวนการจำแนกของเทคนิค (Deep-learning) ผลลัพธ์จะได้เป็นค่าความน่าจะเป็นของการจำแนกรูปภาพเอกซเรย์ปอดว่าเป็นรูปภาพเอกซเรย์ของผู้ติดเชื้อโควิดหรือเป็นภาพเอกซเรย์ทรวงอกของผู้ที่ไม่ติดเชื้อโควิด (COVID-19) กระบวนการเทคนิค CNN (Deep-learning) แสดงดังรูปที่ 2 มีรายละเอียดดังนี้

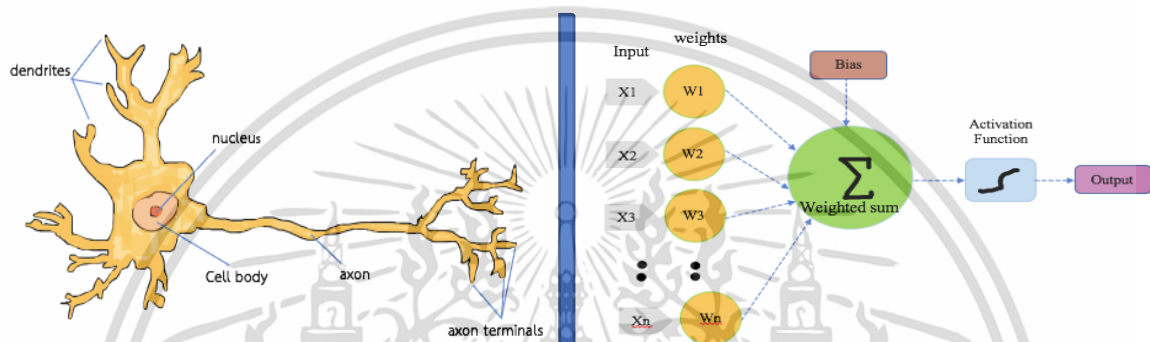


รูปที่ 2.5 แสดงโครงข่ายประสาทเทียมแบบคอนโวลูชัน CNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 เพอร์เซปตรอนหรือนิวรอน

เพอร์เซปตรอน (perceptron) เป็นหน่วยการประมวลผลที่เล็กที่สุดของการเรียนรู้เชิงลึกหรือข่ายประสาทเทียมมีแนวคิดการทำงานคล้ายกับเซลล์ประสาทสมองของมนุษย์ ประกอบด้วยส่วนรับข้อมูลนำเข้า (input) ส่วนค่าถ่วงน้ำหนัก (weights) ส่วนรวมผลคูณของค่าข้อมูลนำเข้าและค่าถ่วงน้ำหนัก (weighted sum) ส่วนค่าเบี่ยงเบน (bias) และส่วนฟังก์ชันกระตุ้น (activation function) แสดงดังภาพ 2.6



$$\text{output} = \text{activation_function}(\text{sum}[\text{inputs} * \text{weights}] + \text{bias})$$

รูปที่ 2.6 แสดงการทำงานของเซลล์ประสาทของสัตว์กับ ANN System (perceptron)

2.8.1 ค่าถ่วงน้ำหนัก

ค่าถ่วงน้ำหนัก (weight) เป็นค่าพารามิเตอร์ที่เป็นส่วนประกอบของเพอร์เซปตรอนซึ่งจะถูกปรับเปลี่ยนค่าเมื่อมีการเรียนรู้ของโมเดลโดยค่าถ่วงน้ำหนักจะถูกคูณเข้ากับค่าข้อมูลสำหรับแต่ละหน่วยของข้อมูลนำเข้าเพื่อแสดงระดับความสำคัญของคุณลักษณะค่าข้อมูลนำเข้าก่อนที่จะถูกรวมและส่งผ่านฟังก์ชันกระตุ้นต่อไปจากสมการ

$$\sum_{i=0}^{i=n} X * \omega + b. \quad (1)$$

$$\text{output} = \text{activation_function}(\text{sum}[\text{inputs} * \text{weights}] + \text{bias})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 ค่าเบี่ยงเบน

ค่าเบี่ยงเบน (bias) เป็นค่าพารามิเตอร์ที่มี 1 ตัว ที่สามารถถูกปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้ของโมเดลโดยค่าเบี่ยงเบนจะถูกรวมเข้ากับผลรวมของผลคูณระหว่างค่าถ่วงน้ำหนักกับค่าข้อมูลนำเข้าทั้งหมดที่เข้าสู่เพอร์เซปตรอน ความสำคัญของค่าเบี่ยงเบนใช้สำหรับบอกความสำคัญของเพอร์เซปตรอนในโครงข่ายที่ประกอบด้วยเพอร์เซปตรอนหลายตัว

2.8.3 ฟังก์ชันกระตุ้น

ฟังก์ชันกระตุ้น(activation function) เป็นส่วนหนึ่งของเพอร์เซปตรอนทำหน้าที่เป็นจุดกำหนดการส่งผ่านค่าข้อมูลไปยังเพอร์เซปตรอนอื่น ๆ ด้วยฟังก์ชันทางคณิตศาสตร์ช่วยให้การส่งผ่านค่าข้อมูลที่ผ่านมาดำเนินการทางคณิตศาสตร์ด้วยค่าถ่วงน้ำหนักและค่าเบี่ยงเบนไปยังเพอร์เซปตรอนอื่นไม่เป็นเชิงเส้นและในระหว่างการสอนโมเดลฟังก์ชันกระตุ้นมีบทบาทสำคัญในการปรับแก้เรียนคุณสมบัติของฟังก์ชันกระตุ้นดังกล่าวนี้ การเปิดใช้งานเป็นฟังก์ชันที่เพิ่มเข้าไปในเครือข่ายประสาทเทียมเพื่อช่วยให้เครือข่ายเรียนรู้รูปแบบที่ซับซ้อนในข้อมูล เมื่อเปรียบเทียบกับโมเดลที่ใช้เซลล์ประสาทที่อยู่ในสมองของเราฟังก์ชันการเปิดใช้งานอยู่ที่จุดสิ้นสุดการตัดสินใจว่าจะถูกยิงไปยังเซลล์ประสาทถัดไป นั่นคือ สิ่งที่ฟังก์ชันการเปิดใช้งานทำในแอกซอนเช่นกัน ใช้ในสัญญาณเอาต์พุตจากเซลล์ก่อนหน้าและแปลงเป็นบางรูปแบบที่สามารถนำมาเป็นอินพุตไปยังเซลล์ถัดไป ขั้นตอนการของฟังก์ชันกระตุ้น มีบทบาทเหมือนประตูทางผ่านของสัญญาณระหว่างชั้นก่อนหน้าและชั้นถัดไป การควบคุมของชั้นฟังก์ชันนี้ขึ้นอยู่กับฟังก์ชันทางคณิตศาสตร์ที่ผู้ใช้งานกำหนด เช่น ฟังก์ชัน ReLU สำหรับชั้นการประมวลผลซ่อน ฟังก์ชัน softmax สำหรับชั้นสุดท้ายของโครงข่ายเพื่อทำนายความน่าจะเป็นของผลลัพธ์ เป็นต้น

สามารถอธิบายได้ตามขั้นตอนด้านล่างดังนี้

1. รับข้อมูล Input ที่ถูกถ่วงน้ำหนัก (Weighted Input) เข้ามาในโหนด
2. นำ Weighted Input ทุก ๆ อันมารวมกัน (Sum of Weighted Input)
3. นำค่าที่ได้จากการรวมกันในโหนดก่อนหน้าไปเข้าโหนด Activation Function
4. และนำค่าที่ออกมาจาก Activation Function ส่งเข้าไปยังโหนดต่อไป

ฟังก์ชัน Sigmoid มีการดำเนินการทางคณิตศาสตร์ ผลลัพธ์จากการดำเนินการของฟังก์ชันมีค่าระหว่าง 0 ถึง 1 ข้อจำกัดของฟังก์ชัน Sigmoid คือระดับความชันของฟังก์ชันน้อย ส่งผลให้การลู่เข้าช้า

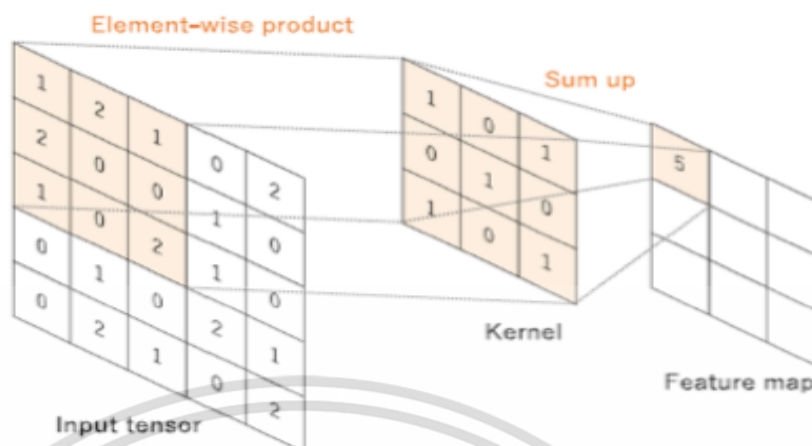
ฟังก์ชัน Tanh มีการดำเนินการทางคณิตศาสตร์ ผลลัพธ์จากการดำเนินการของฟังก์ชันมีค่าระหว่าง -1 ถึง 1 ค่ากึ่งกลางอยู่ที่ 0 ส่งผลให้การ optimization นั้นทำได้ง่ายแต่ก็ยังมีข้อจำกัดที่ระดับความชันของฟังก์ชันน้อย

ฟังก์ชัน Rectified linear unit: (ReLU) มีการดำเนินการทางคณิตศาสตร์ ผลลัพธ์จากการดำเนินการของฟังก์ชันมีค่าเท่ากับ $[0, \infty)$ จุดเด่นของฟังก์ชันคือ มีระดับความชันส่งผลให้การลู่เข้าดีกว่าเมื่อเปรียบเทียบกับฟังก์ชัน Sigmoid และ Tanh ข้อจำกัดคือ ใช้ได้เฉพาะชั้นการประมวลผลที่มองไม่เห็น (Hidden layer) ไม่สามารถใช้ในชั้นข้อมูลขาออกสุดท้ายของโมเดลได้ จึงจำเป็นต้องใช้ฟังก์ชัน SoftMax ในชั้นสุดท้ายแทนเพื่อคำนวณความน่าจะเป็น และอีก 1 ข้อจำกัดของฟังก์ชัน ReLU คือสามารถทำให้เกิดปัญหาการตายของเซลล์ประสาทได้ หากมีค่าข้อมูลสูงไหลผ่าน ReLU อาจทำให้การถ่วงน้ำหนักถูกปรับปรุงมาก ส่งผลให้เซลล์ประสาทหยุดทำงานเมื่อมีชุดข้อมูลใหม่ไหลผ่าน

ฟังก์ชัน Softmax เป็นฟังก์ชันที่มีการดำเนินการทางคณิตศาสตร์ โดยผลลัพธ์อยู่ในรูปแบบความน่าจะเป็น ฟังก์ชัน softmax จะถูกใช้ในชั้นสุดท้ายเมื่อใช้ฟังก์ชัน ReLU สำหรับการแก้ปัญหาการจำแนกประเภทของข้อมูล

2.9 การทำคอนโวลูชัน Convolution Layer/Filter (CONV)

การคอนโวลูชัน (Conv) เป็นขั้นแรก ๆ ของโมเดล CNN ที่มีหน้าที่แยกสกัดคุณลักษณะเด่นของภาพต่าง ๆ ออกจากกัน เช่น เส้นขอบของวัตถุต่าง ๆ เป็นต้น โดยใช้หลักการจากการทำงานของตัวเลขอาร์เรย์ที่เรียกว่า kernel โดยที่การที่โมเดลจะสามารถแยกลักษณะของภาพอย่างมีประสิทธิภาพและแม่นยำได้นั้นตัว kernel จะต้องถูกนำเข้าไปใช้กับ Input ที่เป็นอาร์เรย์ของตัวเลข เรียกว่า tensor ซึ่งใช้รูปแบบการคูณ element-wise กับ Filter matrix ระหว่าง kernel และ tensor แต่ละตำแหน่งของ tensor รวมเข้าด้วยกันเพื่อให้ได้ผลลัพธ์ออกมาในชุดอาร์เรย์ เรียกว่า ฝังคุณลักษณะ (Feature map) กระบวนการนี้จะเลื่อนทำซ้ำไปเรื่อย ๆ โดยมีหลายรูปแบบ สาเหตุมาจาก kernel แต่ละแบบจะทำงานแยกคุณลักษณะที่แตกต่างกันออกไป โดยมีไฮเปอร์พารามิเตอร์ที่ปรับเปลี่ยนได้คือ ขนาดและจำนวนของ kernel, stride and padding เป็นต้น



รูปที่ 2.7 แสดงการทำงานของ Convolutional layer

(ที่มา: https://www.researchgate.net/figure/An-example-of-an-unpadded-convolution-operation-with-a-kernel-of-size-3x3-and-a-stride_fig2_337940311)

นอกเหนือจากความคล้ายคลึงกันทางชีวภาพที่กล่าวถึงก่อนหน้านี้พวกเขายังช่วยในการรักษามูลค่าของผลลัพธ์จากเซลล์ประสาทที่ถูกจำกัดไว้ที่ขีดจำกัดตามข้อกำหนดของเราสิ่งนี้มีความสำคัญเนื่องจากการป้อนข้อมูลลงในฟังก์ชันการเปิดใช้งานคือ $W \cdot x + B$ โดยที่ W คือ น้ำหนักของเซลล์และ x คือ อินพุตและจากนั้นจะมี (Bias) B ที่เพิ่มเข้ามา ค่านี้หากไม่ได้จำกัดอยู่ที่ขีดจำกัดที่แน่นอนสามารถไปได้ในระดับสูงมากโดยเฉพาะอย่างยิ่งในกรณีที่มีเครือข่ายประสาทลึกมากที่มีพารามิเตอร์นับล้าน สิ่งนี้จะนำไปสู่ปัญหาการคำนวณตัวอย่างเช่น มีฟังก์ชันการเปิดใช้งานบางอย่าง (เช่น SoftMax) ที่ออกค่าเฉพาะสำหรับค่าที่แตกต่างกันของอินพุต (0 หรือ 1)

คุณสมบัติที่สำคัญที่สุดในฟังก์ชันการเปิดใช้งาน คือความสามารถในการเพิ่มความไม่เป็นเชิงเส้นลงในเครือข่ายประสาท

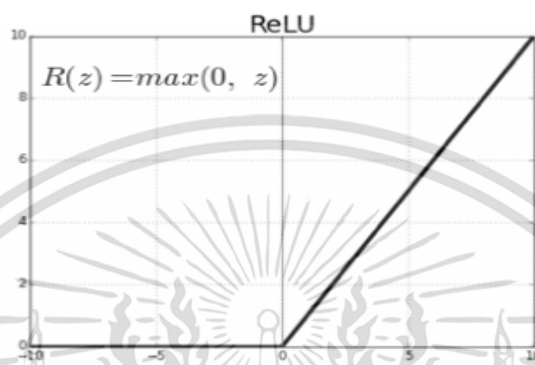
2.10 การขจัดความเป็นเชิงเส้น (ReLU)

คือ ฟังก์ชันเส้นตรงที่ถูกปรับแก้ Rectified ไม่ได้เป็นรูปตัว S เหมือน 2 ฟังก์ชันก่อนหน้านี้ ReLU เป็นฟังก์ชันที่เรียบง่ายกว่าทุก Activation Function ที่ผ่านมาแต่ทรงพลังเนื่องจาก ถ้า Input เป็นบวก Slope จะเท่ากับ 1 ตลอดกาล ทำให้ Gradient ไม่หาย (ไม่เกิด Vanishing Gradient) ส่งผลให้เราเทรนโมเดลได้เร็วขึ้นมาก

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases} \quad (2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติว่าเราแสดงภาพสีของวงกลมอย่างต่อเนื่องเพื่อซีเอ็นเอ็นในระหว่างการฝึกอบรม CNN จะไม่เรียนรู้ว่ามีเส้นตรงอยู่ด้วยเหตุนี้มันจะค่อนข้างสับสนถ้าเราแสดงภาพของสี่เหลี่ยมจัตุรัสในภายหลัง เราสามารถป้องกันกรณีเหล่านี้ได้โดยการเพิ่มเลเยอร์การออกกลางคั่นลงในสถาปัตยกรรมของเครือข่ายเพื่อป้องกันไม่ให้เกิดการ overfitting มากเกินไป



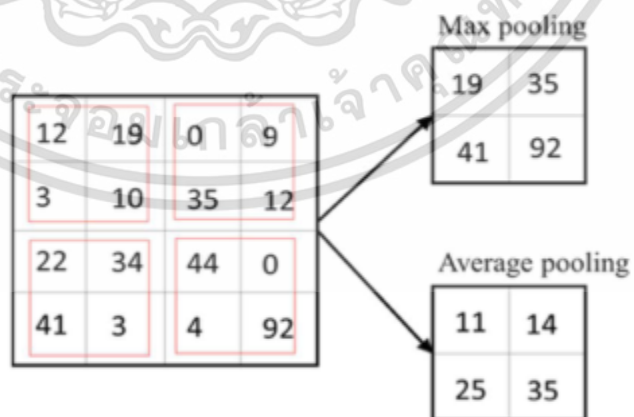
รูปที่ 2.8 แสดง ReLU Activation Function

2.11 SoftMax

เป็นรูปแบบทั่วไปของ sigmoid มากขึ้น มันถูกใช้ในปัญหาการจำแนกประเภทหลายชั้นคล้ายกับ Sigmoid มันจะสร้างค่าในช่วง 0-1 ดังนั้นจึงใช้เป็นเลเยอร์สุดท้ายในรูปแบบการจำแนกประเภท

2.12 การทำพูลลิ่ง (POOL)

เป็นชั้นของ CNN ที่ทำหน้าที่ปรับลดขนาดของรูปภาพโดยคงคุณลักษณะของภาพไว้เพื่อประโยชน์ในการสกัดคุณลักษณะของรูปภาพให้ครอบคลุมคุณลักษณะเด่นที่อยู่ในรูปภาพให้มากขึ้น

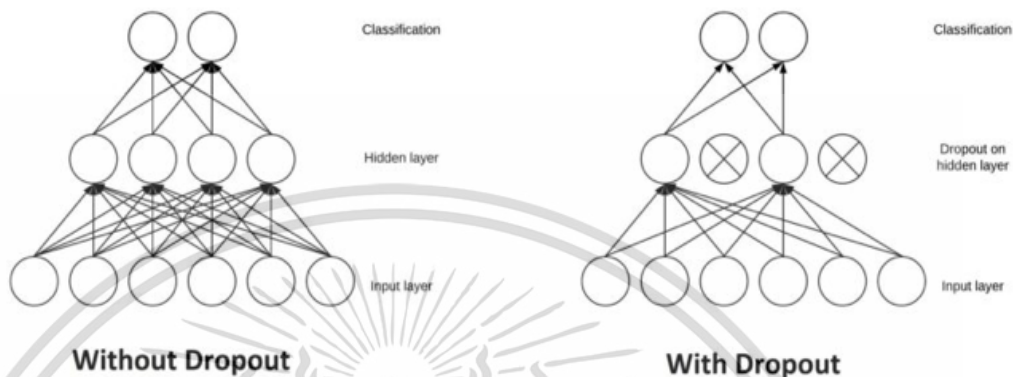


รูปที่ 2.9 แสดงการทำของพูลลิ่ง (Pooling layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 Dropout

เป็นชั้นที่ทำหน้าที่ช่วยลดปัญหาเกิด Overfit โดยการจดจำข้อมูล แบบการสุ่มเอา Node ออก ในระหว่างการเทรนเพื่อให้โมเดลนั้นง่ายขึ้นและเพื่อลดความซับซ้อนของข้อมูลลง

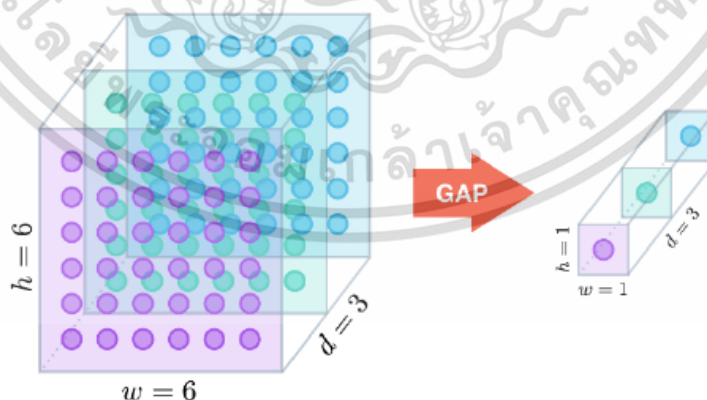


รูปที่ 2.10 แสดงการไม่มี Dropout และมี Dropout

(ที่มา: <https://wenkangwei.github.io/2020/11/13/DL-DropOut/>)

2.14 Global Average Pooling

เป็นการลดจำนวนพารามิเตอร์ทั้งหมดลงในแบบจำลอง มีลักษณะคล้ายคลึงกับ maxing pooling layer เพื่อใช้ลดมิติข้อมูลเชิงพื้นที่ของเมตริกซ์ 3 มิติ อย่างไรก็ตาม maxing pooling layer มีการลดขนาดมากขึ้นโดยที่เมตริกซ์ที่มีขนาด $h \times w \times d$ จะลดขนาดลงเป็นขนาด $1 \times 1 \times d$ เท่านั้น maxing pooling layer จะลดขนาดแผนที่แต่ละตัวลงเหลือเพียงจำนวนเดียวโดยใช้ค่าเฉลี่ยของค่าทั้งหมดของ $h \times w$ ดังภาพที่



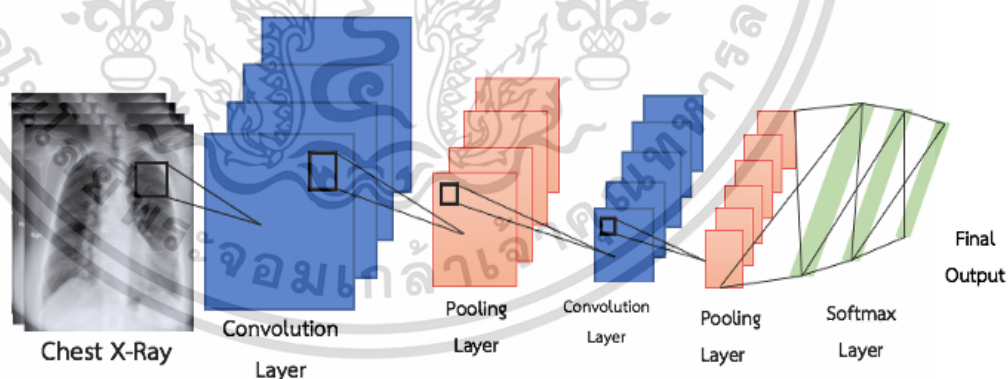
รูปที่ 2.11 Global average pooling layer for object localization

(ที่มา: <https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15 การเชื่อมต่อกันของแต่ละเลเยอร์โดยสมบูรณ์ (Full Connection (FC))

เป็นชั้นที่ทำหน้าที่เชื่อมโยงชั้น CONV และชั้น POOL เพื่อให้เกิดประสิทธิภาพการจำแนกรูปภาพจากเทคนิค (CNN: Deep-learning) สามารถใช้ชั้น CONV และ POOL ที่มีหน้าที่สกัดคุณลักษณะของรูปภาพได้มากกว่า 1 ชั้น Up sampling layer เป็นชั้นการดำเนินการถอดรหัสโดยใช้วิธีการ Deconvolution เพื่อขยายค่าข้อมูลผลลัพธ์จากการดำเนินการ convolution ให้มีขนาดเท่ากับขนาดภาพนำเข้าเริ่มแรกและใช้ชั้น soft max activation function ในชั้นสุดท้ายของโครงข่ายเพื่อทำนายความน่าจะเป็นของแต่ละพิกเซลตามจำนวนคลาสหรือ channel ที่ต้องการ Batch normalization layer เป็นชั้นที่มีการดำเนินการทางคณิตศาสตร์เพื่อทำการปรับข้อมูลให้เป็นมาตรฐานโดยการปรับจะทำการรักษาค่าเฉลี่ย ข้อมูลขาออกให้เข้าใกล้ 0 และค่าเบี่ยงเบนมาตรฐานข้อมูลขาออกให้เข้าใกล้ 1 โดยในระหว่างการสอนชั้น batch normalization จะปรับข้อมูลขาออกให้เป็นมาตรฐาน โดยใช้ค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของชุดข้อมูลนำเข้าปัจจุบัน Merging layers เป็นชั้นที่ดำเนินการผสานผลลัพธ์ระหว่างชั้นดำเนินการสองชั้นหรือมากกว่าเข้าด้วยกัน โดยรับข้อมูลนำเข้าในรูปแบบรายการของ tensor ที่มีรูปร่างทั้งหมดเหมือนกัน แล้วส่งคืนผลลัพธ์เป็น ค่า tensor ตัวเดียว ที่เป็นการผสานกันของข้อมูลที่ถูกนำเข้ามาผสานกันทั้งหมด โดยใช้วิธีการดำเนินการของชั้นต่าง ๆ เช่น Concatenate layer เป็นชั้นการเชื่อมข้อมูลนำเข้าในรูปแบบรายการของ tensor, Add layer เป็นชั้น การรวมข้อมูลนำเข้าในรูปแบบรายการของ tensor, Multiply layer เป็นชั้นการคูณ (element-wise) ข้อมูลนำเข้าในรูปแบบรายการของ tensor เป็นต้น

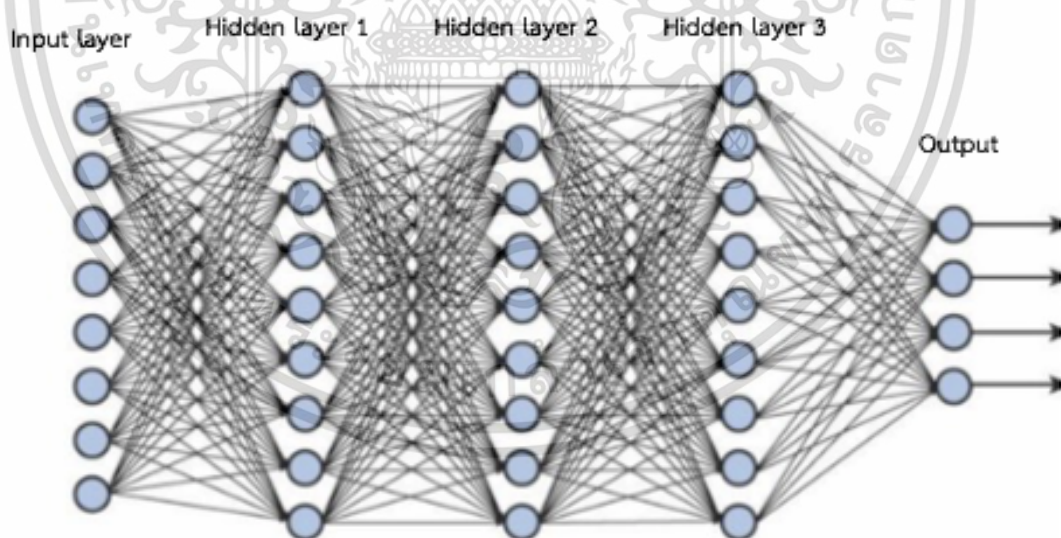


รูปที่ 2.12 การทำคอนโวลูชัน (Convolutional Neural Network: CNN)

2.16 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึกเป็นส่วนหนึ่งของวิธีการการเรียนรู้เครื่องบนพื้นฐานของโครงข่ายประสาทเทียม ซึ่งเป็นแขนงหนึ่งของศาสตร์ด้านปัญญาประดิษฐ์ (AI) ใน ML ถูกสร้างขึ้นโดยใช้หลักการแนวคิดที่เลียนแบบเซลล์ประสาทสมองของมนุษย์ซึ่งทำงานเรียนรู้เป็นรูปแบบอัตโนมัติเชื่อมต่อสลับกันไป-มาของเซลล์เป็นนับ 1000 ล้านเซลล์ ส่งผลให้มนุษย์มีการเรียนรู้ลำดับการคิดพฤติกรรมการแสดงออกที่มีความซับซ้อนด้วยหลักการแนวคิดนี้เองจึงได้ถูกพัฒนาออกแบบและประยุกต์ใช้เข้ากับการประมวลผลของคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์นั้นสามารถประมวลผลจากชุดข้อมูลที่ป้อนเข้าไปในฐานข้อมูลได้เป็นในรูปแบบของตัวเลขโดยใช้หลักการทางตรรกะคณิตศาสตร์ เช่น ข้อมูลภาพ เสียง วิดีโอ ข้อความ เป็นต้น โดยข้อมูลเหล่านี้จะถูกค้นหาจากคุณลักษณะเฉพาะของข้อมูลโดยอัตโนมัติ เมื่อเริ่มการทำงานของตัวประมวลผลโดยไม่จำเป็นต้องอาศัยการกำหนดเงื่อนไขจากผู้ใช้นั้นการทำงานของตัวประมวลผลจะนำไปสู่การทำนายผลและสร้างผลลัพธ์จากการทำนายจะถูกนำไปหาค่าความคลาดเคลื่อนกับค่าความน่าจะเป็น ซึ่งค่าความคลาดเคลื่อนที่ได้ถูกนำมาใช้ปรับปรุงตัวประมวลผล การประมวลผลที่มีชั้นของการค้นหาคุณลักษณะหลายชั้นนี้ถูกจัดให้เป็นรูปแบบของการเรียนรู้เชิงลึกนั่นเองโดย Deep Learning จะมีชั้นที่เรียกว่า hidden layer หลาย ๆ ชั้นที่เป็นตัวแยกคุณลักษณะ

ดังรูป



รูปที่ 2.13 การเรียนรู้เชิงลึก (Deep Learning architecture)

(ที่มา: https://www.researchgate.net/figure/The-structure-of-a-Deep-Neural-Network-with-three-hidden-layers_fig2_352996743)

ซึ่งในโครงข่ายประสาทเทียมอาศัยหลักการเดียวกันกับเซลล์ประสาทของมนุษย์ที่มีการเชื่อมโยงต่อกันเป็นธรรมชาติและอัตโนมัติจากประสบการณ์และการเรียนรู้ที่ซับซ้อนกระทั่งสามารถจำแนกความแตกต่างได้โดยที่คอมพิวเตอร์จะประมวลผลโดยใช้ผลทางคณิตศาสตร์ประมวลผลแบบขนานและประมวลผลได้หลายครั้งละจำนวนมากทำให้เครื่องสามารถแสดงผลลัพธ์ในการตัดสินใจและคาดการณ์ได้ดียิ่งขึ้นซึ่งเปรียบเสมือนการสร้างสัญญาณของการมองเห็น

2.17 อัลกอริทึมปรับปรุงประสิทธิภาพ

อัลกอริทึมปรับปรุงประสิทธิภาพ(Optimization algorithms) เป็นส่วนสำคัญสำหรับการเรียนรู้ของโครงข่ายประสาทเทียมทำหน้าที่สำหรับการปรับปรุงค่าให้โมเดลมีการเรียนรู้ที่เหมาะสม ลดความคลาดเคลื่อนหรือการหาตำแหน่งที่ทำให้มีข้อผิดพลาดน้อยที่สุด รวมถึงขั้นตอนการปรับค่าพารามิเตอร์ของโครงข่าย ตัวอย่างของอัลกอริทึมปรับปรุงประสิทธิภาพพื้นฐาน คือ gradient descents นอกจากนี้ยังมีอีกหลายอัลกอริทึมที่ทำงานคล้ายกัน เช่น Adagrad optimizer, Momentum optimizer, Adam optimizer, Ftrl optimizer RMSProp optimizer เป็นต้น

2.18 ทฤษฎีการ Transfer Learning

เนื่องจาก Deep learning จะมีปัญหาเรื่องใช้ระยะเวลาในการฝึกสอนโมเดลตั้งแต่ต้นจนจบค่อนข้างนานและมีความซับซ้อนมีตัวแปร (weight) จำนวนมากซึ่งในการเริ่มต้นฝึกโมเดล DL ซับซ้อนขนาดนี้ตั้งแต่ต้น (Weight initialization ด้วยค่า random) นั้นต้องใช้ทั้งข้อมูล dataset ขนาดใหญ่และใช้พลังการประมวลผลมหาศาลอีกทั้งต้องใช้เวลานานหลายวัน จนอาจถึงนานหลายสัปดาห์กว่าจะได้โมเดลที่มีประสิทธิภาพ ดังนั้นการใช้ Transfer learning จึงเทคนิคที่ช่วยลดเวลาการฝึกโมเดล Deep learning ด้วยการนำบางส่วนของโมเดลที่ฝึกสอนเรียบร้อยแล้วกับงานที่ใกล้เคียงกันมาใช้เป็นส่วนหนึ่งของโมเดลใหม่

