



รายงานสหกิจศึกษาฉบับสมบูรณ์

การประยุกต์ใช้ปัญญาประดิษฐ์กับภาพทางการแพทย์

Artificial Intelligence For Medical Images

นายอามีน กุลิสถาน

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา การประยุกต์ใช้ปัญญาประดิษฐ์กับภาพทางการแพทย์

ชื่อ-สกุล นักศึกษา นายอามีน กุลิสทาน

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร.พนารัตน์ เชิญถนอมวงศ์

ชื่อ-สกุล ผู้นิเทศงาน นายวลิตะ นาคบัวแก้ว

สถานประกอบการ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

บทคัดย่อ

งานวิจัยนี้มีจุดประสงค์หลักในการนำปัญญาประดิษฐ์มาประยุกต์ใช้กับภาพทางการแพทย์มีเป้าหมายสำหรับการพัฒนาปัญญาประดิษฐ์จากรูปรูปที่ได้จากการสแกนทั้งสองรูปแบบมาใช้ในการวินิจฉัยพร้อมๆ กันซึ่งจะทำให้การวินิจฉัยมีประสิทธิภาพและรวดเร็วมากยิ่งขึ้น โดยงานนี้จะมุ่งเน้นในการตรวจหาความผิดปกติที่เกิดขึ้นบริเวณทรวงอกโดยเฉพาะ การดำเนินงานทั้งหมดจะแบ่งออกเป็น 3 ส่วนหลักๆ คือส่วนของการพัฒนาปัญญาประดิษฐ์เพื่อใช้ในการวิเคราะห์หาความผิดปกติบริเวณทรวงอกจากภาพสแกนเอกซเรย์ 2 มิติ ส่วนของการใช้ปัญญาประดิษฐ์ในการตรวจหาเนื้องอกภายในปอดโดยเฉพาะจากภาพ CT-scan 3 มิติ และส่วนสุดท้ายคือการพัฒนาตัวอย่างซอฟต์แวร์ตลอดกระบวนการตั้งแต่การลงทะเบียนของผู้ป่วยเพื่อยืนยันว่าผู้ป่วยได้ยอมรับเงื่อนไขในการนำรูปภาพทางการแพทย์มาประมวลผล โดยในส่วนนี้มีขึ้นเพื่อให้การดำเนินงานเป็นไปตามหลักยอมรับที่เป็นสากล และอีกส่วนสำหรับซอฟต์แวร์คือ การนำเอาตัวปัญญาประดิษฐ์สำหรับภาพเอกซเรย์มาประมวลผลบนคลาวด์เซิร์ฟเวอร์เพื่อให้การใช้งานง่ายยิ่งขึ้น

คำสำคัญ: การแยกส่วน, การจัดกลุ่ม, โครงข่ายประสาทแบบคอนโวลูชัน, เนื้องอกในปอด, ความผิดปกติบริเวณทรวงอก

Cooperative Title: Artificial Intelligence For Medical Images

Student intern name: Mr.Ameen Gulistan

Faculty: Engineering

Department: Computer Engineering

Advisor name: Asst.Prof.Dr.Panarat Cherntanomwong

Mentor name: Mr.Walita Nakbuakaew

Company: National Science and Technology Development Agency

ABSTRACT

This research work proposes the application of Artificial Intelligence (AI) to diagnose the abnormal thorax through 2D and 3D scan pictures. This makes diagnose faster and more effective. The work is divided into three parts. The first part is developing the AI application to predict an abnormality from 2D X-ray images. The second part is using AI to segment tumor inside the lung through 3D CT-scan (computed tomography) images. The last part is developing the software to support throughout the process. It begins with the registration from patient to fill the consent form permitting researcher to use their X-ray images for diagnosis. Moreover, another feature at the software is using AI to help diagnose the X-ray images.

Keywords: segmentation, classification, Convolution Neural Network, lung tumors, lung diseases

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยการให้ความช่วยเหลือแนะนำของ ผู้ช่วยศาสตราจารย์ พนา รัตน์ เขิญถนอมวงศ์ ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ และ นายวิริยะ นาคบัวแก้วผู้ที่เป็นที่ปรึกษาในขณะ ขณะปฏิบัติสหกิจศึกษาในห้องปฏิบัติการที่ได้กรุณาที่ให้คำแนะนำข้อคิดเห็นตรวจสอบ และแก้ไขร่าง วิทยานิพนธ์ฉบับนี้ให้ถูกต้องและสมบูรณ์มาโดยตลอด ผู้เขียนจึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้

สุดท้ายนี้ ผู้วิจัยขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจผู้วิจัยเสมอมาจนสำเร็จการศึกษา

อามีน กุลิสถาน

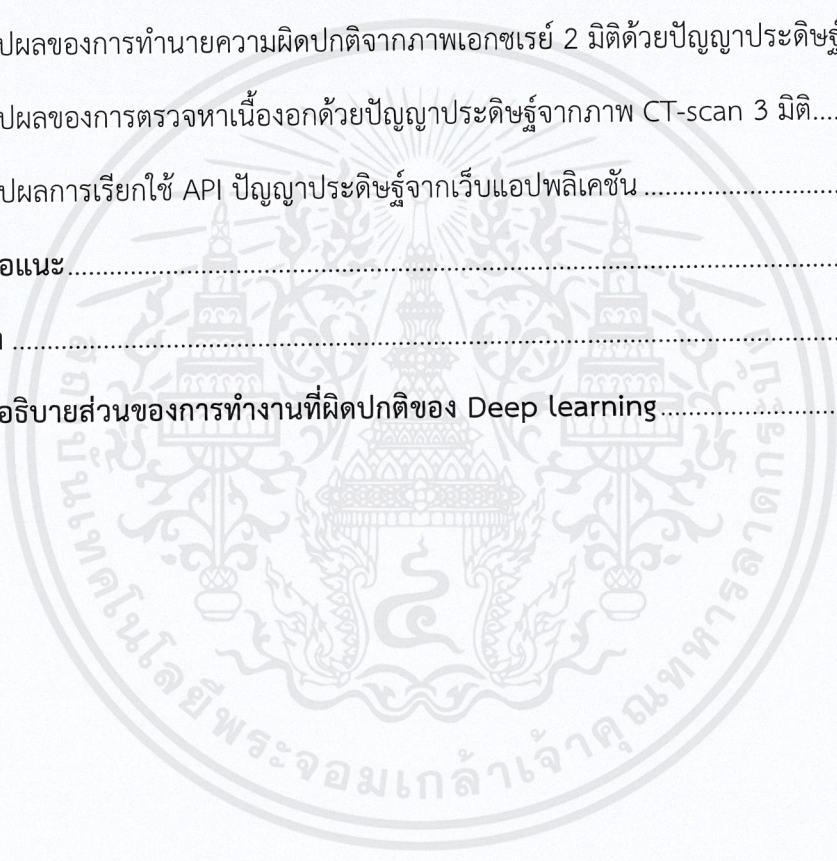


สารบัญ

บทคัดย่อ.....	I
ABSTRACT	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูปภาพ.....	VII
สารบัญตาราง.....	IX
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.1.1 บทบาทของปัญญาประดิษฐ์กับรูปภาพทางการแพทย์.....	1
1.1.2 ความสำคัญของ General Data Protection Regulation กับข้อมูลทางการแพทย์.....	4
1.2 วัตถุประสงค์ของการวิจัย.....	5
1.3 ขอบเขตของการวิจัย.....	5
1.3.1 ขอบเขตของการประยุกต์ใช้ปัญญาประดิษฐ์กับภาพทางการแพทย์.....	5
1.3.2 ขอบเขตสำหรับการพัฒนาตัวอย่างซอฟต์แวร์สำหรับการใช้งานปัญญาประดิษฐ์.....	5
1.4 ขั้นตอนการดำเนินงานวิจัย.....	6
1.5 ประโยชน์ที่ได้รับ.....	7
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	8
2.1 ทฤษฎีที่เกี่ยวข้องกับการเรียนรู้ด้วยเครื่อง.....	8
2.1.1 การเรียนรู้ด้วยเครื่อง (Machine Learning).....	8
2.1.2 การเรียนรู้เชิงลึก (Deep learning).....	10

2.1.3 segmentation.....	13
2.1.4 โมเดล U-Net [3].....	15
2.1.5 เทคนิค transfer learning.....	17
2.1.6 Hyper-parameter tuning.....	17
2.1.7 การพิจารณา early-stop.....	18
2.1.8 ค่า loss ของการประยุกต์ใช้ปัญญาประดิษฐ์กับภาพเอกซเรย์ 2 มิติ	19
2.2 ทฤษฎีในการพัฒนา desktop application.....	19
2.2.1 การทำงานของ face detection ด้วยอัลกอริทึม haar-like	19
2.2.2 Java Native Interface (JNI).....	21
2.2.3 การทำงานของ Django framework.....	22
2.2.4 Image alignment	23
2.2.5 Cross-Origin Resource Sharing (CORS).....	24
2.2.6 การใช้งานฟังก์ชันแฮชในการเก็บรหัสผ่าน.....	24
บทที่ 3 วิธีการดำเนินการ	27
3.1 การดำเนินการส่วนของปัญญาประดิษฐ์.....	27
3.1.1 การทำความเข้าใจข้อมูลที่ใช้ในการทำงาน	27
3.1.2 ข้อมูลในส่วนของ segmentation.....	31
3.1.3 ศึกษาการใช้งานของเครื่องมือ	32
3.1.4 การศึกษาโมเดลที่มีความเหมาะสมกับการทำงานแต่ละส่วน	35
3.2 การดำเนินงานในส่วน of ซอฟต์แวร์ลงทะเบียนและใช้งานโมเดล	38
3.2.1 ผังการทำงานของโปรแกรมสำหรับจัดการไฟล์	38
บทที่ 4 ผลลัพธ์การทำงาน	41
4.1 ผลลัพธ์ของการเรียนรู้ด้วยเครื่องกับการแยกส่วน.....	41

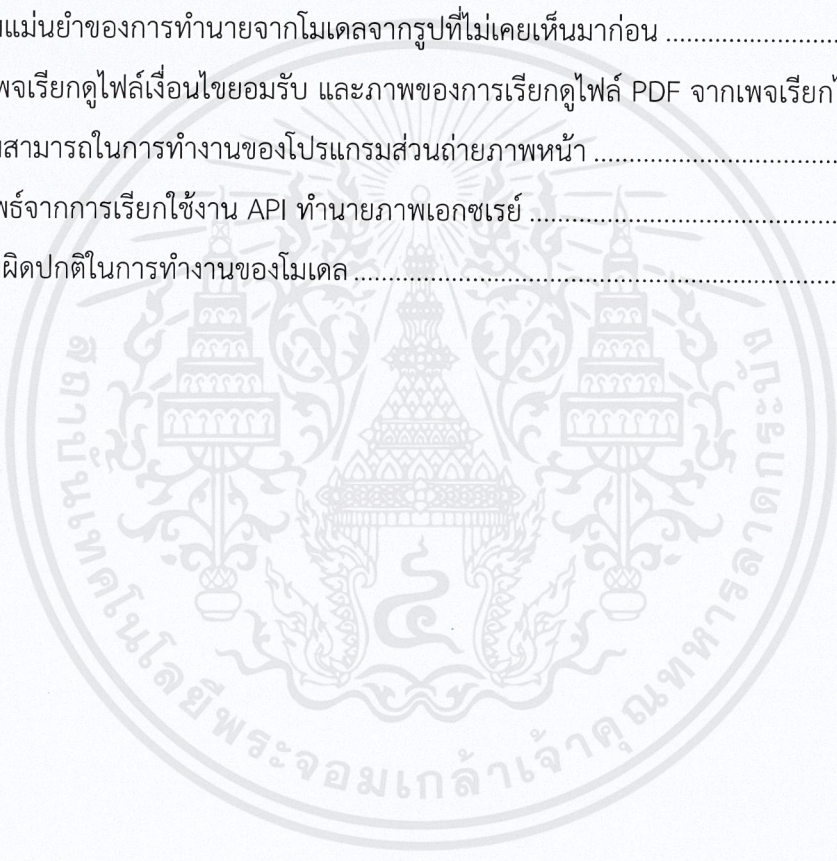
4.2 ผลลัพธ์ของการทำงานในการจำแนกประเภทของอาการในปอด จากภาพเอกซเรย์ 2 มิติ	43
4.3 ผลลัพธ์ของการใช้งานโปรแกรมสร้างไฟล์เงื่อนไขยอมรับ	47
4.4 ผลลัพธ์การทำงานของตัวอย่างเว็บเพจเรียกใช้งานโมเดล DenseNet	48
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	49
5.1 สรุปผลการวิจัย.....	49
5.1.1 สรุปผลการทำงานของซอฟต์แวร์ลงทะเบียน.....	49
5.1.2 สรุปผลของการทำนายความผิดปกติจากภาพเอกซเรย์ 2 มิติด้วยปัญญาประดิษฐ์.....	49
5.1.3 สรุปผลของการตรวจหาเนื้องอกด้วยปัญญาประดิษฐ์จากภาพ CT-scan 3 มิติ.....	49
5.1.4 สรุปผลการเรียกใช้ API ปัญญาประดิษฐ์จากเว็บแอปพลิเคชัน	49
5.2 ข้อเสนอแนะ.....	49
เอกสารอ้างอิง	50
ภาคผนวก ก อธิบายส่วนของการทำงานที่ผิดปกติของ Deep learning.....	53



สารบัญรูปภาพ

รูปที่ 1 การเรียนรู้ในแต่ละชั้นของ Deep learning	2
รูปที่ 2 ฝั่งเวลาทำงานส่วนของการประยุกต์ใช้ปัญญาประดิษฐ์	6
รูปที่ 3 ฝั่งเวลาทำงานส่วนของ consent form application	6
รูปที่ 4 ประเภทของ Machine Learning.....	9
รูปที่ 5 กราฟของ sigmoid function	11
รูปที่ 6 กราฟของ ReLu function	11
รูปที่ 7 กราฟของ Leaky ReLu.....	12
รูปที่ 8 ลักษณะของ deep neural network	12
รูปที่ 9 การ convolution.....	13
รูปที่ 10 ความแตกต่างของการ segment ทั้ง 2 แบบ.....	14
รูปที่ 11 โมเดล U-Net	16
รูปที่ 12 การเรียนรู้แบบ boosting.....	20
รูปที่ 13 Haar-features แบบต่าง ๆ	20
รูปที่ 14 การทำงานของ JNI.....	22
รูปที่ 15 โครงสร้าง MTV	23
รูปที่ 16 homography	24
รูปที่ 17 วงจรของ MD5	25
รูปที่ 18 หลักการทำงานของ การเข้ารหัสแบบ PKBDF2	26
รูปที่ 19 ฝั่งการดำเนินงานคัดเลือกข้อมูล.....	28
รูปที่ 20 กราฟของการเปรียบเทียบอัตราส่วนของคุณค่าข้อมูล.....	30
รูปที่ 21 ลักษณะของชุดข้อมูล medical decathlon 3 มิติ	31
รูปที่ 22 การทำงาน Estimator.....	34
รูปที่ 23 Densely Connected Convolutional Networks	35
รูปที่ 24 การทำงานโดยรวมของโปรแกรมให้เกิดการเรียนรู้ด้วยเครื่อง.....	36
รูปที่ 25 ฝั่งโปรแกรมการเรียนรู้ด้วยเครื่องกับภาพ 3 มิติ.....	37
รูปที่ 26 ลำดับการทำงานของซอฟต์แวร์ลงทะเลเบียน	39

รูปที่ 27	ผังการทำงานของข้อมูลที่มีการเข้ารหัส.....	40
รูปที่ 28	รูปภาพผลลัพธ์จากการแบ่งส่วน	41
รูปที่ 29	กราฟ dice score ของการแยกส่วนภาพ 3 มิติในแต่ละ step	42
รูปที่ 30	ภาพของ convolution activation map.....	43
รูปที่ 32	ความแม่นยำการทำนายในแต่ละ epoch	44
รูปที่ 33	ค่า loss ที่ได้จากการทำนายแต่ละ epoch.....	45
รูปที่ 34	ความแม่นยำจากการทำนายโดยรวม	46
รูปที่ 35	ความแม่นยำของการทำนายจากโมเดลจากรูปที่ไม่เคยเห็นมาก่อน	46
รูปที่ 36	เว็บเพจเรียกดูไฟล์เงื่อนไขยอมรับ และภาพของการเรียกดูไฟล์ PDF จากเพจเรียกไฟล์.....	47
รูปที่ 37	ความสามารถในการทำงานของโปรแกรมส่วนถ่ายภาพหน้า	48
รูปที่ 38	ผลลัพธ์จากการเรียกใช้งาน API ทำนายภาพเอกซเรย์	48
รูปที่ 39	ความผิดปกติในการทำงานของโมเดล	53