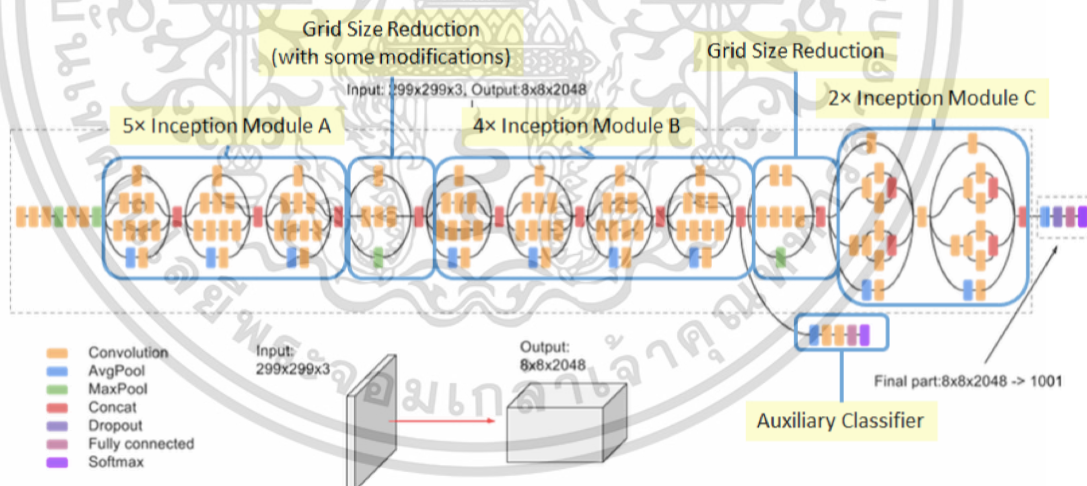
ImageNet dataset กว่า 15,000,000 ภาพความละเอียดสูงซึ่งมีจำนวน 22,000 classes ส่วน ILSVRC (The ImageNet large scale visual recognition challenge) การแข่งขันที่จัดโดย บริษัท google โดยใช้ข้อมูล subset ของ ImageNet โดยจะมีประเภทของภาพ 1,000 classes ซึ่งในแต่ละ classes จะมีรูปภาพอยู่ประมาณ 1,000 ภาพ ทำให้มี dataset ของ ILSVRC รวมประมาณ 1,000,000

ภาพเพื่อใช้ในการฝึกโมเดล โดยโมเดลที่นำมาใช้ในงานวิจัยนี้คือ InceptionV3, ResNet50, VGG16, VGG19, Mobilenet V2 และ InceptionResNet V2

2.19 สถาปัตยกรรมแบบจำลองที่นำมาใช้เพื่อทำนายในงานวิจัย

2.19.1 โมเดล Inception V3

Inception V3 เป็นรูปแบบการเรียนรู้อย่างลึกซึ่งบนพื้นฐานของเครือข่ายประสาท (CNN) ซึ่งใช้สำหรับการจำแนกภาพ Inception V3 เป็นรุ่นที่เหนือกว่าของ Model Basic Inception V1 ซึ่งได้รับการแนะนำให้รู้จักกับ Google Net ในปี 2014 ตามชื่อแนะนำว่ามันได้รับการพัฒนาโดยทีมงานที่ Google โมเดล Inception V3 ประกอบด้วย 42 ชั้นซึ่งสูงกว่ารุ่น V1 และ V2 Inception ก่อนหน้านี้เล็กน้อย แต่ประสิทธิภาพของรุ่นนี้ดีขึ้นมาก โดยการลดโครงสร้างภายในออกเป็น 5 Step คือ Inception Module A จำนวน 5 Module (1), Grid Size of Reduction Step1 จำนวน 1 Module(2), Inception Module B จำนวน 4 Module(3), Grid Size of Reduction Step2 จำนวน 1 Module(4), Inception Module C จำนวน 2 Module(5) และ Head (8x8x2048) สามารถแยก output ได้ 1,000 classes ดังรูปที่ 2.13

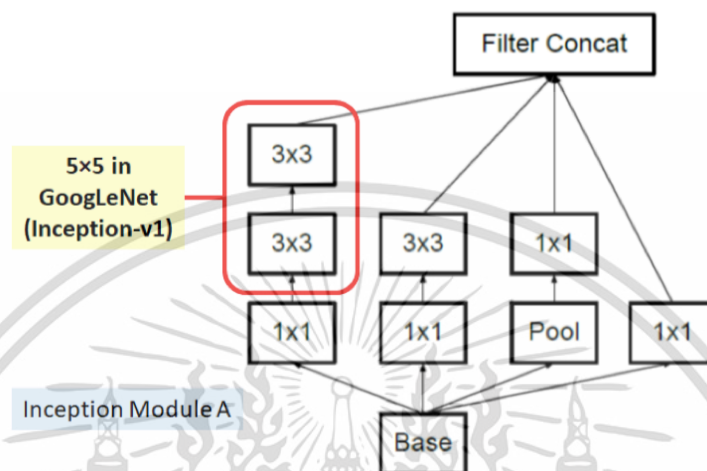


รูปที่ 2.14 แสดงโครงสร้างโมเดล Inception V3 Architecture

พัฒนาให้ Inception-v3 มี parameter ลดลงจากเดิม แต่ยังคงมีประสิทธิภาพสูงโดยการปรับปรุง convolutions เดิมที่ 5x5 ลงเหลือ 3x3 pixels และ maxpooling เดิมที่ 3x3 ลงเหลือ 2x2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pixels ซึ่งวัตถุประสงค์ของการออกแบบ 1×1 convolution ต้องการให้ shape outputs ขาออกเป็น tensor เช่นกำหนด (N, F, H, W) ซึ่ง N คือ batch size, F คือ จำนวนของ convolution filters ส่วน H, W คือ spatial ของมิติ ดังรูปที่ 2.14



รูปที่ 2.15 แสดงโครงสร้างโมเดล Inception V3 ประกอบด้วยชั้น Conv และ Maxpooling

โดยภายในของ inception module กรณี input ข้อมูลเข้ามามี 3 สีคือ RGB ในที่นี้เราสามารถเปรียบเทียบได้กับการทำ feature map ดังรูปด้านซ้าย ซึ่งสมมุติว่าเราต้องการผลลัพธ์ feature map เพียง 1 ค่านั้นถูกทำได้โดยใช้ conv ขนาด 1×1 pixels ที่มีการขยับที่ 1 stride ดังรูปที่ 2.15 โดยค่า weight ที่คูณในแต่ละชั้นของ RGB นั้นจะมีค่าที่ไม่เท่ากันส่งผลให้ค่าที่ได้ในแต่ละพิกเซล หลังการทำ feature map นี้จะมีที่เป็นลักษณะเฉพาะ



รูปที่ 2.16 การทำ Feature map ของตัวอย่าง RGB โดยใช้ 1×1 ชั้น Conv

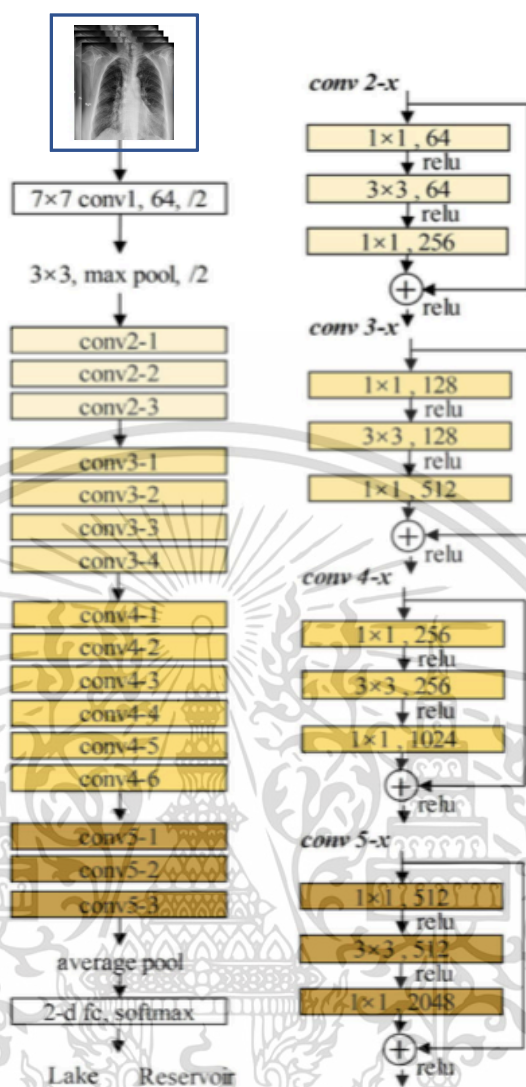
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19.2 โมเดล ResNet50

ResNet ย่อมาจากเครือข่ายที่เหลือและเป็นประเภทเฉพาะของเครือข่ายประสาท (CNN) ที่แนะนำในบทความปี 2015 “การเรียนรู้ที่เหลืออยู่ลึกสำหรับการจดจำภาพ” โดยเขา Kaiming, Zhang Xiangyu, Ren Shaoqing และ Sun Jian CNNs มักใช้ในการใช้งานแอปพลิเคชันการมองเห็นคอมพิวเตอร์

ResNet50 เป็นโครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN) ที่มีความลึก 50 ชั้น Residual Neural Network (ResNet) เป็นโครงข่ายประสาทเทียมที่วางบล็อกที่เหลือทับกันเพื่อสร้างเครือข่าย โดยการทำให้ residual mapping เพื่อแก้ปัญหาชั้นที่มากขึ้นมา ๆ จนทำให้เกิด vanishing/exploding gradients

เนื่องจากเฟรมเวิร์กที่ ResNet50 ได้แก้ไขในจุดนี้นั้นจึงทำให้สามารถฝึกโครงข่าย CNN แบบที่มีชั้นลึกมาก ๆ ได้โดยสามารถมีเลเยอร์ได้หลายร้อยหรือหลายพันเลเยอร์และตัวโมเดลก็ยังคงมีประสิทธิภาพที่ดี โครงสร้าง ResNet50 ประกอบไปด้วยชั้น conv1 กับ kernel ขนาด 7×7 64 kernel ที่แตกต่างกันนั้นจะเลื่อนโดยใช้ค่า stride เท่ากับ 2 ในชั้นนี้จะมีทั้งหมด 1 layer และจะมีการรวมค่ามากที่สุด (max pooling) โดยเลื่อนโดยใช้ค่า stride เท่ากับ 2 ในชั้น conv2.x จะมี kernel ขนาด 1×1 64 kernel ตามมาด้วย kernel ขนาด 3×3 64 kernel และ kernel สุดท้ายขนาด 1×1 256 kernel ซึ่งทั้ง 3 เลเยอร์นี้จะมีการทำซ้ำทั้งหมด 3 ครั้ง จึงได้ทั้งหมด 9 เลเยอร์ ในชั้น conv3.x จะมี kernel ขนาด 1×1 128 kernel ตามมาด้วย kernel ขนาด 3×3 128 kernel และ kernel สุดท้ายขนาด 1×1 512 kernel ซึ่งทั้ง 3 เลเยอร์นี้จะมีการทำซ้ำทั้งหมด 4 ครั้ง จึงได้ทั้งหมด 12 เลเยอร์ ในชั้น conv4.x จะมี kernel ขนาด 1×1 256 kernel ตามมาด้วย kernel ขนาด 3×3 256 kernel และ kernel สุดท้ายขนาด 1×1 1024 kernel ซึ่งทั้ง 3 เลเยอร์นี้จะมีการทำซ้ำทั้งหมด 6 ครั้ง จึงได้ทั้งหมด 18 เลเยอร์ ในชั้น conv5.x จะมี kernel ขนาด 1×1 512 kernel ตามมาด้วย kernel ขนาด 3×3 512 kernel และ kernel สุดท้ายขนาด 1×1 2048 kernel ซึ่งทั้ง 3 เลเยอร์นี้ จะมีการทำซ้ำทั้งหมด 3 ครั้ง จึงได้ทั้งหมด 9 เลเยอร์ หลังจากนั้นที่เลเยอร์ 1 ชั้นสุดท้ายจะใช้การ average pooling เพื่อลดขนาดมิติลงและเข้าสู่ชั้น fully connected layer ที่ประกอบไปด้วย 1000 nodes ผ่าน activation function คือ softmax function เพื่อให้ได้ผลลัพธ์ว่าเป็นคลาสใดโดยสรุปแล้วเมื่อรวมชั้นทั้งหมดจะได้ 50 ชั้นของ Deep learning



รูปที่ 2.17 แสดงโครงสร้างโมเดล ResNet50 Architecture

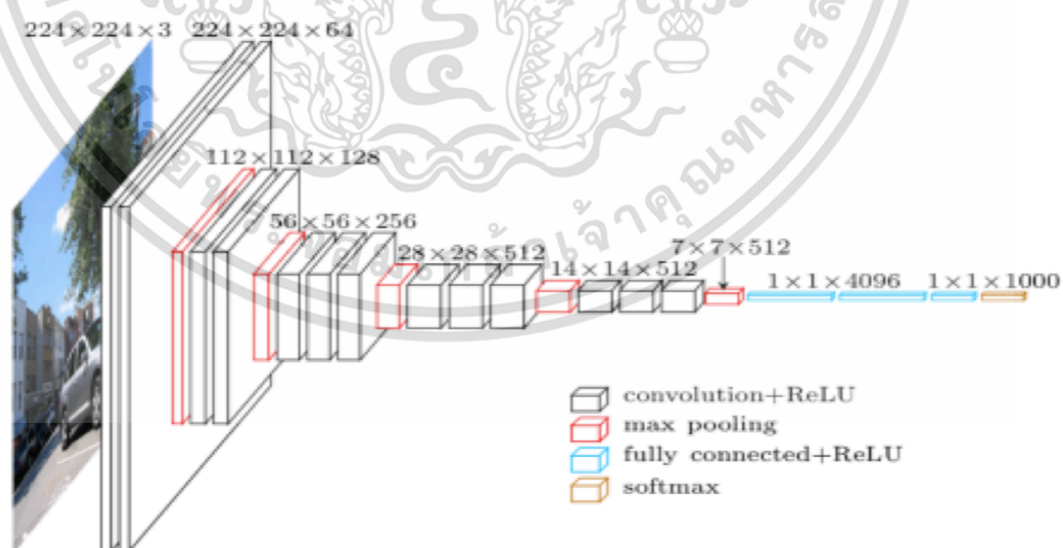
2.19.3 โมเดล VGG16

เครือข่ายประสาทเทียมเป็นที่รู้จักกันในชื่อ Conv Net ซึ่งเป็นเครือข่ายประสาทเทียมชนิดหนึ่ง เครือข่ายประสาทแบบ CNN มีเลเยอร์อินพุตเลเยอร์เอาต์พุตและเลเยอร์ที่ซ่อนอยู่ต่าง ๆ VGG16 เป็นประเภทของ CNN (เครือข่ายประสาท Conv) ที่ถือว่าเป็นหนึ่งในรุ่นคอมพิวเตอร์ที่ดีที่สุดในปัจจุบัน ผู้สร้างโมเดลนี้ประเมินเครือข่ายและเพิ่มความลึกโดยใช้สถาปัตยกรรมที่มีตัวกรองการควบคุมที่เล็กมาก (3 × 3) ซึ่งแสดงให้เห็นถึงการปรับปรุงที่สำคัญในการกำหนดค่าก่อนหน้า พวกเขาผลึกความลึกให้กับชั้นน้ำหนก 16–19 ชั้นทำให้ประมาณพารามิเตอร์ที่สามารถฝึกอบรมได้ 138 พารามิเตอร์ VGG16 เป็นสถาปัตยกรรมโครงข่ายประสาทเทียมแบบสังวัตนาการ(CNN) ที่เรียบง่ายและใช้กันอย่างแพร่หลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งใช้สำหรับ ImageNet ซึ่งเป็นโครงการฐานข้อมูลภาพขนาดใหญ่ที่ใช้ในการวิจัยซอฟต์แวร์การรู้จำภาพ สถาปัตยกรรม VGG16 ได้รับการพัฒนาโดย Karen Simonyan และ Andrew Zisserman จาก University of Oxford ในปี 2014 ผ่านบทความเรื่อง "Very Deep Convolutional Networks for Large-Scale Image Recognition" 'VGG' เป็นตัวย่อของ Visual Geometry Group ซึ่งเป็นกลุ่มนักวิจัยจาก University of Oxford ที่พัฒนาสถาปัตยกรรมนี้ และ '16' บ่งบอกว่าสถาปัตยกรรมนี้มีความลึก 16 ชั้น โดยโมเดล VGG16 ได้รับความแม่นยำในการทดสอบ 5 อันดับสูงสุด 92.7% ใน ImageNet ซึ่งเป็นชุดข้อมูลมากกว่า 14 ล้านภาพที่อยู่ใน 1,000 คลาส เป็นหนึ่งในโมเดลที่มีชื่อเสียงที่ส่งไปยัง ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

โครงสร้าง VGG16 ประกอบไปด้วย 13 conv layer และ 3 Fully connected layer ซึ่ง Conv layer จะทำหน้าที่ดึงคุณลักษณะภาพออกมา (Extract feature) โดยชั้น conv 1-2 จะมี 64 ตัวกรอง ชั้น conv 3-4 จะมี 128 ตัวกรอง ชั้น conv 5-7 จะมี 256 ตัวกรอง และชั้น conv 8-13 มี 512 ตัวกรองขนาดตัวกรองที่ใช้ในแต่ละชั้นคือ 3x3 ชั้น conv 2, conv 4, conv 7, conv 10 และ conv13 จะมีการรวมค่ามากที่สุด (Max Pooling Layer) มีขนาด 2x2 และ จะมีการทำ Batch Normalize เพื่อที่จะช่วยเพิ่มความเร็วในการเรียนรู้ของโมเดลให้มากยิ่งขึ้นมีการใช้ Activation function ที่ใช้ในแต่ละชั้นคือ Relu function ยกเว้นในชั้น FC 3 ที่จะใช้ Activation function เป็น Softmax function เพื่อที่จะให้ในชั้นสุดท้ายของ fully connected layer นี้เป็นชั้น ที่ไว้ทำนายภาพเอกซเรย์ปอดว่าเป็นปกติหรือเป็นวัณโรคปอดหรือไม่

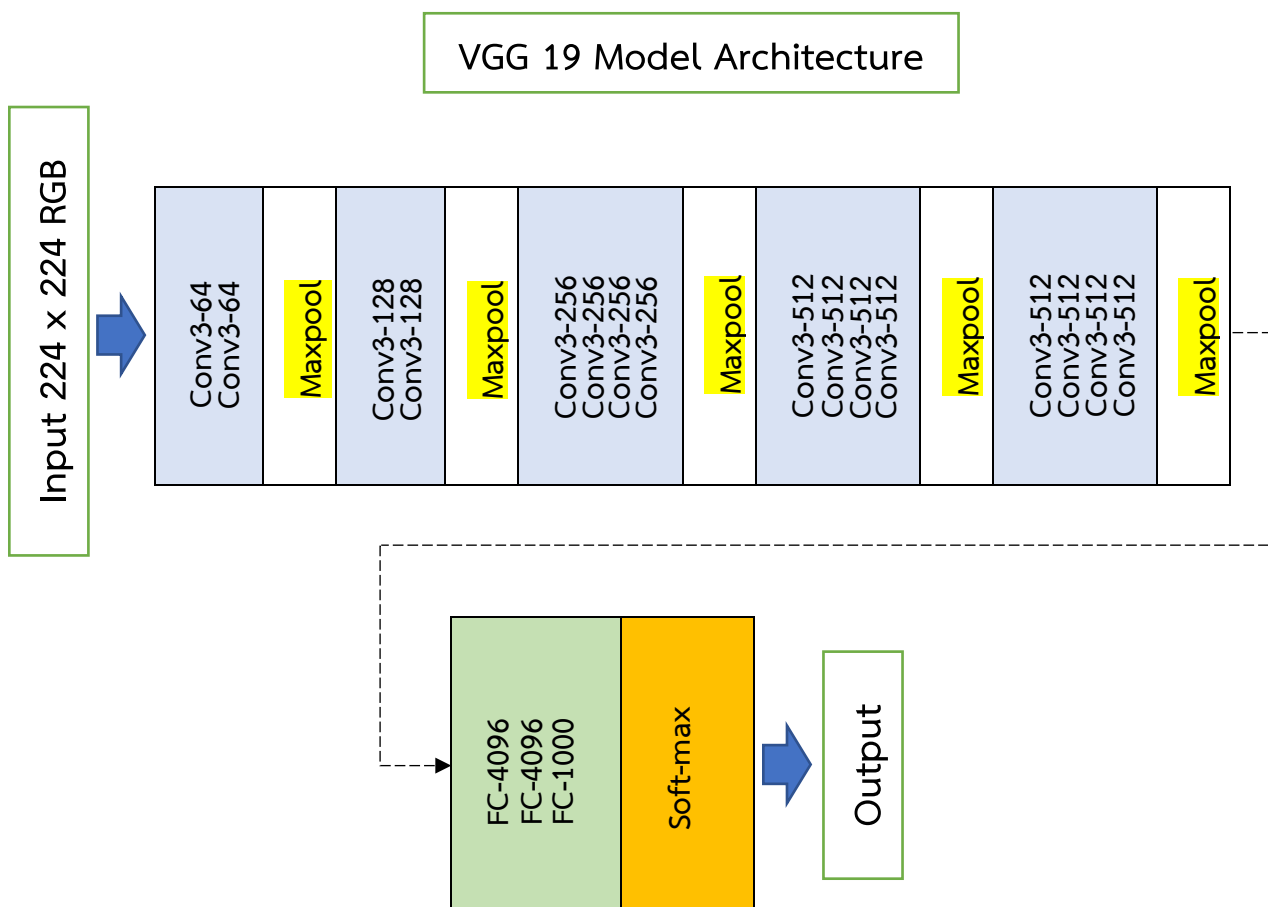


รูปที่ 2.18 แสดงโมเดล VGG16 Architecture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19.4 โมเดล VGG19

VGG19 เป็นรุ่นที่แตกต่างจากรุ่น VGG ซึ่งโดยย่อประกอบด้วย 19 เลเยอร์ (มีชั้น conv 16 ชั้น, 3 ชั้น Fully, 5 ชั้น Maxpool และ 1 ชั้น SoftMax) กำหนดภาพ RGB ขนาดคงที่ (224×224) เป็นอินพุตสำหรับเครือข่ายนี้ ซึ่งหมายความว่าเมทริกซ์มีรูปร่าง (224,224,3) ใช้ kernels ขนาด (3 x 3) ที่มี stride 1 พิกเซล ทำให้สามารถครอบคลุมทั้งหมดของภาพได้ ใช้ spatial padding เพื่อรักษาความละเอียดเชิงพื้นที่ของภาพ ใช้ max pooling ขนาด 2 * 2 พิกเซล stride 2 ตามมาด้วยใช้ฟังก์ชัน (ReLU) เพื่อให้โมเดลจัดประเภทได้ดีขึ้นและปรับปรุงเวลาในการคำนวณเนื่องจากโมเดลก่อนหน้านี้ใช้ฟังก์ชัน tanh หรือ sigmoid ซึ่งพิสูจน์ได้ว่าดีกว่าเหล่านั้นมากที่สุดท้ายคือ ใช้เลเยอร์ที่เชื่อมต่อกัน fully connected ทั้งหมด 3 ชั้น โดยสองชั้นแรกมีขนาด 4096 ช่อง ชั้นที่สามที่มี 1,000 ช่อง สำหรับการจำแนก ILSVRC และชั้นสุดท้ายคือ ฟังก์ชัน Softmax



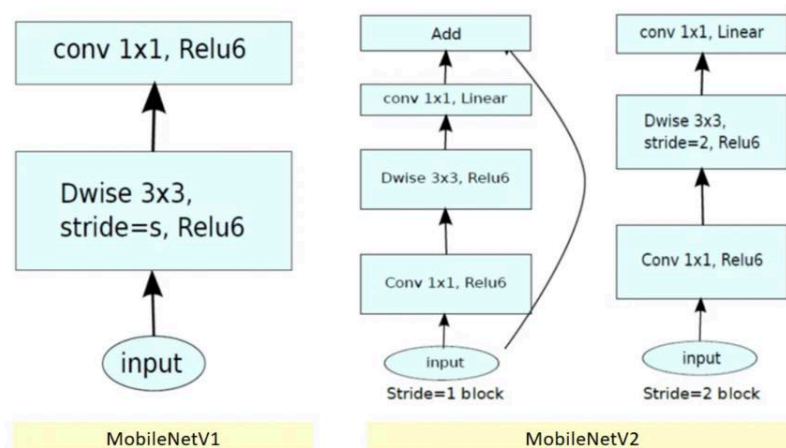
รูปที่ 2.19 แสดงโมเดล VGG19 Architecture

ตารางที่ 2.3 แสดงสถาปัตยกรรมโมเดล VGG16, VGG19

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

2.19.5 โมเดล MobileNet V2

MobileNet คือโมเดลที่มีขนาดเล็ก การทำงานด้วยศักยภาพ (latency) ต่ำ ใช้พลังงานในการประมวลผลพารามิเตอร์น้อย เป็นไปตามข้อจำกัดของทรัพยากรของ Mobilenet กรณีการใช้งานที่หลากหลาย เช่น การจำแนกของข้อมูล (Classification) การแยกข้อมูล (Detection) การเข้าถึงข้อมูล (Embedding) การแบ่งส่วนของข้อมูล (Segmentation)



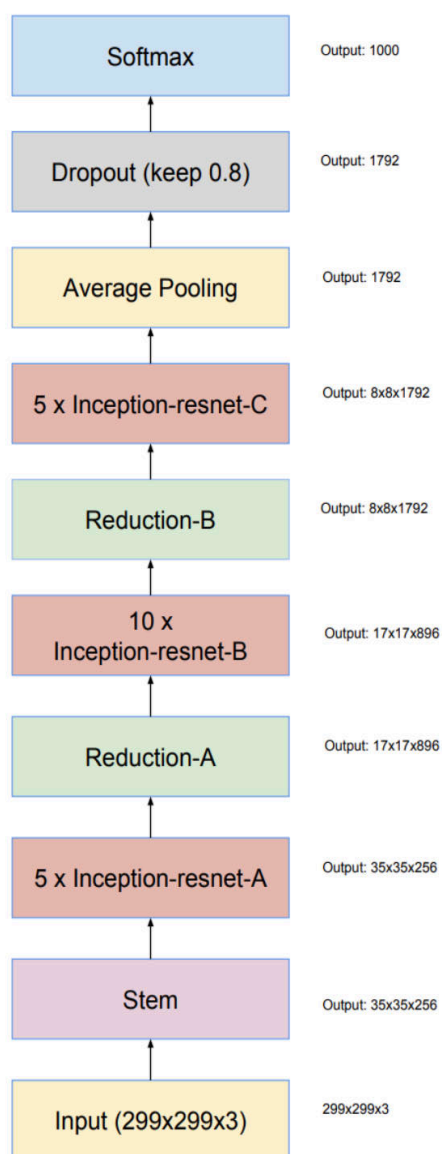
รูปที่ 2.20 แสดงโมเดล MobileNet v2 Architecture

- MobileNetV2 มีบล็อกสองประเภทบล็อกหนึ่ง คือบล็อกที่เหลือซึ่งมี stride 1 อีกบล็อกหนึ่งคือบล็อกที่มี stride 2 สำหรับการลดขนาด
- มี 3 ชั้นสำหรับบล็อกทั้งสองประเภท
- ชั้นแรกคือ 1x1 convolution และ ReLU6
- ชั้นที่สอง คือ depth wise convolution
- ชั้นที่สาม คือ Conv 1 x 1 แต่ไม่มีความไม่เชิงเส้น มีการอ้างว่าหากใช้ ReLU อีกครั้ง เครือข่ายระดับลึกจะมีพลังของตัวแยกประเภทเชิงเส้นในส่วนที่ไม่เป็นศูนย์ของโดเมนเอาต์พุตเท่านั้น

2.19.6 โมเดล InceptionResNet V2

เป็นเครือข่ายประสาทเทียมแบบ conv ที่ได้รับการฝึกฝนมากกว่าหนึ่งล้านภาพจากฐานข้อมูล ImageNet เครือข่ายมีความลึก 164 เลเยอร์และสามารถจำแนกภาพเป็น 1,000 หมวดหมู่วัตถุเช่น แป้นพิมพ์ เมาส์ดินสอและสัตว์ต่าง ๆ มากมาย ด้วยเหตุนี้เครือข่ายจึงได้เรียนรู้การแสดงคุณลักษณะที่หลากหลายสำหรับภาพที่หลากหลายเช่นกัน เครือข่ายมีขนาดอินพุตภาพ 299-299 และเอาต์พุต คือค่าความน่าจะเป็นของแต่ละคลาสโดยประมาณโมเดลถูกกำหนดขึ้นจากการรวมกันของโครงสร้าง Inception และ Residual connection ในบล็อก Inception-Resnet ตัวกรอง Conv หลายขนาดจะรวมกับการเชื่อมต่อที่เหลือการใช้การเชื่อมต่อที่เหลือไม่เพียงแต่หลีกเลี่ยงปัญหาการ

เสื่อมสภาพที่เกิดจากโครงสร้างที่อยู่ลึกเท่านั้นแต่ยัง ช่วยลดเวลาการฝึกอบรมอีกด้วยรูปแสดงสถาปัตยกรรมเครือข่ายพื้นฐานของ InceptionResnetv2 [28]

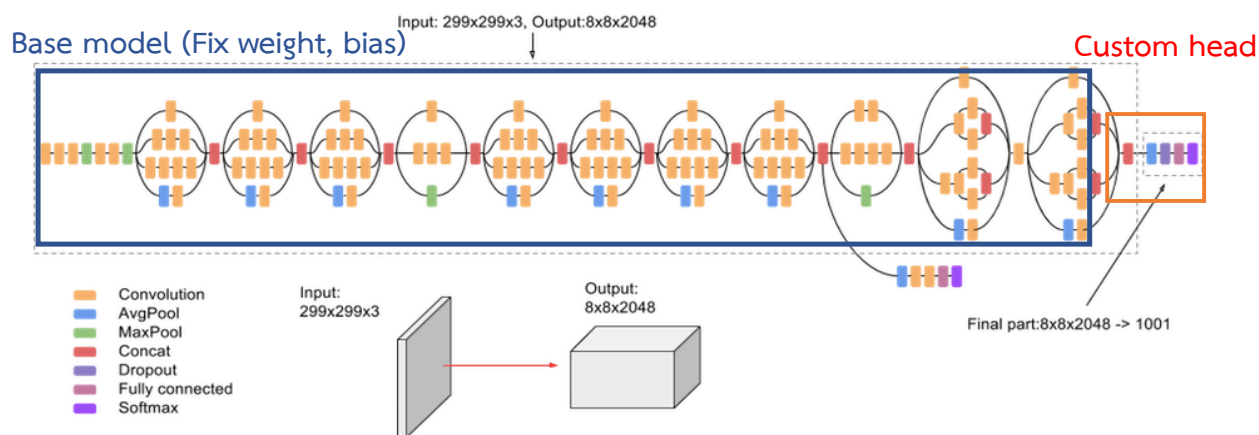


รูปที่ 2.21 แสดงโมเดล InceptionResNet V2 Architecture

2.20 ทฤษฎีการปรับแต่ง

ในที่นี้จะขอกล่าวถึงการใช้อย่างการ Transfer learning โดยใช้ model Inception-v3 ซึ่งสามารถโหลดได้จาก <https://keras.io/api/applications/inceptionv3/> ซึ่งนำมาเป็น base

model inception v3 เข้ามาใช้งานดังนี้ กรณีเปลี่ยน head model ให้เปลี่ยน include top จาก true จากนั้นจึงทำการเพิ่ม layers ที่เราต้องการและทำการฝึกแบบ fix weight และ bias ในส่วนของ base model และทำการตรวจสอบค่าความแม่นยำของโมเดลที่เพิ่ม head layers เข้าไปโดยเทรน ทั้งหมดของ network เป็นดังรูปที่ 2.18



รูปที่ 2.22 แสดงโครงสร้าง networks ของโมเดล Inception V3 ที่มีการ custom head

ตารางที่ 2.4 แสดงรายละเอียดแต่ละของโมเดล

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
IncepResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88