สารบัญตาราง

ตารางที่ 1	หมายเลขคลาส, ชื่อคลาส และความหมาย.....	28
ตารางที่ 2	จำนวนข้อมูลที่ใช้และ อัตราส่วนต่อข้อมูลทั้งหมด.....	29
ตารางที่ 3	จำนวนและสัมประสิทธิ์ของข้อมูลที่ใช้ฝึกฝน.....	29
ตารางที่ 4	จำนวนและสัมประสิทธิ์ของข้อมูลที่ใช้ในการทดสอบ.....	30



บทที่ 1

บทนำ

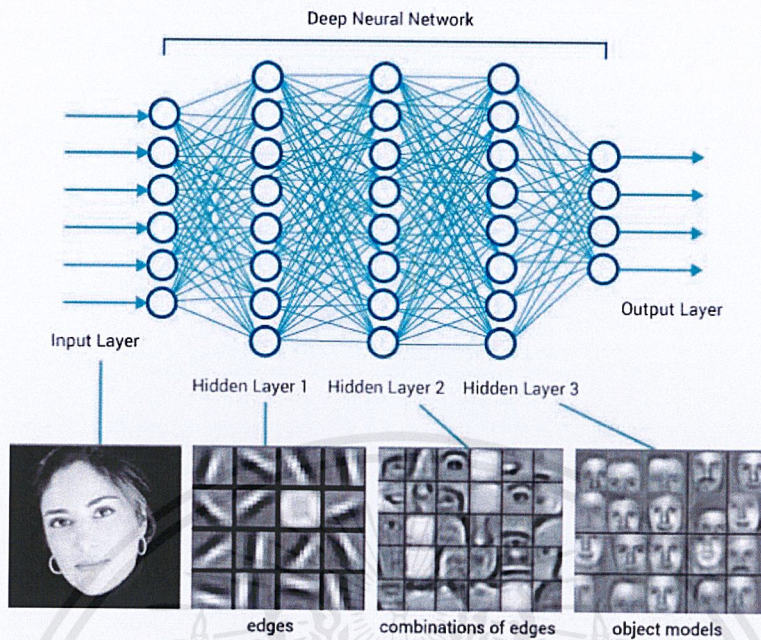
1.1 ที่มาและความสำคัญ

1.1.1 บทบาทของปัญญาประดิษฐ์กับรูปภาพทางการแพทย์

เนื่องจากในปัจจุบันการใช้งานของปัญญาประดิษฐ์เป็นที่นิยมอย่างมากในวงการแพทย์ ทั้งนี้การทำงานของปัญญาประดิษฐ์นั้นมีหลากหลายรูปแบบมาก โดยรูปแบบที่มีประโยชน์ต่อการใช้งานมากที่สุด คือ การนำภาพทางการแพทย์มาใช้เพื่อตรวจหาความผิดปกติที่อยู่ภายในรูปสแกนที่เกิดจากผู้ป่วย ผู้จัดทำจึงได้ศึกษางานวิจัยก่อนหน้าที่ได้มีการใช้งานปัญญาประดิษฐ์กับภาพสแกนทางการแพทย์ มีดังนี้

- CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, Andrew Y. Ng [5]

CheXNet เป็นงานวิจัยที่มุ่งเน้นในการค้นหาผู้ป่วยที่เป็นโรคปอดบวม โดยมีวิธีการออกแบบตัวโมเดล CheXNet ที่ให้ระดับความแม่นยำมากกว่ารังสีแพทย์ด้วยการแข่งขันระหว่างโมเดลและนักรังสีจำนวน 3 คนโครงสร้างของโมเดลเป็น CNN ที่มี 121 layers สำหรับจำนวนของ layers เป็นขั้นของการเรียนรู้ใน deep learning model คือยังมี layers มากก็ยังมีโอกาสที่โมเดลจะเห็นส่วนสำคัญมากกว่าทั้งนี้จำนวนของ layers ที่มาก ๆ ก็มีทั้งผลดีและผลเสีย เนื่องจากถ้ามันมีจำนวนมากเกินไปมันอาจจะเรียนรู้อะไรที่แปลก ๆ ไปได้ เช่นในงานวิจัยของ google Deepmind [6] ได้แสดงให้เห็นว่าโมเดลที่มีการเรียนรู้โดยการปล่อยผ่าน weak feature เมื่อมันถูกรูปที่มีการแทรก noise เข้าไป (ตัวอย่างของภาพในงานวิจัยนี้คือรูปหมูที่ถูกเพิ่ม noise เข้าไปซึ่งตามมนุษย์ยังรับรู้ว่าเป็นรูปของหมูเช่นเดิม) มันกลับมองเห็นเป็นรูปเครื่องบิน เช่น การจัดกลุ่มรูปสุนัขและแมว ในขั้นแรกอาจจะมองเห็นแค่เส้นหรือขอบภาพ ในขั้นถัดมาอาจจะมีการเรียนรู้ที่จะจับภาพของเงาความสูงต่ำที่ต่างกันให้เกิดเป็นรูปจุกสุนัขปากแมว เป็นต้น ในขั้นสุดท้ายอาจจะเรียนรู้ว่านี่คือสุนัขหรือแมวโดยความแตกต่างของการแยกส่วนในการเรียนรู้ของขั้นก่อน ๆ หน้า โดยในการฝึกโมเดล CheXNet จะใช้ชุดข้อมูลจำนวน 100,000 รูปภาพในมุมมอง frontal view สำหรับการฝึก โดยงานนี้เป็นงานวิจัยที่มีความมุ่งเน้นในการหาปอดบวม โดยมีเทคนิคในการพลอต heatmap [7] สำหรับแสดงส่วนที่เป็นจุดสนใจของตัวโมเดลเองในรอบการฝึก ทำให้ผู้วิจัยได้รู้พฤติกรรมของโมเดล



รูปที่ 1 การเรียนรู้ในแต่ละชั้นของ Deep learning [8]

- Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists [9]

CheXNeXt เป็นงานวิจัยที่เกี่ยวกับการตรวจจับโรคที่ผิดปกติบริเวณทรวงอกด้วยรูปภาพเอกซเรย์ รวมไปถึง วัณโรค และมะเร็งปอด การวิจัยนี้มีขึ้นเพื่อทำให้ในพื้นที่ที่ไม่มีรังสีแพทย์สามารถใช้ตัวโมเดลนี้จัดการตรวจหาความผิดปกติในบริเวณทรวงอกได้ และมุ่งเน้นวิธีการวัดผลทางสถิติมากกว่าจะมุ่งเน้นไปในเรื่องเทคนิคที่จะทำให้โมเดลทำนายผลได้อย่างแม่นยำ ในการพัฒนาจะใช้รูปภาพในด้าน frontal-view วิเคราะห์หาความผิดปกติ การวัดผลด้วย AUC (area under the receiver operating characteristic curve) ตัวโมเดลสามารถแยกแยะโรคได้ในระดับของรังสีแพทย์ที่ 11 โรค โดยการวิเคราะห์โรคปอดแพบเป็นส่วนที่ให้ความแม่นยำมากกว่ารังสีแพทย์ (model AUC 0.862, รังสีแพทย์ที่ 0.808) และเวลาในการแยกแยะของรังสีแพทย์ใช้ประมาณ 4 ชั่วโมงในขณะที่ตัวโมเดลใช้เวลาเพียง 1 นาทีครึ่ง ในงานวิจัยนี้เป็นงานวิจัยที่มุ่งเน้นในการหาโรคที่พิจารณาได้ยาก เทคนิคที่สำคัญคือการ annotation ของข้อมูลซึ่งทำให้ผลลัพธ์สุดท้ายขึ้นไปอยู่ในระดับที่สามารถใช้งานได้จริง

- Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification [10]

เทคนิค transfer learning เป็นการใช้งานของโมเดลที่ได้มีการฝึกไว้แล้วนำมาเปลี่ยนตัว header เพื่อให้จำนวน outputs ออกมาเท่ากับที่เราต้องการเทคนิคนี้เป็นสิ่งที่ทำให้เราสามารถพัฒนาตัวโมเดลให้มีความแม่นยำได้สูงมากและทำให้ประหยัดเวลา

งานวิจัยนี้มีการใช้เทคนิค transfer learning ในการทำให้ประสิทธิภาพของการจำแนกอาการจากชุดข้อมูลเพิ่มมากขึ้นอย่างมาก ใช้เทคนิค transfer learning แบบ feature extraction ซึ่งเป็นวิธีที่ได้ผลดีและประหยัดเวลามากที่สุดเนื่องจากการ implement ไม่จำเป็นที่จะต้องทำการเปลี่ยนแปลงค่าถ่วงน้ำหนักของทั้งโมเดล ในรายงานวิจัยนี้ได้ฝึกบนโมเดล CNN Resnet-50 ในชุดข้อมูลมีการแปะป้ายประเภทของอาการทั้งหมด 14 อาการไม่ต่างจาก CheXNet สำหรับงานวิจัยนี้เป็นงานวิจัยที่มีความน่าสนใจตรงที่ตัวโมเดลที่มีการฝึกแบบสำเร็จไว้แล้วกับรูปภาพทางการแพทย์หาได้ยากโดยเทคนิค transfer learning จริง ๆ แล้วสามารถที่จะให้มันใช้งานข้ามกันได้ระหว่างรูปภาพ grayscale และภาพ RGB ได้ตัวอย่างเช่นใน Pytorch จะมีฟังก์ชัน copy ที่ใช้จัดการในเรื่อง input channels ไม่ตรงกับ pre-trained model ได้

สำหรับเทคนิค transfer learning เป็นเทคนิคที่เป็นที่นิยมอย่างมากสำหรับผู้ที่มีทรัพยากรไม่เพียงพอหรือมีชุดข้อมูลเล็กน้อยไปจะทำให้โมเดลรูปแบบ deep learning สามารถเรียนรู้ได้อย่างมีประสิทธิภาพการทำงานในส่วนนี้คือการแปลง layer สุดท้ายของโมเดลให้ output เท่ากันกับจำนวนของประเภทข้อมูลที่เราจำเป็นต้องทำการแยก

- CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison

CheXpert เป็นชุดข้อมูลที่รวบรวมภาพเอกซเรย์ทรวงอกไว้ถึง 224,316 รูปภาพจากผู้ป่วยจำนวน 65,240 คนทำการ labeled ที่ 14 อาการโดยมีจุดประสงค์เพื่อใช้ข้อมูลชุดนี้ในการเปรียบเทียบแข่งขันระหว่างโมเดลที่ดีที่สุดของผู้วิจัยกับรังสีแพทย์ 3 คน กระบวนการใช้งานจริง ๆ คือการทดลองใช้กับโมเดลต่าง ๆ คือ ResNet152, DenseNet121, Inception-v4, and SEResNeXt101 และพบว่า DenseNet คือตัวที่ให้ผลลัพธ์ได้ดีมากที่สุดด้วยการ classify ที่ 3 class คือ negative, positive และ uncertain มีเทคนิคในการจัดการกับชุดข้อมูลที่โมเดลไม่สามารถวิเคราะห์ที่ได้แบบแน่นอนด้วยการแบ่งเป็น u-labels

- Lung Nodules Detection and Segmentation Using 3D Mask-RCNN [11]

งานวิจัยนี้ใช้ model Mask-RCNN ในการทำ segmentation ภาพมะเร็งปอดด้วยความที่งานนี้ใช้โมเดลที่ไม่ได้มีความจำเพาะกับภาพทางการแพทย์ทำให้เป็นงานวิจัยที่น่าสนใจมากกว่ามีการจัดการในเรื่องของ class imbalance ได้อย่างไร ซึ่งก็ได้แสดงให้เห็นแล้วว่าสามารถจัดการกับปัญหานี้ได้ด้วยการอาศัย focal loss ในการทำให้เกิดการเรียนรู้ซึ่งการทำงานจัดทำบนชุดข้อมูล LUNA16 ซึ่งเป็นชุดข้อมูลที่มีการปล่อยให้โหดโดยสาธารณะ และงานวิจัยนี้มีขึ้นเพื่อเปรียบเทียบการใช้งานระหว่างตัวโมเดล Mask-RCNN กับรังสีแพทย์ และทำการเปรียบเทียบกับโมเดล 2D-RCNN + 3DCNN

จากงานวิจัยที่กล่าวมาทั้งหมดจะเห็นได้ว่าในปัจจุบันมีการใช้งานปัญญาประดิษฐ์กับทางการแพทย์อยู่มากทั้งนี้ผู้จัดทำได้ตั้งใจที่จะยกตัวอย่างภาพสแกนบริเวณปอดซึ่งผู้จัดทำมีความสนใจเป็นพิเศษและแน่นอนว่าผู้จัดทำได้สังเกตเห็นว่าการตรวจสอบความผิดปกติที่เกิดขึ้นบริเวณปอดนั้นสามารถตรวจสอบได้จาก 2 อย่างหลักด้วยกัน คือ การทำ CT-scan และการ X-ray บริเวณทรวงอก ซึ่งการ X-ray เป็นวิธีการที่สะดวกและรวดเร็วซึ่งใช้เป็นวิธีในการตรวจสอบร่างกายในระดับที่ไม่มีความจำเป็นที่จะต้องการการวิเคราะห์ที่สูงมากนัก

1.1.2 ความสำคัญของ General Data Protection Regulation (GDPR) กับข้อมูลทางการแพทย์

ประเทศไทยมีข้อกำหนดสำหรับข้อมูลผู้ป่วยที่มีการประกาศอย่างชัดเจนในประเทศจะมีในลักษณะตามที่อธิบาย ดังนี้ ข้อมูลที่มีการนำไปใช้งานในการประมวลผลจะต้องได้รับความยินยอมจากเจ้าของข้อมูลก่อน โดยผู้ประมวลผลข้อมูลหรือผู้ใช้งานข้อมูลต้องจัดเก็บหรือใช้งานภายใต้เงื่อนไขที่ทำการตกลงเอาไว้เท่านั้น ในการจัดเก็บข้อมูลต้องมีการปกปิดข้อมูลตามหลักการที่เป็นที่ยอมรับ ในกรณีที่เจ้าของข้อมูลไม่สามารถดูแลตนเองได้ไม่ว่าจะเป็นผู้ที่ยังไม่บรรลุนิติภาวะ ข้อตกลงทั้งหมดสามารถให้ผู้พิทักษ์จัดการแทนได้ เมื่อการจัดเก็บข้อมูลของประมวลผลข้อมูลต้องการจัดเก็บผ่านตัวกลางซึ่งเป็นสินค้าขององค์กรอื่น ผู้ให้บริการในการจัดเก็บข้อมูลต้องมีการรับรู้ และทำตามเงื่อนไขที่เป็นที่ยอมรับทั้งหมด

ดังที่กล่าวมาทั้งหมดการทำงานของปัญญาประดิษฐ์กับข้อมูลทางการแพทย์เป็นงานที่มีประโยชน์อย่างแพร่หลายและมีความน่าสนใจอย่างมาก ไม่ว่าจะเป็นการเพิ่มประสิทธิภาพของข้อมูลด้วย super-resolution [12] หรือการตรวจหาความผิดปกติด้วยการ detection แบบทั่วไป อย่างไรก็ตามแต่เมื่อเป็นการทำงานที่ต้องมีการนำข้อมูลของผู้ป่วยมาใช้ร่วมด้วยก็ต้องทำให้ถูกต้องตามเงื่อนไขของกฎหมายสากล ดังนั้นในงานนี้จึงต้องมีการจัดสร้างระบบที่รองรับส่วนของเอกสารหลักฐานด้วยควบคู่ไปกับการทำงานของการใช้งานปัญญาประดิษฐ์

1.2 วัตถุประสงค์ของการวิจัย

1. การศึกษาแนวทางปัญญาประดิษฐ์กับภาพปอด 2 มิติจากเครื่อง DR และการศึกษาแนวทางการใช้ปัญญาประดิษฐ์กับภาพถ่าย 3 มิติจากเครื่อง CT scan
2. ทำการศึกษาและออกแบบพัฒนาซอฟต์แวร์เพื่อเก็บข้อมูลส่วนบุคคลให้เป็นไปตามหลักสากล

1.3 ขอบเขตของการวิจัย

1.3.1 ขอบเขตของการประยุกต์ใช้ปัญญาประดิษฐ์กับภาพทางการแพทย์

การศึกษาส่วนของการใช้งานปัญญาประดิษฐ์จะเน้นไปที่การทำงานของการเรียนรู้เชิงลึก (deep learning) ซึ่งมีความสามารถในการเรียนรู้มากกว่าอัลกอริทึมแบบทั่วไป การใช้งานจะเน้นไปที่การปรับปรุงให้ตัวโมเดลของการเรียนรู้เชิงลึกนี้สามารถที่จะทำนายผลออกมาได้อย่างมีประสิทธิภาพมากที่สุด เนื่องจากว่าในชุดข้อมูลที่นำมาใช้มีปัญหาที่ทำให้การเรียนรู้ไม่เป็นไปได้อย่างที่สมควรอยู่ ในการศึกษาวิจัยสำหรับส่วนของภาพ 3 มิตินี้ได้มีการนำชุดข้อมูลที่เป็นภาพของความผิดปกติในบริเวณทรวงอกมาใช้งานเป็นชุดข้อมูลที่ได้ปล่อยให้ใช้งานได้โดยไม่ต้องมีคำขออนุญาตจากโครงการ MedicalDecathlon โดยจำนวนของชุดข้อมูลนี้มีจำนวน 63 ไฟล์สำหรับการฝึกฝน และ 10 ไฟล์สำหรับทดสอบสาเหตุที่ตัดสินใจนำข้อมูลชุดนี้มาใช้งานเนื่องจากว่าเป็นไฟล์ข้อมูลที่ไม่ได้มีขนาดใหญ่มากและชุดข้อมูลนี้ได้มีการจัดทำสัญลักษณ์ให้อยู่ในลักษณะเดียวกันกับข้อมูลสำหรับการทดสอบ ในการศึกษาส่วนของภาพเอกซเรย์ 2 มิติได้เลือกนำชุดข้อมูล Chexpert dataset[2] มาใช้เนื่องจากเป็นชุดข้อมูลที่มีการทำ labels ไว้โดยละเอียดกว่าชุดข้อมูลภาพเอกซเรย์ทรวงอกชุดอื่น ๆ เช่น ส่วนที่ไม่แน่ใจว่าเป็นอาการใดกันแน่จะมีการติด labels ในกับข้อมูลแบบนี้ให้ด้วย

1.3.2 ขอบเขตสำหรับการพัฒนาตัวอย่างซอฟต์แวร์สำหรับการใช้งานปัญญาประดิษฐ์

การพัฒนาซอฟต์แวร์มีภาพรวมคือเริ่มจากการลงทะเบียนผู้ป่วยด้วยซอฟต์แวร์ลงทะเบียนเพื่อเก็บข้อมูลลงในการใช้งานรูปภาพสแกนของผู้ป่วย เมื่อการลงทะเบียนสำเร็จจะมีการเก็บข้อมูลของผู้ป่วยลงในฐานข้อมูลในลักษณะที่เป็นการยอมรับตามหลักสากล ในส่วนถัดมาจะเป็นการนำข้อมูลรูปที่ได้ของผู้ป่วยมาใช้ในการประมวลผล และส่วนสุดท้าย คือเว็บไซต์สำหรับใช้ Application Programming Interface (API) ของตัวโมเดลปัญญาประดิษฐ์กับรูปที่ได้จากผู้ป่วยแล้วส่งผลลัพธ์การทำนายที่ได้ออกมาแสดงผลผ่านเว็บเบราว์เซอร์

1.4 ขั้นตอนการดำเนินงานวิจัย

งานวิจัยนี้มีขั้นตอนการดำเนินงานจะแยกออกเป็นสองส่วนหลัก ๆ คือ

- 1) การพัฒนาส่วนของโมเดลการเรียนรู้เชิงลึกสำหรับการทำนายความผิดปกติจากรูปภาพการสแกน

การดำเนินงาน	ระยะเวลาในการทำงาน
สืบค้นข้อมูลที่เกี่ยวข้อง	1 สิงหาคม – 31 ตุลาคม 2019
ศึกษาเครื่องมือที่ใช้ในการทำงาน	16 สิงหาคม – 1 กันยายน 2019
ศึกษาข้อมูลและวิเคราะห์	2 – 8 กันยายน 2019
จัดทำส่วนของโมเดล	9 – 23 กันยายน 2019
ทำการปรับค่าให้ตัวโมเดล	24 – 30 กันยายน 2019
เปรียบเทียบผลของโมเดล	3 ตุลาคม – 24 ตุลาคม 2019
พัฒนาเว็บแอปพลิเคชัน	25 ตุลาคม – 4 พฤศจิกายน 2019
ทดสอบการใช้งานเว็บไซต์	5 – 12 พฤศจิกายน 2019
จัดทำรายงาน	12 – 22 พฤศจิกายน 2019

รูปที่ 2 ฝั่งเวลาทำงานส่วนของการประยุกต์ใช้ปัญญาประดิษฐ์

- 2) การพัฒนา desktop application สำหรับการสร้างไฟล์คำยินยอมให้ใช้ข้อมูล

การดำเนินงาน	ระยะเวลาในการทำงาน
กำหนดและออกแบบการทำงานของโปรแกรม	10-17 กรกฎาคม 2019
ศึกษาเครื่องมือที่ใช้	12-17 กรกฎาคม 2019
พัฒนาโปรแกรม	18-26 กรกฎาคม 2019
พัฒนา desktop application ลงเว็บไซต์	28 กรกฎาคม-10 สิงหาคม 2019
ทดสอบการใช้งาน	27 กรกฎาคม-1 สิงหาคม 2019
พัฒนาเว็บไซต์สำหรับการเรียกดูไฟล์	28 กรกฎาคม-10 สิงหาคม 2019
ทดสอบการใช้งานทั้งหมดของซอฟต์แวร์	10-16 สิงหาคม 2019

รูปที่ 3 ฝั่งเวลาทำงานส่วนของ consent form application

1.5 ประโยชน์ที่ได้รับ

การทำงานในส่วนนี้ทำให้มีโอกาสได้ทดลองใช้งาน deep learning กับไฟล์รูปที่คนทั่วไปจะไม่ได้คุ้นเคยมากนัก ทำให้ได้ประสบการณ์ในการทำความเข้าใจข้อมูลมากยิ่งขึ้น และทำให้รู้ถึงจุดที่ขาดความเข้าใจที่มากพอในบางส่วนซึ่งทำงานล่าช้า



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

การทำงานทั้งหมดจะมีทฤษฎีที่เกี่ยวข้องเป็นสามส่วนหลัก ๆ โดยผู้เขียนจะไล่ตามลำดับลงไป ตั้งแต่รายละเอียดของการเรียนรู้ด้วยเครื่องไปตลอดจนโมเดลที่ใช้ในการทำงานในครั้งนี้อย่างไรและวิธีการปรับแต่งเทคนิคต่าง ๆ ในการทำให้โมเดลสามารถเรียนรู้ได้มากยิ่งขึ้นจนมีประสิทธิภาพที่ดีขึ้นกว่าเดิม โดยจะอธิบายออกเป็นสามส่วน คือ ส่วนของการพัฒนาโมเดลสำหรับภาพ 2 มิติ ส่วนการพัฒนาโมเดลกับภาพ 3 มิติ และส่วนสุดท้ายคือการพัฒนาซอฟต์แวร์สำหรับการลงทะเบียนและเรียกใช้งานโมเดลมาทำนายผล

2.1 ทฤษฎีที่เกี่ยวข้องกับการเรียนรู้ด้วยเครื่อง

2.1.1 การเรียนรู้ด้วยเครื่อง (Machine Learning)

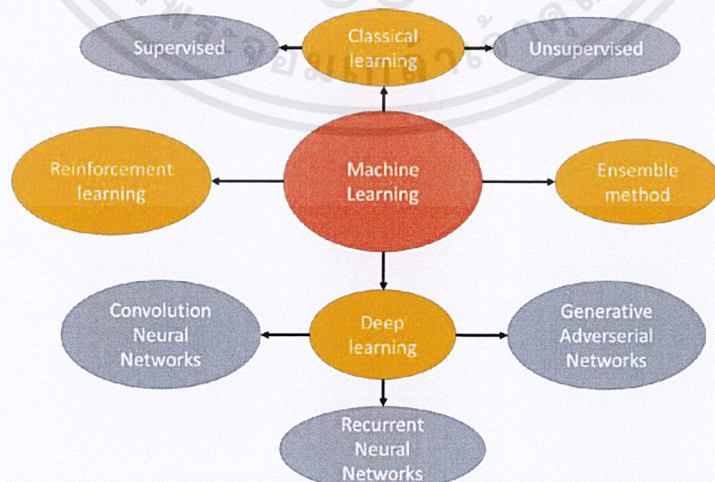
เริ่มตั้งแต่แรกการทำงานของเครื่องเรียนรู้ด้วยเครื่องมีรากฐานมาจาก สถิติ และความน่าจะเป็นอาจกล่าวอย่างง่ายได้ว่าเป็นการทำซ้ำ ๆ จากสิ่งเดิมเพื่อหวังให้เกิดสิ่งใหม่ก็ได้เหมือนกับพฤติกรรมของมนุษย์ เช่น การแยกแยะระหว่างบริบทของคำว่า “ปู” จากประโยค “ปูในห้องนอน” เราคงจะไม่นึกถึงปูที่มีกำลังเดินในห้องนอน แต่จะเข้าใจว่าเป็นการปูผ้าหรืออะไรในทำนองนี้ ซึ่งเราตระหนักถึงผลลัพธ์ในทำนองนี้ได้ก็เกิดมาจากการเรียนรู้ถึงคำที่มีโดยรอบคำนั้น คอมพิวเตอร์ก็เช่นกันเมื่อมันเจอคำว่า “ปู” , “ใน” , “ห้องนอน” ผลลัพธ์จากการทำนายถ้าเป็นเชิงสถานที่ก็ควรจะออกมาเป็นบ้านมากกว่าทุ่งนาด้วยการให้น้ำหนัก (weight) กับคำว่า “ห้องนอน” มากกว่า “ปู”

การเรียนรู้ด้วยเครื่องแล้วแบ่งได้เป็นสี่ประเภทหลัก ดังนี้

1. Classical learning เป็นการเรียนรู้ด้วยอัลกอริทึมที่ไม่เป็นที่นิยมมากนักในปัจจุบัน เช่น Support Vector Machine (SVM) หรือ Decision Trees เป็นต้น
2. Reinforcement Learning เป็นการเรียนรู้ที่มีคนรับรู้ถึงผลงานมากในระดับหนึ่ง เช่น openAI [13] ที่ใช้ในการเล่นเกม E-sport ด้วยการที่มันอาศัยการเรียนรู้ด้วยประสบการณ์ ทำให้มันไม่ได้เพียงแค่วางรูปแบบการทำงานของตัวมันเองเท่านั้นแต่มันยังคาดคะเนสิ่งที่ฝั่งตรงข้ามต้องการจะทำได้ใช้อัลกอริทึมในวิธีนี้ เช่น Genetic Algorithm

3. Ensemble method เป็นการรวมกลุ่มกันเรียนดังนั้นวิธีนี้จะพบคีย์เวิร์ดเป็นคำว่า learners อธิบายให้เห็นภาพ คือ ในห้องเรียนจะมีนักเรียนมากกว่า 1 เมื่อมีคำถามเข้ามาแล้วไม่มีเฉลย วิธีการตัดสินคำตอบอย่างง่ายที่สุดก็คือเอาคำตอบที่มีคนตอบมากที่สุดเป็นผลเฉลย ตัวอย่างของ อัลกอริทึมที่นิยม เช่น XGBoost [14]
4. Deep learning เป็นวิธีการในการเรียนรู้ที่ทำให้ตัวโมเดลมีความสามารถในการพิจารณาได้มาก แต่ก็ทำให้มันต้องการข้อมูลจำนวนมากในการฝึกฝนให้มันเรียนรู้ได้อย่างมีประสิทธิภาพด้วยความที่มันเป็นส่วนที่คนให้ความสนใจมากที่สุดในขณะนี้ทำให้การพัฒนาทำง่ายตัวอย่าง เช่น Convolution Neural Networks (CNN) ซึ่งเป็นตระกูลที่นิยมใช้กับงานจำพวก Computer Vision (CV) หรือ Recurrent Neural Network ซึ่งเป็นโมเดลที่นิยมใช้ในการประมวลผลภาษา เป็นหลัก

ดังที่กล่าวมาทั้งหมดเป็นแค่ส่วนที่ทำให้เครื่องได้เกิดการเรียนรู้เท่านั้นแต่ปัจจัยที่สำคัญที่ทำให้เครื่องเกิดการเรียนรู้ได้ที่แท้จริงจะมีเพียงแค่อัลกอริทึมไม่ได้ เปรียบเทียบกับตัวเราคือ ถ้ามีแต่ตัวไม่มีข้อมูล input เข้ามาที่ประสาททั้ง 5 เราก็จะไม่สามารถเรียนรู้อะไรได้เลย ข้อมูลจึงเป็นส่วนสำคัญอีกอย่างที่ทำให้โมเดลได้เกิดการเรียนรู้ที่มีประสิทธิภาพเป็นสาเหตุที่ทำให้การพยากรณ์สภาพอากาศในประเทศไทยญี่ปุ่น ถูกต้องมากกว่า 90 เปอร์เซ็นต์ในระยะเวลาสามชั่วโมงสาเหตุคือมีการติดตั้งเซนเซอร์รับ input ข้อมูลได้แทบจะครอบคลุมทั้งประเทศด้วยความที่ประเทศมีขนาดเล็กทำให้ไม่ใช่เรื่องยากนี่คือตัวอย่างที่ทำให้เห็น ความสำคัญของข้อมูลได้มากที่สุด



รูปที่ 4 ประเภทของ Machine Learning

2.1.2 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึกเป็นหนึ่งในวิธีการเรียนรู้ของเครื่อง ซึ่งมันมีต้นแบบแรกสุดมาจาก perceptron ในยุคเริ่มแรกเป็นการทำงานที่อาศัยการ operate ด้วย and gate ทั้งหมดด้วยการอาศัยชุดข้อมูลเหมือนกับที่มีใน MNIST คือเป็นการเรียนรู้ตัวเลขแต่ 0 ถึง 9 ซึ่งการทำงานทั้งหมดเป็นเพียงการทำงานของฮาร์ดแวร์เท่านั้นแตกต่างกับในปัจจุบันที่การเรียนรู้เกือบจะทั้งหมดเป็นการทำงานของซอฟต์แวร์ ในปี 2015 ได้เกิด framework สำหรับการพัฒนา deep learning model การทำงานของ deep learning เป็นการทำงานโดยได้ต้นแบบมาจากเซลล์ประสาทของมนุษย์โดยการที่มันจะใช้ node หรือ neuron ภายในเก็บค่าถ่วงน้ำหนัก (weight) สำหรับการเรียนรู้แต่ละ layers สมการสำหรับการเรียนรู้ที่มีความสัมพันธ์อย่างง่ายที่สุดก็คือ สมการเส้นตรง (linear function) ซึ่งเป็นฟังก์ชันทางเดียว (monotonic function)

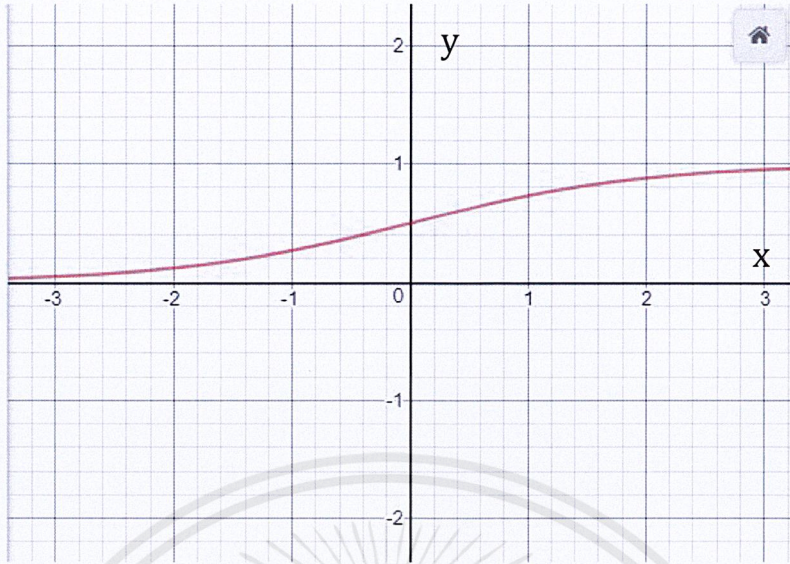
$$\text{output} = \text{weight} * \text{input} + \text{bias}$$