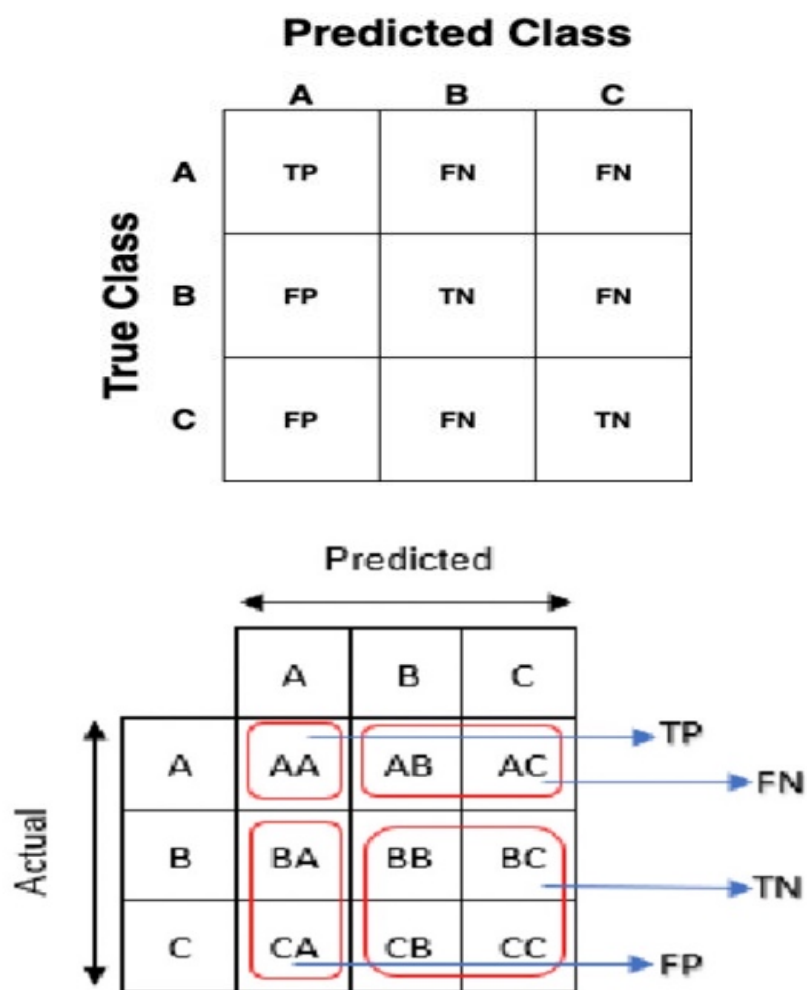
2.21 นิยามการทดสอบประสิทธิภาพและการประเมินผลในงานวิจัย

2.21.1 การทดสอบการทำงานของโมเดล

ในการทดลองเราใช้ตัวชี้วัดที่มีประสิทธิภาพการฝึกสอนโมเดล จากการวัดค่าความถูกต้อง (Accuracy) และความแม่นยำ (Precision) ซึ่งเป็นค่าความน่าจะเป็นโดยคิดเป็นค่าร้อยละ (%) ที่ได้

จากการทดสอบ เพื่อหาค่าพยากรณ์ความถูกต้องของข้อมูลโดย Confusion matrix 3x3 มีสูตรการคำนวณดังตาราง

ตารางที่ 2.5 นิยาม Confusion matrix 3x3 ใช้พิจารณาในงานวิจัยและแสดงตัวอย่างการหาค่า Confusion Matrix แบบละเอียด



คำอธิบาย

TP → True Positive	TN → True Negative
FP → False Positive	FN → False Negative

True Positive (A) = AA	True Negative (A) = BB+BC+CB+CC
True Positive (B) = BB	True Negative (B) = AA+AC+CA+CC

Ture Positive (C) = CC	Ture Negative (C) = AA+AB+BA+BB
False Positive (A) = BA+CA	False Negative (A) = AB+AC
False Positive (B) = AB+CB	False Negative (B) = BA+BC
False Positive (C) = AC+BC	False Negative (C) = CA+CB
<p>Accuracy (Acc) = $AA+BB+CC/N$</p> <p>Precision(A) = $TP/(TP+FP) = AA/(AA+(BA+CA))$</p> <p>Precision(B) = $TP/(TP+FP) = BB/(BB+(AB+CB))$</p> <p>Precision(C) = $TP/(TP+FP) = CC/(CC+(AC+BC))$</p> <p>Recal/Sensitivity = $TPR(A) = TP/(TP+FN) = AA/(AA+(AB+AC))$</p> <p>Recal/Sensitivity = $TPR(B) = TP/(TP+FN) = BB/(BB+(BA+BC))$</p> <p>Recal/Sensitivity = $TPR(C) = TP/(TP+FN) = CC/(CC+(CA+CB))$</p> <p>Specificity = $TNR(A) TN/(TN+FP) = (BB+BC+CB+CC)/((BB+BC+CB+CC) + (BA+CA))$</p> <p>Specificity = $TNR(B) TN/(TN+FP) = (AA+AC+CA+CC)/((AA+AC+CA+CC) + (AB+CB))$</p> <p>Specificity = $TNR(C) TN/(TN+FP) = (AA+AB+BA+BB)/((AA+AB+BA+BB) + (AC+BC))$</p>	

2.21.2 ตัวชี้วัดการประเมินผล

การประเมินผลเพื่อประเมินประสิทธิภาพของโมเดลที่แตกต่างกันกับปัญหาที่มีของงานวิจัยโดย Confusion Matrix ถือว่าเป็นเครื่องมือสำคัญในการชี้วัดการประเมินผลลัพธ์ของการทำนายหรือ (Prediction) โมเดลที่สร้างขึ้นใน ML โดยใช้การวัดค่าสิ่งที่โมเดลทำนายกับสิ่งที่เกิดขึ้นจริงมีสัดส่วนเป็นอย่างไร แต่ในการวิจัยนี้ ค่าความถูกต้อง(Accuracy), ค่าความแม่นยำ(Precision), ค่าการเรียกคืน(Recal/Sensitivity), ค่าตัวชี้วัดF1 (F-1 score) และความจำเพาะ(Specificity) ถูกนำมาใช้ในการประเมินผลการปฏิบัติของแบบจำลองแบบ Classification

เพื่อให้การวัดประสิทธิภาพที่ดีและเชื่อถือได้ข้อมูลที่จะใช้ในการฝึกอบรมโมเดลจะต้องมีความสมดุลความแม่นยำถูกคำนวณโดยการรวมคลาส Positive (TP) และ TRUE เชิงลบ (TN) ที่แท้จริงหารด้วยการรวมของคลาสบวกจริงค่าลบจริงค่าบวกเท็จ (FP) และคลาสเท็จลบ (FN) ดังแสดงในสูตรด้านล่างคือ

2.21.2.1 ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้องเป็นตัวชี้วัดการประเมินที่ใช้สำหรับการวัดประสิทธิภาพของการจำแนกประเภทหรืออัลกอริทึม ความถูกต้องอาจเป็นตัวชี้วัดประสิทธิภาพที่เป็นปัญหาหรือทำให้เข้าใจผิดเมื่อใช้ในการประเมินแบบจำลองที่ได้รับการฝึกฝนเกี่ยวกับข้อมูลที่ไม่สมดุล สำหรับตัวชี้วัดการประเมินนี้ เพื่อให้การวัดประสิทธิภาพที่ดีและเชื่อถือได้ข้อมูลที่จะใช้ในการฝึกอบรมโมเดลจะต้องมีความสมดุล ความแม่นยำถูกคำนวณโดยการรวมคลาสบวกจริง (TP) และคลาส True Negative (TN) หารด้วยการรวมของคลาสบวกจริงค่าลบจริงค่าบวกเท็จ (FP) และชั้นเรียนเท็จลบ (FN) ดังแสดงใน สูตรด้านล่าง

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive} + \text{False Negative} + \text{True Negative}}$$

2.21.2.2 ค่าความแม่นยำ (Precision Metric)

ค่าความแม่นยำนี้คำนวณอัตราส่วนระหว่างกรณีที่ต้องการที่คาดการณ์ว่าเป็นบวกมากกว่าที่คาดการณ์ไว้ทั้งหมดเป็นกรณีบวก ความแม่นยำช่วยให้คาดเดาความสามารถในการค้นหาทุกกรณีที่เกี่ยวข้องกับชุดข้อมูล โดยตัวชี้วัดความแม่นยำจะเกี่ยวข้องโดยตรงกับค่าบวกเท็จขนาดใหญ่แสดงว่าคิดเป็นเชิงบวกที่แท้จริงสูงกว่า

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

2.21.2.3 ค่าการเรียกคืน (Recall Metric)

การประเมินนี้จะคำนวณอัตราส่วนระหว่างกรณีที่ต้องการที่คาดการณ์ว่าเป็นบวกมากกว่ากรณีบวกจริงและการเรียกคืนแสดงความสามารถในการค้นหาทุกกรณีที่เกี่ยวข้องในชุดข้อมูลเท่านั้น เป็นตัวชี้วัดนี้เกี่ยวข้องกับค่าเชิงลบหลายประการ

$$\text{Sensitivity/Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

2.21.2.4 ค่าตัวชี้วัด F1 (F1-Score Metric)

การประเมินคะแนน F1 ใช้ในการวัดประสิทธิภาพของตัวจำแนกโดยการรวมตัวชี้วัดการเรียกคืนและการประเมินความแม่นยำเป็นหนึ่งตัวชี้วัดการประเมินผลการวัดประสิทธิภาพเดียว การประเมิน F1-Score คำนวณโดยการคูณผลิตภัณฑ์ของความแม่นยำและการเรียกคืนโดย 2 และหารด้วยผลรวมของการเรียกคืนและการประเมินความแม่นยำ

$$F1\text{-Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

2.21.2.5 ค่าความจำเพาะ (Specificity)

ความจำเพาะเป็นสมรรถภาพของการทดสอบในการตรวจหาที่ที่ไม่มีภาวะนั้น ๆ ลองพิจารณาตัวอย่างการทดสอบภาพคนไข้ของเราค่าความจำเพาะของการทดสอบก็คือ สัดส่วนของภาพบุคคลปกติที่ไม่มีโรคผู้จะทดสอบได้ผลลบ ซึ่งสามารถเขียนเป็นสมการ ดังนี้

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}}$$

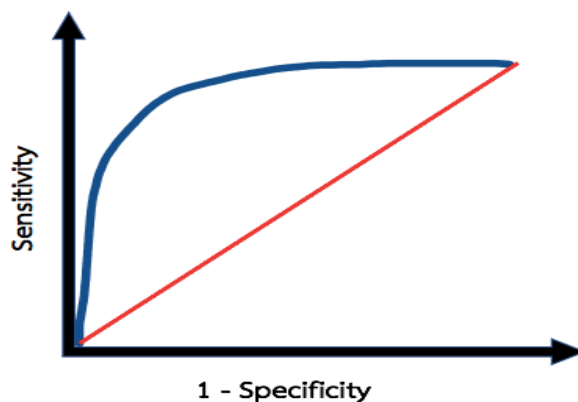
การได้ผลบวกจากการทดสอบจำเพาะสูงมีประโยชน์ในการวินิจฉัยว่าเป็นภาพที่เป็นโรค เพราะว่าการทดสอบนี้ไม่ค่อยให้ผลบวกในภาพคนปกติ เมื่อผลทดสอบเป็นบวกการทดสอบที่จำเพาะ 100% แสดงว่า ทั้งหมดเป็นภาพผู้ป่วยโดยไม่มีคนปกติ

2.22 ศัพท์เฉพาะและคำอนุพันธ์

ศัพท์เฉพาะและคำอนุพันธ์	
Positive (P) คือ การสังเกตผลเป็นบวก	Negative (N) คือ การสังเกตผลเป็นลบ
True Positive (TP) คือ ค่าที่พยากรณ์ถูกต้องเชิงบวก	True Negative (TN) คือ ค่าที่พยากรณ์ถูกต้องเชิงลบ
False Positive (FP) คือ ค่าที่พยากรณ์ผิดต้องเชิงบวก	False Negative (FN) คือ ค่าที่พยากรณ์ผิดต้องเชิงลบ
Accuracy คือ สัดส่วนของจำนวนคำทำนายทั้งหมดที่ถูกต้อง	
Precision คือ สัดส่วนของกรณีบวกที่เกิดขึ้นจริงต่อกรณีที่เป็นบวกทั้งหมด	
Sensitivity or Recall คือ สัดส่วนของกรณีบวกที่เกิดขึ้นจริงซึ่งระบุได้อย่างถูกต้อง	
Specificity คือ สัดส่วนของกรณีเชิงลบที่เกิดขึ้นจริงซึ่งระบุได้อย่างถูกต้อง	

2.23 พื้นที่ใต้เส้นโค้ง ROC (AUC - ROC)

ROC (Receiver Operation Characteristic) เป็นการพลอตกราฟระหว่าง sensitivity และ 1- specificity โดยที่ 1- specificity นั้นก็คือ false positive rate และ sensitivity นั้นก็คือ True Positive rate นั่นเอง ตามรูปที่ 3.12 เป็นตัวอย่าง ROC curve



รูปที่ 3.12 แสดง ROC Curve

2.24 งานวิจัยที่เกี่ยวข้อง

เราได้ศึกษาหาข้อมูลเพิ่มเติมและทำความเข้าใจเกี่ยวกับงานวิจัยต่าง ๆ ทางด้านการใช้เทคนิค CNN (Deep learning), โมเดล และผลลัพธ์ของแต่ละงานวิจัย ซึ่งประกอบไปด้วย

งานวิจัยของ Rachna Jain, Meenu Gupta, Soham Taneja และ D.Jude Hemanth และกลุ่มใช้ชุดข้อมูลจากฐานข้อมูลสาธารณะเพื่อทดสอบกับ 3 โมเดล คือ Inception V3, Xception Net และ ResNeXt สรุปผลได้ว่าโมเดลที่ให้ค่าความถูกต้องสูงสุด คือ Xception โดยทำคะแนนสูงถึง 97.97%[6]

งานวิจัยของ Joaquim de Moura, Jorge Novo และ Marcos Ortega ได้ปรับปรุงประสิทธิภาพการจำแนกรูปภาพด้วย CNN โดยใช้ชุดข้อมูลภาพจากสาธารณะ สรุปว่าผลลัพธ์ที่ได้นั้นมีประสิทธิภาพสามารถช่วยเพิ่มความสะดวกให้กับแพทย์ผู้เชี่ยวชาญ[7]

งานวิจัยของ Khandaker Foysal และ Ahmed Abdelgawad ได้ใช้ 3 โมเดล คือ ResNet 50, VGG-16, VGG-19 กับชุดข้อมูล 2 ชุด สรุปผลการทดสอบที่มีประสิทธิภาพมีค่าความถูกต้อง 98.3% ความแม่นยำ 96.72% ค่าพื้นที่ใต้กราฟ (ROC) 0.983 และ F1-score (ค่าความแม่นยำ*ค่าการเรียกคืน*2/ค่าความแม่นยำ+ค่าการเรียกคืน) ของชุดข้อมูลที่ 2 ที่มีจำนวนภาพมากกว่า ซึ่งให้ข้อมูลที่มีประโยชน์ในการใช้สำหรับวิจัยโดยเฉพาะอย่างยิ่งการประมวลผลภาพเพื่อจำแนกโควิด (COVID-19)[8]

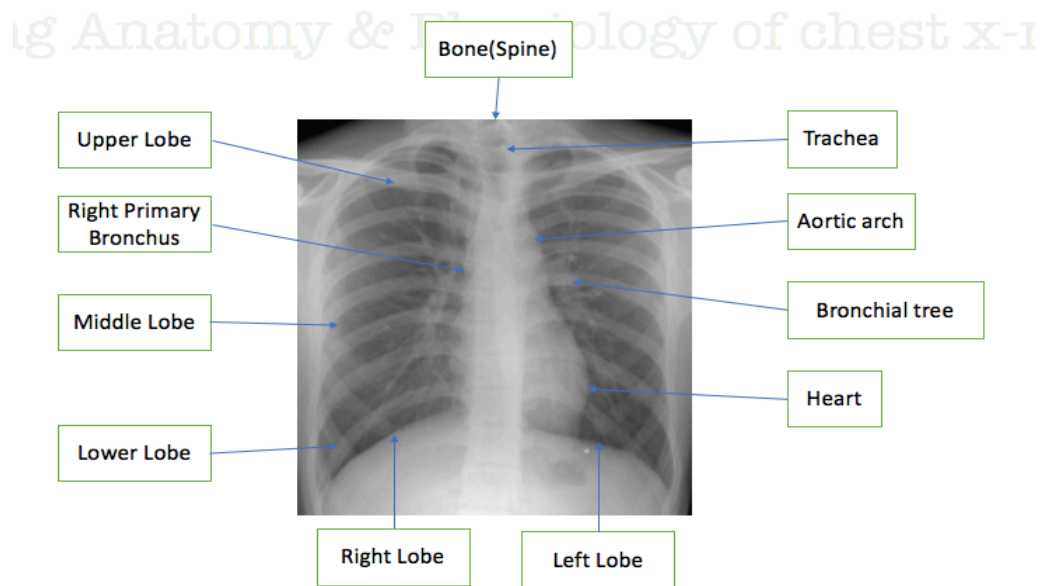
จากการศึกษางานวิจัยที่เกี่ยวข้องข้างต้นพบว่าสามารถนำ (CNN: Deep learning) มาใช้ในการจำแนกภาพเอกซเรย์เพื่อวินิจฉัยโรคโควิด (COVID-19) ได้เป็นอย่างดี

บทที่ 3

ขั้นตอนและวิธีดำเนินการวิจัย

3.1 ลักษณะความผิดปกติของภาพเอกซเรย์ปอดอักเสบโควิด (COVID-19)

จากการศึกษารายละเอียดของงานวิจัยลักษณะของภาพเอกซเรย์ปอดที่เกิดจากการติดเชื้อไวรัส COVID-19 พบว่าในภาพเอกซเรย์ปอด (CXR) ด้วยพลังงานจากรังสีเอกซ์ได้ให้ทำให้สีของภาพที่แตกต่างกันออกไปตามหลักการตกกระทบของพลังงานอิเล็กตรอนไปยังบนแผ่นรับภาพ ดังนั้นจะเห็นว่าสีที่แสดงบนแผ่นรับภาพ เช่น ที่ขาว เป็นส่วนของกระดูก ทา เป็นส่วนของเนื้อเยื่อและดำคือ บริเวณที่เป็นส่วนที่บางกลวงหรือบาง เช่น ปอด จะสามารถบ่งบอกถึงคุณลักษณะความหนาแน่นของอวัยวะที่รังสีไปตกกระทบพื้นผิวแตกต่างกันด้วย ปัจจัยดังกล่าวจึงช่วยทำให้แพทย์สามารถวินิจฉัยและแปลผลถึงความแตกต่างระหว่างภาพที่ปกติและแยกความเป็นโรคได้ด้วยคุณลักษณะดังกล่าวของภาพเอกซเรย์ แต่การจะแยกภาพที่ผิดปกติอย่างภาพปอดอักเสบทั่วไปกับปอดอักเสบที่ติดเชื้อโควิดทำได้ค่อนข้างยากเนื่องจากภาพเอกซเรย์จะแสดงภาพปอดที่ติดเชื้อเป็นสีขาวเหมือนกันทั้งปอดอักเสบทั่วไปและปอดที่ติดเชื้อโควิด ทำให้ต้องใช้เทคนิคอื่นร่วมด้วย เช่น การตรวจ ATK ทางจมุกหรือปาก เป็นต้น ข้อเสนอแนะควรให้แพทย์ผู้เชี่ยวชาญตรวจสอบเพื่อเป็นการสร้างความมั่นใจและประกอบการพิจารณาการรักษา ดังตัวอย่างภาพเอกซเรย์ปอดปกติ รูปที่ 3.1



รูปที่ 3.1 แสดงภาพเอกซเรย์ปอดปกติ (Normal) และ Anatomy ของปอด (Lung)

ลักษณะภาพปอดโควิด COVID-19 จากภาพถ่ายเอกซเรย์ปอด (CXR) ที่แสดงความผิดปกติที่เข้าได้กับภาวะปอดอักเสบจากโรคโควิด COVID-19 ที่มีลักษณะเป็นฝาบาง ๆ หรือ รูปร่างกลมกระจายทั่วบริเวณปอดทั้ง 2 ข้าง ดังรูปที่ 3.2



รูปที่ 3.2 แสดงภาพเอกซเรย์ปอดอักเสบที่ติดเชื้อโควิด (COVID-19)

ลักษณะปอดภาพปอดอักเสบทั่วไป (Pneumonia) ที่แสดงความผิดปกติที่ภาวะปอดอักเสบที่มีลักษณะเป็นฝาบาง ๆ ปื้น ๆ กระจายทั่วปอด ดังรูป 3.3

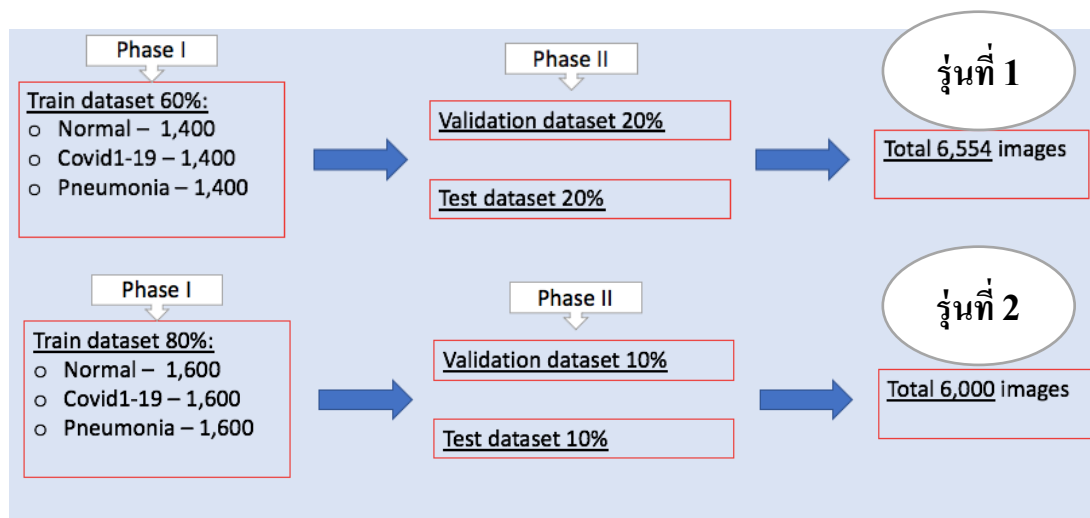


รูปที่ 3.3 แสดงภาพเอกซเรย์ปอดอักเสบทั่วไป (Pneumonia)

3.2 การเตรียมชุดข้อมูล

จากการดำเนินงานวิจัยนี้เราใช้ภาพเอกซเรย์ปอด 2 มิติ ชนิดไฟล์ PNG และ JPEG ขนาด 400x400 pixel และเป็นข้อมูลภาพเอกซเรย์สาธารณะแบบ (PA),(AP) โดยใช้เทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN: Deep-learning) ในการคัดกรองผู้ป่วยที่ติดเชื้อโควิด

COVID-19 เราได้จำแนกชุดข้อมูลภาพออกเป็น 3 กลุ่ม สำหรับฝึกสอนโมเดลคือ (Training) ตรวจสอบโมเดล (Validation) และกลุ่มสุดท้ายสำหรับทดสอบโมเดล (Testing) ประกอบด้วยกลุ่ม Negative คือภาพกลุ่มเอกซเรย์ปอดปกติ 1 กลุ่มและ Positive 2 กลุ่ม คือภาพเอกซเรย์ปอดที่ติดเชื้อโควิด (COVID-19) และกลุ่มภาพเอกซเรย์ปอดติดเชื้อที่ปอดอักเสบทั่วไป ตามลำดับ โดยที่จะแบ่งชุดข้อมูลภาพการทดลองเป็นสองรุ่น คือรุ่นแรกแบ่ง 60:20:20 ไม่มี Augmentation และรุ่นที่สอง 80:10:10 เพิ่ม Augmentation ดังรูปที่ 3.4



รูปที่ 3.4 แสดง Flow Chart แสดงขั้นตอนการแบ่งข้อมูลภาพรุ่นที่ 1 และ 2

3.3 เครื่องมือ+สเปกคอมพิวเตอร์ที่ใช้ในการวิจัย

- ไลบรารี Keras เวอร์ชัน 2.4.3
- Tensorflow เวอร์ชัน 2.3.0
- ภาษา Python เวอร์ชัน 3.8.11
- หน่วยประมวลผลกลาง (Central Processing Unit; CPU) เป็น Intel (R) Core (TM) i5-1035G1 CPU @ 1.00 GHz 1.19 GHz หน่วยความจำหลัก (Random Access Memory; RAM) ขนาด 8GB ระบบปฏิบัติการ (Operating System; OS)
- Windows 11 และหน่วยประมวลผลด้านกราฟิก (Graphics Processing Unit; GPU) เป็น NVIDIA GeForce MX350
- Computer laptop รุ่น LENOVO ปี 2020

3.4 การออกแบบอัลกอริทึมของโปรแกรม

ในส่วนของการประมวลผลภาพนั้น เราได้ใช้ไลบรารีจาก Tensorflow ซึ่งใช้พัฒนา ML (เขียนด้วยโปรแกรม Python) ที่พัฒนาตัวรูปแบบโดย Google ซึ่งมีความยืดหยุ่นและมีประสิทธิภาพในการประมวลผลได้รวดเร็ว สะดวกต่อการใช้งานเป็นอย่างมากเริ่มต้นจากการติดตั้งโปรแกรม Anaconda จากเว็บไซต์ไปยังตัวคอมพิวเตอร์ที่ทำการสอนโมเดลและใช้ CPU ในการรันโปรแกรม

หมายเหตุ : โปรแกรมภาษา python และวิธีการลงไลบรารีอยู่ในภาคผนวก ก.

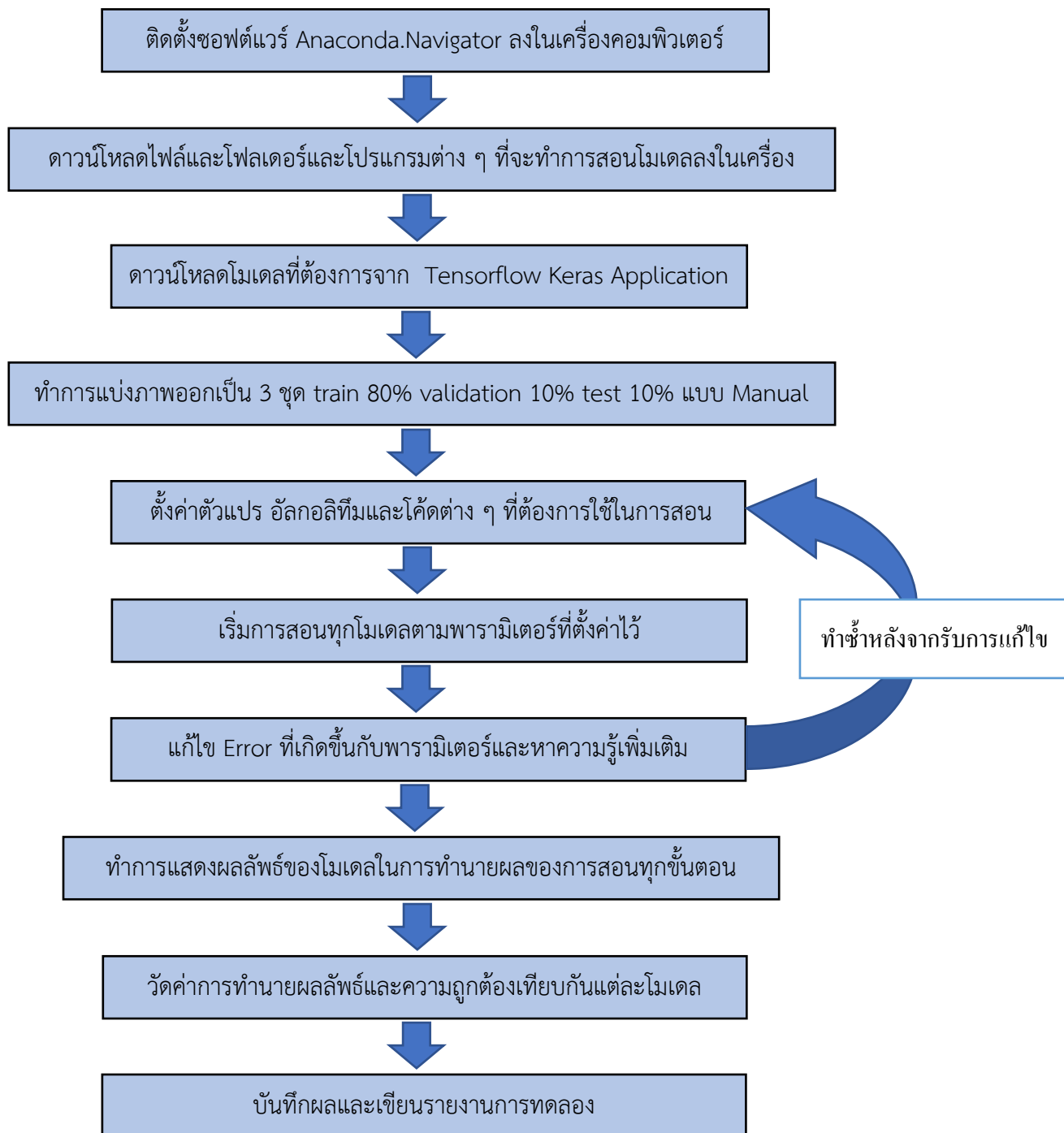
3.4.1 เทคนิคที่เราได้ทำการสอนโดยมีตัวแปรในการสอนโมเดล ดังนี้

- learning rate = 0.0001
- Train batch size = 32
- Epochs = 50
- Optimizer = Adam (Adaptive Moment Estimation)

3.4.2 โมเดลที่ใช้ในงานวิจัย

- Inception V3
- Resnet 50
- VGG16
- VGG19
- Mobilenet v2
- Inception Resnet V2

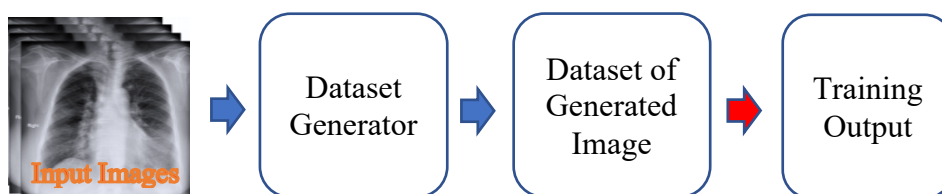
3.5 ขั้นตอนและกระบวนการเตรียมงานวิจัย



รูปที่ 3.5 แสดง Flow Chart ขั้นตอนการสอนโมเดลงานวิจัย

3.6 ขั้นตอนกระบวนการฝึกสอน

กระบวนการฝึกสอนแบบจำลองของงานวิจัยนี้แสดงขั้นตอนคร่าว ๆ ดังรูปที่ 3.13 ดังนี้



รูปที่ 3.6 แสดงกระบวนการนำข้อมูลภาพเข้าฝึกสอน

วิจัยเรื่องนี้มีวัตถุประสงค์เพื่อศึกษาการจำแนกรอยโรค 2 ชนิด คือภาพปอดอักเสบทั่วไปและภาพปอดอักเสบโควิดจากภาพปอดปกติ โครงสร้างเครือข่ายประสาทเทียม CNNs และเพื่อเปรียบเทียบการทำงานของแต่ละโมเดล โดยจะเป็นการนำเสนอโมเดลที่มีการแยกชั้นที่ลึกโดยอ้างอิงรูปแบบมาจาก 6 โมเดล คือ Inception V3, ResNet 50, VGG16, VGG19, Mobilenet V2 และ InceptionResNet V2 ซึ่งสามารถโหลดได้จาก <https://keras.io/api/applications/inceptionv3/> และชุดข้อมูลสาธารณะจากเว็บไซต์ Kaggle.com

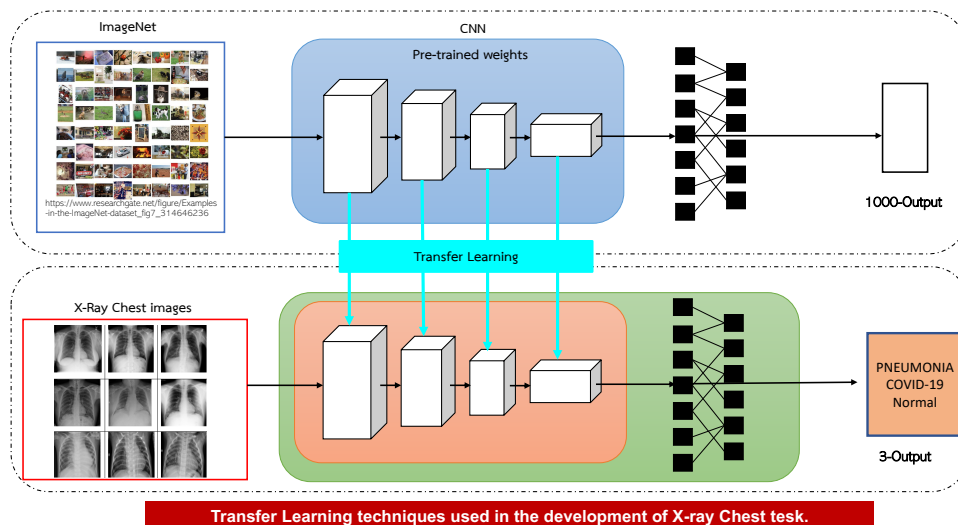
3.6.1 ศึกษาการทำงานของแบบจำลองกับการจำแนกภาพปอดทั้ง 3 ชนิด