จากสมการเส้นตรงด้านบนจะเห็นได้ว่าการเรียนรู้ทำให้ค่า weight เกิดการเปลี่ยนแปลง graph ที่อยู่ภายในโมเดลก็จะเปลี่ยนแปลงตามไปด้วย สำหรับค่า bias เป็นส่วนที่ทำให้กราฟเปลี่ยนแปลง offset ไปดังนั้นค่านี้จะไม่นิยมให้มีการเปลี่ยนแปลงที่มากนักเมื่อทั้งสองค่านี้ปรับตามกันไปแล้วมันจะได้สมการที่มีค่าสัมประสิทธิ์ที่เหมาะสมกับการกระจายของข้อมูลหรือ element ที่กระจายอยู่บน coordinate plane ออกมา เมื่อมีข้อมูลใหม่เข้ามา มันก็จะอาศัยชุดของสมการพวกนี้ในการทำนายผลลัพธ์ที่ตรงกับค่าสัมประสิทธิ์ออกมา สมการที่ยกตัวอย่างมานี้เรียกกันว่า activation function หรือ transfer function นั่นเองซึ่งมันไม่ได้มีแค่สมการนี้แค่สมการเดียวผู้เขียนจะขอยกตัวอย่างอีกเล็กน้อยเพื่อความเข้าใจที่ตีมากยิ่งขึ้น ดังนี้

1. Sigmoid function

- Sigmoid หรือ Logistic function เป็นฟังก์ชันที่ไม่ได้เป็นฟังก์ชันทางเดียวดังนั้นการเรียนรู้ให้ได้ผลลัพธ์ออกมาก็จะเป็นไปได้ทั้งสองค่ามันจึงนิยมใช้กับการเรียนรู้ที่ต้องการผลทำนายออกมาเป็นสองค่า สมการ คือ

$$y = \frac{1}{1 + e^{-x}}$$

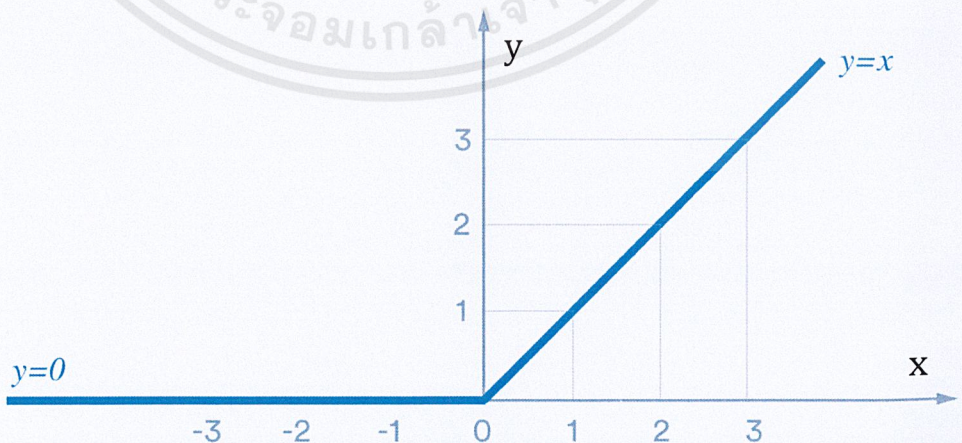


รูปที่ 5 กราฟของ sigmoid function

2. ReLu function

- Rectified Linear unit (ReLU) เป็นสมการที่มีการข้อดีตรงที่ส่วนที่จะมีค่าแค่ในส่วน
ของ x เป็นบวกเท่านั้นในส่วนลบจะถูกพิจารณาให้เป็น 0 เสมอซึ่งมันก็มีข้อดีตรงที่ทำให้
การเรียนรู้เป็นไปได้แบบรวดเร็วแต่ข้อเสียก็คือมันทำให้ข้อมูลที่สำคัญในส่วนลบหายไป
ด้วย

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$



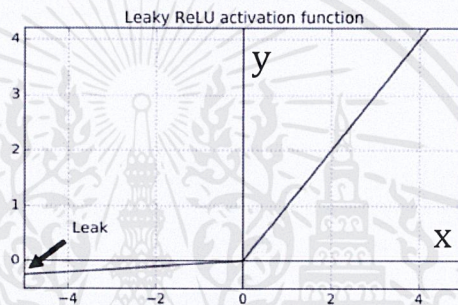
รูปที่ 6 กราฟของ ReLu function [24]

3. Leaky ReLu

- เป็นฟังก์ชันที่มีขึ้นมาเพื่อแก้ไขข้อด้อยของ ReLu โดยการเติมค่าสัมประสิทธิ์ให้กับมันลงไปในด้านลบทำให้มันพิจารณาค่าที่อยู่ในด้านลบได้ด้วย

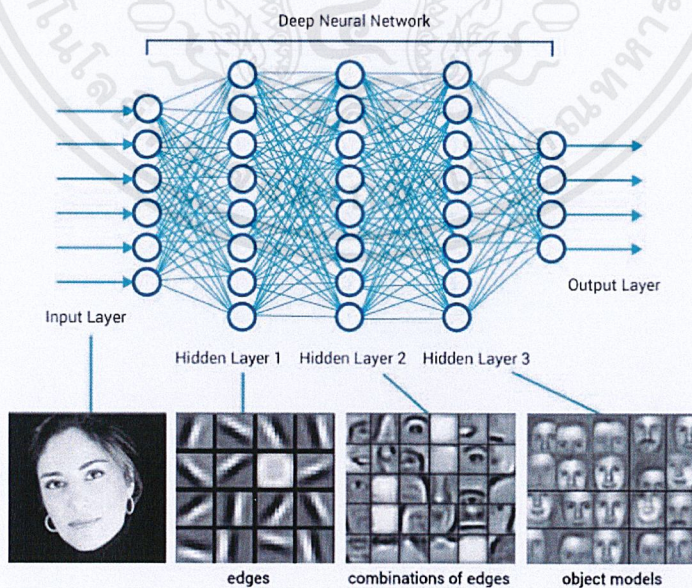
$$f(x) = \begin{cases} x\epsilon, & x < 0 \\ x, & x \geq 0 \end{cases}$$

โดยที่ ϵ มีค่าน้อยมาก ๆ



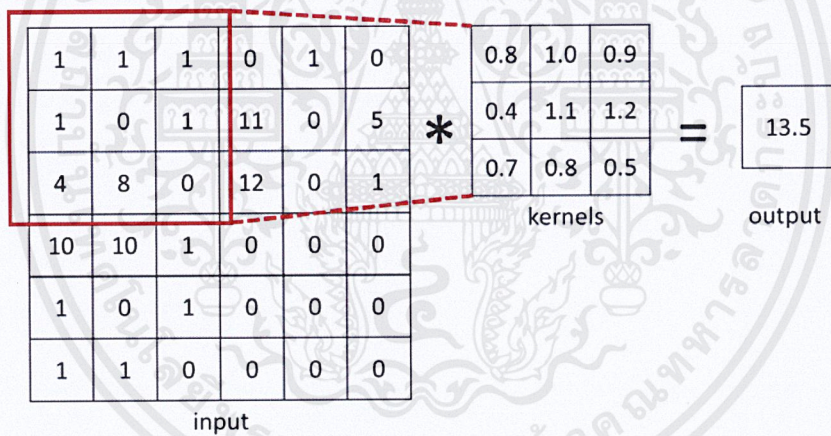
รูปที่ 7 กราฟของ Leaky ReLu

สำหรับการเรียนรู้เชิงลึกจะมีโครงสร้างดัง รูปที่ 8



รูปที่ 8 ลักษณะของ deep neural network [8]

ดังที่ได้กล่าวไปข้างแล้วในตอนต้นว่าประเภทย่อยของการเรียนรู้เชิงลึกมีแบ่งอีกมากมาย แต่โดยหลักแล้วในปัจจุบันนี้ ก็จะมีกลุ่มของ Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) และ Generative Adversarial Networks (GAN) ซึ่งแต่ละแบบก็มีความเหมาะสมต่างกันไป เช่น GAN ที่เป็นที่ยอมรับในขณะนี้ก็สามารถที่จะทำการเพิ่มประสิทธิภาพของรูปภาพ (enhance) จากความละเอียดน้อยให้มาเป็นความละเอียดมากได้โดยที่ทำการสร้างรูปใหม่เลยไม่ใช้การเปลี่ยนขนาดรูปภาพเรียกว่า super-resolution ในรูปภาพจะเห็นได้ว่า การเรียนรู้แต่ละชั้นจะมีความสามารถในการรับรู้ถึงวัตถุต่างกันโดยตัวอย่างของ CNN จะมีการนำพวกการ pooling มาเพื่อใช้ลดเวลาที่ต้องใช้ในการเรียนรู้จากแต่ละชั้นและการสังวัตนาการ (convolution) ซึ่งเป็นส่วนหลักที่ทำให้เกิดการเรียนรู้ได้จริง ๆ เนื่องจากทั้งในสัญญาณต่อเนื่องและไม่ต่อเนื่อง (digital) จะใช้วิธีการนี้ในการค้นหาส่วนที่สำคัญของรูปภาพออกมาจะเห็นได้ว่าแต่ละชั้นของการเรียนรู้จะปิดท้ายด้วย activation function เสมอเนื่องจากมันเป็นเหมือนกับส่วนที่ใช้ตัดสินใจว่าผลลัพธ์สุดท้ายที่ออกมาควรจะเป็นค่าใดในการ convolution จะมีสิ่งที่เรียกว่า kernels อยู่ภายในซึ่งจะคอยถ่วงน้ำหนักให้เกิด ค่าที่สำคัญจริง ๆ จากรูปรูปที่ 9



รูปที่ 9 การ convolution

2.1.3 segmentation

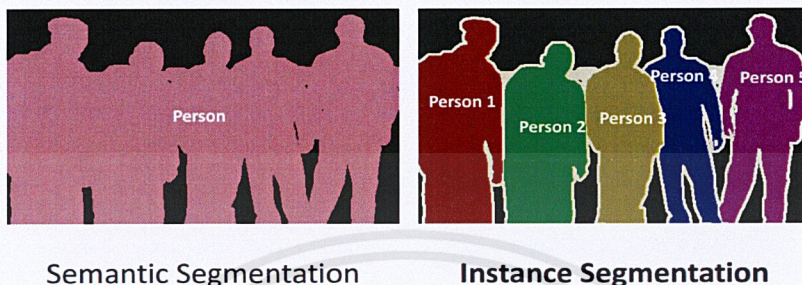
การแบ่งส่วน (segmentation) คือการระบุตำแหน่งและขอบภาพของรูปภาพ โดยประเภทของการ segmentation มี 2 รูปแบบด้วยกัน คือ

1. Semantic segmentation

- การแบ่งส่วนด้วยรูปแบบนี้จะสามารถบอกได้แค่ส่วนที่แบ่งออกมาเป็นคลาสไหนเพียงเท่านั้น เป็นเหมือนกับ classification ปกติ

2. Instance segmentation

- การแบ่งส่วนด้วยวิธีนี้เป็นการแบ่งตามคลาสโดยที่สามารถระบุได้ด้วยว่าแต่ละส่วนนั้นเป็นของวัตถุชิ้นไหน เป็นเหมือนกับการ detection



รูปที่ 10 ความแตกต่างของการ segment ทั้ง 2 แบบ [15]

โดยโมเดลที่นิยมใช้งานกับทั้งสองประเภทก็มีแตกต่างกันไป โดยการทำงานแบบ instance segmentation จะมีโมเดลที่ใช้งานซึ่งมีการทำ pre-trained ให้เรียบร้อยแล้ว เช่น Mask-RCNN ทั้งนี้โมเดลสำหรับการ segmentation มีหลากหลายโมเดลมากทั้งในแบบที่เป็นการทำงานของอัลกอริทึมที่ไม่ใช่การเรียนรู้เชิงลึก เช่น watershed algorithm ซึ่งเป็นการเรียนรู้โดยไม่ต้องมีคนคอยป้อนข้อมูล (unsupervised learning) การทำงานของอัลกอริทึมนี้จะใช้เรื่องของ contour เข้ามาเกี่ยวข้องด้วยให้ลองจินตนาการว่ารูปภาพหนึ่งเมื่อแปลงเป็น grayscale แล้วมันจะมีระดับความสูงต่ำของรูปภาพต่างกัน เหมือนกับหุบเขา เมื่อมีฝนตกลงไปมันจะเกิดแอ่งน้ำแล้วแบ่งออกเป็นส่วนชัดเจนขึ้นเมื่อระดับน้ำสูงขึ้นไปเรื่อย ๆ ในส่วนของการเรียนรู้เชิงลึกกับ Mask-RCNN จะเป็นโมเดลที่มีประสิทธิภาพมากในเรื่องของความแม่นยำเนื่องจากการทำนายจะทำนายทั้งส่วนที่เป็น mask อย่างเดียวพร้อมกับการทำนายจากรูป input ที่ใส่เข้าไปด้วยว่าขอบเขตของวัตถุที่เราสนใจอยู่บริเวณไหน ดังนั้นค่า loss ที่ได้ออกมาจะมีประสิทธิภาพกับการเรียนรู้ที่สูงกว่า

ตัววัดผลที่ใช้ได้ในการพิจารณาความถูกต้องของการแยกส่วน (Metrics)

ในส่วนนี้ไม่ได้ต้องการจะพูดถึงแค่การแสดงค่าที่ได้จากการเปรียบเทียบการทำนาย (predictions) และค่าความจริง (ground truth) แต่ค่าในส่วนนี้จะมีความสำคัญกับการเรียนรู้แบบย้อนกลับ (backward-propagation) ซึ่งเป็นการใช้กฎลูกโซ่ (chain rule) มาหาค่า weight กลับของแต่

ละ node โดยปกติค่าเริ่มต้น (initialize) จะเป็นแบบสุ่มยกเว้นแต่จะใช้ He หรือ Xavier initialize มาควบคุมความ variance ของการสุ่มเริ่มต้น

ตัววัดผลที่ได้พูดถึงตรงนี้ไม่ได้มีไว้เพื่อการวัดความถูกต้องของการทำ segmentation เพียงแค่ว่าอย่างเดียว แต่ยังสามารถที่จะใช้วัดผลในส่วนอื่นได้อีกด้วยจากสมการของ F1-score จะเห็นได้ว่ามันจะพิจารณาจากค่าที่ซ้ำมากที่สุด ยกตัวอย่าง คือ ร้านหนึ่งมีลูกค้าเข้ามา 100 คนซื้อของออกไป 80 คนที่เหลือเข้ามาแค่ดูของ เมื่อลูกค้าคนที่ 101 เข้ามามันจะทำนายว่าคนนี้จะซื้อด้วยทันที ที่กล่าวมาคือการวัดค่าแบบกรณีตัวอย่างในการศึกษาเพื่อให้เข้าใจง่ายมากยิ่งขึ้น

1. Dice coefficient

จากสมการของ

$$diceScore = \frac{2|A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN}$$

โดยที่ A คือ predict regions และ B คือ ground truth โดยค่าที่ได้ออกมาจะเป็นค่าเดียวกับ F1score เสมอ(เนื่องจากทั้งสองค่านี้เป็นแค่การจัดรูปสมการใหม่)

2. Jaccard index

มีอีกชื่อว่า IoU : Intersection over Union

$$Jaccard = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{FP + FN - TP}$$

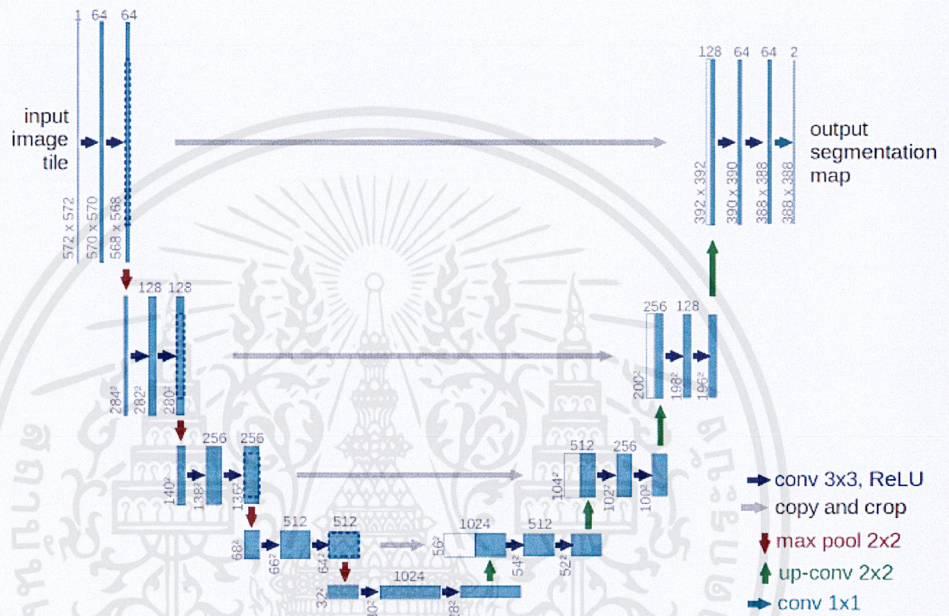
จะเห็นได้ว่าค่าของ dice และ Jaccard จะมีความแตกต่างกันตรงที่เซตของส่วนใน Jaccard จะไม่มีส่วนที่ intersect กัน

2.1.4 โมเดล U-Net [3]

เนื่องจากข้อมูลภาพทางการแพทย์ (ส่วนใหญ่เป็น grayscale) จะมีความสามารถในการให้โมเดลได้เรียนรู้ค่อนข้างน้อยกว่ารูปที่เป็น RGB ดังนั้นการใช้งานของโมเดลจะต้องมีความต่างออกไปในตัวโมเดล U-Net นี้ถูกออกแบบมาเพื่อรับมือกับปัญหาการถ่ายภาพทางการแพทย์ที่มีปัญหาทั้งเรื่องของจำนวน

channels จำนวนคลาสที่ไม่เท่าเทียมกัน (class imbalance) ตัวโมเดลนี้จะทำงานได้อย่างมีประสิทธิภาพเมื่อมีการเพิ่มจำนวนข้อมูล (data augmentation) เยอะ ๆ

ส่วนของการใช้งานจะเป็นโมเดลที่ออกมาแบบมาเพื่อการแยกส่วนของรูปภาพทางการแพทย์ โดยเฉพาะซึ่งมันมีความเหมาะสมกับรูปภาพระดับ grayscale ทั้งแบบ 2 มิติและ 3 มิติ ที่มีจำนวนของ samples น้อยดังนั้นโมเดลนี้จะมีประสิทธิภาพกับการดัดแปลงรูปภาพเพื่อให้เกิดการเรียนรู้เยอะๆ



รูปที่ 11 โมเดล U-Net [3]

โครงสร้างของโมเดล จากรูปภาพจะเห็นได้ว่าตัวโมเดลมีทั้งหมด 3 ส่วนหลักๆ ก็คือ ทางด้านซ้ายสุดจะเป็นส่วนที่เรียกว่า down convolution หรือ encoder ซึ่งมีหน้าที่หลักๆในการหาส่วนสำคัญของรูปภาพซึ่งในส่วนนี้จะใช้ sliding convolution ตามปกติและเพิ่มจำนวนของ feature channels เป็นสองเท่าในทุกๆ step ตรงกลางหรือ bottleneck และจะมีส่วนของการ และทางด้านขวาสุด คือ up convolution จะทำการลดจำนวนของ feature channels ทีละครึ่งในทุก step และมีการ concatenate รูปที่ที่เกิดการ crop แล้วมาจากส่วนของการ down-sampling ซึ่งการทำงานนี้ทำให้โมเดลสามารถที่จะแยกแยะได้ชัดเจนมากยิ่งขึ้นเมื่อในขั้นสุดท้ายเป็นการ convolution 1x1 เพื่อที่จะปรับให้ขนาดของ output เป็นไปตามที่เราต้องการ

2.1.5 เทคนิค transfer learning

เทคนิค transfer learning มีความนิยมกันอย่างมาในช่วงหลังมานี้เนื่องจากทำให้ได้ผลลัพธ์ที่มีคุณภาพมากถึงขนาดที่คอมพิวเตอร์ปกติไม่สามารถทำได้เนื่องจากตัว pre-trained model บางตัวและยังเป็นการประหยัดเวลาด้วยโดยเทคนิคนี้แบ่งออกเป็นสองประเภทหลัก ๆ คือ feature extraction และ fine-tuning โดยเทคนิคทั้งสองอย่างนี้มีความแตกต่างกันเพียงเล็กน้อยแต่มีวัตถุประสงค์ในความต้องการใช้งานเป็นผลพวงจากกัน อธิบายเริ่มด้วยตัว pre-trained โมเดลที่มีการทำไว้แล้วไม่ว่าจะเป็นโมเดลที่มีค่าถ่วงน้ำหนักจากการชุดข้อมูลของ COCO หรืออะไรก็ตามแต่มันในแต่ละ framework ก็จะมีการจัดเตรียมในส่วนนี้ไว้ให้ต่างกันสำหรับ Pytorch จะมีการเตรียม pre-trained ไว้ให้เลือกมากจากตัวมันเอง ส่วนใน Tensorflow ส่วนใหญ่จะเป็นการทิ้ง checkpoint ไว้ให้ดาวน์โหลดและต้องมาทำการปรับ variables ด้วยตัวเอง สำหรับ feature extraction จะเป็นการตัดส่วนหัวของโมเดลออก (model surgery) และเปลี่ยนจำนวนของ output layers เท่ากับจำนวนของคลาสที่เราต้องการเมื่อโมเดลเดิมมีความเหมาะสมกับ output อยู่แล้วมันจึงไม่มีความจำเป็นที่จะต้องฝึกฝนเพิ่มเติมแต่อย่างใด

2.1.6 Hyper-parameter tuning

ในบางกรณีโมเดลที่เราได้เริ่มต้นฝึกฝนมาในตอนแรกบางครั้งก็อาจมีความไม่เหมาะสมของค่าบางค่าในสมการอยู่เนื่องจากค่าบางค่าเช่น ค่าถ่วงน้ำหนักที่จะถูกกำหนดเริ่มต้นแบบสุ่มถึงแม้มันจะเปลี่ยนด้วยตนเองไปจนถึงจุดที่เหมาะสมได้ แต่มันก็จะเปลี่ยนแปลงภายใต้ปัจจัยของ Hyper-parameter ซึ่งเป็นค่าที่ตัวเครื่องไม่สมควรเปลี่ยนแปลงด้วยตัวเอง ค่าที่นิยมในการปรับมีทั้งหมดด้วยกัน คือ 4 ค่าดังนี้

1. Learning Rate (lr)

เป็นค่าที่ส่งผลกับการเรียนรู้ของโมเดลอย่างเห็นได้ชัด ทำให้ถูกนิยมปรับเปลี่ยนเป็นค่าแรกสุดของทุก ๆ งานซึ่งขอบเขตที่นิยมจะเป็น 0.1 ไปจนถึง 1e-06 โดยค่าที่ lr ส่งผลด้วย คือ weight (W) และ bias (B) ซึ่งเป็นค่าที่อยู่ภายใต้ activation function เมื่อมีการปรับปรุงจากการเรียนรู้แบบย้อนกลับจะได้สมการ

$$W = W - lr * dW$$

$$B = B - lr * dB$$

เมื่อได้ทำการพิจารณาจากสมการแล้วจะเห็นได้ว่าค่า learning rate ไม่ได้ส่งผลกับปัจจัยทางกายภาพทำให้มันไม่ส่งผลใด ๆ กับความเร็วในการฝึกฝนในการปรับเปลี่ยนจะใช้วิธีการแบบ grid search ซึ่งเป็นวิธีที่ใช้การปรับค่าใน list ไปตรง ๆ และพิจารณาว่าค่าใดที่เหมาะสมที่สุด ทั้งนี้ส่วนที่อธิบายข้างต้นเป็นส่วนของ gradient descent ซึ่งเป็นการปรับตามค่า loss ให้เข้าสู่ศูนย์ ซึ่งการปรับปรุงดังกล่าวจะถูก momentum เข้ามาควบคุมด้วย

2. Batch size

เป็นค่าที่ใช้กำหนดการเปลี่ยนแปลงของค่าถ่วงน้ำหนักให้มีจำนวนกี่ครั้งในแต่ละรอบการฝึกฝน เช่นกำหนดให้รอบการฝึกฝน 1 รอบมีขนาด batch เป็น 4 นั้นหมายความว่าในหนึ่งรอบการฝึกฝนจะเกิดการเปลี่ยนแปลงค่าถ่วงน้ำหนักจะเกิดที่ 4 รอบดังนั้นค่านี้จะเป็นตัวที่ส่งผลโดยตรงกับความเร็วในการเรียนรู้ของตัวโมเดลทั้งนี้การที่บอกว่าปรับค่าถ่วงน้ำหนัก 4 รอบมันเกิดจากการนำรูปภาพไปในหน่วยความจำชั่วคราวกับแต่ละรอบการฝึกฝนดังนั้นขีดจำกัดของค่านี้จะเกิดจากความจุของตัวประมวลผล

3. Momentum

เป็นค่าที่ควบคุม learning rate อีกทีโดยในการขยับแต่ละครั้งที่เกิดจาก learning rate มีโอกาสที่เจอปัญหาในตอนที่เจอกับพวก local minima บน gradient map ได้ค่านี้จะทำหน้าที่ในการเปลี่ยนแปลงค่าอนุพันธ์ของค่าถ่วงน้ำหนัก และค่า bias แต่ละครั้งของรอบการปรับค่าถ่วงน้ำหนักด้วยความที่การเรียนรู้จะพิจารณาจาก gradient ของค่า loss เป็นหลักดังนั้นค่านี้จะทำให้การเคลื่อนผ่านจุดพวก saddle point จะเป็นไปได้ด้วย

2.1.7 การพิจารณา early-stop

การทำ early-stop คือการสั่งให้หยุดฝึกฝนโมเดลเพิ่มเติมเพื่อป้องกันการเกิดปัญหา overfit ของตัวโมเดลกับชุดข้อมูลที่ใช้ในการฝึกฝนได้โดยการพิจารณาที่ใช้ในงานครั้งนี้ คือ การพิจารณาจาก validation loss เป็นหลักด้วยความที่ validation loss จะเป็นตัวที่ใช้บอกว่าข้อมูลที่โมเดลไม่เคยเห็นมาก่อนนั้นมีการทำนายถูกหรือผิดไปมากเพียงใด ดังนั้นถ้าโมเดลยังพิจารณาข้อมูลในระดับที่ยังยึดหยุ่นพอประมาณผลลัพธ์ของ validation loss จะยังลดลงต่อไปเรื่อย ๆ แต่เมื่อ validation loss กลับเพิ่มขึ้นมาได้ แสดงว่าข้อมูลที่โมเดลเคยทำนายจนมันถูกไปกลับถูกพิจารณาให้ผิดไปอีกรอบ ดังรูปที่ 31 ดังนั้นการทำ early stop จึงเป็นส่วนสำคัญในการฝึกฝนของตัวโมเดลด้วย

2.1.8 ค่า loss ของการประยุกต์ใช้ปัญญาประดิษฐ์กับภาพเอกซเรย์ 2 มิติ

ค่า loss สำหรับงานนี้คือ binary cross entropy ซึ่งเป็นค่าที่ใช้การคำนวณจาก entropy ของค่าที่เป็นความจริง (labels) และค่าที่เป็นผลมาจากการทำนาย (logits) ดังสมการด้านล่าง ด้วยความที่งานนี้ต้องการผลลัพธ์ในลักษณะของ multi-label classification ทำให้ในส่วนของ output layer จะตัดสินใจโดยใช้ softmax function ไม่ได้ หรือก็คือในทุก output class จะต้องถูกตัดสินใจเป็นเหตุการณ์ “ใช่” หรือ “ไม่ใช่” ทั้งหมด

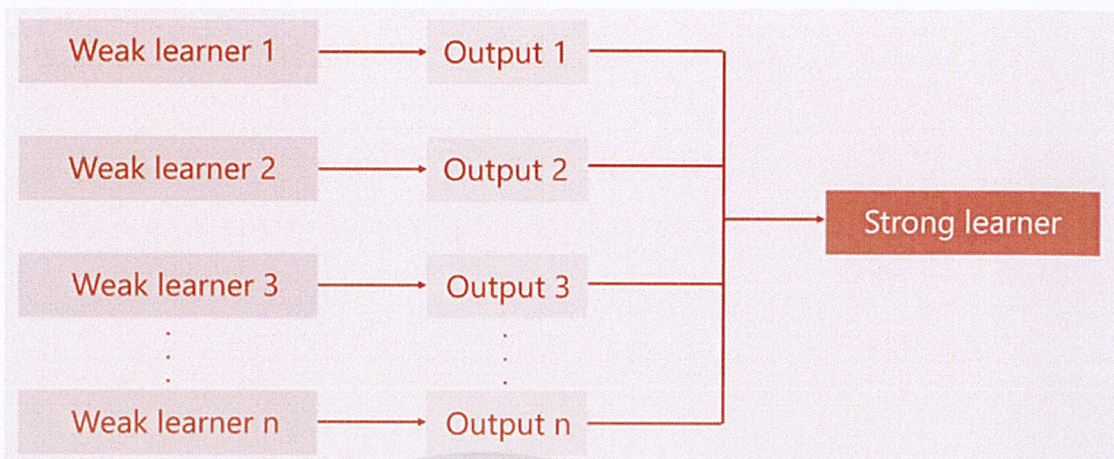
$$Loss = - \sum_{i=1}^N Prob_i(true) \log Prob_i(predict)$$

2.2 ทฤษฎีในการพัฒนา desktop application

2.2.1 การทำงานของ face detection ด้วยอัลกอริทึม haar-like

อัลกอริทึม Haar-like เป็นหนึ่งในการเรียนรู้แบบ boosting ซึ่งเป็นหนึ่งในประเภทของ Ensemble method ซึ่งมีวิธีการแบ่งได้ 3 หลัก คือ Bagging, Stacking และ Boosting

1. Bagging เป็นการเรียนรู้แบบที่การให้เรียนรู้แบบที่เปรียบให้เห็นภาพไปแล้วคือเอาคำตอบของนักเรียนมาหาว่าส่วนใหญ่ักเรียนตอบอะไรกัน วิธีการนี้จะจัดการความ variance ได้ดี ตัวอย่าง เช่น พวกอัลกอริทึม Forest
2. Boosting เป็นวิธีการที่ใช้ในการทำงาน face detection นี้ด้วยเป็นการเรียนรู้โดยอาศัยความผิดพลาดจากรุ่นที่มาปรับปรุงให้รุ่นใหม่เรียนรู้ได้ดีขึ้น วิธีนี้ให้ความแม่นยำที่สูงเพราะมุ่งเน้นจะจัดการกับ error อย่างเดียวมุ่งเน้นที่จะเปลี่ยน offset เป็นหลัก ข้อเสียที่ชัดเจนก็คือถ้ามุ่งจะลดแต่ error ต้องมีการพิจารณา early stop ที่ดีเพราะมีโอกาสที่จะเกิด overfit สูง (เป็นปัญหาที่โมเดลจะเรียนรู้แบบเถรตรงเกินไปจนปฏิเสธความเป็นไปได้ของข้อมูลใหม่ ๆ มีโมเดลที่เกิดปัญหานี้จะมีชื่อเล่นว่า grandmother model)

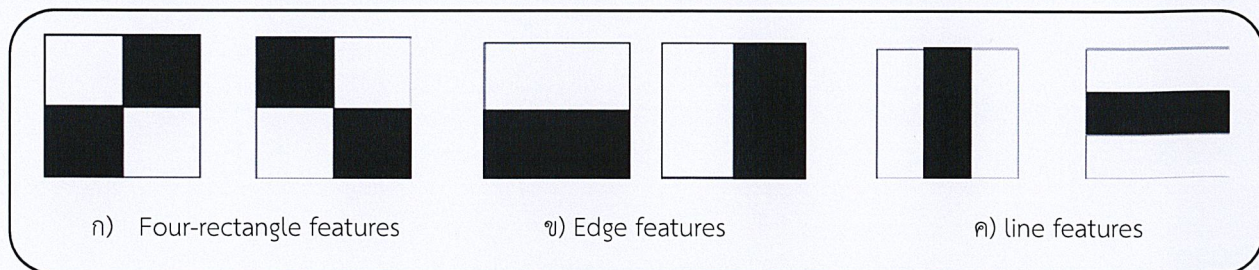


รูปที่ 12 การเรียนรู้แบบ boosting [15]