3.6.2 ออกแบบโครงสร้างเครือข่ายประสาทเทียม CNNs ปรับปรุงแบบมาจากแบบจำลองทั้ง 6 แบบ

การฝึกโมเดลการเรียนรู้เชิงลึกที่ซับซ้อน (transfer learning) ตั้งแต่เริ่มต้น (การเริ่มต้นด้วยน้ำหนักด้วยค่าสุ่ม) ต้องใช้ทั้งข้อมูลการฝึกอบรม ขนาดใหญ่และพลังในการคำนวณสูง ดังนั้นการใช้การเรียนรู้แบบถ่ายทอดจึงเป็นเทคนิคที่ได้รับความนิยม เนื่องจากจะช่วยลดเวลาในการฝึก โมเดลขนาดใหญ่ที่ผ่านการฝึกแล้วบางส่วนกับงานที่คล้ายกัน เป็นส่วนหนึ่งของโมเดลใหม่ ผลลัพธ์ของโมเดลที่ได้รับการ pre-trained จะถูกใช้เพื่อดึงคุณสมบัติผ่าน ตัวแปรใหม่แบบเชื่อมทุกเลเยอร์ (fully dense) จะถูกดึง latent feature (สร้างโดยโมเดล transfer learning ผ่านการ load weight จาก imagenet) คุณสมบัติเหล่านี้ใช้เป็นอินพุตสำหรับโมเดล machine learning ในชั้น fully connected

เรานำชุดข้อมูลภาพจำนวนกว่า 6,000 ภาพ ภาพ 2 มิติ ประเภท JPEG และ PNG โครงสร้างเครือข่ายประสาทเทียม CNNs โดยจะเป็นการนำเสนอโมเดลที่มีปรับแต่งบางส่วนโดยอ้างอิงรูปแบบมาจากโมเดลทั้ง 6 ในที่นี้จะขอล่าถึงการใช้ตัวอย่างการ Transfer learning โดยใช้ model

Inception-v3 ซึ่งสามารถโหลดได้จาก <https://keras.io/api/applications/inceptionv3/> ซึ่งจะประกอบด้วย 2 ส่วนคือ base model และ head model โดยมีคำสั่งที่ใช้ในการโหลด base model inception v3 เข้ามาใช้งานดังนี้กรณีเปลี่ยน head model ให้เปลี่ยน include top จาก true จากนั้นจึงทำการเพิ่ม layers ที่เราต้องการและทำการฝึกแบบ fix weight และ bias ในส่วนของ base model และทำการตรวจสอบค่าความแม่นยำของโมเดลที่เพิ่ม head layers เข้าไปโดยเทรนทั้งหมดของ network



รูปที่ 3.7 แสดงกระบวนการนำข้อมูลภาพเข้าฝึกสอนโดยใช้โมเดล CNNs network

บทที่ 4

การทดลองและผลการทดลอง

4.1 ระบบปฏิบัติการงานวิจัยและภาษาที่ใช้ในการทดลองในงานวิจัย

จากงานวิจัยการสร้างแบบจำลองโครงข่ายประสาทเทียม (CNN: Deep learning) เราได้ใช้ Computer รุ่น LENOVO ปี 2020 ติดตั้ง Software ไลบรารีที่ประกอบด้วย keras เวอร์ชัน 2.4.3 Tensorflow เวอร์ชัน 2.3.0 กับกระบวนการฝึกสอนด้วยโมเดลทั้ง 6 โมเดลคือ InceptionV3, ResNet50, VGG16, VGG19, Mobilenet V2 และ InceptionResNet V2 จาก keras application และ Tensorflow เป็นไลบรารีที่พัฒนาด้วยภาษา Python ที่มีประสิทธิภาพในกระบวนการฝึกสอนเพื่อจำแนกรูปภาพ ซึ่งถูกใช้ในงานวิจัยก่อนหน้านี้และภาษา Python เวอร์ชัน 3.8.11 คอมพิวเตอร์ที่ใช้หน่วยประมวลผลกลาง (Central Processing Unit; CPU) เป็น Intel (R) Core (TM) i5-1035G1 CPU @ 1.00 GHz 1.19 GHz หน่วยความจำหลัก (Random Access Memory; RAM) ขนาด 8GB ระบบปฏิบัติการ (Operating System; OS) เป็น Windows 11 และหน่วยประมวลผลด้านกราฟิก (Graphics Processing Unit; GPU) เป็น NVIDIA GeForce MX350

4.2 ชุดข้อมูลที่ใช้ในการทดลองและวิธีแบ่งข้อมูลเพื่อสอนโมเดล

ชุดข้อมูลที่ใช้ในการศึกษาวิจัยนี้ถูกดาวน์โหลดจากที่เก็บชุดข้อมูลสาธารณะ Kaggle.com ชุดข้อมูลแรก (CXR: Images Pneumonia (ภาพปอดอักเสบทั่วไป)), ภาพปอดอักเสบโควิด COVID-19 และภาพปอดปกติ (Normal) ภาพเอกซเรย์กว่าจำนวน 6,000 ภาพ ในรูปแบบ JPEG และ PNG ประกอบด้วยภาพเอกซเรย์ปอดที่ถ่ายเอกซเรย์จากด้านหน้า(AP) และด้านหลังไปหน้า(PA) ของผู้ป่วย โดยสามารถสังเกตจากตัวmaker R และ L ที่ติดในข้อมูลภาพจะสามารถแยกด้านขวาหรือซ้ายของผู้ป่วยในภาพได้ซึ่งใช้ในการทดลองเป็นข้อมูลภาพเอกซเรย์ปอดจากฐานข้อมูลสาธารณะจากรอยโรคโควิด COVID-19 กว่า จำนวน 2,000 ภาพ (โดยได้ทำการนิรนามข้อมูลแล้ว) ชุดข้อมูลถูกแบ่งออกเป็นสามส่วน คือการฝึกอบรม(Training) การตรวจสอบ(Validation) และการทดสอบ(Testing) การทดลองนี้แบ่งข้อมูลภาพ 2 ครั้งโดยใช้ชุดข้อมูลเดียวกันแต่แบ่งเป็นสัดส่วนที่แตกต่างกันคือ 60:20:20 ในรุ่นที่ 1 และ 80:10:10 ในรุ่นที่ 2 แต่ละส่วนได้รับการแบ่งย่อยเป็นสามคลาสคือ คลาสปกติ(Normal), ปอดอักเสบโควิด(COVID-19) และโรคปอดอักเสบ(Pneumonia) ใช้ภาพเอกซเรย์ปอด 6,554 ภาพ เป็นข้อมูลการฝึกอบรม(Training) ภาพเอกซเรย์ปอด 4,200 ภาพ สำหรับการตรวจสอบความถูกต้อง (Validation) 1,182 ภาพและภาพสำหรับการทดสอบ(Testing) โดยที่ 1,172 ภาพ เป็นโรคปอดอักเสบทั่วไป(Pneumonia) 2,250 ภาพ เป็นปอดปกติ(Normal) 2,180 ภาพ และปอดอักเสบโควิด (COVID-19) 2,124 ภาพ ในการทดลองรุ่นแรก

ชุดที่สองแบ่งข้อมูลภาพมีภาพเอกซเรย์ปอดโควิด(COVID-19) เอกซเรย์ปอดปกติและโรคปอดอักเสบทั้งหมดจำนวนคลาสละ 2,000 ภาพ จากภาพทั้งหมด 6,000 ภาพ เป็นข้อมูลการฝึกอบรม (Training) ภาพเอกซเรย์ปอดทั้งสิ้น 4,800 ภาพ สำหรับการตรวจสอบความถูกต้อง(Validation) 600 ภาพ และภาพสำหรับการทดสอบ(Testing) 600 ภาพ รายละเอียดของข้อมูลภาพมีอยู่ในตารางที่ 1,2,3 และตัวอย่างแสดงภาพเอกซเรย์ปอดจากโควิด(COVID-19) โรคปอดอักเสบทั่วไปและปอดปกติ จะแสดงที่รูป 4.1 รายละเอียดของข้อมูลแต่ละฐานข้อมูลเป็นดังนี้

ตารางที่ 4.1 แสดงขั้นตอนการแบ่งภาพและชนิดของภาพ รุ่นที่ 1

ชนิดของภาพ	Train Data	Validation Data	Test Data	Total of images
Normal	1,400	360	460	2,180
COVID-19	1,400	362	362	2,124
Pneumonia	1,400	460	390	2,250
ภาพทั้งหมด	4,200	1,182	1,172	6,554

ตารางที่ 4.2 แสดงขั้นตอนการแบ่งภาพและชนิดของภาพ รุ่นที่ 2

ชนิดของภาพ	Train Data	Validation Data	Test Data	Total of images
Normal	1,600	200	200	2,000
COVID-19	1,600	200	200	2,000
Pneumonia	1,600	200	200	2,000
ภาพทั้งหมด	4,800	600	600	6,000

ตารางที่ 4.3 รายละเอียดของภาพเอกซเรย์ทรวงอกแยกตามฐานข้อมูล



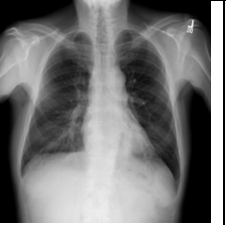
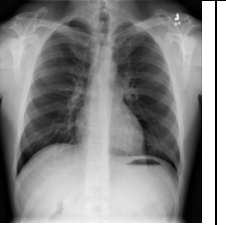
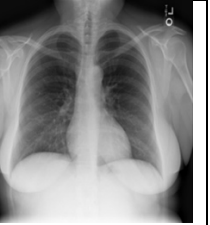

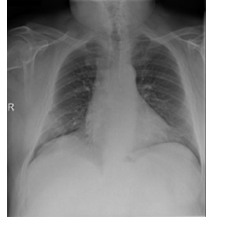

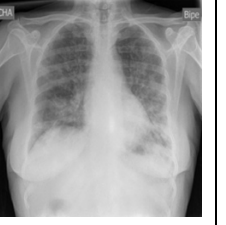

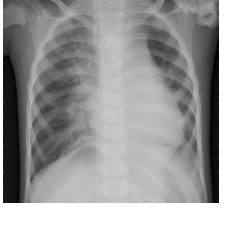
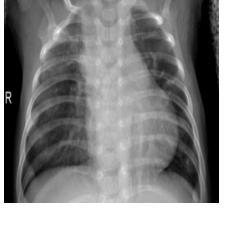

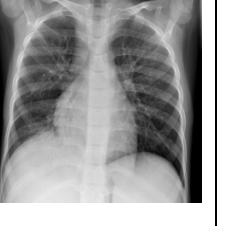
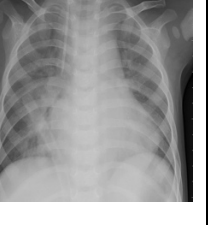
แหล่งข้อมูล	จำนวนภาพเอกซเรย์ (ปกติ)	จำนวนภาพเอกซเรย์ (เป็นปอดอักเสบทั่วไป)	จำนวนภาพเอกซเรย์ (เป็นปอดอักเสบ COVID-19)	ภาพรวม
ชุดที่ 1 รวม	2,180	2,250	2,124	6,554
ชุดที่ 2 รวม	2,000	2,000	2,000	6,000

4.3 การเตรียมชุดข้อมูลและโมเดลที่ใช้ในการทดลอง

เพื่อทำการทดลองโครงข่ายประสาทเทียม(CNN: Deep learning) ด้วยสถาปัตยกรรมทั้ง 6 แบบคือ Inception V3, ResNet50, VGG16, VGG19, Mobilenet V2และInceptionResNet V2 มีการเตรียมชุดข้อมูลด้วยวิธี Data augmentation และการ resampling ดังต่อไปนี้เพื่อการเพิ่มจำนวนภาพให้สมดุลช่วยลดปัญหา Overfitting ของข้อมูลและช่วยเพิ่มประสิทธิภาพการทำงานของโมเดลให้ดีขึ้น

ในงานวิจัยนี้เราได้ใช้จากการทดลองข้อมูลภาพเอกซเรย์ปอดสาธารณะจากเว็บ Kaggle.com โดยจะแบ่งข้อมูลออกเป็น 3 ชุดคือ Training dataset, Validation dataset และ Testing dataset แบ่งออกเป็นชุดข้อมูลสำหรับการสอน (Training dataset) สำหรับการตรวจสอบ (Validation dataset) และข้อมูลสำหรับการทดสอบ (Test dataset) แบบจำลองชุดแรก 60:20:20, ชุดที่ 2 เป็น 80:10:10 โดยใช้ของจำนวนข้อมูลภาพทั้งหมดเดียวกัน เพื่อเปรียบเทียบการทำงานของแต่ละโมเดลที่ให้ค่าความถูกต้อง (Accuracy) และความแม่นยำ (Precision) ที่ดีที่สุด

ตารางที่ 4.4 แสดงตัวอย่างภาพเอกซเรย์ปอดปกติ ปอดอักเสบทั่วไปและปอดอักเสบโควิด COVID-19

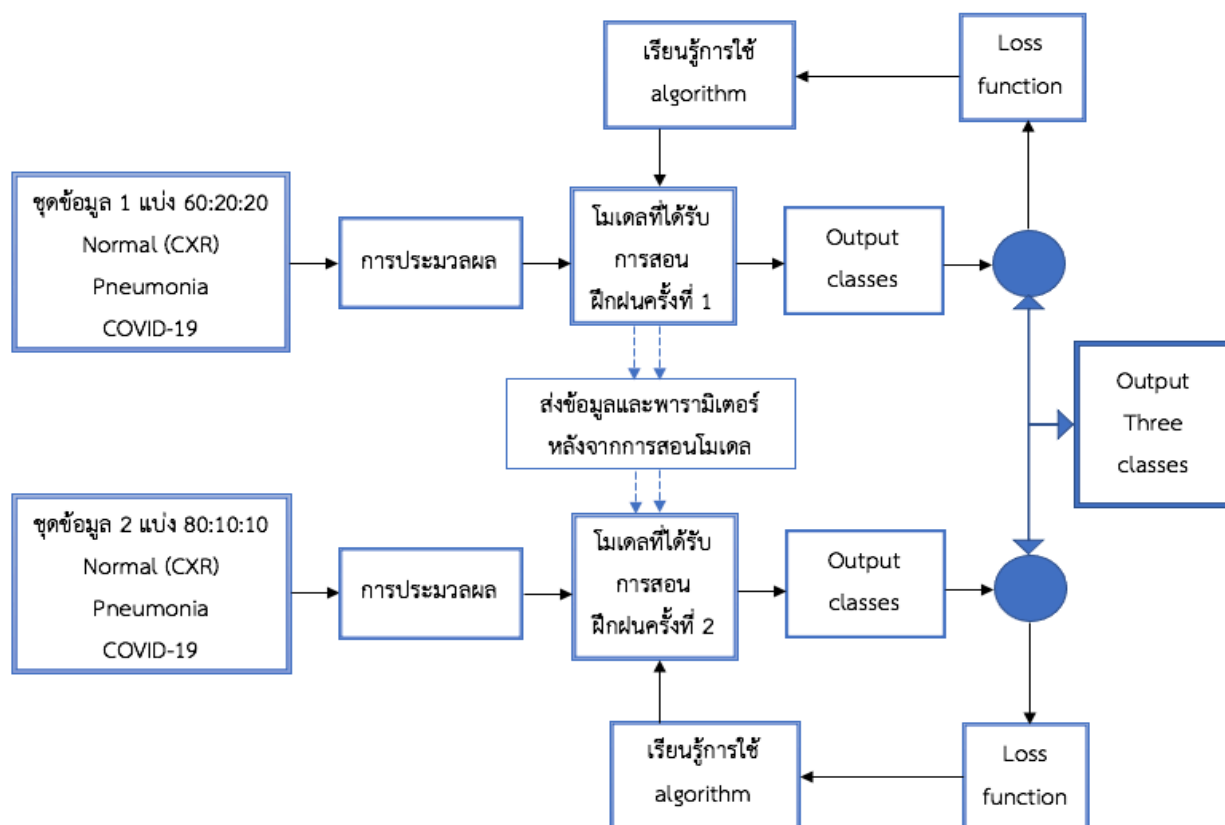
Normal					
COVID-19					
Pneumonia					

4.4 วิธีการดำเนินการทดลอง

การดำเนินการทดลองในงานวิจัยนี้จะแบ่งเป็น 2 ช่วง โดยช่วงแรกเป็นการทดลองเพื่อหาโมเดลที่ดีที่สุดในการทำนายผลที่เหมาะสมกับชุดข้อมูลภาพเอกซเรย์ปอด (CXR: Chest X-ray) จากในแต่ละสถาปัตยกรรมจะให้ผลที่ดีกับภาพแตกต่างกันแต่ละโมเดล เราจึงต้องทดสอบภาพเอกซเรย์ปอดซึ่งเป็นสีขาว ดำ เทา เป็นส่วนใหญ่และมีการใช้การถ่ายทอดการเรียนรู้มาเพื่อช่วยลดระยะเวลาในการเรียนรู้ของแต่ละแบบจำลองโดยในงานวิจัยได้เลือกใช้แบบจำลองซึ่งเป็นที่นิยมได้แก่ InceptionV3, ResNet50, VGG16, VGG19, MobileNetV2 และ InceptionResNetV2 เพื่อเปรียบเทียบดูว่าโมเดลใดที่จะเหมาะสมที่สุด มีขั้นตอนดังต่อไปนี้

1. การเตรียมชุดข้อมูลการวิจัยที่มีโรคปอดอักเสบทั่วไป ภาพเอกซเรย์ปอดปกติ และภาพอักเสบปอดจากโควิด (COVID-19) การแบ่งภาพของภาพในการฝึกอบรมการตรวจสอบและชุดทดสอบด้วยเทคนิค manual โดยแบ่ง 60:20:20
2. ประมวลผลภาพอินพุตล่วงหน้าใช้ชุดข้อมูลภาพแบบดั้งเดิม
3. การใช้โครงสร้าง CNN: Deep learning ที่สร้างขึ้นสำหรับการคำนวณเอาต์พุตของโมเดลเป้าหมาย/โดยใช้ภาพเอกซเรย์หน้าอกของโรคปอดอักเสบทั่วไป ปอดปกติ และปอดอักเสบกับการติดเชื้อโควิด COVID-19
4. การคำนวณคลาสเอาต์พุต
5. เปรียบเทียบเอาต์พุตปัจจุบันกับคลาสที่ต้องการและการคำนวณฟังก์ชันการสูญเสีย
6. ใช้ฟังก์ชันการสูญเสียและอัลกอริทึมการฝึกอบรมเพื่อปรับพารามิเตอร์ของ CNN
7. การใช้ขั้นตอนที่ 3-6 สำหรับชุดข้อมูลทั้งหมดและรวมถึงจำนวน epochs ทั้งหมด
8. ใช้โมเดลที่ได้รับการฝึกฝน/ต้นฉบับ (รุ่นแรก) ซึ่งได้รับการฝึกฝนด้วยชุดข้อมูลที่ประมวลผลล่วงหน้ารวมถึงโรคปอดอักเสบ ภาพปอดปกติและภาพปอดอักเสบจากโควิด COVID-19 เป็นสถาปัตยกรรมแบบจำลองพื้นฐานสำหรับการใช้งานแบบจำลองที่สองของเทคนิคการเรียนรู้การถ่ายโอน
9. ปรับข้อมูลภาพและเตรียมชุดข้อมูลการวิจัยจาก 60:20:20 เป็น 80:10:10 รวมถึงการแบ่งภาพเป็นการฝึกอบรมการตรวจสอบและชุดทดสอบ
10. การประมวลผลภาพอินพุตล่วงหน้ารวมถึงการปรับขนาดภาพการเพิ่มข้อมูลและการสุ่มข้อมูลใหม่
11. ใช้โครงสร้าง CNN ที่สร้างขึ้นสำหรับการคำนวณเอาต์พุตของโมเดลเป้าหมาย/โดยใช้ภาพเอกซเรย์ปอดของโควิด COVID-19, โรคปอดอักเสบและปอดปกติ
12. ใช้อัลกอริทึมการฝึกอบรมสำหรับการอัปเดตพารามิเตอร์ CNN ของโมเดล pretrained

13. ใช้ขั้นตอนที่ 10–12 สำหรับชุดข้อมูลทั้งหมดและจำนวนของ epochs ทั้งหมด
14. เอาต์พุตโมเดลเป้าหมายเป็นส่วนการตัดสินใจของสถาปัตยกรรมแบบจำลองที่กำหนดว่าภาพเอกซเรย์เป็นของโควิด COVID-19, ปอดอักเสบ (Pneumonia) หรือภาพปอดปกติสุขภาพดีกรณีปกติ

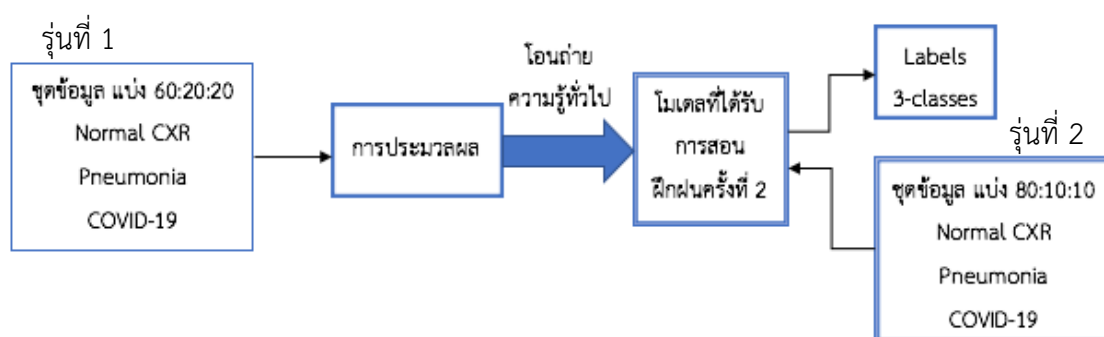


รูปที่ 4.1 แสดงกระบวนการฝึกสอนโมเดลรุ่นที่ 1 ไปยังรุ่นที่ 2

4.5 ขั้นตอนการโอนถ่ายข้อมูล

วิธีการถ่ายโอนข้อมูลถูกนำมาใช้ในการพัฒนาแบบจำลองในงานวิจัยนี้ เราได้เริ่มการฝึกสอนโดยใช้ชุดข้อมูลเดียวกันคือ รุ่นแรกได้รับการฝึกสอนโดยแบ่งชุดข้อมูลเป็นภาพปอดปกติ ภาพโรคปอดทั่วไปและปอดอักเสบโควิด (COVID-19) อัตราส่วน 60:20:20 โดยไม่มีการ augmentation รุ่นที่สองใช้รุ่นแรกเป็นรุ่นพื้นฐานคือ ใช้ความรู้ที่เรียนรู้และพารามิเตอร์จากรุ่นแรกในการฝึกสอนโมเดลในชุดข้อมูลโดยที่เปลี่ยนการแบ่งข้อมูลภาพเป็นอัตราส่วน 80:10:10 แสดงให้เห็นถึงการเรียนรู้การถ่ายโอนในการสร้างแบบจำลองขั้นสุดท้ายการถ่ายโอนการเรียนรู้เป็นวิธีการนำรูปแบบการเรียนรู้ของเครื่องกลับมาใช้ใหม่

ซึ่งก่อนหน้านี้ได้รับการฝึกสอนด้วยข้อมูลจำนวนมากในงานเฉพาะเพื่อฝึกสอนตัวจำแนกใหม่ของงานที่คล้ายกันหรือแตกต่างกันโดยการปรับอัตราส่วนของข้อมูลและการเพิ่มการจัดรูปแบบของข้อมูลภาพ เช่น การ augmentation การ Cropping การหมุน การย่อ ขยายขนาดและการปรับแสงของรูปภาพเพื่อบอกถึงกระบวนการการฝึกอบรมแล้วดังที่แสดงในรูปที่ 4.5 การถ่ายโอนข้อมูลการเรียนรู้ช่วยในการฝึกสอนโครงข่ายประสาทแบบ Conv: CNN ด้วยชุดข้อมูลที่มีการปรับเปลี่ยนใหม่และการเรียนรู้การถ่ายโอนรูปแบบที่สร้างขึ้นด้วยโมเดลที่ผ่านการฝึกสอนด้วยชุดข้อมูลขนาดใหญ่เพื่อให้ทำงานได้ดีจากคุณสมบัติพื้นฐานทั้งหมดได้รับการเรียนรู้แล้วในรูปแบบที่ได้รับการสอนแล้ว เวลาที่ต้องใช้ในการฝึกสอนนั้นน้อยกว่าโดยไม่ต้องใช้วิธีการเรียนรู้การถ่ายโอนหน่วยความจำและทรัพยากรการคำนวณจะลดลงเมื่อเทียบกับการฝึกสอนตั้งแต่เริ่มต้นอัลกอริทึมที่ได้รับการสอนมากมาย การฝึกสอนด้วยชุดข้อมูลขนาดใหญ่ เช่น ชุดข้อมูล ImageNet [25] ซึ่งมีภาพมากกว่า 15 ล้านภาพจากประมาณ 22,000 หมวดหมู่ อัลกอริทึมที่ได้รับการฝึกฝนเหล่านั้นส่วนใหญ่จะใช้เป็นฐานในวิธีการเรียนรู้การถ่ายโอน อย่างไรก็ตามในงานวิจัยนี้แบบจำลองที่ใช้แล้วที่ใช้ได้รับการฝึกสอนด้วยชุดข้อมูลเอกซเรย์ปอด (โรคปอดอักเสบ) และปอดที่ติดเชื้อโควิด (COVID-19) ดังรูป



รูปที่ 4.2 แสดงกระบวนการโอนถ่ายข้อมูลไปยังรุ่นที่ 2

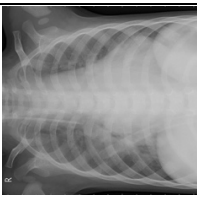
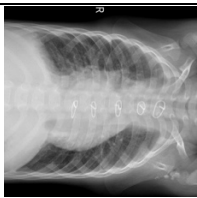

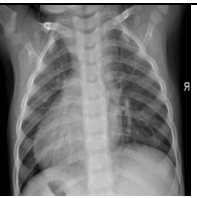






4.6 การเพิ่มข้อมูล (Data Augmentation)

กระบวนการการเพิ่มข้อมูลเป็นการเทคนิคหนึ่งของการฝึกสอนโมเดลในโครงข่ายประสาทเทียม (CNN) เพื่อสร้างข้อมูลการฝึกสอนให้มีจำนวนมากขึ้นจากข้อมูลเดิมที่มีอยู่ เพื่อเพิ่มประสิทธิภาพในชุดข้อมูล (Training set) 80% ปรับเพิ่มคุณภาพรูปภาพเพื่อเพิ่มประสิทธิภาพของชุดข้อมูลฝึกโดยใช้ Tensorflow import keras ผ่าน ฟังก์ชัน image data generator ซึ่งเป็นการทำ real-time augmentation และเป็นการเพิ่มข้อมูลบางคลาสที่มีจำนวนข้อมูลน้อยการใช้เทคนิคการประมวลผลภาพง่าย ๆ เช่น การตัดส่วนภาพที่ไม่ต้องการออก การขยับภาพ การย่อ-ขยายภาพ การพลิกกลับซ้าย-

ขวา การปรับเปลี่ยนสี โดยเทคนิค manual และการปรับด้วย code โดยลดภาระ Ram ของ เครื่อง ระหว่างการ augmentation ซึ่งกำหนดค่าและแสดงตัวอย่างภาพดังนี้

- ทำการปรับ Rotation range = 40 degree
- ทำการปรับ Width shift range = 20%
- ทำการปรับ Height shift range = 20%
- ทำการปรับ Brightness range = 20-40%
- ทำการปรับ Shear range = 20%
- ทำการปรับ Horizontal flip = True
- ทำการปรับ Vertical flip = True

ตารางที่ 4.5 แสดงตัวอย่าง Cropping การขยายภาพ การย่อ-ขยายภาพ การพลิกกลับซ้าย-ขวา การปรับเปลี่ยนสี

Pneumonia	Pneumonia	Pneumonia	Pneumonia	Pneumonia
				
หมุนขวา 90°	หมุนซ้าย 90°	หมุนขวา 180°	หมุนกลับหลัง	ปรับสีให้สว่าง
				
การย่อภาพ	การขยายภาพ	การ Cropping	การขยายภาพขึ้น	ปรับสีให้มีมืดลง
COVID-19	COVID-19	COVID-19	Normal	Normal

4.7 ขั้นตอนการ Training โมเดลและ Code

เป้าหมายหลักของงานวิจัยนี้ คือการพัฒนาแบบจำลองที่สามารถวินิจฉัยการปรากฏตัวของรอยโรคโควิด (COVID-19) แยกออกจากโรคปอดอักเสบทั่วไป (Pneumonia) และปอดปกติสุขภาพดี

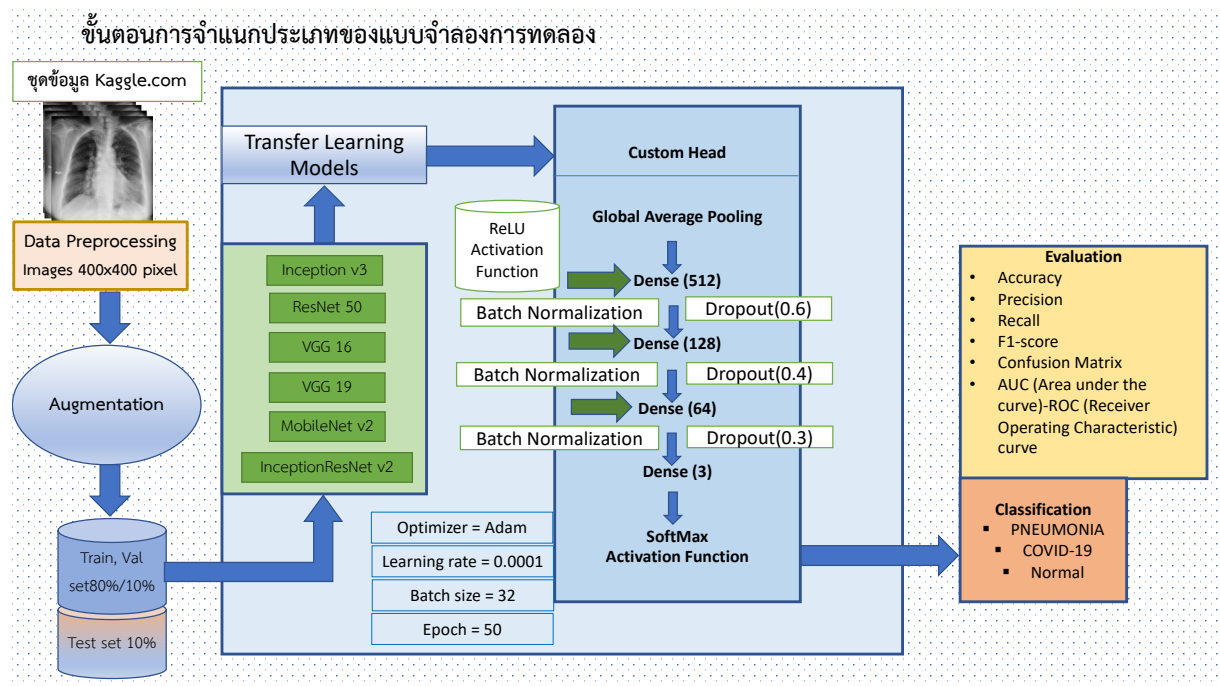
(Normal) โดยใช้ข้อมูลภาพเอกซเรย์ ชนิด 2 มิติ ประเภท PNG และ JPEG ขนาดของภาพ 400x400 pixel จำนวน 6,000 ภาพ ขั้นตอนวิธีการเรียนรู้การถ่ายโอนข้อมูลถูกนำมาใช้ในการพัฒนาแบบจำลอง โดยในงานวิจัยแบ่งการสอนโมเดลออกเป็นสองรุ่นด้วยกัน ใช้เทคนิคการปรับจำนวนข้อมูลภาพของแต่ละชุดให้แตกต่างกัน เริ่มจากรุ่นแรกกำหนดชุดข้อมูลอัตราส่วน 60:20:20 ใช้ภาพปอดปกติ โรคปอดอักเสบทั่วไป (Pneumonia) และปอดอักเสบอักเสบโควิด (COVID-19) รุ่นที่สองใช้โมเดลแรกเป็นแบบจำลองพื้นฐาน คือใช้เทคนิคการเรียนรู้มาจากรุ่นแรกมาใช้ในการฝึกสอนและการทดสอบแบบจำลองรุ่นที่สอง เครื่องมือที่ใช้ในการดำเนินการ เช่น Keras และ TensorFlow Framework ในภาษาการเขียนโปรแกรม ภาษา Python ประมวลผลโดยใช้ CPU

โครงสร้างของงานวิจัยใช้เทคนิคการ Transfer learning Model 6 โมเดลคือ Inception V3, ResNet 50, VGG16, VGG19, Mobilenet V2 และ InceptionResNet V2 และ Global Average Pooling เพื่อลดขนาดการ mapping และส่งต่อไปเลเยอร์ Dense ทั้งหมด 4 ชั้น แต่ละชั้นมีตัวกรอง (Filters) ขนาด (512, 128, 64, 3) การทำ Batch Normalize เพื่อช่วยเพิ่มความเร็วในการเรียนรู้ของ โมเดลให้มากยิ่งขึ้น Dropout เพื่อทำให้โมเดลทำงานได้มีประสิทธิภาพมากขึ้นโดยภายในชั้นจะมีสุ่ม เพื่อปิดการทำงานชั่วคราวในขณะการ Training โมเดล การทำ Activation function ใช้ในแต่ละชั้นคือ Relu function ยกเว้นในชั้นสุดท้ายที่จะใช้ Activation function เป็น Softmax function เพื่อไว้ทำนายผลลัพธ์ภาพเอกซเรย์ปอดว่าเป็นปกติหรือเป็นอักเสบทั่วไปหรือปอดอักเสบโควิด โดยการตั้งค่าพารามิเตอร์ต่าง ๆ คือ อัตราการเรียนรู้ (Learning rate) 0.0001 และ อัลกอริทึม Optimizer ที่ใช้คือ Adam และ batch size = 32 กับฝึก 50 epochs

4.8 Custom head

การนำโมเดลทั้ง 6 โมเดลคือ Inception V3, ResNet50, VGG16, VGG19, Mobilenet V2 และ InceptionResNet V2 จากขั้นตอน pre-trained โดยใช้ weight จาก ImageNet มาทำการ custom head และทำการ flatten โดยเพิ่มชั้น Dense 512, 128 และ 64 โดยใช้ activation = relu และกำหนดชั้นสุดท้ายเป็น dense 3 เพื่อกำหนด output โดยใช้ activation softmax เพื่อจำแนกภาพแยกเป็น 3 ชนิด ทำการ validation split = 10% และ compile model โดยใช้ optimizer = adam และ batch size = 32 และมีการใช้ loss แบบ sparse categorical crossentropy เพื่อให้ค่า output เป็น onehot encoding วัดประสิทธิภาพระหว่างการฝึกแต่ละรอบโดยใช้ metrics = accuracy และ learning rate ที่ 0.0001 และทำการฝึก 50 epochs และกำหนดสิ้นสุดการฝึกโดย

ใช้ earlyStopping โดยดู จากค่า loss ใน validation set ไม่ลดลงใน 3 รอบสุดท้ายของการฝึก (patience = 3) ดังรูปที่ 4.3



รูปที่ 4.3 แสดงโครงสร้างการทดลองร่วมกับกระบวนการต่าง ๆ ที่ใช้ในการวิจัย

หมายเหตุ : โปรแกรมภาษา python และวิธีการลงไลบรารีอยู่ในภาคผนวก ก.