- Stacking เป็นการเรียนรู้โดยการถ่ายทอดความรู้จากรุ่นพี่กลุ่มหนึ่งมายังรุ่นน้องอีกกลุ่มหนึ่งมีความแตกต่างจาก Boosting ตรงที่มันใช้ learners มากกว่า 1 สำหรับแต่ละ iteration

อัลกอริทึม Harr-like [16] หรืออีกชื่อว่า Harr-cascades เป็นอัลกอริทึมที่ใช้ในการฝึกฝนตัวเครื่องให้มีความสามารถในการเรียนรู้วัตถุชนิดหนึ่งได้ (Object Detection) โดยข้อจำกัดของอัลกอริทึมนี้จะต้องใช้ข้อมูลในการฝึกฝนจำนวนมาก ในการเรียนรู้จะเริ่มโดยการแบ่งรูปภาพออกเป็นสองชนิดด้วยกัน คือ positive images ซึ่งเป็นส่วนที่มีองค์ประกอบที่เราสนใจอยู่ภายในบริเวณนั้นของรูป และ negative images คือรูปภาพที่ไม่มีส่วนที่เราสนใจอยู่ภายในนั้น ในขณะที่ทำการฝึกฝนตัวโมเดลมันจะสามารถเรียนรู้ได้จากภาพทั้งสองลักษณะนี้ โดยการเรียนรู้ตาม

การแบ่งลักษณะรูปภาพดังกล่าวจะเกิดได้จากการทำงานของ Haar-features หรือ sub-windows ซึ่งมันจะเอาส่วนดำและขาวของมันในการ operate กับรูปภาพ input ให้เกิดลักษณะที่เหมือนกับค่าถ่วงน้ำหนักใน convolution kernels ขึ้นมา ลักษณะของ Haar-features เป็นดังรูปที่ 13



รูปที่ 13 Haar-features แบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานโดยรวมของอัลกอริทึมนี้แบ่งออกทั้งหมดได้ 3 ขั้นตอน

1. การคำนวณหา Integral image
2. ค้นหา contour ที่สนใจด้วย AdaBoost
3. จัดกลุ่มข้อมูลแบบ Cascade classifier

ในขั้นตอนแรกคือการคำนวณหา integral image จากรูปที่ 13 แสดงให้เห็นว่าเราสามารถคำนวณ ส่วนต่างของแต่ละ Haar-features ได้โดยการที่มันจะใช้ผลลัพธ์ระหว่างส่วน positive และ negative ใน Haar-features มาทำการคำนวณ threshold และ polarity ถัดมาเป็นการหา contour ที่อยู่ภายใน รูปภาพด้วยการที่มันสามารถแบ่งรูปออกเป็น ส่วน ขั้วบวก และลบ (ขาว-ดำ) ได้จากหน้าต่างย่อย หลังจากนั้นเมื่อได้รูปที่มีขั้วออกมาแล้ว เครื่องจะทำการเรียนรู้โดยการนำรูปภาพพวกนั้นมาใช้ในการดึงภาพพวก นั้นออกมาจำนวน n samples แล้วเริ่มทำการปรับค่าถ่วงน้ำหนัก (weight) จากรูปที่เป็นแบบลบโดยค่า weight เริ่มต้นจะเกิดมาจากการสุ่มแล้วทำการปรับปรุงไปเรื่อย ๆ จนกว่า error จะต่ำสุด เมื่อถึงขั้นตอน สุดท้ายได้ผลลัพธ์ออกมาก็จะทำการจัดกลุ่มโดยอัลกอริทึมแบบก่อนหน้าจะใช้วิธีการจัดกลุ่มโดยที่ต่อให้ค่า threshold ยังอยู่ในระดับที่ไม่แน่นอนมันก็จะตัดสินใจไปทั้งแบบนั้นเลยซึ่งถ้ามันเป็นข้อมูลในกลุ่มที่ต้องการ แต่มันโดนคัดออกไปก็จะเสียประโยชน์ในส่วนนี้ไป แต่ใน cascade classifier จะใช้การเลื่อนการพิจารณา ออกไปแทนทำให้มันสามารถที่จะเรียนรู้ได้อย่างมีประสิทธิภาพมากขึ้นและรวดเร็วกว่าอัลกอริทึมก่อน ๆ หน้า

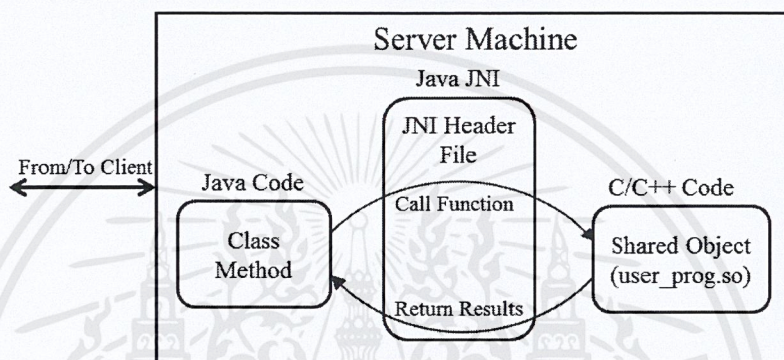
สำหรับ face detection ใน อัลกอริทึมชนิดนี้เมื่อมีข้อมูลเป็น input ด้วยรูปภาพของใบหน้า เริ่มแรกสุดที่มันจะทำการหา features แยกออกมาแต่ละส่วนได้รูปที่ปรับรูปที่เป็นสองขั้วออกมา คือ ส่วนที่มีรูป ที่เราสนใจซึ่งก็คือใบหน้าซึ่งมันจะเกิดซ้ำ ๆ ในทุกรูปให้เป็นภาพขั้วบวก และภาพส่วนที่เราไม่ได้สนใจซึ่ง เป็นภาพขั้วลบ หลังจากนั้นมันจะทำการปรับปรุงค่า weight ไปเรื่อย ๆ จนกระทั่งค่า error มีความ ชัดเจนที่จะรู้เข้าหาค่าใดค่าหนึ่ง จะสังเกตได้ว่าการทำงานของอัลกอริทึมจำพวก boosting จะมีความ คล้ายกับ deep learning มาก

2.2.2 Java Native Interface (JNI)

JNI เป็นเหมือนกับ library ที่ถูกสร้างขึ้นมาเพื่อให้โปรแกรมสามารถทำงานแบบ native ได้ หรือก็คือมันสามารถรันข้อมูลบนคนละภาษากันได้นั่นเอง ตัวอย่างเช่น เขียนภาษา C แต่ส่งข้อมูลไปเรียก ฟังก์ชันบน Java class เป็นต้น โดยตัว JNI ถูกพัฒนามาให้ Java execution engine สามารถรัน C, C++ (Native Library) ได้ ความจำเป็นสำหรับการใช้งาน JNI ส่วนใหญ่จะเกิดจากการที่ทั้งสามภาษานี้

ไม่ได้เอื้อต่อการทำงานทั้งหมดจะเห็นได้ว่า Java เป็นภาษาที่เหมาะสมกับการพัฒนาซอฟต์แวร์ ในขณะที่ C หรือ C++ เป็นภาษาที่เหมาะสมกับการติดต่อกับฮาร์ดแวร์ (low-level resource)

ในส่วนนี้อาจมีความเข้าใจผิดได้ว่าเมื่อมีการใช้งาน JNI แล้วการทำงานของ application จะถูกแบ่งออกเป็น 2 การทำงานซึ่งความเป็นจริงแล้วไม่ใช่อย่างนั้นการทำงานของโปรแกรมที่มีการเรียกใช้ JNI จะถูกทำงานเป็น process เดียวกันเลยตั้งแต่การเรียกใช้เมธอดที่เกิดขึ้นมาจากฝั่ง Java ไปจนถึงการส่งกลับผลลัพธ์ที่ได้จาก low-level resource



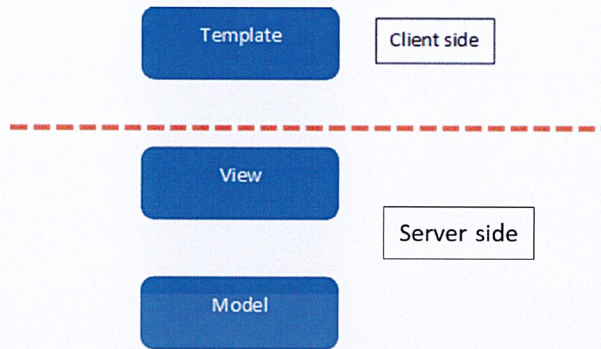
รูปที่ 14 การทำงานของ JNI [21]

2.2.3 การทำงานของ Django framework

Django เป็นชุดเครื่องมือที่ใช้ในการพัฒนา website ซึ่งเป็นชุดเครื่องมือที่เขียนด้วยภาษา python ด้วยความที่ Django เป็นภาษาที่ถูกพัฒนาขึ้นเพื่อให้ง่ายต่อการทำงานติดต่อกับพวก REST API นอกจากนี้ยังมีการทำงานของหลายส่วนไม่ว่าจะเป็นการจัดการส่วนของ admin ให้โดยที่ไม่จำเป็นต้องมานั่งเขียนตั้งแต่เริ่ม

การทำงานของ Django จะเป็นการทำงานคล้ายกับ framework อื่นโดยแพทเทิร์นการทำงานก็คือ MTV (Model-Template-View) ดังรูปที่ 15

องค์ประกอบหลักของการทำงานใน framework นี้คือ จะใช้ View ในการ response ผ่านไปบน HTTP ด้วยการสร้างฟังก์ชันผ่านไฟล์นี้ทั้งหมด ส่วนของการจัดการกับ database จะทำผ่าน Model โดยทำได้ทั้งการสร้าง table เพิ่มหรือลบออกหรือการปรับปรุงก็สามารถทำผ่านไฟล์นี้ได้ สำหรับส่วนของ controller จะเหมือนกับไฟจราจรที่คอยควบคุมการลิ่งระหว่างเพจโดยสามารถใช้ regex ในการตรวจผ่านลิงค์ได้ (regular-expression-based URL) จาก รูปที่ 15 ส่วนของ template ก็คือฝั่งของ frontend ซึ่งก็คือพวกไฟล์ html



รูปที่ 15 โครงสร้าง MTV

2.2.4 Image alignment

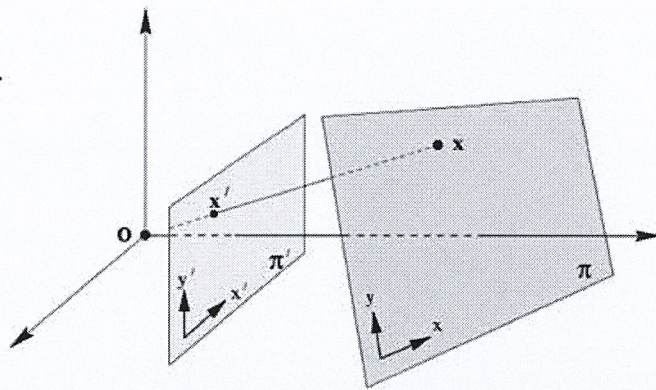
Image alignment หรือ image registration เป็นการทำให้รูปภาพที่อยู่บนมุมมองที่ต่างได้มีการปรับเปลี่ยนขึ้นมาอยู่ภายใต้ มุมมองเดียวกันทั้งนี้ในรูปที่ใช้เป็นตัวอ้างอิงจำเป็นที่จะต้องให้มันมีความเด่นชัดของข้อมูลเพื่อที่เมื่อมีข้อมูลเข้ามาใหม่แต่มีลักษณะต่างกันไปเล็กน้อยก็ยังสามารถที่จะทำงานได้

หัวใจหลักของการทำงานนี้คือการหา homography ซึ่งเป็นการหาตำแหน่งเดียวกันจากภาพ 2 ภาพโดยการทำงานของมันเป็นจะอาศัย kernels ขนาด 3x3

$$\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

เมทริกซ์ด้านบนแทนด้วยหน้าต่าง homography ขนาด 3x3 และการคูณกันของเมทริกซ์จะเป็นดังแสดงด้านล่าง

$$\begin{bmatrix} x^{\wedge}_1 \\ x^{\wedge}_2 \\ x^{\wedge}_3 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$



รูปที่ 16 homography [22]

จากรูปที่ 16 แสดงให้เห็นได้ว่าการหาผลลัพธ์สุดท้ายที่ออกมาจะเป็นได้ด้วยถ้าให้ x_1, x_2, x_3 เป็นจุดที่ที่อยู่ในตำแหน่งเดียวกันกับของอีกภาพตั้งนั้นไม่ว่าจะมองภาพมาจากมุมไหน ผลลัพธ์ที่ได้จากอัตราส่วนของค่า x คูณกับ h ในการขยับ window ไปที่ละหนึ่งจุด ๆ หนึ่งจะได้ผลออกมา 9 ครั้งโดยค่าของแต่ละครั้งจะต่างกันไปอย่างไรก็แล้วแต่อัตราส่วน (Degree Of Freedom) ของมัน ซึ่งกันและกันจะต้องใกล้เคียงกัน

2.2.5 Cross-Origin Resource Sharing (CORS)

WebSwing คือบริการที่ทำให้สามารถใช้งาน Java application และ Java applet ได้บนเว็บเบราว์เซอร์ ทั้งนี้การใช้งานจะต้องมีการ config server ส่วนของ CORS ซึ่งเป็นกลไกที่ทำให้สามารถเข้าถึงทรัพยากรของ domain อื่น ๆ ที่อยู่คนละ domain กับที่ใช้งานในขณะนั้น ตัวอย่างเช่น การร้องขอข้อมูลจาก facebook จาก website ที่สร้างขึ้นมาเองเป็นต้น ในกรณีนี้ก็คือการร้องขอจากหน้าเว็บเพจที่จะเข้าไปใช้งานทรัพยากรในตัว server เป็นต้น ในกรณีที่เว็บเซิร์ฟเวอร์ไม่ได้มีการรองรับส่วนนี้มาให้หรือเป็นการเข้าผ่านตัวเว็บที่รันอยู่บน

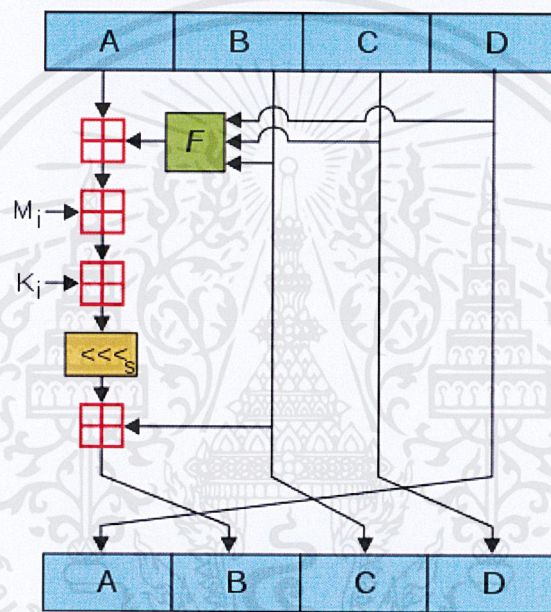
2.2.6 การใช้งานฟังก์ชันแฮชในการเก็บรหัสผ่าน

เนื่องจากฟังก์ชันแฮชเป็นฟังก์ชันทางเดียวจึงนิยมใช้ในการเก็บรักษาหัสผ่านของผู้ใช้ เนื่องจากว่ามันทำให้ตัวผู้ดูแลเองก็ไม่สามารถที่จะรับรู้ได้ว่ารหัสที่แท้จริงคืออะไรด้วยลักษณะนี้ทำให้การที่จะถอดรหัสย้อนกลับมาเป็นไปได้ในวิธีเดียว คือ การทำ brute-force attack ซึ่งเป็นการลองแทนที่ทุกตัวอักษรว่าจะได้ตัวแฮชตรงออกมากับตัวแฮชที่มีหรือไม่ ในฟังก์ชันแฮชแบบปกติจะมีความยาวที่ชัดเจน

เสมอเป็นส่วนหนึ่งที่ทำให้เกิดจุดอ่อนของการเข้ารหัสแบบนี้ ในครั้งนี้มีการใช้งานของฟังก์ชันแฮช 2 แบบ ซึ่งให้ความเหมาะสมกับข้อมูลที่ต้องการเก็บ