จากภาพที่ 4.4 เป็นการดึงโมดูลและการเรียกใช้ไลบรารีที่จำเป็นออกมาในการฝึกสอนโมเดล

```
import numpy as np
from sklearn.metrics import accuracy_score, f1_score, precision_score,
confusion_matrix
from sklearn.model_selection import StratifiedKFold
from PIL import Image
import random
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
import warnings
import os
import shutil
from PIL import ImageFile
warnings.simplefilter('error', Image.DecompressionBombWarning)
ImageFile.LOAD_TRUNCATED_IMAGES = True
from PIL import Image
Image.MAX_IMAGE_PIXELS = 1000000000
config = ConfigProto()
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
```

ภาพที่ 4.4 แสดงการดั่งโมดูลที่จำเป็นต้องใช้ออกมาในการฝึกสอนโมเดล

- เราจึงต้องลดขนาดลงให้สามารถนำเข้ามา Train ใน Model ได้โดยใช้เวลาในการ Train แต่ละ Epoch ไม่มากนัก แต่ยังคงมีประสิทธิภาพที่ดี นอกจากนี้การปรับ Batch Size ก็มีผลต่อประสิทธิภาพของ Model เช่นกัน เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด ในการทดลองจึงควรมีการปรับขนาด Batch Size ค่อย

```
Width = 180
```

```
Batch_size 32
```

ภาพที่ 4.5 ชุดคำสั่งปรับค่าถ่วงน้ำหนักเท่ากับ 180 และการปรับขนาด Batch size 32

```
datasetFolderName='C:/chest 80_10_10%/'
MODEL_FILENAME="model_200_InceptionResNetV2_0.0001.h5"
sourceFiles=[]
classLabels=['NORMAL','PNEUMONIA','PNEUMONIA COVID 19']

img_rows, img_cols = 150, 150 # input image dimensions
train_path=datasetFolderName+'/train_80%/'
validation_path=datasetFolderName+'/validation/'
test_path=datasetFolderName+'/test/'
```

ภาพที่ 4.6 ชุดคำสั่งดึงภาพจากโฟลเดอร์เข้าโปรแกรมและกำหนดการแบ่งภาพเป็น 3 คลาสคือ ภาพปกติ ภาพปอดอักเสบและปอดอักเสบโควิด COVID-19 โดยกำหนดขนาดของภาพ Input = 150,150,3

```
def prepareNameWithLabels(folderName):
    sourceFiles=os.listdir(datasetFolderName+'/train/'+folderName)
    for val in sourceFiles:
        X.append(val)
        for i in range(len(classLabels)):
            if(folderName==classLabels[i]): Y.append(i)
```

ภาพที่ 4.7 ชุดคำสั่งการกำหนดชื่อของแต่ละคลาสของ Input นำเข้าฝึกสอน

```

def transferBetweenFolders (source, dest, splitRate):
    global sourceFiles
    sourceFiles=os.listdir(source)
    if(len(sourceFiles)!=0):
        transferFileNumbers=int(len(sourceFiles)*splitRate)
        transferIndex=random.sample(range(0, len(sourceFiles)),
transferFileNumbers)
        for eachIndex in transferIndex:
            shutil.move(source+str(sourceFiles[eachIndex]),
dest+str(sourceFiles[eachIndex]))
    else:
        print("No file moved. Source empty!")

def transferAllClassBetweenFolders(source, dest, splitRate):
    for label in classLabels:
        transferBetweenFolders(datasetFolderName+'/'+source+'/'+label+ '/',
                                datasetFolderName+'/'+dest+'/'+label+ '/',
                                splitRate)
'''

```

ภาพที่ 4.8 ชุดคำสั่งการส่งข้อมูลภาพเข้าไปใน Folder

- training_datagen สำหรับอ่านไฟล์ภาพ แล้วทำ Scaling และทำ Image Augmentation เพื่อขยายขนาด Dataset และเพิ่มความหลากหลายของภาพ ด้วยการบิดภาพ (Shear) แบบสุ่มไม่เกิน 20 องศา ขยายภาพ (Zoom) แบบสุ่มไม่เกิน 20% และพลิกภาพแนวนอน (Horizontal Flip) ซ้าย-ขวา แบบสุ่ม
- val_test_datagen สำหรับอ่านไฟล์ภาพ แล้วทำ Scaling โดยไม่มีการทำ Image Augmentation เนื่องจากเป็นชุดภาพสำหรับการ Validate และ Test
- test_DIR การดึงข้อมูลภาพจากฐานข้อมูลที่กำหนด ดังแสดง Code ด้านล่าง

```

TRAINING_DIR = 'C:/chest 80_10_10%/train_80%/
training_datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
VALIDATION_DIR = "C:/chest 80_10_10%/Validation_10%/
validation_datagen = ImageDataGenerator(rescale = 1./255)
train_generator = training_datagen.flow_from_directory(
    TRAINING_DIR,
    target_size=(width,width),
    class_mode='categorical')
validation_generator = validation_datagen.flow_from_directory(
    VALIDATION_DIR,
    target_size=(width,width),
    class_mode='categorical')
test_DIR = "C:/chest 80_10_10%/Test_10%/
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory(
    test_DIR,
    target_size=(width,width),
    class_mode='categorical',
    shuffle=False)

```

ภาพที่ 4.9 ชุดคำสั่งการอ่านไฟล์ภาพ การดึงข้อมูลภาพและการปรับภาพในรูปแบบต่าง ๆ

```

from keras. applications import ResNet50

Inception_base_model = Inceptionv3(input_shape= (180,180,3),
include_top=False, weights='imagenet')

```

ภาพที่ 4.10 ชุดคำสั่งการดึงโมเดลจาก keras applications และปรับ Input ที่ 180,180,3

```
Inception_base_model. summary ()
```

ภาพที่ 4.11 คำสั่งดึงโครงข่ายโมเดลมาใช้ในโปรแกรม เช่น โมเดล Inception

```

import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense,
Activation, BatchNormalization
from keras.layers import Dense, GlobalAveragePooling2D
resnet_model = tf.keras.Sequential([
    resnet_base_model,
    GlobalAveragePooling2D(),
    Dense(512, activation="relu"),
    BatchNormalization(),
    Dropout(0.6),
    Dense(128, activation="relu"),
    BatchNormalization(),
    Dropout(0.4),
    Dense(64,activation="relu"),
    BatchNormalization(),
    Dropout(0.3),
    Dense(3,activation="softmax")
])
opt = tf.keras.optimizers.Adam(learning_rate=0.0001)

```

```

METRICS = [
    'acc',
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall')
]
resnet_model.compile(optimizer=opt,loss='categorical_crossentropy',metrics=METRICS)

```

ภาพที่ 4.12 ชุดคำสั่งการดึงอัลกอริทึมเข้ามาปรับใช้กับโมเดลโครงข่ายประสาทเทียมที่มีอยู่

```

history = Inception_model.fit_generator(train_generator, epochs=50, validation_data =
validation_generator, verbose = 1)

Inception_model.save("rps.h5")

Epoch 45/50
150/150 [=====] - 3607s 24s/step - loss: 0.1502 - acc: 0.9492 - precision: 0.9520 - recall: 0.9452 - v
al_loss: 0.1073 - val_acc: 0.9583 - val_precision: 0.9583 - val_recall: 0.9567
Epoch 46/50
150/150 [=====] - 5845s 39s/step - loss: 0.1366 - acc: 0.9565 - precision: 0.9592 - recall: 0.9548 - v
al_loss: 0.0705 - val_acc: 0.9700 - val_precision: 0.9716 - val_recall: 0.9683
Epoch 47/50
150/150 [=====] - 2393s 16s/step - loss: 0.1316 - acc: 0.9583 - precision: 0.9608 - recall: 0.9556 - v
al_loss: 0.0484 - val_acc: 0.9900 - val_precision: 0.9900 - val_recall: 0.9883
Epoch 48/50
150/150 [=====] - 5467s 36s/step - loss: 0.1192 - acc: 0.9625 - precision: 0.9648 - recall: 0.9606 - v
al_loss: 0.0721 - val_acc: 0.9783 - val_precision: 0.9799 - val_recall: 0.9767
Epoch 49/50
150/150 [=====] - 2557s 17s/step - loss: 0.1233 - acc: 0.9625 - precision: 0.9651 - recall: 0.9613 - v
al_loss: 0.0953 - val_acc: 0.9567 - val_precision: 0.9615 - val_recall: 0.9567
Epoch 50/50
150/150 [=====] - 1251s 8s/step - loss: 0.1185 - acc: 0.9629 - precision: 0.9646 - recall: 0.9596 - va
l_loss: 0.0655 - val_acc: 0.9750 - val_precision: 0.9750 - val_recall: 0.9733

```

ภาพที่ 4.13 ชุดคำสั่งการสอนโมเดลตามที่กำหนดการฝึกสอนจำนวน 50 epochs

```
def my_metrics (y_trued, y_pred):
    accuracy=accuracy_score(y_trued, y_pred)
    precision=precision_score(y_trued, y_pred,average='weighted')
    f1Score=f1_score(y_trued, y_pred, average='weighted')
    print("Accuracy : {}".format(accuracy))
    print("Precision : {}".format(precision))
    print("f1Score : {}".format(f1Score))
    cm=confusion_matrix(y_trued, y_pred)
    print(cm)
    return accuracy, precision, f1Score

print("=====TEST RESULTS=====")

predictions = resnet_model.predict(test_generator, verbose=1)
yPredictions = np.argmax(predictions, axis=1)
true_classes = test_generator.classes

testAcc,testPrec, testFScore = my_metrics(true_classes, yPredictions)
resnet_model.save(MODEL_FILENAME)
```

ภาพที่ 4.14 แสดงการทำนายผล Testing ของค่าความถูกต้อง แม่นยำและ F1-Score

```

import matplotlib.pyplot as plt
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(acc))
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend(loc=0)
plt.figure()
plt.show()

```

ภาพที่ 4.15 ชุดคำสั่งสร้างกราฟแสดงความสัมพันธ์ระหว่าง accuracy กับ จำนวน epochs

```

Loss = history.history['loss']
val_loss = history.history['val_loss']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
plt.plot(epochs, loss, 'r', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend(loc=1)
plt.figure()
plt.show()

```

ภาพที่ 4.16 ชุดคำสั่งสร้างกราฟแสดงความสัมพันธ์ระหว่าง loss กับ จำนวน epochs

```

def cm_plot_test(cm_test, labels):
    x = labels

```

```

y = labels
z_text = [[str(y) for y in x] for x in cm_test]
fig = ff.create_annotated_heatmap(cm, x=x, y=y, annotation_text=z_text,
colorscale='oranges')
fig.update_layout(title_text='Confusion Matrix')
fig.add_annotation(dict(font=dict(color="black", size=13),
                        x=0.5,
                        y=-0.15,
                        showarrow=False,
                        text="Predicted Value",
                        xref="paper",
                        yref="paper"
                        ))
fig.add_annotation(dict(font=dict(color="black", size=13),
                        x=-0.20,
                        y=0.5,
                        showarrow=False,
                        text="Real Value",
                        textangle=-90,
                        xref="paper",
                        yref="paper"
                        ))
fig.update_layout(margin=dict(t=50, l=200))
fig['layout']['yaxis']['autorange'] = "reversed"
fig['data'][0]['showscale'] = True
fig.show()

```

ภาพที่ 4.17 นิยาม cm_plot Function

```
cm_plot_test(cm_test, classLabels)
```

ภาพที่ 4.18 คำสั่งแสดง Plot Confusion Matrix

```
C_test = Counter(y_true_test)
```

```
C_test
```

ภาพที่ 4.19 ทดลองนับจำนวนภาพของ Test Dataset ในแต่ละ Class

```
Cm_test = confusion_matrix(y_true_test, predicted_classes_test)
```

```
Cm_test
```

ภาพที่ 4.20 แสดงค่านวนค่า Confusion Matrix

```

hovertemplate = 'False Positive Rate = %{:4f}<br>True Positive
Rate=%{:4f}<br>Threshold=%{:text:.4f}'
fig = go.Figure()
fig.add_shape(type='line', line = dict(dash='dash'), x0=0, x1=1, y0=0, y1=1)
for i in range(predicted_score.shape[1]):
    y_real = y_val[:, i]
    y_score = predicted_score[:, i]
    fpr, tpr, threshold = roc_curve(y_real, y_score)
    auc_score = auc(fpr, tpr)
    name = f"{classLabels[i]}, AUC={auc_score:.4f}"
    fig.add_trace(go.Scatter(x=fpr, y=tpr, name=name, mode='lines', text=threshold,
hovertemplate=hovertemplate))
fig.update_layout(
    title = 'ROC Curve and AUC',
    xaxis_title = 'False Positive Rate',
    yaxis_title = 'True Positive Rate',)
fig.show()

```

ภาพที่ 4.21 คำสั่งแสดง Confusion matrix ของ AUC กราฟสัมพันธ์ระหว่าง (TP)กับ (FP)

4.9 ผลการทำนายที่ได้จากการทดลอง

จากการฝึกสอนแบบจำลองโดยใช้เทคนิค (CNN: Deep-learning) เพื่อจำแนกภาพเอกซเรย์ปอดสำหรับวินิจฉัยโรคโควิด COVID19 จากงานวิจัยนี้ได้ใช้ 6 โมเดล คือ Inception V3, ResNet50, VGG16, VGG19, Mobilenet V2, InceptionResNet V2 ของ CNN โดยกำหนดให้คอมพิวเตอร์เรียนรู้ในขั้นตอนการฝึกสอนจำนวน 50 epochs ประสิทธิภาพการฝึกสอน (Training) ในแต่ละเทคนิคของโมเดลการตรวจสอบผล (Validation) และการทดสอบ (Testing) แต่ละโมเดล โดยแสดงผลการทดลองที่ได้ระหว่างการฝึกสอนโมเดลทั้งสองรุ่น ดังตารางตามลำดับ

ตารางที่ 4.6 แสดงชนิดของโมเดลและค่า Training จากการศึกษา รันที่ 1

Architecture	Training Result 1				
	loss	accuracy	precision	recall	F1-score
Inception V3	0.0291	0.9921	0.9921	0.9919	0.9919
ResNet 50	0.0863	0.9714	0.9723	0.9707	0.9715
VGG16	0.1112	0.9660	0.9680	0.9638	0.9630
VGG19	0.2593	0.9164	0.9213	0.9110	0.9161
Mobilenet V2	0.2462	0.9080	0.8901	0.8910	0.8905
InceptionResnet V2	0.0560	0.9833	0.9833	0.9831	0.9832

ตารางที่ 4.7 แสดงชนิดของโมเดลและค่า Training จากการศึกษา รันที่ 2

Architecture	Training Result 2				
	loss	accuracy	precision	recall	F1-score
Inception V3	0.0820	0.9764	0.9754	0.9737	0.9745
ResNet 50	0.1185	0.9629	0.9646	0.9596	0.9620
VGG16	0.1432	0.9517	0.9528	0.9496	0.9512
VGG19	0.2593	0.9164	0.9213	0.9110	0.9161
Mobilenet V2	0.2462	0.9080	0.8901	0.8910	0.8905
InceptionResnet V2	0.0503	0.9835	0.9837	0.9835	0.9836

ตารางที่ 4.8 แสดงชนิดของโมเดลและค่า Validation จากการฝึกสอนรุ่นที่ 1

Architecture	Validation Result 1				
	Val_loss	Val_acc	Val_Precision	Val_recall	Val_F1-score
Inception V3	0.3006	0.9078	0.9094	0.9078	0.9086
ResNet 50	0.4871	0.8299	0.8340	0.8249	0.8295
VGG16	0.4440	0.8452	0.8465	0.8443	0.8440
VGG19	0.5159	0.8249	0.8275	0.8240	0.8220
Mobilenet V2	1.1916	0.6590	0.6492	0.6511	0.6502
InceptionResnet V2	1.4389	0.7346	0.7359	0.7346	0.7353

ตารางที่ 4.9 แสดงชนิดของโมเดลและค่า Validation จากการฝึกสอนรุ่นที่ 2

Architecture	Validation Result 2				
	Val_loss	Val_acc	Val_Precision	Val_recall	Val_F1-score
Inception V3	0.0560	0.9817	0.9817	0.9817	0.9817
ResNet 50	0.0655	0.9750	0.9750	0.9733	0.9742
VGG16	0.1824	0.9233	0.9247	0.9217	0.9232
VGG19	0.5159	0.8249	0.8275	0.8240	0.8258
Mobilenet V2	1.1916	0.6590	0.6492	0.6511	0.6502
InceptionResnet V2	0.0164	0.9967	0.9967	0.9967	0.9967

ตารางที่ 4.10 แสดงชนิดของโมเดลและค่า Testing Result จากการสอนรุ่นที่ 1

Architecture	Testing Result 1		
	Test_acc	Test_Precision	Test_F1-score
Inception V3	0.9095	0.9155	0.9089
ResNet 50	0.8003	0.8601	0.7884
VGG16	0.8626	0.8658	0.8635
VGG19	0.8217	0.8265	0.8237
Mobilenet V2	0.6328	0.6490	0.5525
InceptionResnet V2	0.7749	0.8549	0.7431

ตารางที่ 4.11 แสดงชนิดของโมเดลและค่า Testing Result จากการสอนรุ่นที่ 2

Architecture	Testing Result 2		
	Test_acc	Test_Precision	Test_F1-score
Inception V3	0.9516	0.9521	0.9516
ResNet 50	0.9333	0.9428	0.9326
VGG16	0.8850	0.8843	0.8840
VGG19	0.8217	0.8265	0.8237
Mobilenet V2	0.6328	0.6490	0.5525
InceptionResnet V2	0.9550	0.9558	0.9594

ตารางที่ 4.12 แสดงชนิดของโมเดลค่า Testing Result และ ค่า AUC จากการสอนโมเดลรุ่นที่ 1

Architecture	Testing Result 1		
	AUC (Area under the curve)		
	Normal	Pneumonia	Pneumonia Covid 19
Inception V3	0.9523	0.9357	0.9923
ResNet 50	0.9516	0.9446	0.9999
VGG16	0.9487	0.9308	0.9983
VGG19	0.9288	0.8778	0.9787
Mobilenet V2	0.8004	0.8632	0.9767
InceptionResnet V2	0.9376	0.9030	0.9998

ตารางที่ 4.13 แสดงชนิดของโมเดลค่า Testing Result และ ค่า AUC จากการสอนโมเดลรุ่นที่ 2

Architecture	Testing Result 2		
	AUC (Area under the curve)		
	Normal	Pneumonia	Pneumonia Covid 19
Inception V3	0.9979	0.9979	1.0000
ResNet 50	0.9992	0.9989	1.0000
VGG16	0.9832	0.9848	0.9997
VGG19	0.9963	0.9956	1.0000
Mobilenet V2	0.9326	0.9426	0.9951
InceptionResnet V2	0.9997	0.9999	1.0000

ตารางที่ 4.14 แสดงเปรียบเทียบประสิทธิภาพ Transfer learning 6 models/ รุ่นที่ 1

Models	Accuracy	Precision	Epochs
Inception V3	0.9078	0.9094	50
ResNet 50	0.8299	0.8340	
VGG16	0.8452	0.8465	
VGG19	0.8249	0.8275	
Mobilenet V2	0.6590	0.6492	
InceptionResNet V2	0.9078	0.9094	

ตารางที่ 4.15 แสดงเปรียบเทียบประสิทธิภาพ Transfer learning 6 models/ รุ่นที่ 2

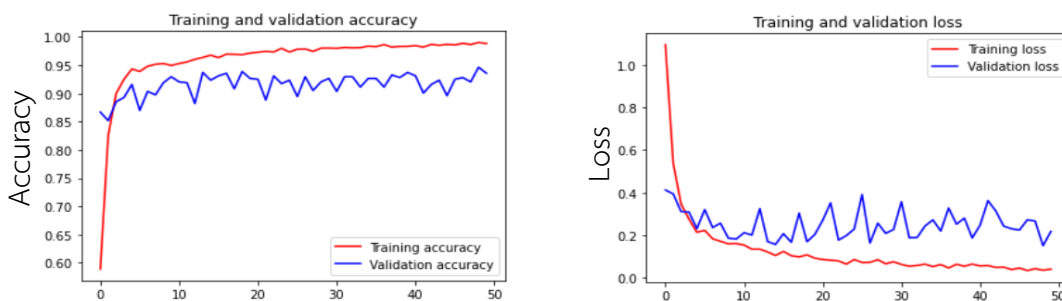
Models	Accuracy	Precision	Epochs
Inception V3	0.9516	0.9521	50
ResNet 50	0.9333	0.9528	
VGG16	0.8850	0.8843	
VGG19	0.8217	0.8265	
Mobilenet V2	0.6328	0.6490	
InceptionResNet V2	0.9550	0.9558	

4.10 การวิเคราะห์ความสัมพันธ์จากข้อมูลหลังทำการฝึกสอนโมเดลรุ่นที่ 1

จากการทดลองพบว่า

Inception V3

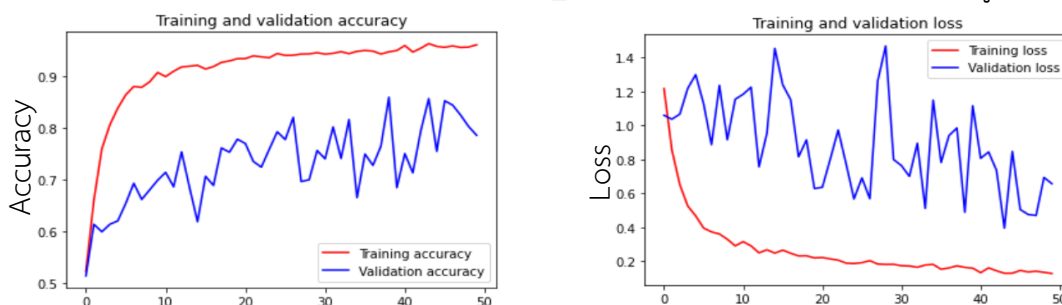
Accuracy Val (ความแม่นยำ = 0.9078), Loss_val (ค่าความผิดพลาด = 0.3006) ดังรูปที่ 4.23



รูปที่ 4.22 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

VGG19

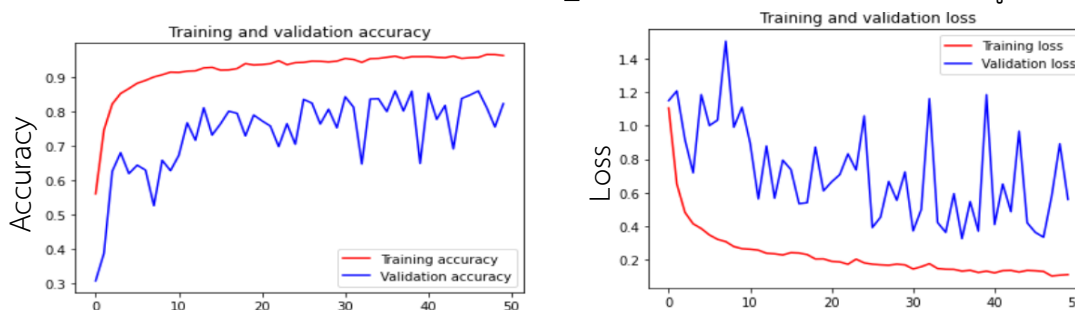
Accuracy Val (ความแม่นยำ = 0.8249), Loss_val (ค่าความผิดพลาด = 0.5159) ดังรูปที่ 4.24



รูปที่ 4.23 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

VGG16

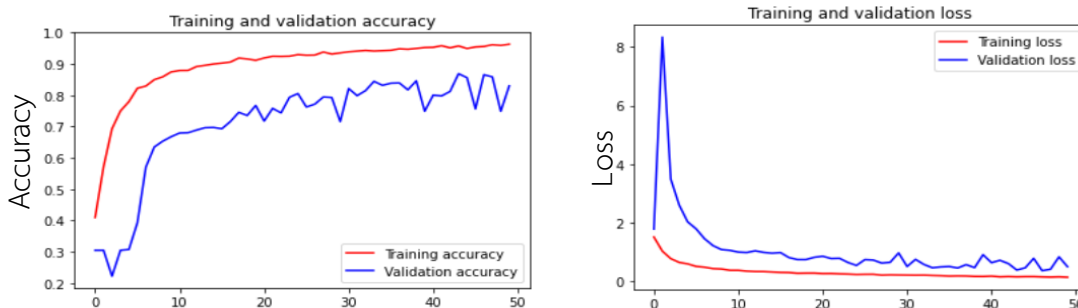
Accuracy Val (ความแม่นยำ = 0.8452), Loss_val (ค่าความผิดพลาด = 0.4440) ดังรูปที่ 4.25



รูปที่ 4.24 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

ResNet 50

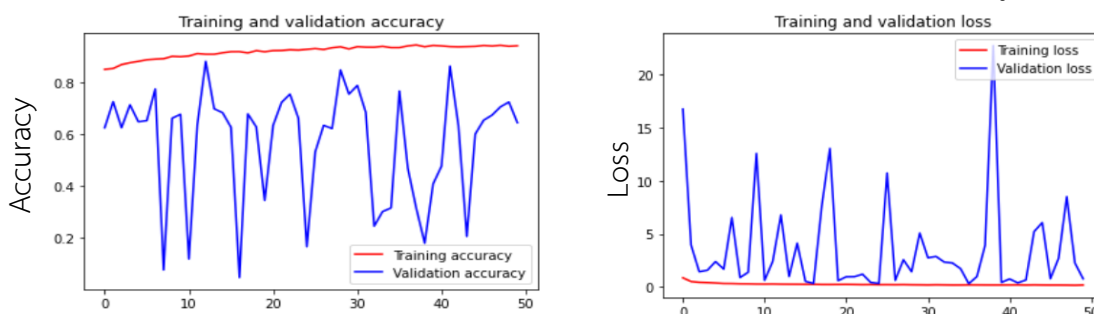
Accuracy_val (ความแม่นยำ = 0.8299, Loss_val (ค่าความผิดพลาด = 0.4871) ดังรูปที่ 4.26



รูปที่ 4.25 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

MobileNet V2

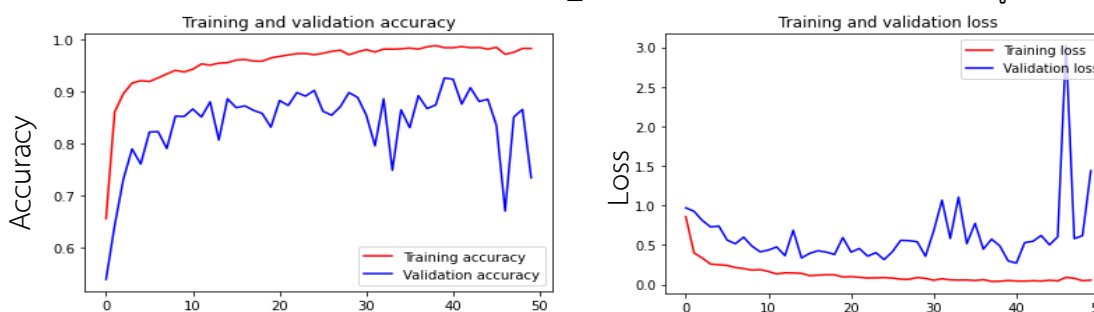
Accuracy Val (ความแม่นยำ = 0.6590), Loss_val (ค่าความผิดพลาด = 1.1916) ดังรูปที่ 4.27



รูปที่ 4.26 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

InceptionResNet V2

Accuracy Val (ความแม่นยำ = 0.7346), Loss_val (ค่าความผิดพลาด = 1.4389) ดังรูปที่ 4.28



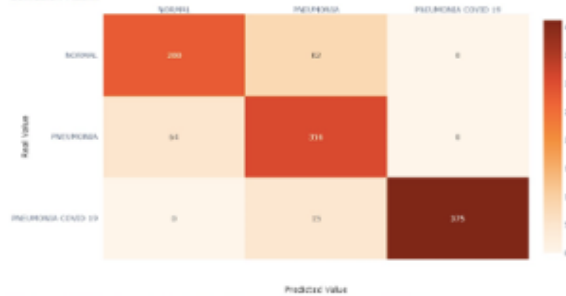
รูปที่ 4.27 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs



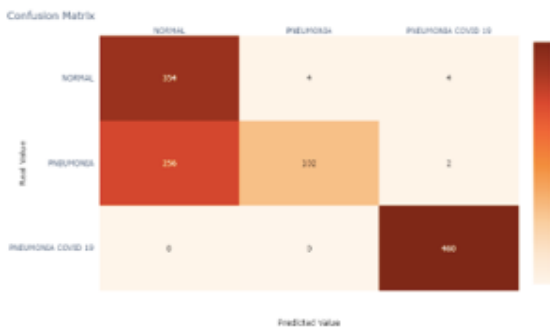
(a) Confusion matrix of Inception V3



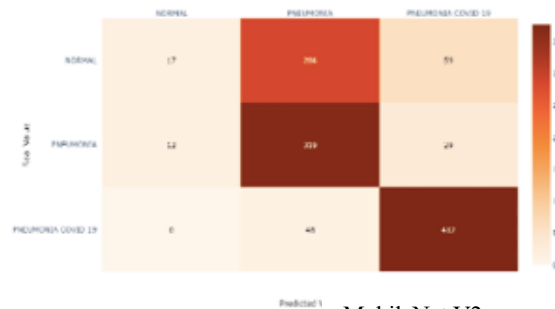
(b) Confusion matrix of VGG 19



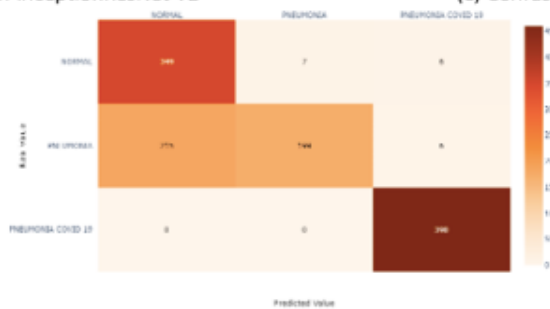
(c) Confusion matrix of VGG 16



(d) Confusion matrix of InceptionResNet V2



(e) Confusion matrix MobileNet V2



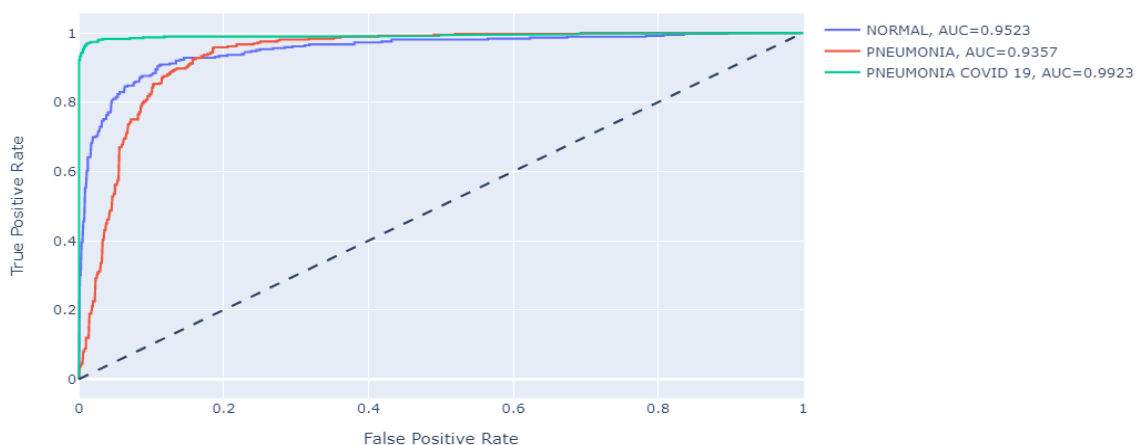
(f) Confusion matrix of Resnet 50

รูปที่ 4.28 เมทริกซ์ Validation แสดงความสัมพันธ์ระหว่างประเภทของรอยโรคการทำนายผลของทั้ง 6

โมเดลสำหรับการวิเคราะห์ด้วยชุดข้อมูลการตรวจสอบความถูกต้อง: (a) Inceptionv3, (b) VGG19, (c) VGG16, (d) InceptionResnet V2, (e) MoblieNet V2 และ (f) Resnet 50 โดยอ้างอิงถึงภาพเอกซเรย์ของผู้ป่วย COVID-19 ปกติ ตามลำดับ

-กราฟ ROC Curve และ AUC แสดงความสัมพันธ์ของประสิทธิภาพการทำงานของแต่ละโมเดลที่ผลลัพธ์ออกมาแตกต่างในพารามิเตอร์เดียวกัน อัลกอริทึมเดียวกัน มีค่าการแสดงผลกำหนดตามสี โดยสีเขียวคือ ข้อมูลภาพปอดอักเสบโควิด COVID-19, สีแดงคือ ข้อมูลภาพปอดอักเสบทั่วไป และสีน้ำเงินแสดงข้อมูลภาพปอดปกติ

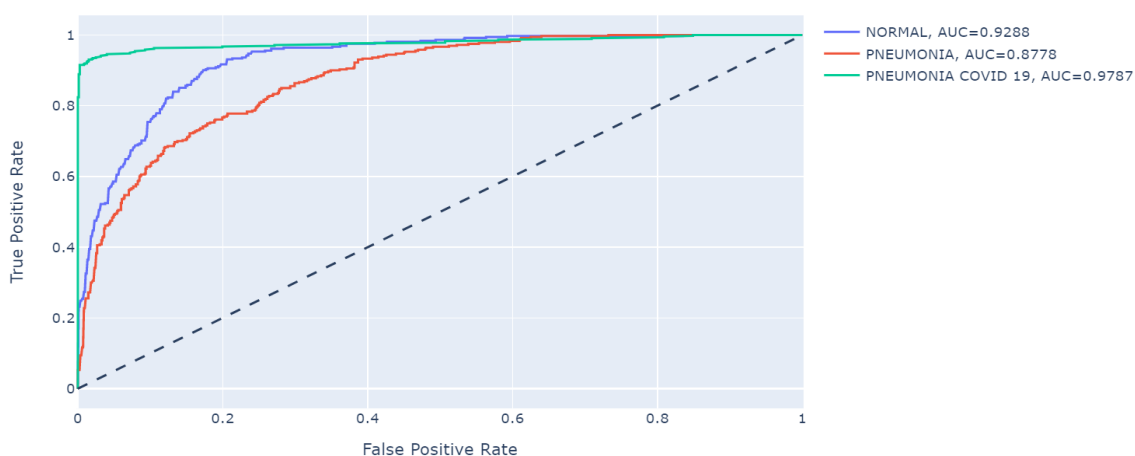
ROC Curve and AUC



รูปที่ 4.29 Confusion matrix ของ Inception V3 โมเดล

(Normal, AUC = 0.9523, Pneumonia, AUC = 0.9357, Pneumonia, AUC = 0.9923)

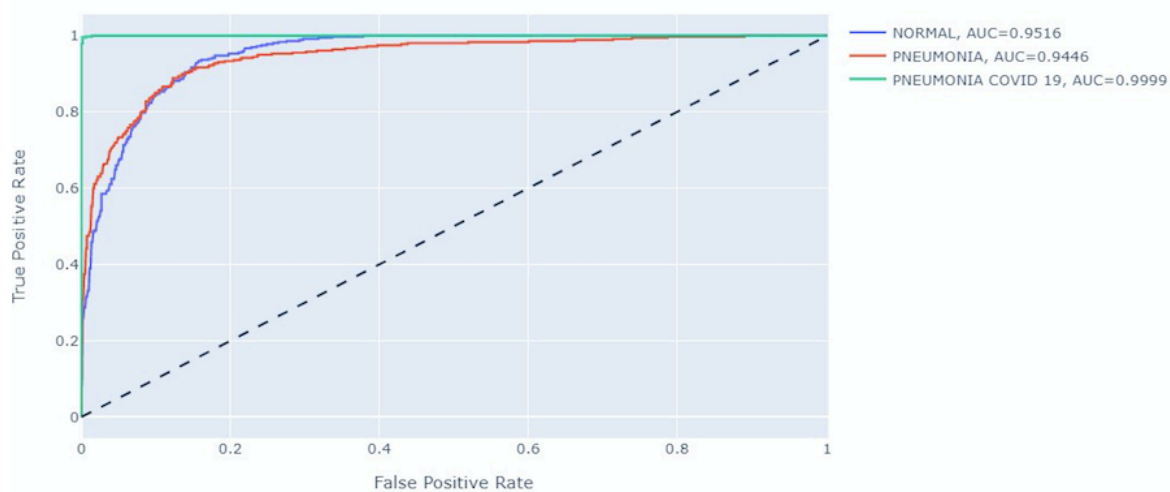
ROC Curve and AUC



รูปที่ 4.30 Confusion matrix ของ Vgg19 โมเดล

(Normal, AUC = 0.9288, Pneumonia, AUC = 0.8778, Pneumonia, AUC = 0.9787)

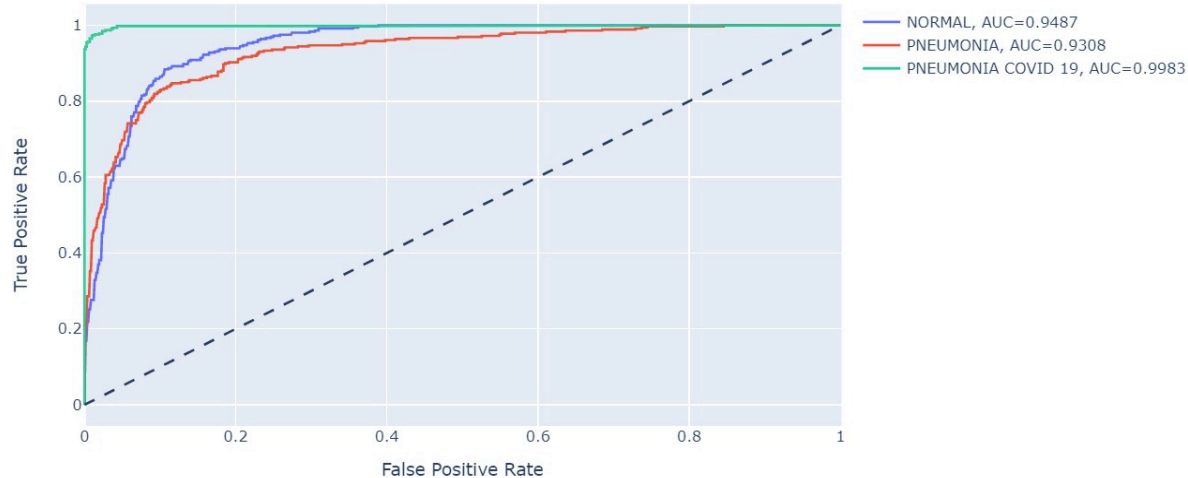
ROC Curve and AUC



รูปที่ 4.31 Confusion matrix ของ ResNet 50 โมเดล

(Normal, AUC = 0.9516, Pneumonia, AUC = 0.9446, Pneumonia, AUC = 0.9999)

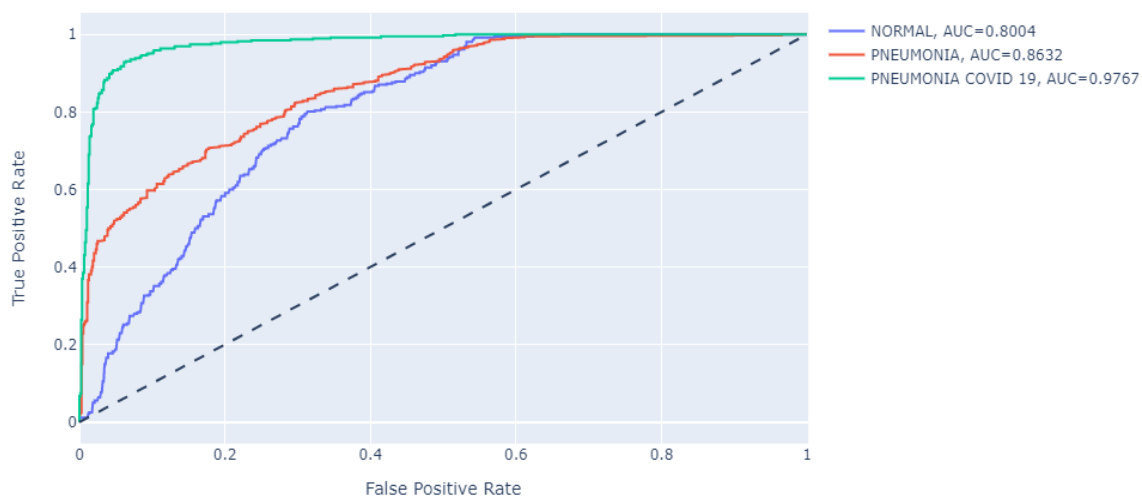
ROC Curve and AUC



รูปที่ 4.32 Confusion matrix ของ Vgg16 โมเดล

(Normal, AUC = 0.9487, Pneumonia, AUC = 0.9308, Pneumonia, AUC = 0.9983)

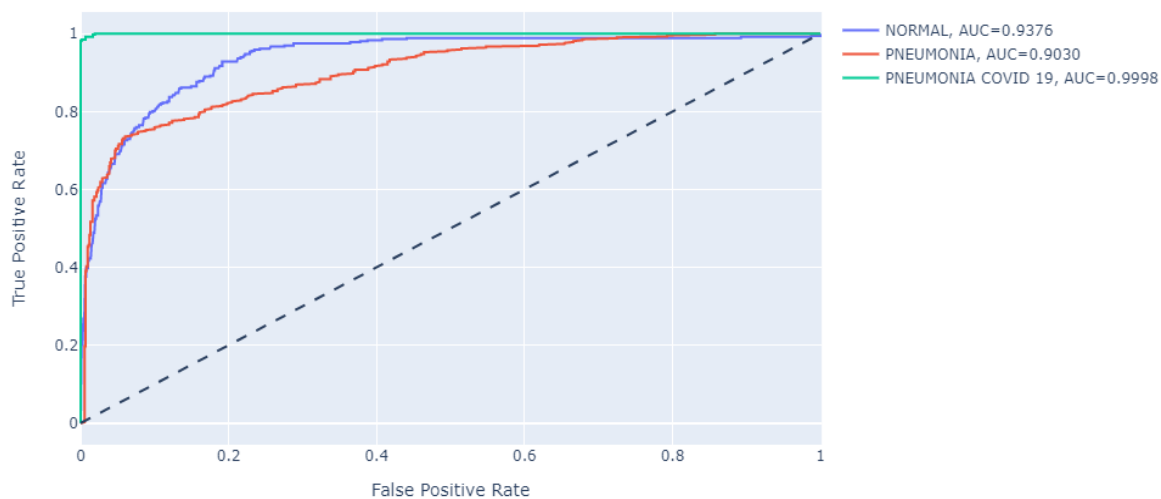
ROC Curve and AUC



รูปที่ 4.33 Confusion matrix ของ MobileNet V2 โมเดล

(Normal, AUC = 0.8004, Pneumonia, AUC = 0.8632, Pneumonia, AUC = 0.9767)

ROC Curve and AUC



รูปที่ 4.34 Confusion matrix ของ InceptionResnet V2 โมเดล

(Normal, AUC = 0.9376, Pneumonia, AUC = 0.9030, Pneumonia, AUC = 0.9998)

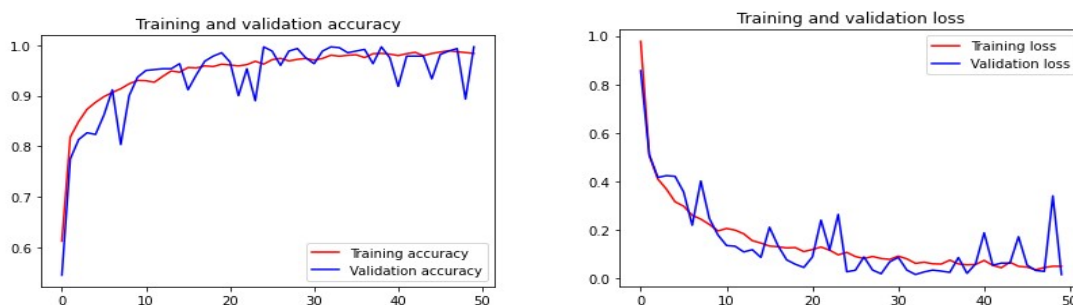
-จะเห็นว่าค่าของ AUC ของแต่ละโมเดลที่แสดงค่าการทำนายของข้อมูลภาพปอดอักเสบโควิด
ไม่ได้แตกต่างกันมากโดยค่าเพิ่มขึ้นเข้าใกล้หนึ่งอย่างมีนัยสำคัญ ดังตารางที่ 4.12

4.11 การวิเคราะห์ความสัมพันธ์จากข้อมูลหลังทำการฝึกสอนโมเดลรุ่นที่ 2

จากการทดลองพบว่าหลังจากมีการปรับและจัดเรียงจำนวนข้อมูลภาพใหม่ด้วยเทคนิคต่าง ๆ แล้วนำกลับไปเข้ากระบวนการฝึกสอนโมเดลอีกครั้ง ผลลัพธ์ที่ได้เป็นที่น่าพอใจพอสมควรด้วยค่าความถูกต้องและค่าความแม่นยำของโมเดล InceptionResNet V2 ทำคะแนนได้สูงมีค่าความถูกต้อง 99.67% ค่าความแม่นยำ 99.67% และโมเดลที่ทำคะแนนได้น้อยที่สุดเช่นเคยยังคงเป็น MobileNet V2 ค่าความถูกต้อง 65.90% ค่าความแม่นยำ 64.92% ดังรูปล่าง

InceptionResNet V2

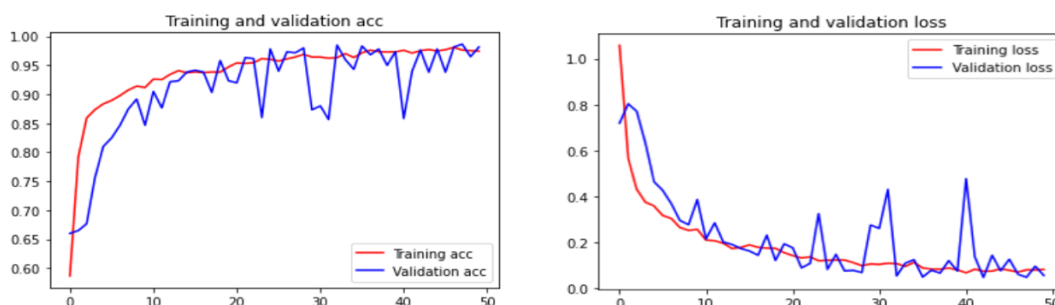
Accuracy Val (ความแม่นยำ = 0.9967, Loss_val (ค่าความผิดพลาด = 0.0164) ดังรูปที่ 4.36



รูปที่ 4.35 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

Inception V3

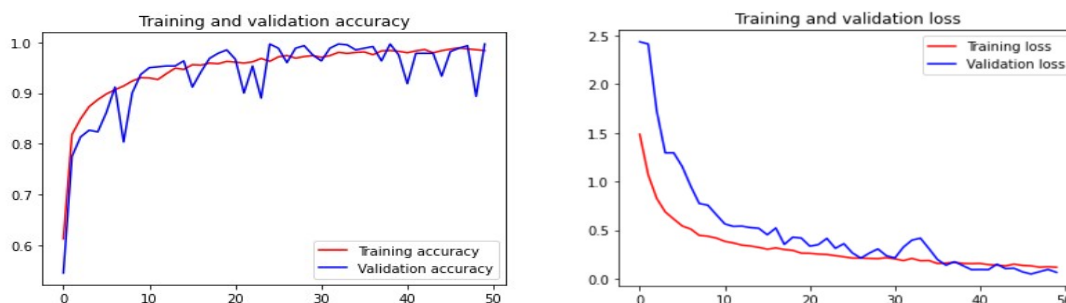
Accuracy Val (ความแม่นยำ = 0.9817), Loss_val (ค่าความผิดพลาด = 0.0560) ดังรูปที่ 4.37



รูปที่ 4.36 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

ResNet 50

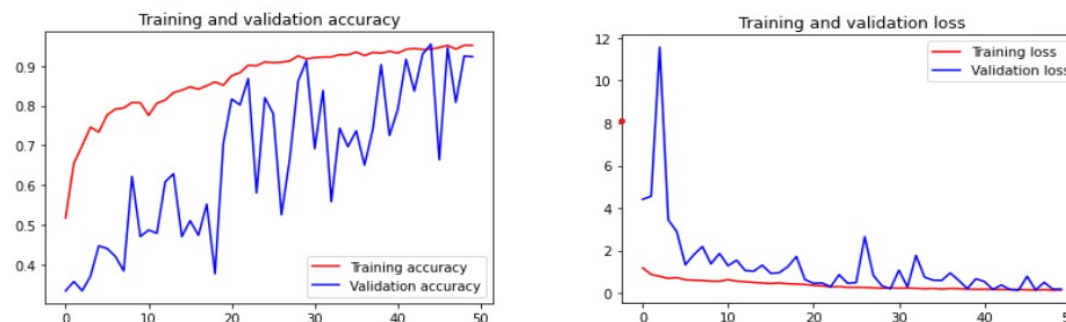
Accuracy Val (ความแม่นยำ = 0.9750), Loss_val (ค่าความผิดพลาด = 0.0655) ดังรูปที่ 4.38



รูปที่ 4.37 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

VGG16

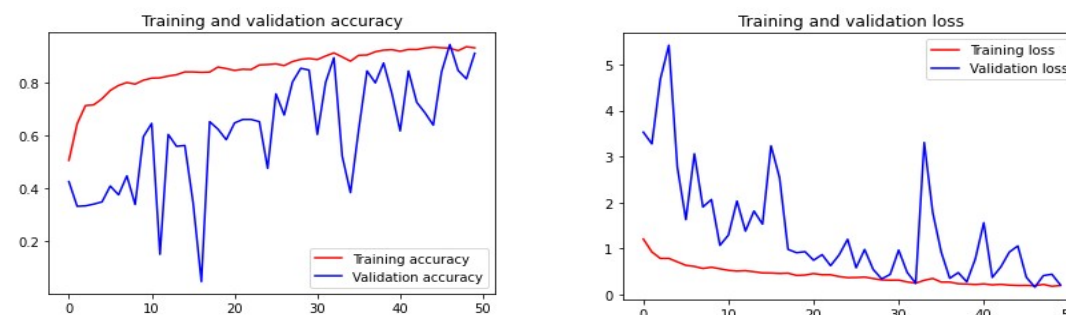
Accuracy Val (ความแม่นยำ = 0.9233), Loss_val (ค่าความผิดพลาด = 0.1824) ดังรูปที่ 4.39



รูปที่ 4.38 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

VGG19

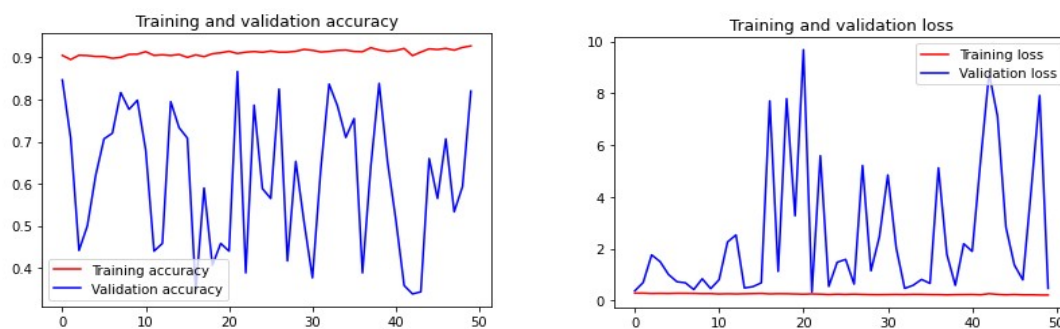
Accuracy Val (ความแม่นยำ = 0.8249), Loss_val (ค่าความผิดพลาด = 0.5159) ดังรูปที่ 4.40



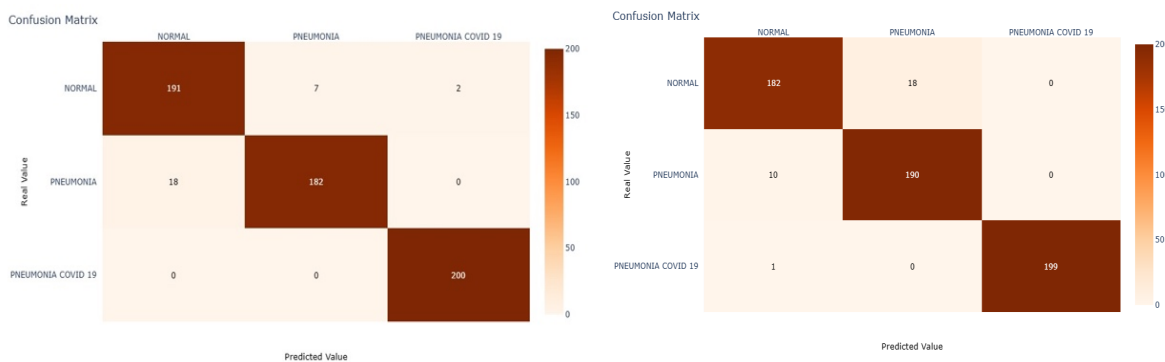
รูปที่ 4.39 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

MobileNet V2

Accuracy Val (ความแม่นยำ = 0.6590), Loss_val (ค่าความผิดพลาด = 1.1916) ดังรูปที่ 4.41

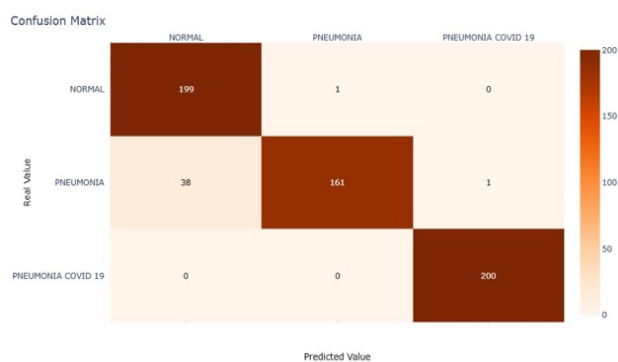


รูปที่ 4.40 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ จำนวน epochs

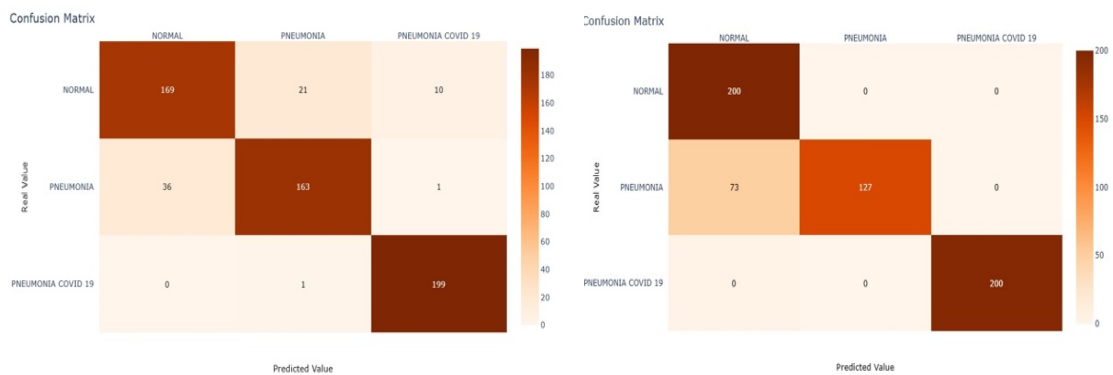


(a) Confusion matrix of InceptionResNet V2

(b) Confusion matrix of Inception V3

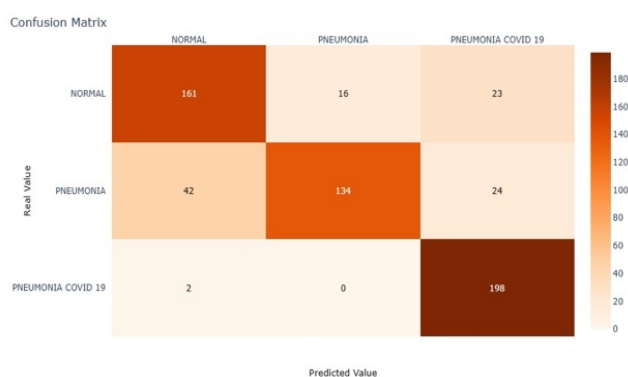


(c) Confusion matrix of ResNet 50



(d) Confusion matrix of VGG16

(e) Confusion matrix of VGG19



(f) Confusion matrix of MobileNet V2

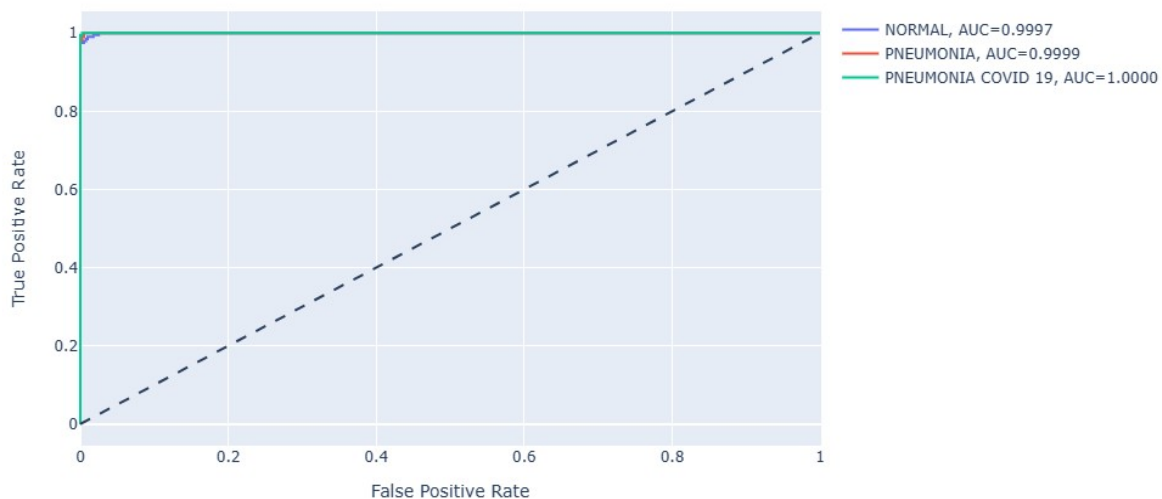
รูปที่ 4.41 เมตริกซ์ Validation แสดงความสัมพันธ์ระหว่างประเภทของรอยโรคการทำนายผลของทั้ง 6 โมเดลสำหรับการวิเคราะห์ด้วยชุดข้อมูลการตรวจสอบความถูกต้อง: (a) Inceptionv3, (b) VGG19, (c) VGG16, (d) InceptionResnet V2, (e) MoblieNet V2 และ (f) Resnet 50 โดยอ้างอิงถึงภาพเอกซเรย์ของผู้ป่วย COVID-19 ปกติ ตามลำดับ

- กราฟ ROC Curve และ AUC ของรุ่นที่สอง แสดงความสัมพันธ์ของประสิทธิภาพการทำงานของแต่ละโมเดลที่ผลลัพธ์ออกมาแตกต่างในพารามิเตอร์และอัลกอริทึมเดียวกัน มีค่าการแสดงผลกำหนดตามสี โดยสีเขียวแสดงข้อมูลภาพปอดอักเสบ COVID-19, สีแดงแสดงข้อมูลภาพปอดอักเสบทั่วไปและสีน้ำเงินแสดงข้อมูลภาพปอดปกติ ดังรูป 4.40-4.45

- จะเห็นว่าค่าของ AUC ของแต่ละโมเดลที่แสดงค่าการทำนายของข้อมูลภาพปอดอักเสบโควิดที่แสดงในเส้นสีเขียวไม่ได้แตกต่างกันมากโดยค่าเพิ่มขึ้นเข้าใกล้หนึ่งอย่างมีนัยสำคัญ รายละเอียดดังตารางที่ 4.13

สรุปได้ว่าสถาปัตยกรรมทั้ง 6 โมเดลที่ใช้มีประสิทธิภาพในการทำนายค่อนข้างดี โดยเฉพาะการทำนายภาพที่ตัดห้องแยกความแตกต่างระหว่างสีขาว ดำ มีค่าเพิ่มขึ้นเข้าใกล้หนึ่งอย่างมีนัยสำคัญ

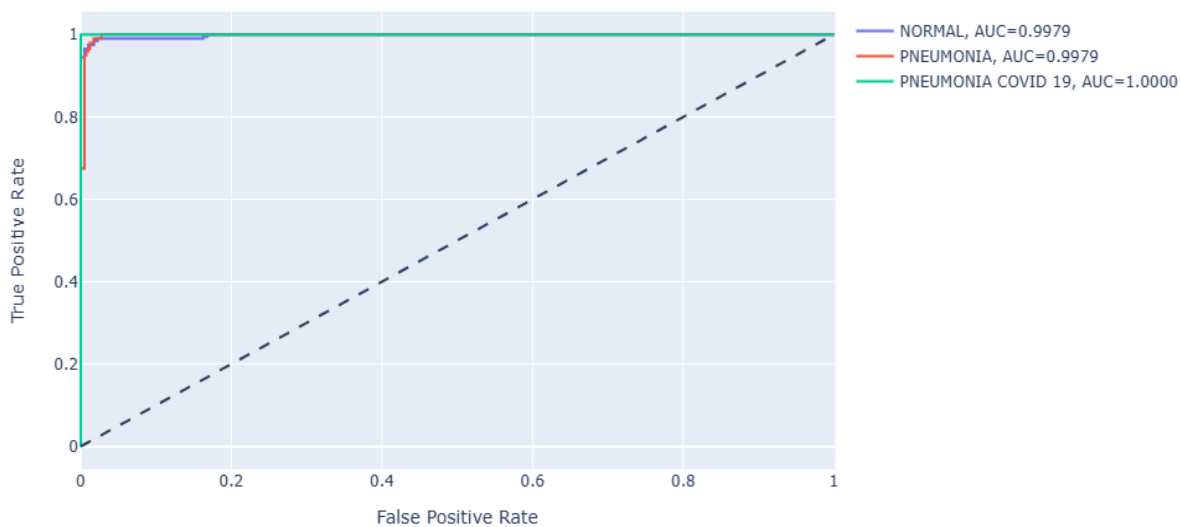
ROC Curve and AUC



รูปที่ 4.42 Confusion matrix ของ InceptionResnet V2 โมเดล

(Normal, AUC = 0.9997, Pneumonia, AUC = 0.9999, Pneumonia, AUC = 1.0000)

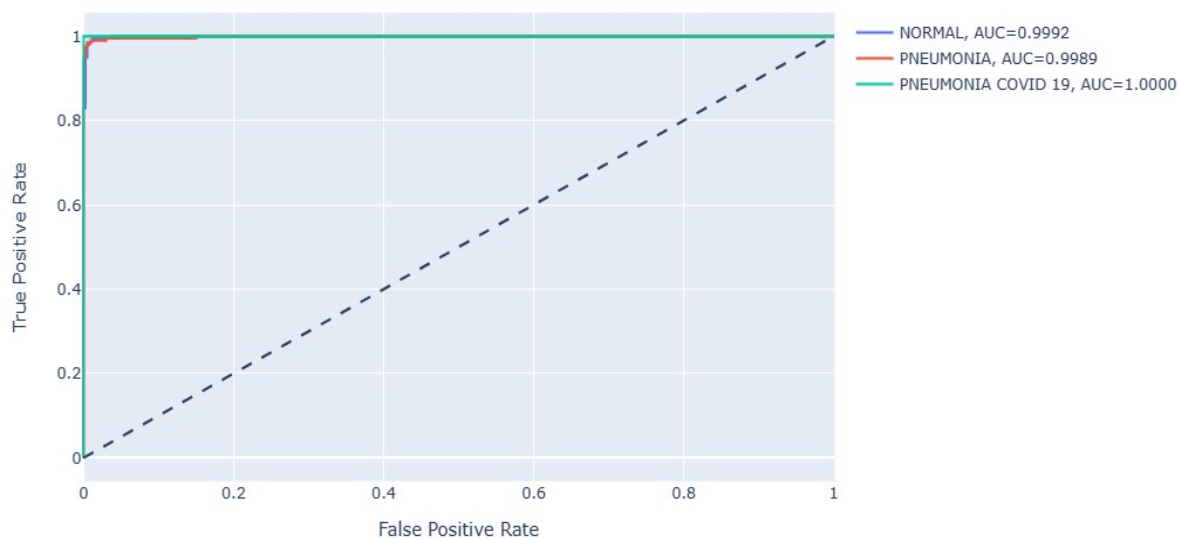
ROC Curve and AUC



รูปที่ 4.43 Confusion matrix ของ Inception V3 โมเดล

(Normal, AUC = 0.9979, Pneumonia, AUC = 0.9979, Pneumonia, AUC = 1.0000)

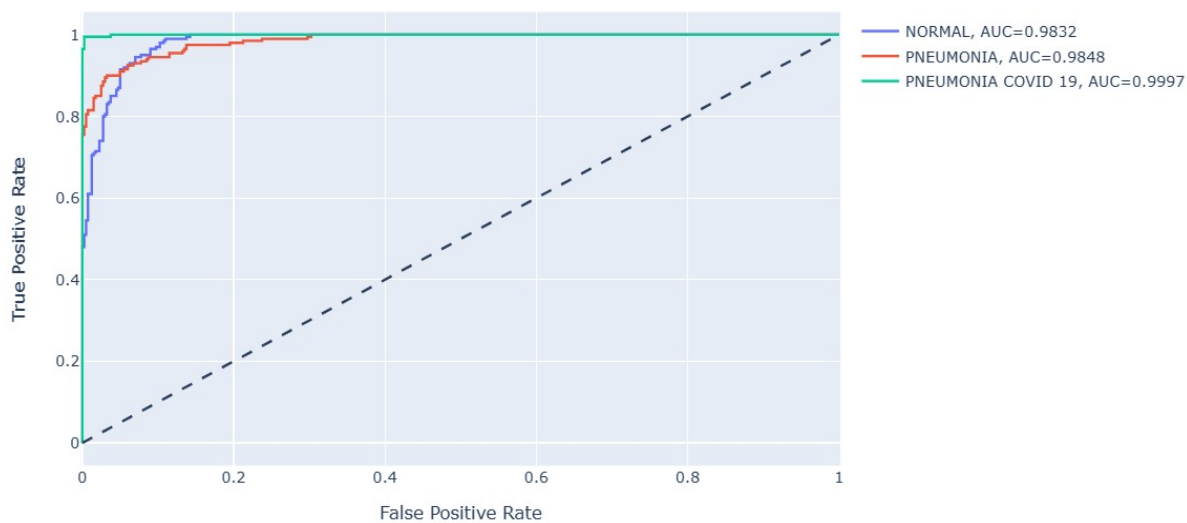
ROC Curve and AUC



รูปที่ 4.44 Confusion matrix ของ Resnet 50 โมเดล

(Normal, AUC = 0.9992, Pneumonia, AUC = 0.9989, Pneumonia, AUC = 1.0000)

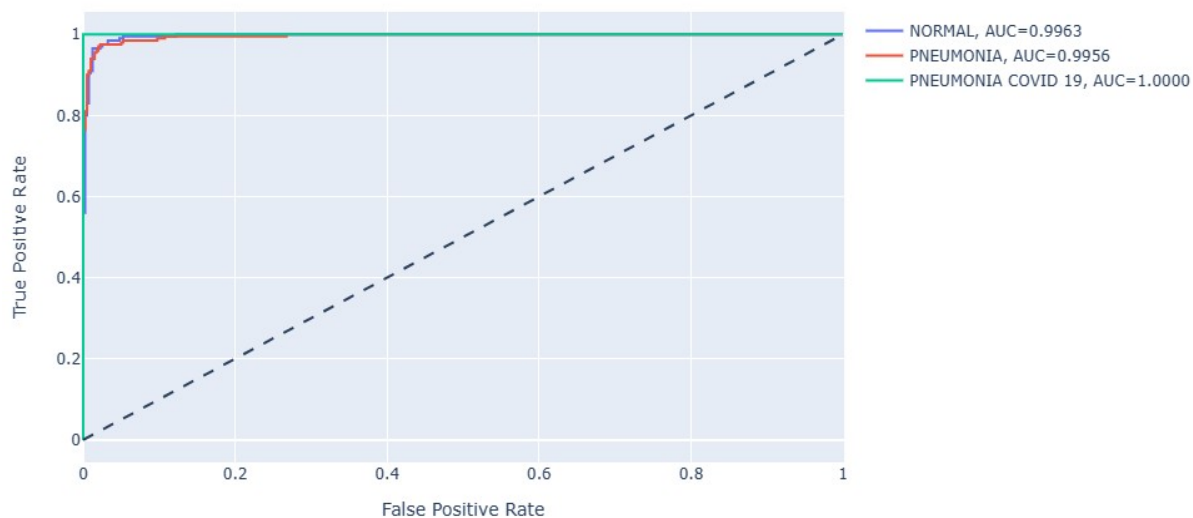
ROC Curve and AUC



รูปที่ 4.45 Confusion matrix ของ VGG16 โมเดล

(Normal, AUC = 0.9832, Pneumonia, AUC = 0.9848, Pneumonia, AUC = 0.9997)

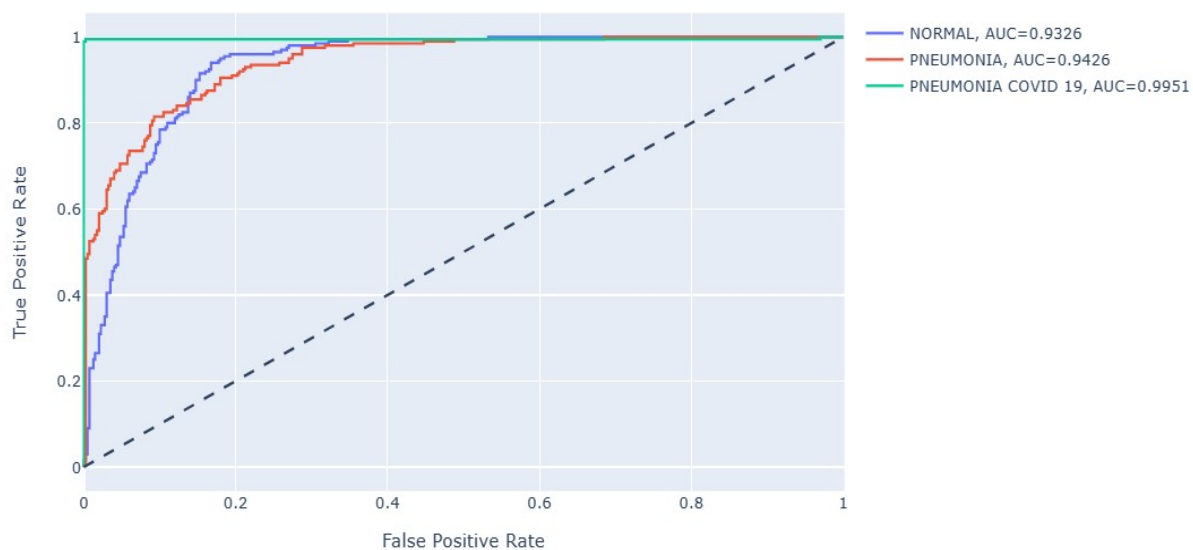
ROC Curve and AUC



รูปที่ 4.46 Confusion matrix ของ VGG19 โมเดล

(Normal, AUC = 0.9963, Pneumonia, AUC = 0.9956, Pneumonia, AUC = 1.0000)

ROC Curve and AUC



รูปที่ 4.47 Confusion matrix ของ MobileNet V2 โมเดล

(Normal, AUC = 0.9326, Pneumonia, AUC = 0.9426, Pneumonia, AUC = 0.9951)

4.12 การประเมินประสิทธิภาพบนฐานข้อมูล

จากงานวิจัยพบว่าการฝึกสอนโมเดลผลลัพธ์ที่ได้ของรุ่นที่ 1 โดยเทคนิคการแบ่งข้อมูลภาพ 60:20:20 โมเดล Inception V3 กับชุดข้อมูลสำหรับการตรวจสอบ มีค่าความแม่นยำ ค่าความถูกต้อง และค่าการเรียกคืน ได้คะแนนสูงที่สุดในทุกโมเดลคือ 0.9155, 0.9095 และ 0.9089 ตามลำดับ ในขณะที่ในรุ่นที่สองเราได้ปรับเปลี่ยนเทคนิคการแบ่งข้อมูลภาพ 80:10:10 การใช้เทคนิค Data augmentation เพื่อเพิ่มจำนวนภาพในบางคลาส เช่น การหมุน การตัดบางส่วนของภาพที่ไม่ต้องการออก(Cropping) ภาพหรือการปรับเพิ่มแสงร่วมกับเทคนิคการ Transfer พารามิเตอร์จากการฝึกสอนโมเดลในรุ่นแรกมารุ่นที่สองนอกจากจะช่วยลดเวลาในการฝึกสอนโมเดลแล้วยังทำให้ค่าความแม่นยำและถูกต้องเพิ่มขึ้นอย่างมีนัยสำคัญอีกด้วยและแบบจำลองที่ทำคะแนนได้ดีที่สุดในรุ่นที่สองคือ InceptionResnet V2 กับชุดข้อมูลสำหรับการตรวจสอบ(Data test) เมื่อนำมาคำนวณหาค่าความแม่นยำ(Precision) ค่าความถูกต้อง(Accuracy) และค่าการเรียกคืน(Recall) ได้คะแนนสูงถึง 0.9516, 0.9521 และ 0.9516 ตามลำดับ แสดงให้เห็นว่าเทคนิคที่ใช้เพิ่มขึ้นในรุ่นที่สองทั้งหมดรวมถึงโมเดลที่เลือกใช้ในงานวิจัยนี้เหมาะสมกับชุดข้อมูลที่เรามีอยู่ทำให้ได้ผลลัพธ์เป็นที่น่าพอใจ

บทที่ 5

บทสรุปและข้อเสนอแนะจากงานวิจัย

5.1 อภิปรายผลและสรุป

จากการทดลองในงานวิจัยเพื่อเปรียบเทียบประสิทธิภาพและผลลัพธ์ที่ได้จากโครงข่ายประสาทเทียม (CNN: Deep learning) กับภาพเอกซเรย์ปอดทั่วไป (CXR) ร่วมกับวิธีการใช้โมเดลรูปแบบต่าง ๆ เพื่อการทำนายโรคปอดอักเสบโควิด COVID-19 จำแนกออกจากโรคปอดอักเสบทั่วไปและปอดปกติ โดยใช้ชุดข้อมูลเปิดจากฐานข้อมูลจากสาธารณะที่นิยมใช้กันแพร่หลาย การนำฐานข้อมูลเปิดมาใช้เป็นข้อมูลในการเรียนรู้ของสถาปัตยกรรมแบบจำลองทั้งจำนวน 6 โมเดล โดยได้เลือกใช้แบบจำลองซึ่งเป็นที่นิยมได้แก่ Inception V3, ResNet50, VGG16, VGG19, Mobilenet V2 และ InceptionResNet V2 เนื่องจากในแต่ละสถาปัตยกรรมของแต่ละแบบจำลองจะให้ผลลัพธ์ที่ดีกับภาพที่แตกต่างกัน โดยงานวิจัยนี้จะเป็นภาพจากการเอกซเรย์ปอด (CXR) มีรอยโรคโควิด COVID-19 ปอดอักเสบทั่วไปและภาพปอดปกติโดยที่แบบจำลองจะหาบริเวณที่เป็นรอยสีขาวที่ปรากฏขึ้นในภาพเอกซเรย์ที่แตกต่างเพื่อทำการทำนายผล จากผลลัพธ์ที่ได้จากการทดลองนั้นพบว่าแบบจำลองที่มีประสิทธิภาพในการทำนายรอยโรคปอดอักเสบที่เกิดจากการติดเชื้อโควิด COVID-19 เรียงจากมากไปน้อยคือ Inception V3, VGG16, ResNet50, VGG19, InceptionResNet V2 และ MobileNet V2 ตามลำดับในการฝึกโมเดลรุ่นแรก แต่กลับน่าประหลาดหลังจากในรุ่นที่สองที่มีการปรับอัตราส่วนจากรุ่นแรกจาก 60:20:20 เป็น 80:10:10 และเพิ่มการ augmentation ของข้อมูลและการ Transfer พารามิเตอร์จากรุ่นแรกเพื่อลดระยะเวลาในการฝึกสอนให้สั้นลง พบว่าผลลัพธ์ที่ได้นั้นมีค่าความแม่นยำ (precision) และค่าความถูกต้อง (accuracy) ดีขึ้นอย่างมาก โดยเรียงลำดับจากมากไปน้อยคือ InceptionResNet V2, Inception V3, ResNet50, VGG19, VGG16 และ Mobilenet V2 สาเหตุส่วนหนึ่งอาจเกิดจากการเลือกใช้แบบจำลองพบว่าแบบจำลองใหม่มีประสิทธิภาพได้ดีในการจำแนกสิ่งมีชีวิต เช่น ภาพสัตว์หรือสิ่งของ ซึ่งเป็นภาพถ่ายทั่วไปอย่าง InceptionResNet V2, Inception V3, ResNet50 เป็นต้น แสดงให้เห็นว่าสถาปัตยกรรมใหม่ ๆ ที่ออกแบบมาโดยหลักการทำงานที่ซับซ้อนแต่อาจจะเหมาะกับงานบางชนิดเท่านั้นดังนั้นการเลือกโมเดลจึงจำเป็นต้องพิจารณาเลือกแบบจำลองให้เหมาะสมกับงานนั้น ๆ ด้วย จะเห็นได้ว่าโครงข่ายประสาทเทียมแบบจำลอง InceptionV3 ในรุ่นแรกและ InceptionResNet V2 ในรุ่นที่สองมีประสิทธิภาพที่ดีในการใช้งานกับภาพเอกซเรย์ปอด เมื่อนำมาใช้กับการเรียนรู้บนฐานชุดข้อมูลสาธารณะ พบว่าประสิทธิภาพในการทำนายรอยโรคปอดอักเสบที่ติดเชื้อโควิด COVID-19 เพิ่มขึ้นและเมื่อนำแบบจำลองมาทดสอบแต่ละ

ฐานข้อมูลพบว่าการทดสอบโรคได้ประสิทธิภาพทั้งความถูกต้อง ความแม่นยำ และความไว ในการทำนายโรคดี ความแม่นยำคือ 0.9095, 0.9155, และ0.9089 ตามลำดับ ในการฝึกสอนรุ่นแรกและในรุ่นที่ 2 คือ 0.9516, 0.9521และ0.9516 ตามลำดับ ที่เป็นเช่นนี้นั้นทางผู้วิจัยได้วิเคราะห์ว่าอาจมาจากเหตุผลดังต่อไปนี้ คือ

- 1) จากแบบจำลองที่มีการเรียนรู้จากฐานชุดข้อมูลสาธารณะในปริมาณมาก จึงทำให้แบบจำลองมีความคุ้นเคยกับชุดข้อมูล
- 2) จากฐานข้อมูลสาธารณะที่หลากหลายและประสิทธิภาพจากโมเดลที่มีความแตกต่างกันจึงส่งผลให้บางโมเดลได้ผลลัพธ์ที่ดีและไม่ดีต่างกันด้วย
- 3) ปัจจัยที่อาจจะเกิดขึ้นได้ เนื่องจากชุดข้อมูลเป็นสาธารณะตัวแปรที่อาจทำให้ภาพแตกต่าง เช่น เครื่องเอกซเรย์ที่ใช้ อายุของเครื่องถ่ายเอกซเรย์ที่ไม่เท่ากันถ่ายภาพออกมาไม่เหมือนกัน การปรับแต่งค่าพลังงานรังสีเอกซ์ แผ่นรับตัวสัญญาณมีการกระจายของแสงที่ไม่เหมือนกันและเทคนิคที่ใช้ในการฉายภาพนั้นแตกต่างกัน ส่งผลให้กับคุณภาพของภาพเอกซเรย์ทั้งสิ้น

ประโยชน์ที่จะได้รับจากงานวิจัยการใช้แบบจำลองเพื่อทำนายโรคโควิด COVID-19 โดยใช้ภาพเอกซเรย์ปอด จากพลังงานรังสีเอกซ์ ทำให้ช่วยลดเวลาในการอ่านฟิล์มของแพทย์ที่จะคัดกรองและช่วยลดภาวะตึงเครียดของบุคลากรทางการแพทย์หน้างาน ส่งผลในมีเวลาไปดูแลผู้ป่วยได้เพิ่มมากขึ้น การที่เรากัดกรองผู้ที่มีความเสี่ยงสูงออกมาได้ก่อนยังช่วยลดอัตราโอกาสการแพร่กระจายโรคไปยังคนอื่น ๆ อีกทั้งการคัดกรองผู้ที่มีความเสี่ยงสูงออกมาก่อนจะช่วยในการตัดสินใจของบุคลากรทางการแพทย์เพื่อการจัดการบริการรวมถึงทรัพยากรและการส่งตรวจอื่น ๆ ทำได้คล่องตัว รวดเร็วมากขึ้น การตรวจเอกซเรย์ปอดทั่วไปที่ทำการเป็นพื้นฐานในการตรวจสอบสุขภาพอยู่แล้วนั้นยังถือว่าเป็นวิธีการวินิจฉัยโรคโควิด COVID-19 อีกวิธีหนึ่ง เป็นเทคนิคที่ให้ความถูกต้องในการประเมินผลการติดเชื้อค่อนข้างดีเป็นวิธีบ่งชี้พื้นฐานที่สามารถจะทำได้ง่ายเหมือนการตรวจร่างกายร่วมด้วยกับการนำเทคโนโลยีปัญญาประดิษฐ์เข้ามาช่วยสร้างระบบคอมพิวเตอร์ผู้ช่วยอัจฉริยะเพื่อคิดวิเคราะห์และทำการแปลผลภาพถ่ายรังสีจากการถ่ายภาพเอกซเรย์ปอด การดำเนินการได้รวดเร็วมากขึ้น แต่อย่างไรก็ตามการตรวจด้วยดังกล่าวนี้ก็ยังไม่ถือเป็นการตรวจที่มาทดแทนการตรวจสอบด้วยชุดตรวจปกติได้แต่อย่างไรวิธีดังกล่าวแนะนำให้มีการทำซ้ำและอ่านภาพที่เกิดขึ้นหลาย ๆ รอบ เพื่อหารอยโรคในปอดที่มีจำนวนมากและมีความเปลี่ยนแปลงอย่างรวดเร็วของรอยโรคที่เกิดกับปอดอักเสบเพื่อความแม่นยำสูงสุดเพื่อให้เกิดประโยชน์กับคนไข้มากที่สุด งานในอนาคตผู้วิจัยอาจจะมีการพัฒนาการฝึกสอนแบบจำลองด้วยเทคนิค (CNN: Deep-learning) จะพัฒนาด้วยอัลกอริทึมและโมเดลใหม่อื่น ๆ เช่น

DenseNet121 เพื่อหาเทคนิคที่ดีที่สุดสำหรับฝึกสอนแบบจำลองด้วยเทคนิค CNN เพื่อจำแนกภาพเอกซเรย์ปอด สำหรับวินิจฉัยโรคต่อไป

5.2 บทวิจารณ์และข้อเสนอแนะ

จากการศึกษางานวิจัยจะพบได้ว่าจำนวนข้อมูลภาพสามารถช่วยเพิ่มประสิทธิภาพการทำงานของโมเดลให้กับแบบจำลองได้เป็นอย่างดี จากแหล่งข้อมูลภาพจากสาธารณะดังกล่าวที่ทำให้ภาพของเอกซเรย์มีคุณภาพของภาพที่ไม่เหมือนกันและต้องศึกษาเพิ่มเติมอัลกอริทึมให้กับโมเดลสำหรับการคัดกรองภาพที่มีคุณภาพไม่เข้าเกณฑ์ออกก่อนการนำภาพไปใช้ในแบบจำลองในการทำนายโรค ทั้งนี้เพื่อเพิ่มประสิทธิภาพให้กับแบบจำลอง แล้วนำภาพบางส่วนที่ไม่เข้าเกณฑ์สาเหตุจากคุณภาพของภาพไม่ดีหรือมีความสว่างน้อย ถ่ายผิดประเภทหรือการผิดพลาดทางเทคนิค ควรส่งไปให้กับทางรังสีแพทย์ช่วยพิจารณาเลือกและแปลผล นอกจากนี้อาจมีการใช้เทคนิคอื่น ๆ เพื่อมาช่วยเพิ่มประสิทธิภาพการนำมาประมวลผลกับภาพเช่น การใช้โปรแกรมตัดภาพเฉพาะส่วนของปอดหรือชักประวัติอาการของผู้ป่วยร่วมกับผลทางห้องปฏิบัติการเข้ามาช่วยในการพิจารณาการตัดสินใจเพื่อเพิ่มความแม่นยำให้กับโมเดลในการทำนายได้เป็นอย่างดี

บรรณานุกรม

เอกสารอ้างอิง

- [1] K.U. Ahamed, M. Islam, A. Uddin, Akhter A, B.K. Paul, M.A.Yousuf, et al. A deep learning approach using effective preprocessing techniques to detect COVID-19 from chest CT-scan and X-ray images. *Comput Biol Med.* 2021; 139:105014.
- [2] T.B. Alakus, I. Turkoglu. Comparison of deep learning approaches to predict COVID-19 infection. *Chaos Solitons Fractals.* 2020; 140:110120.
- [3] I. Ahmed, M. Ahmad, G. Jeon. Social distance. monitoring framework using deep learning architecture to control infection transmission of COVID-19 pandemic. *Sustain Cities Soc.* 2021; 69:102777.
- [4] A.P. Adedigba, S.A. Adeshina, O.E. Aina, A.M. Aibinu. Optimal hyperparameter selection of deep learning. models for COVID-19 chest X-ray classification. *Intel Based Med.* 2021; 5:100034.
- [5] L.R. Baltazar, M.G. Manzanillo, J. Gaudillo, E.D. Viray, M. Domingo, B. Tiangco, et al. Artificial intelligence on COVID-19 pneumonia detection using chest xray images. *PLoS One.* 2021; 16(10): e0257884.
- [6] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images:Results of 10 convolutional neural networks. *Comput Biol Med.* 2020; 121:103795.
- [7] Hemanth RJMGSTDJ. Deep learning-based detection and analysis of COVID-19 on chest X-ray images. 2020.
- [8] J. de Moura, J. Novo, M. Ortega. Fully automatic deep convolutional approaches for the analysis of COVID-19 using chest X-ray images. *Appl Soft Comput.* 2022; 115:108190.
- [9] A. Abdelgawad and F.Ha. Khandaker. A Deep Learning Approach to Detect COVID-19 Patients from Chest X-ray Images †. 2020.
- [10] X-ray Tube Working Principle [ออนไลน์]. 2566 [ค้นเมื่อวันที่ 15 มกราคม 2566].

- จาก <http://www.gpmanesar.ac.in/GPContent/X-RAY%20MACHINE-converted.pdf>
- [11] Working Principle of X-Ray Machine [ออนไลน์]. 2566 [ค้นเมื่อวันที่ 16 มกราคม 2566].
จาก <https://studiousguy.com/working-principle-x-ray-machine/>
- [12] การเชื่อมโยงของเซลล์ประสาท [ออนไลน์]. 2566 [ค้นเมื่อวันที่ 16 มกราคม 2566]. จาก
https://www.google.co.th/search?q=การเชื่อมโยงของเซลล์ประสาท&tbm=isch&sa=X&ved=2ahUKEwi3zun8_838AhVuxjgGHcwlCXAQ0pQJegQIChAB&biw=1005&bih=724&dpr=1#imgrc=7TBJ43K0gzRUUM&imgdii=ys4oTpDQasYZIM
- [13] คู่มือเกณฑ์การคัดแยกระดับความผิดปกติจากภาพรังสีทรวงอกเพื่อใช้สำหรับการวินิจฉัยภาวะปอดอักเสบในผู้ป่วยโรคโควิด 19 (เวอร์ชัน 1) [ออนไลน์]. 2564 [ค้นเมื่อวันที่ 18 มกราคม 2566].
จาก
<https://www.rama.mahidol.ac.th/radiology/sites/default/files/public/knowledge/20210505050251.pdf>
- [14] Training Deep Neural Networks [ออนไลน์]. 2561 [ค้นเมื่อวันที่ 18 มกราคม 2566]. จาก
<https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964>
- [15] Learn about AUC ROC Curve [ออนไลน์]. 2561 [ค้นเมื่อวันที่ 25 มกราคม 2566]. จาก
<https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152>
- [16] The Annotated Resnet-50 [ออนไลน์]. 2564 [ค้นเมื่อวันที่ 26 มกราคม 2566]. จาก
<https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>
- [17] Inception V3 Model Architecture [ออนไลน์]. 2564 [ค้นเมื่อวันที่ 26 มกราคม 2566]. จาก
<https://iq.opengenus.org/inception-v3-model-architecture/>
- [18] Everything you need to know about VGG16 [ออนไลน์]. 2564 [ค้นเมื่อวันที่ 26 มกราคม 2566]. จาก
<https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>
- [19] How ReLU and Dropout Layers Work in CNNs [ออนไลน์]. 2564 [ค้นเมื่อวันที่ 26 มกราคม 2566]. จาก <https://www.baeldung.com/cs/ml-relu-dropout-layers>
- [20] ReLu Function คืออะไร [ออนไลน์]. 2561 [ค้นเมื่อวันที่ 26 มกราคม 2566]. จาก
<https://www.bualabs.com/archives/1355/what-is-relu-function-why-popular-deep-learning-training-deep-neural-network-activation-function-ep-3/>
- [21] waenghospital.com [ออนไลน์]. 2560 [ค้นเมื่อวันที่ 30 มกราคม 2566]. จาก

- http://www.waenghospital.com/newdetail_Health.php?id=H018
- [22] Python คืออะไร [ออนไลน์]. 2566 [ค้นเมื่อวันที่ 31 มกราคม 2566]. จาก <https://aws.amazon.com/th/what-is/python/#:~:text=Python%20เป็นภาษาการเขียน,ประเภท%20และเพิ่มความเร็วใน>
- [23] DICOM file image และ CT Image [ออนไลน์]. 2562 [ค้นเมื่อวันที่ 31 มกราคม 2566]. จาก <https://parinkittipongdaja.medium.com/รู้จักกับ-dicom-file-image-และ-ct-image-65115d7b693d>
- [24] การจำแนกผลึกน้ำตาลด้วยวิธีการเรียนรู้เชิงลึก [ออนไลน์]. 2564 [ค้นเมื่อวันที่ 8 กุมภาพันธ์ 2566]. จาก https://www.cp.eng.chula.ac.th/~prabhas/thesis/iThesis_suriya_2020.pdf
- [25] COVID-19 and Pneumonia Diagnosis in X-Ray Images Using Convolutional Neural Networks.2021; 10.1155:3281135.
- [26] โครงข่ายประสาทเทียม [ออนไลน์]. 2565 [ค้นเมื่อวันที่ 20 กุมภาพันธ์ 2566]. จาก <https://th.wikipedia.org/wiki/โครงข่ายประสาทเทียม>
- [27] Understanding the VGG19 Architecture [ออนไลน์]. 2565 [ค้นเมื่อวันที่ 2 มีนาคม 2566]. จาก <https://iq.opengenus.org/vgg19-architecture/>
- [28] InceptionResNetV2 Simple Introduction [ออนไลน์]. 2563 [ค้นเมื่อวันที่ 10 มีนาคม 2566]. <https://medium.com/@zahraelhamraoui1997/inceptionresnetv2-simple-introduction-9a2000edcdb6>

ประวัติผู้เขียน

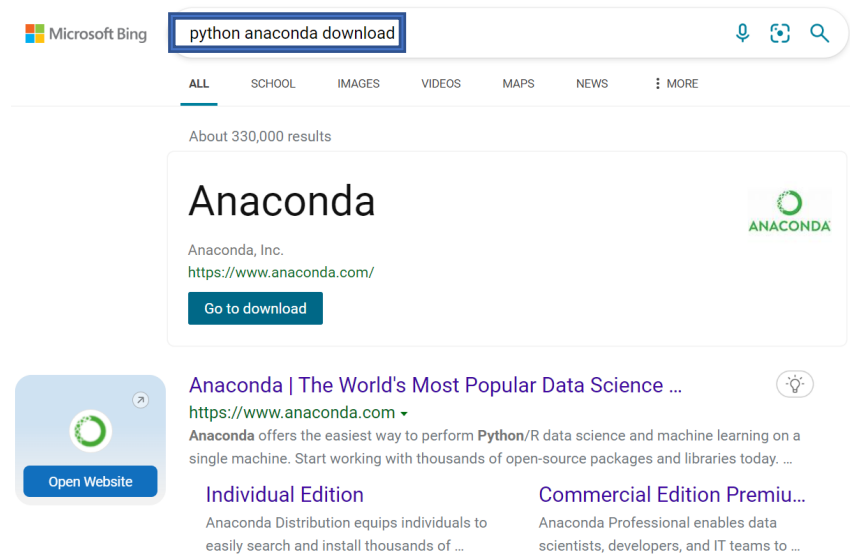
ชื่อ-นามสกุล	นางสาววรรณิกา ศรีอารา
วัน เดือน ปีเกิด	20 สิงหาคม พ.ศ.2529 ที่ ราชบุรี
ที่อยู่	คอนโด เอส-สปาย ดีก คอปเปอร์โรชน สาธร-ตากสิน ซอยสวนหลวง แขวง บางค้อ เขตจอมทอง กรุงเทพฯ 10150
ประวัติการศึกษา	2550 อนุปริญญาสาขารังสีเทคนิค โรงเรียนรังสีเทคนิคโรงพยาบาล จุฬาลงกรณ์สภากาชาดไทย 2558 วิทยาศาสตรบัณฑิต สาขารังสีเทคนิค มหาวิทยาลัยรามคำแหง
ความชำนาญเฉพาะด้าน	1.) เอกซเรย์คอมพิวเตอร์ 2.) เครื่องแม่เหล็กกำลัง
ประสบการณ์ทำงาน	2550-ปัจจุบัน ตำแหน่งนักรังสีเทคนิค 5 โรงพยาบาลจุฬาลงกรณ์ สภากาชาดไทย (Full time) 2553-2563 ตำแหน่งนักรังสีเทคนิค โรงพยาบาลวิภาวดี ปากเกร็ด (Part time) 2554-2556 ตำแหน่งนักรังสีเทคนิค โรงพยาบาลบางโพ บางซื่อ (Part time) 2556-2558 ตำแหน่งนักรังสีเทคนิค โรงพยาบาลลาดพร้าว (Part time) 2553-2557 ตำแหน่งนักรังสีเทคนิค โรงพยาบาลเดชา (Part time)

ภาคผนวก ก.