2.2.6.1 แฮชฟังก์ชัน MD5

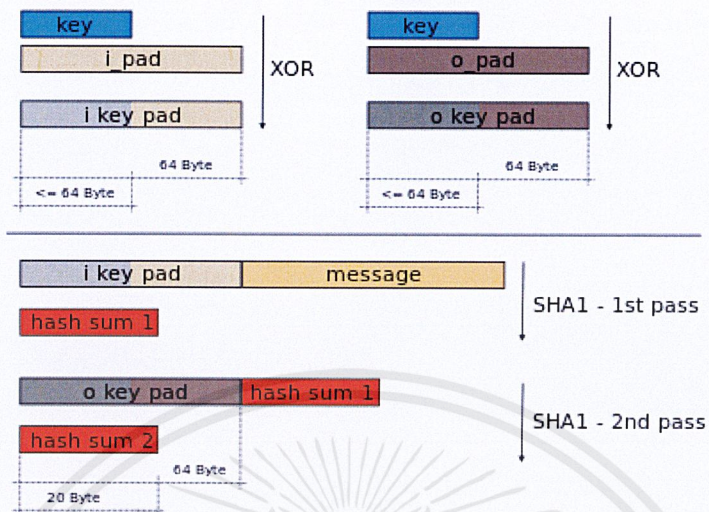
MD5 เป็นฟังก์ชันแฮชที่มีความยาว 128 บิตซึ่งเป็นที่นิยมกับการใช้งานในยุคก่อนถึงขั้นมีการใช้เพื่อเป็นการยืนยันการเข้าถึงของ admin บนเราเตอร์ แต่อย่างไรก็ตามในปัจจุบันมีตัวประมวลผลที่มีความสามารถมากขึ้นไปจากยุคก่อนมาก การทำ brute-force กับบางรหัสสามารถสำเร็จได้ในหน่วยมิลลิวินาที



รูปที่ 17 วงจรของ MD5 [17]

ข้อมูลที่จะออกไปจะถูกบังคับที่ 128 บิตซึ่งส่วนที่เป็นการดำเนินการของวงจรจะถูกแบ่งออกเป็นบล็อก ความยาวบล็อกละ 32 บิตทั้งหมด 4 บล็อก A, B, C และ D ซึ่งจะได้ความยาวรวม 128 บิต หลังจากนั้นจะนำมาเข้า ฟังก์ชันที่ไม่เป็นเชิงเส้น(F) และ M_i คือ ข้อความที่ใส่เข้ามาความยาว 32 บิต K_i คือ ค่าคงที่ยาว 32 บิตที่จะเปลี่ยนไปทุกรอบ $\ll s$ คือ ขยับบิตไปทางซ้ายเป็นจำนวน s ครั้ง \boxplus แทนด้วยมอดุโล 2^{32}

2.2.6.2 Password-Based Key Derivation Function 2



รูปที่ 18 หลักการทำงานของ การเข้ารหัสแบบ PKBDF2 [18]

PKBDF2 เป็นการแฮชข้อมูลที่ได้รับการยอมรับในปัจจุบันเนื่องจากการทำงานของมันจะเป็นการเข้ารหัสที่มีการใช้งานของ salt ที่เป็นส่วนช่วยจัดการในการมีอยู่ของ rainbow table เนื่องจาก salt จะเกิดจาก PRF : Pseudo Random Function ซึ่งมันจะกำเนิดชุดข้อมูลแบบสุ่มซึ่งในรูปก็คือ “key” หลังจากที่ได้ชุดรหัสสุ่มนี้มาก็จะนำมาทำการ padding ให้มีความยาวตาม length ที่กำหนดไว้โดยตลอด กระบวนการจะทำการดำเนินการแบบ XOR หลังจากนั้นจึงเอาชุดของ key ที่ผ่านการเติม pad แล้วมาต่อเข้ากับข้อความแล้วแฮชด้วย SHA อีก 2 รอบโดยการเข้ารหัสมีการวนรอบ (iteration) ไปเรื่อย ๆ จนกว่าจะถึงจำนวนรอบที่กำหนด

บทที่ 3

วิธีการดำเนินการ

3.1 การดำเนินการส่วนของปัญญาประดิษฐ์

3.1.1 การทำความเข้าใจข้อมูลที่ใช้ในการทำงาน

3.1.1.1 ข้อมูลในส่วนของ chest X-rays

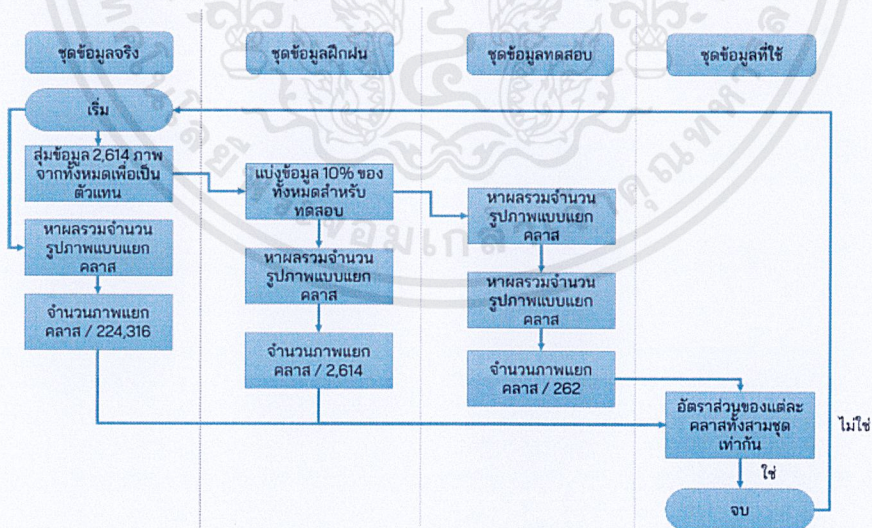
ข้อมูลในส่วนของ chest X-rays โดยส่วนใหญ่แล้วจะปล่อยออกมาในรูปแบบที่คล้ายๆกันโดยข้อมูลที่ได้นำมาใช้ในครั้งนี่คือ Chexpert ซึ่งได้กล่าวถึงไปบ้างแล้วในบทที่ 1 เป็นชุดข้อมูลที่มีขนาดใหญ่มากโดยรวมประมาณ 500 GB ซึ่งต้องใช้ทรัพยากรที่มากในการฝึกฝนโมเดลกับชุดข้อมูลปริมาณขนาดนี้ ดังนั้นผู้จัดทำจึงได้ตัดสินใจเลือกใช้ข้อมูลที่มีขนาดย่อจากภาพปกติที่ไม่มีการบีบอัดและขนาดปกติเป็น 1024x1024 เป็นข้อมูลที่มีการบีบอัดแบบ JPEG และ ย่อขนาดเหลือ 320x320 แต่ด้วยจำนวนของรูปภาพขนาด 224,316 จากผู้ป่วยจำนวน 65,240 คนรูปภาพนั้นยังมากเกินไปสำหรับทรัพยากรของผู้จัดทำจึงได้ลดจำนวนของข้อมูลลงโดยยึดตามจำนวนคนไข้จาก 65,240 คนให้เหลือจำนวน 1,000 คน ซึ่งเมื่อตรวจสอบรูปภาพสแกนแล้วเหลือจำนวนประมาณ 3,000 รูปภาพซึ่งเป็นปริมาณข้อมูลที่เหมาะสมกับทรัพยากรที่มีในการฝึกฝน สำหรับข้อมูล Chexpert เป็นข้อมูลที่ได้มีการจำแนกอาการไว้ทั้งหมดจำนวน 14 อาการ คือ No Finding, Enlarged Cardiomedastinum, Cardiomegaly, Lung Opacity, Lung Lesion, Edema, Consolidation, Pneumonia, Atelectasis, Pneumothorax, Pleural Effusion, Pleural Other, Fracture, Support Devices คลาส “Support Devices” คือ คลาสที่มีการเจาะท่อระบายปอดซึ่งมันอาจใช้ช่วยในการทำนายเกี่ยวกับอาการปอดบวม (Pneumonia) ได้ ซึ่งแต่ละรูปภาพอาจมีได้มากกว่า 1 อาการ

ตารางที่ 1 หมายเลขคลาส, ชื่อคลาส และความหมาย

หมายเลขคลาส	คลาส	ความหมาย
[0]	No Finding	ไม่พบความผิดปกติ
[1]	Enlarged Cardiome-diastinum	กล้ามเนื้อหัวใจโตมาก
[2]	Cardiomegaly	หัวใจโต
[3]	Lung Opacity	มีรบบริเวณปอด
[4]	Lung Lesion	แผลในปอด
[5]	Edema	น้ำในปอด
[6]	Consolidation	ปอดแข็ง
[7]	Pneumonia	ปอดบวม
[8]	Atelectasis	ปอดแฟบ
[9]	Pneumothorax	ปอดรั่ว
[10]	Pleural Effusion	น้ำในช่องเยื่อหุ้มปอด
[11]	Pleural Other	มีความผิดปกติในเยื่อหุ้มปอด
[12]	Fracture	กระดูกแตก/หัก
[13]	Support Devices	ใช้อุปกรณ์ทางการแพทย์

3.1.1.2 การเตรียมความพร้อมข้อมูลในงานภาพเอกซเรย์จากชุดข้อมูล chest X-rays

ลำดับการทำงานจะเป็นไปตามผังการทำงานดังรูปที่ 19 ซึ่งจะเริ่มจากการที่นำชุดข้อมูลทั้งหมดมาเลือกสุ่มทีละ 2,614 รูปภาพ หลังจากนั้นจึงนำชุดข้อมูลนั้นมาแบ่งออกอีกอีก 10% สำหรับการทดสอบ เมื่อได้ทำการสุ่มและแบ่งเรียบร้อยแล้วจะนำมาหาจำนวนผลรวมคลาสแบบแยกกันซึ่งผลลัพธ์ที่ได้ออกมาจะถูกนำมาเปรียบเทียบกันในรูปแบบของกราฟแท่งดังรูปที่ 20



รูปที่ 19 ผังการดำเนินงานคัดเลือกข้อมูล

ตารางที่ 2 จำนวนข้อมูลที่ใช้และ อัตราส่วนต่อข้อมูลทั้งหมด

คลาส	ผลรวมของแต่ละอาการ	อัตราส่วนของข้อมูลทั้งหมด
No Finding	22381	0.10
Enlarged Cardiomeastinum	10789	0.05
Cardiomegaly	27000	0.12
Lung Opacity	105581	0.47
Lung Lesion	9186	0.04
Edema	52246	0.23
Consolidation	14783	0.07
Pneumonia	6039	0.03
Atelectasis	33376	0.15
Pneumothorax	19448	0.09
Pleural Effusion	86187	0.39
Pleural Other	3523	0.02
Fracture	9040	0.04
Support Devices	116001	0.52

จากตารางที่ 2 แสดงถึงจำนวนอาการที่พบในรูปภาพทั้งหมดจำนวน 223,414 รูปภาพ และแสดงสัมประสิทธิ์ของรูปภาพเพื่อใช้ในการพิจารณาหากกลุ่มข้อมูลที่สามารถเป็นตัวแทนของทั้งชุดข้อมูลได้

ตารางที่ 3 จำนวนและสัมประสิทธิ์ของข้อมูลที่ใช้ฝึกฝน

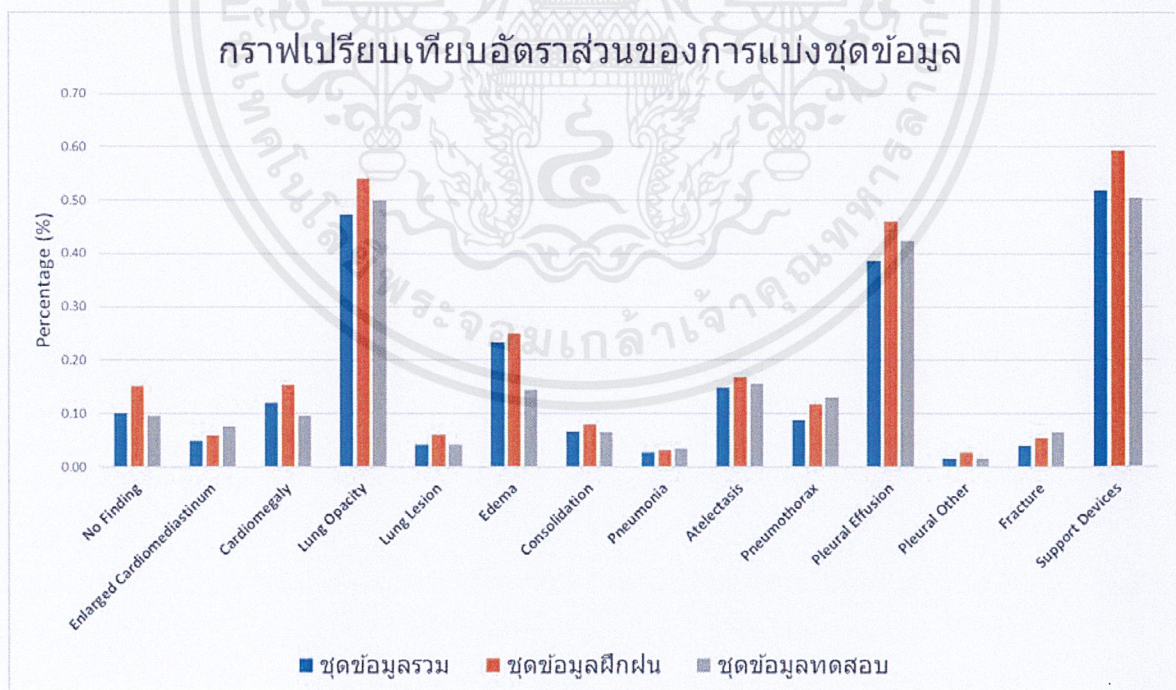
คลาส	ผลรวมของข้อมูลที่ใช้ฝึกฝน	อัตราส่วนของข้อมูลที่ใช้
No Finding	268	0.15
Enlarged Cardiomeastinum	122	0.06
Cardiomegaly	320	0.15
Lung Opacity	1197	0.54
Lung Lesion	104	0.06
Edema	595	0.25
Consolidation	164	0.08
Pneumonia	61	0.03
Atelectasis	374	0.17
Pneumothorax	270	0.12
Pleural Effusion	1020	0.46
Pleural Other	41	0.03
Fracture	111	0.05
Support Devices	1382	0.59

ตารางที่ 3 แสดงถึงอัตราส่วนของแต่ละคลาสในการพิจารณาตัวแทนชุดข้อมูลสำหรับการฝึกฝน โดยคลาสที่มีจำนวนรูปภาพน้อยที่สุดคือ Pleural Other ซึ่งมีจำนวน 41 รูปภาพ

ตารางที่ 4 จำนวนและสัมประสิทธิ์ของข้อมูลที่ใช้ในการทดสอบ

คลาส	ผลรวมข้อมูลที่ใช้ทดสอบ	อัตราส่วนข้อมูลที่ทดสอบ
No Finding	25	0.10
Enlarged Cardiomeastinum	20	0.08
Cardiomegaly	25	0.10
Lung Opacity	131	0.50
Lung Lesion	11	0.04
Edema	38	0.15
Consolidation	17	0.06
Pneumonia	9	0.03
Atelectasis	41	0.16
Pneumothorax	34	0.13
Pleural Effusion	111	0.42
Pleural Other	4	0.02
Fracture	17	0.06
Support Devices	132	0.50

ตารางที่ 4 แสดงถึงอัตราส่วนของจำนวนรูปภาพแต่ละคลาสต่อจำนวนรูปภาพทั้งหมดสำหรับชุดข้อมูลทดสอบโดยคลาสที่มีจำนวนรูปภาพน้อยที่สุดจะยังเป็นคลาส Pleural Other เช่นเดิมซึ่งในข้อมูลทดสอบนี้มีเพียงแค่ 4 รูปภาพเท่านั้น