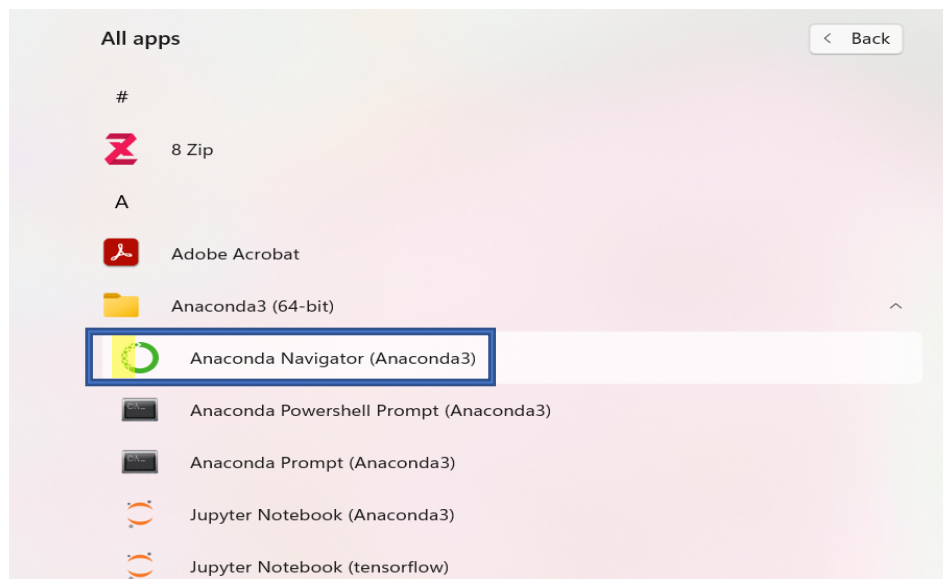
วิธีการลงโปรแกรมสำหรับฝึกสอนโมเดล

วิธีการลงไลบรารีในการสอนโมเดล

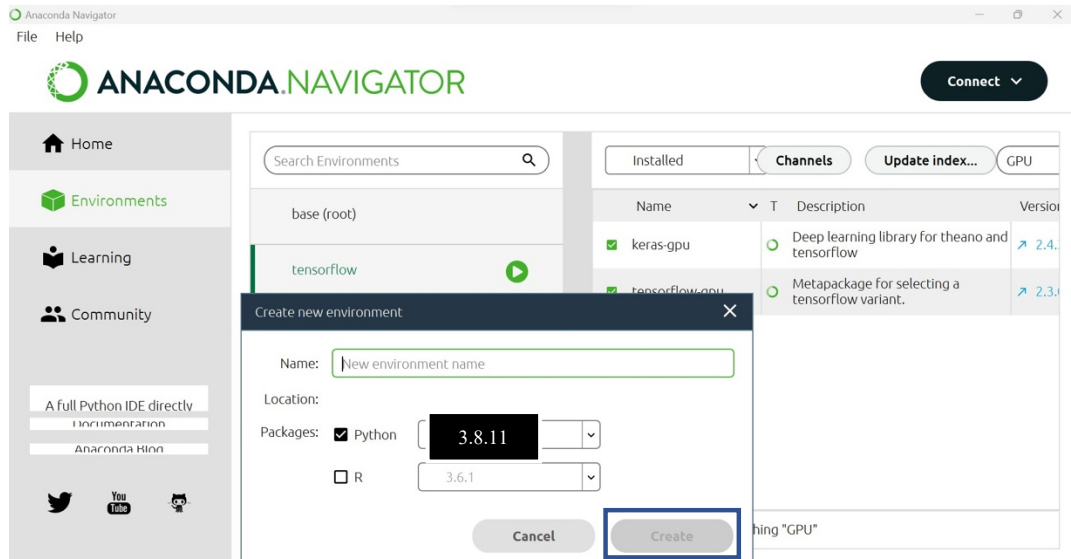
1. การค้นหาดาวน์โหลดโปรแกรมจากเว็บไซต์และทำการติดตั้งซอฟต์แวร์ Anaconda



2. ติดตั้ง-กดเลือกใช้โปรแกรมจากไอคอน

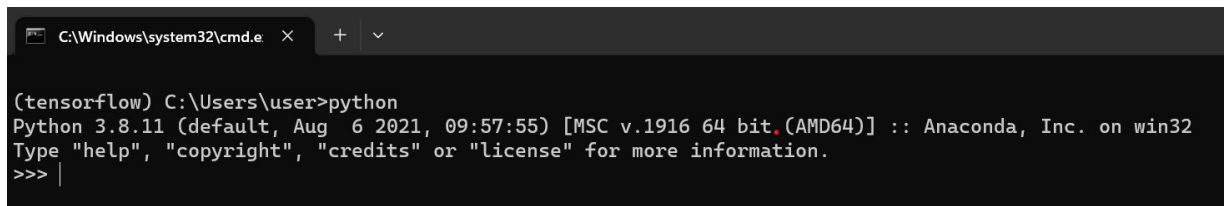


3. เปิดเข้า ANACONDA.NAVIGATOR ทำการสร้าง Environments โดยกดเข้าไปที่ Environments → Create จากนั้นตั้งชื่อเป็น tensorflow เราทำการติดตั้งตัวไลบรารีอื่น ๆ ที่จำเป็นต้องใช้ในการฝึกสอนโมเดลไว้เช่น ภาษา Python 3.8.11

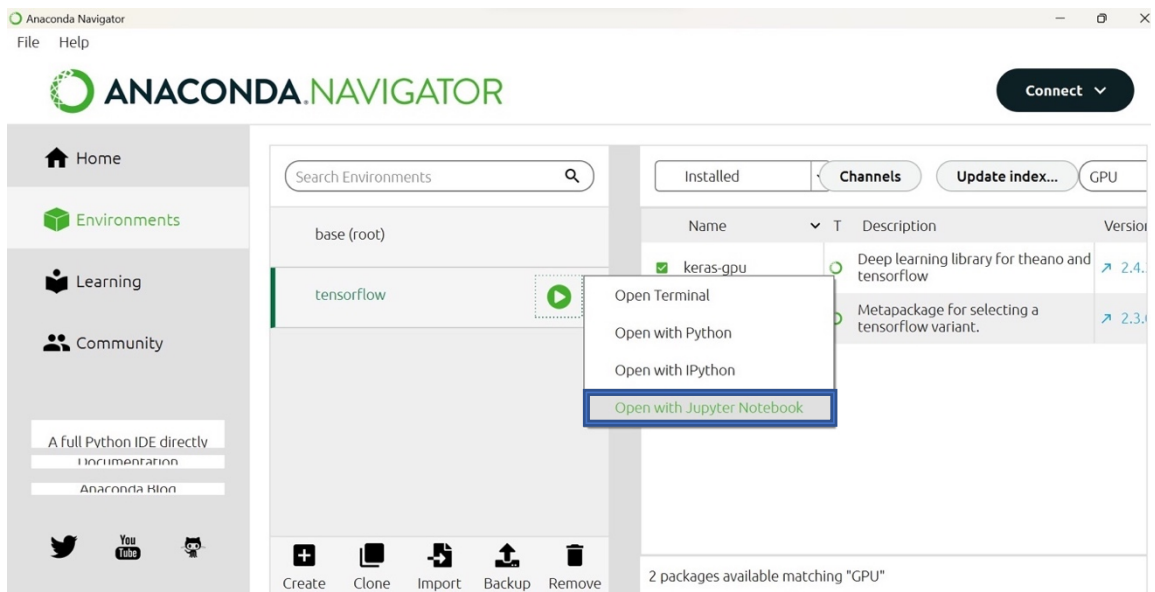


4. กดเข้าหน้า terminal แล้วทำการตรวจสอบ version ของ python ว่าตรงตามที่ต้องการหรือไม่ด้วยคำสั่ง python

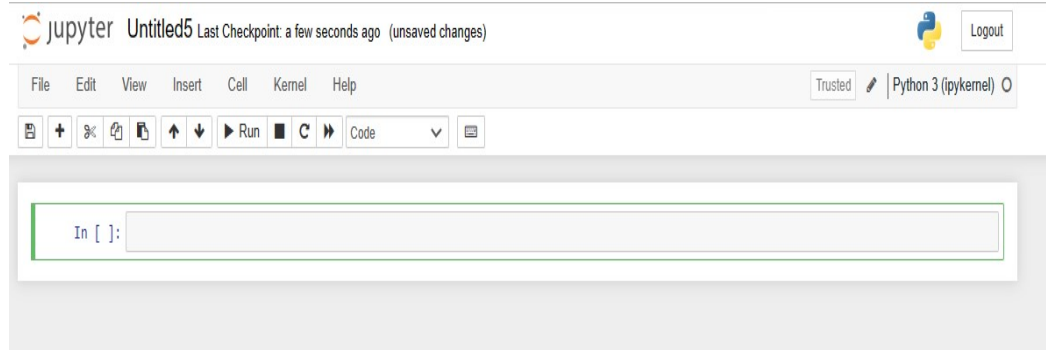
หมายเหตุ: เราสามารถลงไลบรารีอื่น ๆ ได้โดยเลือก Installed บน Anaconda



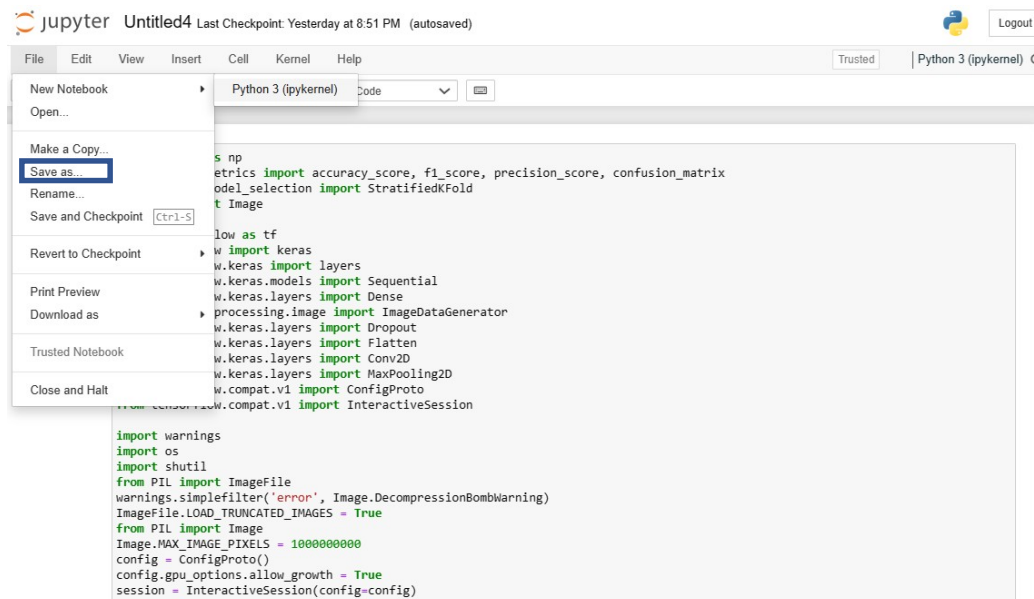
5. คลิกที่ลูกศรสีเขียว คลิกขวาเลือก Jupyter notebook



6. หน้าต่างสำหรับการเขียน Code พร้อมใช้งาน บน Jupyter Anaconda



7. การบันทึก Code ที่เขียนหลังการสอนโมเดลแล้ว



ภาคผนวก ข.
บทความทางวิชาการ

Detection and Classification of COVID-19 Chest X-rays by the Deep Learning Technique

Wannika Sonarra, Naphatsawan Vongmanee, Nutthanan Wanluk, Chuchat Pintavirooj and Sarinporn Visitsattapongse*
Department of Biomedical Engineering, School of Engineering,
King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok 10520, Thailand.
*Correspondence author's, Email: sarinporn.vi@kmitl.ac.th

Abstract—The Coronavirus disease (COVID-19) infection has become a pandemic, and this is the most critical problem that has occurred in Thailand and also expanded all over the world. As such, it is not astonishing to know that this virus has had a direct effect on hospitals with the delayed screening of patients because of the increasing number of daily cases and the shortage of medical personnel and restricted treatment space. Due to such restrictions, in this study, we used a clinical decision-making system with predictive algorithms. Predictive algorithms could potentially ease the strain on healthcare systems by identifying the diseases. Moreover, image classification is one interesting aspect of image processing. Convolutional neural network (CNN) is a widely used algorithm for image classification by separating the images of the COVID-19 disease, images with a lung infection, and normal images. To evaluate the predictive performance of our models, precision, F1-score, recall, receiver operating characteristic (ROC) curve (area under the ROC curve), and accuracy scores were used. It was observed that the predictive models trained on the laboratory findings could be used to predict the COVID-19 infection as well and could be helpful for medical experts to appropriately prioritize the resources. This could be employed to assist medical experts in validating their initial laboratory findings and could also be used for clinical prediction studies.

Keywords—Convolutional neural network (CNN), image of COVID-19, coronavirus, deep learning

I. Introduction

At present, the super spreading and infection of Coronavirus disease 2019 or COVID-19 is a major problem in Thailand and other countries throughout the world. [1, 2] Therefore, surveillance is essential for monitoring the infection situation closely and preventing further infection as this virus mutates and outbreaks become airborne and spread more quickly. Moreover, spreading the infection, such as coughing, sneezing, and talking without wearing a mask [3] would exacerbate the situation of this pandemic and affect Thai society in all aspects, such as economic, social, health, etc. However, the primary first is health; thus, a lot of people have been infected,

and the numbers are increasing everyday, as infection from human to human is very easy and quick; consequently, this has seen a rising, mortality rate. To directly initiate the treatment provided, medical and public health agencies must increase their workforce to help care for the infected people in the hospitals. Hence, the methods of screening the general patients for infection has been divided into two types: Molecular sampling secretions and serum test kit or antigen test kit through the patient's nose and mouth or reverse transcription polymerase chain reaction (RT-PCR). The swap from these samples is thus able to detect the specific genetic code of the virus that causes COVID-19, e.g., Coronavirus, severe acute respiratory syndrome or SARS-CoV-2. Therefore, this has caused delays with patients receiving incomplete and/or inaccessible good quality treatment because of the lack of specialist medical personnel. As a result, some patients have died. On the other hand, the test by serum versus the blood sample of the infected patients for detection of any antibodies or immunity has demonstrated the disadvantage of this technique, which is a complex self-examination procedure and requires several hours of waiting for the results. As a consequence, it would be necessary to train new medical staff amid the pandemic. Hence, an alternative technique that could increase the screening for COVID-19 is a chest X-ray. Furthermore, medical experts would have increased confidence with this method of detection. However, the principle of an essential check-up for detecting a patient's disease used the chest X-ray method for screening. [4, 5] In this study, we used convolutional neural network (CNN; deep learning) model techniques to classify chest X-ray images.[3] The advantage was increasing the efficiency of the decisions for the medical experts and decreasing the time for repeating the results because the virus could change quickly just like, their bodies. Additionally, this would help reduce the rate of infection and number of deaths as well.

II.Theories and Related research

Convolutional Neural Networks (CNN)

A CNN was developed from the neural networks, as the method could be used for image processing [6] and be able to simulate human vision by extracting features of the images brought into the training process by the CNN technique. The image featured the extraction and analysis, so to define the image properties (input) and then classify the image type a category, such as a chest X-ray image. The learning process was divided into three categories consisting of two positive groups, infected with COVID-19, (Fig. 1 a) and chest X-ray of non-COVID-19 infection (Fig. 1 b), and a negative group with a normal chest X-ray (Fig. 1 c).

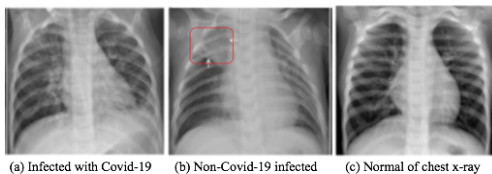


Fig. 1. Data set of the chest X-ray images.

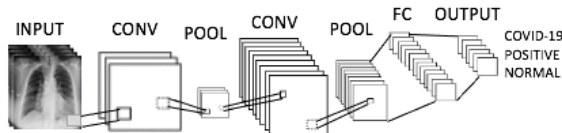


Fig. 2. Convolutional neural networks (CNN)

The CNN technique of the image classification process was applied to the network. Multiple neural layers were examined for the attributes and iterations until finding the appearance. The most accurate classification was expressed as a probability value. For example, when a chest X-ray image was placed into the CNN classification process, the result had the probability of classifying a chest X-ray image appropriately. This would be shown as a chest X-ray with a COVID-19 infection or being uninfected with COVID-19. The CNN technique process is as follows: (Fig. 2)

(1) Convolution Layer (CONV) is a layer that is responsible for extracting the outstanding features of an image.

(2) Pooling Layer (POOL) is a layer that acts to reduce the size of an image for the benefit of the extraction of the image's feature to cover more features contained in the image.

(3) Dropout is a layer that reduces the overfit issue by the random data and removes the nodes during training to make the model easier and also to reduce the complexity of the data.

(4) Full Connection (FC) is a layer that connects the CONV layer and the POOL layer to achieve the efficiency of the image's classification. The CNN

technique can also use the CONV and POOL layers that are responsible for extracting the characteristics of the image beneath more than one layer.

III.Literature review

The literature to be included in this review was obtained as follows: We learned and found the knowledge of various related other research studies from PubMed that had used the CNN techniques, conducted models, and recorded the results.

Jain et al. [7] used a data set from the general public for training and testing three models: Inception V3, Xception Net, and ResNeXt. Xception displayed the highest accuracy, and the result showed that the approach had a high score of 97.97%.

De Moura et al. [8] improved the efficiency of the CNN technique followed by classifying the images by using publicly available image data sets. The results proved that this could effectively help with the convenience and increase the rapidity of the analysis of the images for medical professionals.

Foysal and Abdelgawad [9] used three models (ResNet 50, VGG-16 and VGG-19) and two data sets to classify the images. The summary of the available test results showed that the efficiency had 98.39% accuracy, 96.72% accuracy, receiver operating characteristic (ROC) curve 0.983, F1-score (accuracy value) and recall value ($\frac{2}{\text{precision value} + \text{recall value}}$). More images of the second data set useful information that could be used in specific research, especially for the image processing to classify COVID-19. According to the above-related research studies, the technique of the CNN model could classify the X-ray images for the diagnosis of COVID-19 quite effectively.

IV.Material and Method

Data set

In this study, we established a public database of chest X-ray images of patients with lung disease. As we had obtained the chest X-ray images from public resources so not to violate the patient's confidentiality, we then classified them with the confirmed results from special or medical radiology in the respective category by using, a 2D image, or a PNG file as the image data, in particular the chest X-rays of COVID-19 patients. The positions of the chest X-ray were the posteroanterior (PA) and anteroposterior (AP) by using CNN network techniques for screening patients infected with COVID-19. We divided the image data set into three categories: 1,400 images each for the model training, 360 images each for validation, and the last group comprised 400 images for the testing model. The images consisted of the three categories of patient: The negative group with a normal chest x-ray, one positive group infected with

COVID-19 and the other positive group of non-COVID-19 infection, respectively.

Instrument

In this study, we used an online library consisting of Keras version 2.4.3 with Tensorflow version 2.3.0 to train the model, such as Inception V3, ResNet 50, VGG-16, VGG-19, MobileNet 50, and InceptionResNet V2. Both Keras and Tensorflow are libraries developed in the Python programs, which are efficient in the training process to classify images. We used it in a previous study and in Python version 3.9.0 put the details of the installation of both Keras and Tensorflow on a computer that used an Intel (R) Central Processing Unit (CPU). Core (TM) i5-1035G1 CPU @ 1.00 GHz 1.19 GHz 8GB RAM, 8GB Operating System (OS) and a Graphics Processing Unit GPU, and NVIDIA GeForce MX350.

Evaluation Criteria of the Equations

The measurement of the effectiveness of coaching in this study was to measure the accuracy, which was a percentage of the probability (%) obtained from the test to find the forecast accuracy of the data set with the formula for calculating. the evaluation of the classification, such as the accuracy, precision, recall/sensitivity, and F1-score is found in equations (1-4):

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

$$\text{Recall/Sensitivity} = \frac{TP}{(TP+FP)} \quad (2)$$

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

$$\text{F1} = \frac{2*(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

Experiment



Fig. 3. The training model.

A data set of the generated images of the three groups was taken, which consisted of two positive groups with the classification between the chest X-rays of COVID-19 and chest X-rays of non-COVID-19 patients. The negative group was a chest X-ray image of a group of patients not infected with COVID-19 was placed into the system. The system took the original X-ray image to create an artificial image in the data set generator in each technique with 1,400 images per group. We also used the Keras and Tensorflow libraries and wrote in Python. Then, we set the ratio of the coaching model 80:10:10

that was divided into training 80%, validation 10%, and testing 10, which was the artificial visualization in each technique. The experiments were conducted on six models: InceptionV3, ResNet50, VGG-16, VGG-19, MobileNet 50, and InceptionResNet V2 of CNN, with the computer learning in 50 epochs of training, Optimizer Adam algorithm, a learning rate of 0.00001, and the last epoch for predicting each training, validation, and test data.

V.Results

In this study, the results obtained from the CNN experiments, the measurement of the experiments, and the transfer learning of data is presented as follows:

Results of the Experiments

As mentioned above, from the experiments using the CNN technique to classify the chest X-ray images for the diagnosis of the COVID-19 primary in this study, the six models were used:

InceptionV3, ResNet50, VGG-16, VGG-19, MobileNet 50 and InceptionResNet V2 by requiring the computer to learn in the training process of the 50 epochs, training efficiency in each technique of each model, validation, and testing, as shown in Table 1-5, respectively.

TABLE 1. MODEL AND TRAINING PROCESS.

Architecture	Epoches	Optimizer	Learning Rate
Inception V3	50	adam	0.00001
ResNet 50			
VGG-16			
VGG-19			
MobileNet 50			
InceptionResNet V2			

TABLE 2. MODEL AND TRAINING PROCESS.

Architecture	Training Result				
	loss	acc	precision	recall	F1-score
Inception V3	0.0291	0.9921	0.9921	0.9919	0.9919
ResNet 50	0.0863	0.9714	0.9723	0.9707	0.9715
VGG-16	0.1112	0.9660	0.9680	0.9638	0.9630
VGG-19	0.2593	0.9164	0.9213	0.9110	0.9161
MobileNet 50	0.2462	0.9080	0.8901	0.8910	0.8905
Inception Resnet V2	0.0560	0.9833	0.9833	0.9831	0.9832

TABLE 3. MODEL AND TESTING PROCESS.

TABLE 4. MODEL AND TESTING PROCESS.

Architecture	Testing Result			
	Test_acc	Test_Precision	Test_F1-score	Sensitivity/recall
Inception V3	0.9095	0.9155	0.9089	0.9744
ResNet 50	0.8003	0.8601	0.7884	1.0000
VGG-16	0.8626	0.8658	0.8635	0.9250
VGG-19	0.8217	0.8265	0.8237	0.9410
MobileNet 50	0.6328	0.6490	0.5525	0.8956
Inception Resnet V2	0.7749	0.8549	0.7431	1.0000

TABLE 5. MODEL AND TESTING PROCESS.

Architecture	Testing Result		
	AUC		
	Normal	Pneumonia	Pneumonia Covid 19
Inception V3	0.9523	0.9357	0.9923
ResNet 50	0.9516	0.9446	0.9999
VGG-16	0.9487	0.9308	0.9983
VGG-19	0.9288	0.8778	0.9787
MobileNet 50	0.8004	0.8632	0.9767
InceptionResnet V2	0.9376	0.9030	0.9998

In this study, the accuracy and F1-score defined how close the generated result was to the actual value, whereas the precision measured the percentage (%) of the pertinent results. The recall/sensitivity was an essential factor for evaluating the CNN model. It was defined by the percentage (%) of the total pertinent results that the model could correctly classify. The F1-score combined both the precision and recall, and it was designated as the weighted average of these two factors. Equations (1-4) represented the accuracy, precision, recall, and F1-score, respectively.

The InceptionV3 model from this experiment had the highest accuracy of 90.95%, contrarily, MobileNet50 performed with the lowest accuracy of 63.28%. The accuracy of the proposed model was 90.95% with the precision and recall value of 91.55% and 97.44% for the parameters, respectively. The overall performance including the accuracy and F1-score could be improved further by training the model with a larger data set. (Figs. 4-9).

ROC curve and AUC

To further examine the efficacy of the proposed approach, we evaluated the overall comparison between the models for all the threshold values. In this study, receiver operating characteristics (ROC) curves were used to demonstrate the diagnostic potential of the classifier system when the threshold for distinction was varied. They provided the true positive rate (TPR) as a function of the false positive rate (FPR) and plotted a graph between the two. Additionally, classification was used on the six models of the CNN technique that

Architecture	Validation Result				
	Val_loss	Val_acc	Val_Precision	Val_recall	Val_F1-score
Inception V3	0.3006	0.9078	0.9094	0.9078	0.9086
ResNet 50	0.4871	0.8299	0.8340	0.8249	0.8295
VGG-16	0.4440	0.8452	0.8465	0.8443	0.8440
VGG-19	0.5159	0.8249	0.8275	0.8240	0.8220
MobileNet 50	1.0000	0.6590	0.6492	0.6511	0.6555
Inception Resnet V2	0.9888	0.7346	0.7359	0.7346	0.7353

consisted of InceptionV3, VGG-19, VGG-16, InceptionResNet V2, MoblieNet 50, and ResNet 50 (Fig. 11-16).

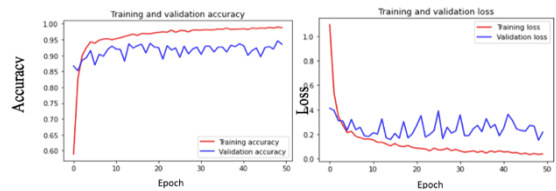


Fig. 4. Training and validation of the Inception V3 model.

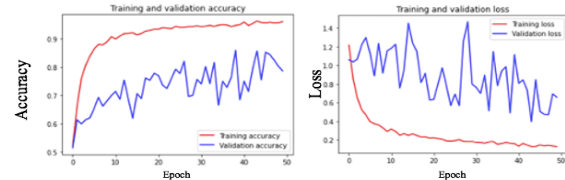


Fig.5. Training and validation of the VGG-19 model.

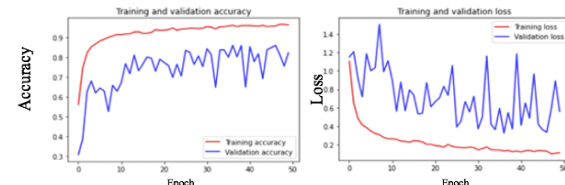


Fig.6. Training and validation of the VGG-16 model.

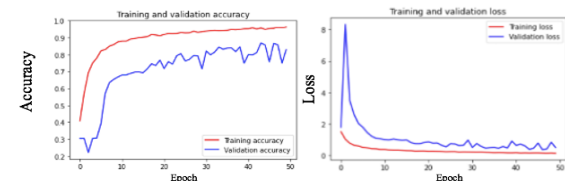


Fig.7. Training and validation of the ResNet 50 model.

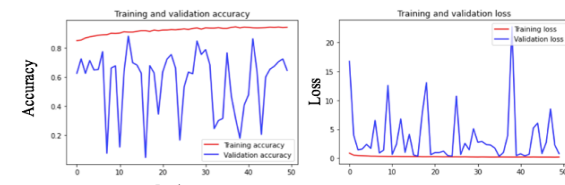


Fig.8. Training and validation of the MobileNet 50 model.

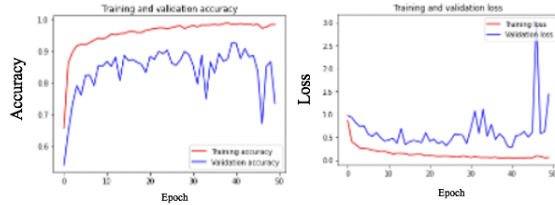


Fig.9. Training and validation of the InceptionResNet V2 model.

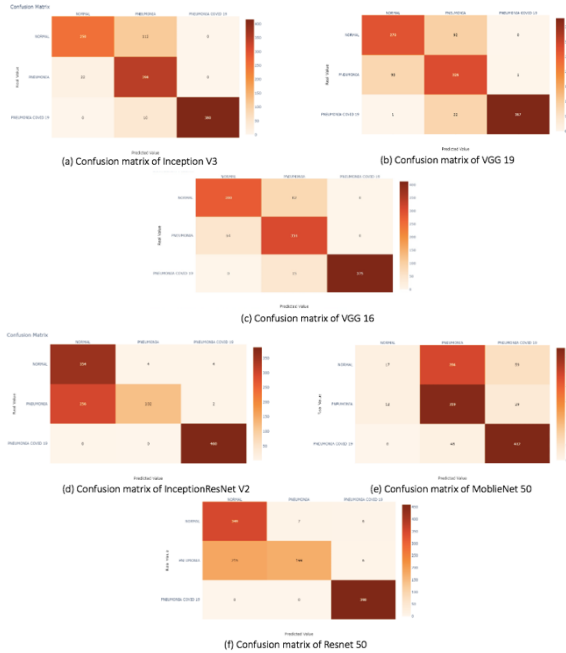


Fig.10. Confusion matrices of all six models for analysis with the validation data set: (a) InceptionV3, (b) VGG-19, (c) VGG-16, (d) InceptionResNet V2, (e) MobileNet 50, and (f) ResNet 50 with reference to the normal, pneumonia and COVID-19 patient's X-ray images, respectively.

VI. Conclusions and future scope

In this study, we collected chest X-ray images from an available public data set and through improving by expert radiology, enabled further refinements to the systems employing them. The highest test accuracy was 90.95% with, precision of 91.55%, and the sensitivity and F1- scores were 97.44% and 90.89%, respectively. Furthermore, this could reduce the workload on strained health workers, and provide better healthcare entrance to under-equipped areas. Nevertheless, the first issue for consideration was the most accurate methods for diagnosing COVID-19. The evaluation of the infection proved to be very effective, which was a basic indication that could be as simple as a physical examination. In addition, artificial intelligence technology could be used to create intelligent computer systems to help radiography to analyze and evaluate the results from chest X-ray imaging, as well as with speed. However, this chest X-ray examination was not a substitute for chemotherapy. Therefore, it was

recommended to conduct a check with a normal test kit or antigen test kit. Moreover, conducting an examination by this method is recommended, as more images could be conducted by the specialist or radiology expert. As such lesions in the lungs that would be large and rapidly changing would be examined in a future study. Furthermore, CNN model coaching results would be developed with other logarithms, such as Lenet to determine the most appropriate logarithm for the training of a model with CNN to classify the chest X-ray images for the diagnosis of COVID-19.

ACKNOWLEDGMENT

The authors would like to thank you Sarinporn Visitsattapongse, Ph. D. and Chuchat Pintavirooj, Ph. D. for suggestions on principle of research.

REFERENCES

- [1] K.U. Ahamed, M. Islam, A. Uddin, Akhter A, B.K. Paul, M.A. Yousuf, et al. A deep learning approach using effective preprocessing techniques to detect COVID-19 from chest CT-scan and X-ray images. *Comput Biol Med.* 2021; 139:105014.
- [2] T.B. Alakus, I. Turkoglu. Comparison of deep learning approaches to predict COVID-19 infection. *Chaos Solitons Fractals.* 2020;140:110120.
- [3] I. Ahmed, M. Ahmad, G. Jeon. Social distance monitoring framework using deep learning architecture to control infection transmission of COVID-19 pandemic. *Sustain Cities Soc.* 2021; 69:102777.
- [4] A.P. Adedigba, S.A. Adeshina, O.E. Aina, A.M. Aibinu. Optimal hyperparameter selection of deep learning models for COVID-19 chest X-ray classification. *Intell Based Med.* 2021; 5:100034.
- [5] L.R. Baltazar, M.G. Manzanillo, J. Gaudillo, E.D. Viray, M. Domingo, B. Tiangco, et al. Artificial intelligence on COVID-19 pneumonia detection using chest xray images. *PLoS One.* 2021;16(10):e0257884.
- [6] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Comput Biol Med.* 2020; 121:103795.
- [7] Hemanth RJMGSTDJ. Deep learning-based detection and analysis of COVID-19 on chest X-ray images. 2020.
- [8] J. de Moura, J. Novo, M. Ortega. Fully automatic deep convolutional approaches for the analysis of COVID-19 using chest X-ray images. *Appl Soft Comput.* 2022; 115:108190.
- [9] A. Abdelgawad and F.Ha. Khandaker. A Deep Learning Approach to Detect COVID-19 Patients from Chest X-ray Images †. 2020.