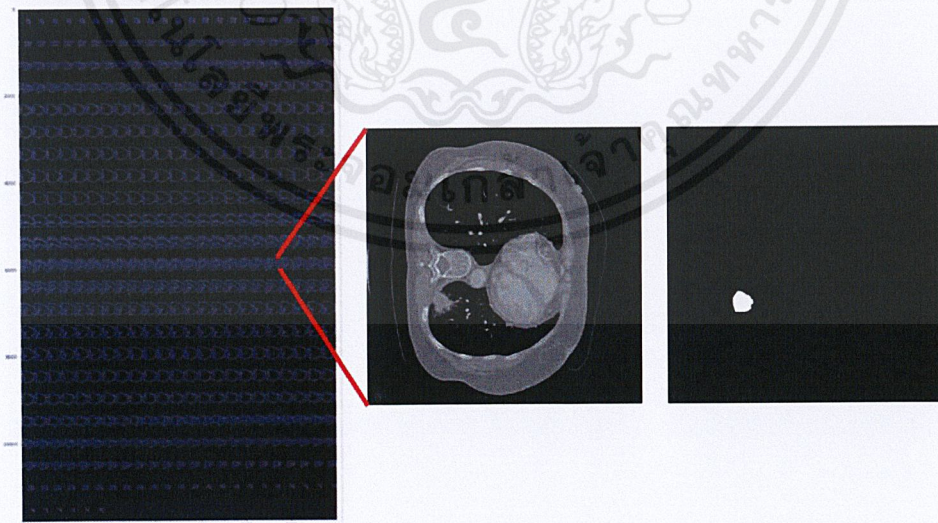
รูปที่ 20 กราฟของการเปรียบเทียบอัตราส่วนของข้อมูล

จากรูปที่ 20 จะเห็นได้ว่าชุดข้อมูลสำหรับการทดสอบ และการฝึกฝนมีความใกล้เคียงกันมากพอสมควรดังนั้นในขั้นตอนของการทดสอบตัวโมเดลการมีโอกาสที่จะวัดผลคลาดเคลื่อนได้จะเป็นไปได้น้อย

3.1.2 ข้อมูลในส่วนของการ segmentation

ข้อมูลในการทำ segmentation ในเริ่มแรกได้ทำการเปรียบเทียบความแตกต่างของประเภทข้อมูลก่อนโดยส่วน เช่น LUNA16 และ medicalDecathlon Dataset[4] ในชุดข้อมูลทั้งสองนี้ความแตกต่างที่ชัดเจนที่สุด คือ นามสกุลของไฟล์ คือ NIFTI และ DICOM ไฟล์ ซึ่งเครื่องมือใน python ที่รองรับการโหลด mesh ให้เป็น 3 มิติจะมีรองรับแค่ใน DICOM เท่านั้นแต่นั้นไม่ใช่ประเด็นสำคัญในการฝึกฝนโมเดล ส่วนสำคัญคือการพิจารณาขนาดของรูปภาพ และปริมาณของรูปภาพใน medicalDecathlon เป็นชุดข้อมูลที่มีชิ้นส่วนของร่างกายให้พิจารณามากกว่า เช่น บริเวณปอด, สมอง หรือ ตับ เป็นต้นแต่ใน LUNA16 จะเป็นแค่ข้อมูลบริเวณปอดเท่านั้นทำให้ตัดสินใจเลือกใช้ชุดข้อมูล medicalDecathlon เพื่อที่เมื่อต้องการให้ตัวโมเดลสามารถที่จะตรวจหาก้อนเนื้อที่บริเวณอื่นได้อีกจะได้สามารถใช้โปรแกรมตัวเดิมได้ในการฝึกฝนได้เลย ดังนั้นผลลัพธ์สุดท้ายของการทำนายจะเป็น binary class คือ '1' และ '0' ว่าตรงไหนที่เป็นเนื้องอกและไม่เป็นเท่านั้น

สำหรับรูปภาพสามมิติในที่นี้จะป็นรูปที่เกิดจากภาพ 2 มิติหลายๆ รูปซ้อนกันจนเกิดเป็นก้อนขึ้นมา ในแต่ละไฟล์จะมีความหนาแตกต่างกันออกไป



รูปที่ 21 ลักษณะของชุดข้อมูล medical decathlon 3 มิติ

3.1.3 ศึกษาการใช้งานของเครื่องมือ

3.1.3.1 การใช้งานของ Pytorch

Pytorch เป็นเครื่องมือที่ใช้ในการพัฒนาโมเดลสำหรับการเรียนรู้เชิงลึก ซึ่งมีต้นแบบจาก Torch ที่ใช้ภาษาในการพัฒนาคือ Lua โดย Pytorch จะถูกสร้างให้มีความคล้ายคลึงกับ Numpy แต่ในการประมวลผลจะใช้ tensors แทน numpy arrays ความแตกต่างคือ tensors สามารถที่จะส่งไปประมวลผลบน Graphic Processing Unit (GPU) ได้และความที่มันเป็น dynamic computational graph (โครงสร้างข้อมูลที่เป็นกราฟการดำเนินการด้วยวิธีการทางคณิตศาสตร์ในแต่ละโหนด) ทำให้การจัดการตัว variables (ที่อยู่ของ parameters ที่อยู่ภายในตัวโมเดล) ใน graph ทำได้ง่ายผิดกับ tensorflow ที่เป็นการกำหนดแบบ static computational graph แต่ในส่วนที่อธิบายมานี้ไม่ได้มีความสำคัญมากนักกับการทำงานด้วยรูปภาพทำให้การปรับค่า variable ระหว่างการรันนั้นไม่จำเป็นมากนักเพราะข้อมูลที่เข้ามาส่วนใหญ่จะถูกเตรียมการให้มีความเหมาะสมอยู่ก่อนแล้ว คือ ขนาดโดยทั่วไปจะเท่ากัน แต่ในการประมวลผลภาษามันเป็นไปไม่ได้ที่จะทำให้ประโยคมีความยาวเท่ากันตลอดเวลาเทคนิคอย่างง่าย คือ การตัดคำในประโยคทิ้งไป แต่ด้วยความสามารถของการ define-by-run ทำให้มันไม่จำเป็นที่จะต้องทำลายข้อมูลไปแบบนั้น

การศึกษาเครื่องมือที่ใช้กับการทำงานของไฟล์รูปภาพที่เหมาะสมกับการทำงานของ medicalDecathlon ในส่วนแรกได้เริ่มจากการใช้เครื่องมือ Medicaltorch ซึ่งเป็น high-level API ที่เขียนบน Pytorch อีกทีในเครื่องมือนี้สำหรับการทำงานของเครื่องมือจะทำการแยกทำนายผลจากแต่ละ slice ของไฟล์รูปภาพในแบบ 2 มิติข้อดีของมันก็คือมีการเตรียมฟังก์ชันสำหรับการวัดผล และมีฟังก์ชัน transform (data augmenter) ที่สามารถใช้งานกับรูปภาพทางการแพทย์ได้เหมาะสม หลังจากที่ได้ทำการ training ตัวโมเดลไปมากพอสมควรและได้ทำการปรับค่าพารามิเตอร์ให้มีความเหมาะสม (hyperparameter tuning) ด้วยการทำ grid search และเห็นว่าถึงจุดที่ไม่สามารถจะเรียนรู้เพิ่มได้อีกแล้ว (loss ลู่เข้าค่าได้อย่างเห็นได้ชัด) เมื่อวิธีการทางการเรียนรู้ไม่สามารถทำให้เกิดการเรียนรู้เพิ่มได้อีกจึงต้องมีการนำเทคนิคที่เปลี่ยนวิธีการเรียนรู้โดยตรงด้วยการทำ transfer learning แบบ fine-tuning เนื่องจากตัวโมเดลต้นแบบ (pre-trained model) ไม่ได้ถูกฝึกฝนมาให้เหมาะสมกับรูปสแกนทางการแพทย์ที่เป็น grayscale และมี feature ที่แตกต่างกันมากใน pre-trained model จาก Pytorch ที่ถูกฝึกไว้บนชุดข้อมูลของ COCO หรืออื่นๆ ซึ่งไม่ได้มีความใกล้เคียงกับภาพทางการแพทย์ทำให้ผลลัพธ์ที่ออกมาได้ไม่เป็นอย่างที่สมควร

3.1.3.2 การใช้งานของ Tensorflow

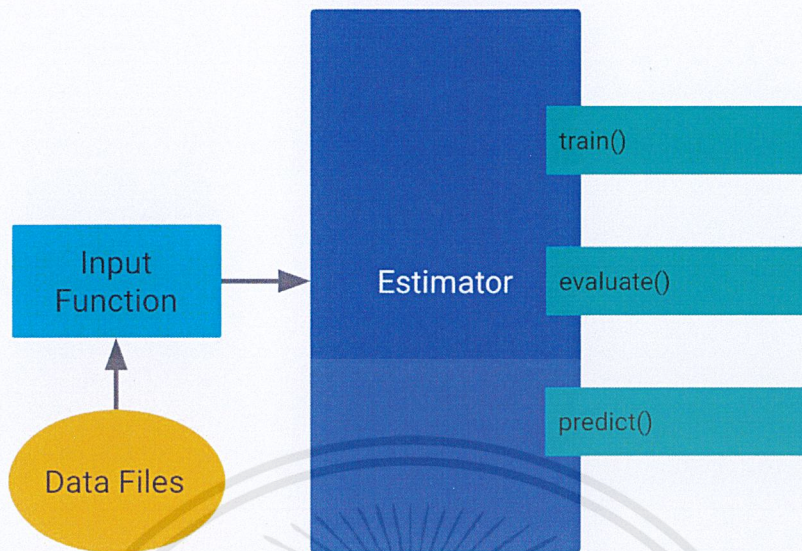
Tensorflow เป็นอีกหนึ่ง deep learning framework ที่ถูกพัฒนาขึ้นมาด้วย google และเป็น framework สำหรับการเรียนรู้เชิงลึกที่ได้รับความนิยมสูงสุดในปัจจุบัน โดยการทำงานของ Tensorflow เป็นแบบ static computational graph คือต้องมีการสร้างกราฟทั้งหมดก่อนแล้วค่อยรันภายในทีเดียว นั้นทำให้เป็นเรื่องยากที่จะแก้ไขได้เมื่อขั้นตอนของตัวโครงข่ายมีปัญหาแต่ตัว Tensorflow เองก็มี high-level API เช่น Keras, estimator เป็นต้น ซึ่งทำให้การทำงานง่ายขึ้นอย่างมากเนื่องจากมันสามารถรันแบบ eager ได้ทำให้การเขียนโค้ดแบบ boilerplate (โค้ดที่ต้องเขียนเป็นจำนวนมากเพื่อให้ได้ผลลัพธ์ที่น้อยนิด) มีโอกาสเกิดขึ้นน้อย แก้ไขบั๊กที่เกิดขึ้นได้ง่ายขึ้นและกำหนดโครงสร้างได้ง่ายมากขึ้น

ในการทำงานครั้งนี้ได้มีการเรียกใช้ high-level API Estimator ซึ่งเป็น API ที่อยู่ภายใน Tensorflow ที่มีการห่อหุ้มฟังก์ชันสำหรับการทำงาน ดังนี้

- การฝึกฝนตัวโมเดล
- การทดสอบตัวโมเดล
- การทำนายผลโดยเรียกใช้โมเดลที่ได้จากการฝึกฝน
- การส่งออกตัวโมเดลเพื่อการใช้งาน

ส่วนของการทำงาน estimators ที่เด่นๆ เช่น บันทึกผลลัพธ์ทั้งหมดลงบน Tensorboard ได้เลย บันทึก checkpoint และมี WarmStartSettings ทำให้การทำ fine-tuning สะดวกมากยิ่งขึ้น ส่วนของการโหลดข้อมูลมาเพื่อฝึกฝนสามารถแบ่งเป็นข้อมูลสำหรับฝึกฝนและทดสอบได้ด้วยการกำหนดเพียงแค่ key parameter เท่านั้นและด้วยความที่มันแยกส่วนออกมาโดยสมบูรณ์การจะเปลี่ยนชุดข้อมูลระหว่างการ train ก็เลยเป็นไปได้ด้วย

```
def read_fn(file_references, mode, params=None):  
# ให้อ่านไฟล์สำหรับทั้ง cycle ของการทำงาน  
# file_references เป็นพารามิเตอร์สำหรับเก็บไฟล์ที่จะใช้ในการ train และ validate  
# mode เป็นตัวกำหนด key คือ TRAIN, EVAL, PREDICT  
# params คือ การกำหนดว่า output จะเป็นแบบไหนพวก ขนาดรูปภาพ เป็นต้น
```



รูปที่ 22 การทำงาน Estimator [23]

Scaffold และ SessionHooks เป็น object ที่มีความสำคัญในการทำงานของ estimator สำหรับ scaffold เป็นตัว initialize config โมเดลในส่วนของการส่งกลับค่าของ EstimatorSpec ในตอนฝึกฝนตัวโมเดล

```

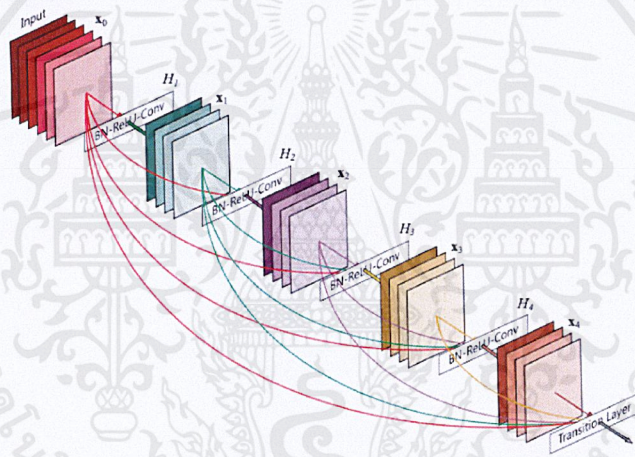
return tf.estimator.EstimatorSpec(mode=mode,
                                   predictions=net_output_ops,
                                   loss=loss,
                                   train_op=train_op,
                                   eval_metric_ops=None,
                                   )
  
```

ในตัวอย่างจะเห็นได้ว่าการส่งกลับค่าต่างๆ ที่เกิดขึ้นระหว่างการฝึกฝนตัวโมเดล ดังที่ได้กล่าวไปแล้วข้างต้นว่าในกรณีที่มีการเริ่มต้นจาก checkpoint จะต้องมีการกำหนดค่า weight ต่างๆ มาก่อนจะทำการ fine-tuning SessionRunHooks จะต่างกับ scaffold ตรงที่มันสามารถเรียกใช้ได้ทั้งใน training, validating และ testing สำหรับกรณีที่มันมี argument เรียกใช้มากกว่าหนึ่งมันจะดึง method จาก super class มา override เช่น ฟังก์ชัน before_run, after_create_session เป็นต้น

3.1.4 การศึกษาโมเดลที่มีความเหมาะสมกับการทำงานแต่ละส่วน

3.1.4.1 โมเดลสำหรับการจัดกลุ่มของรูปภาพบริเวณทรวงอก

ในการดำเนินการได้ทำการสืบค้นถึงตัวโมเดลที่มีความสามารถในการจัดกลุ่มกับรูปภาพเอกซเรย์บริเวณทรวงอก จากงานวิจัย ChexPert ได้มีการทดสอบกับโมเดลที่หลากหลาย จากการทดลองพบว่าโมเดลที่มีประสิทธิภาพกับชุดข้อมูลมากที่สุด คือ DenseNet ซึ่งเป็นโมเดลที่มีประสิทธิภาพสูงด้วยการวัดค่าจาก AUC สูง ตัว DenseNet ให้ผลลัพธ์ในการเอาชนะปัญหา Vanishing-gradient (ปัญหาที่เกิดในการเรียนรู้แบบต่อเนื่องใน Residual Network เวลาที่มีการเรียนรู้ออนกลับแต่ค่า gradient ต่ำมากๆ จนทำให้มันเข้าสู่ศูนย์) ด้วยความที่มันเป็นโมเดลแบบ feature reuse ทำให้จำนวนพารามิเตอร์ที่ต้องใช้ลดจำนวนลงไปด้วย โมเดลนี้เป็นโมเดลที่มีประสิทธิภาพสูงสุดโมเดลหนึ่งด้วยการคำนึงถึง SOTA (state-of-the-art) เพียงอย่างเดียว



รูปที่ 23 Densely Connected Convolutional Networks [1]

จากรูปภาพจะเห็นได้ว่าการเรียนรู้ของโมเดลจะเกิดการส่งต่อ feature ตั้งแต่ชั้นแรกตลอดไปจนถึงชั้นสุดท้าย ถึงแม้ว่าจะเกิดการเรียนรู้ที่ดูเหมือนจะมากจนอาจทำให้เกิดปัญหา overfitting นั้นทำให้โมเดลของ DenseNet จะต้องมีการใช้ parameters จำนวนน้อยๆ ในการทำงานได้มีการนำเข้าโมเดลและพารามิเตอร์ที่ได้มีการทำการฝึกฝนไว้แล้วกับชุดข้อมูลที่เป็น RGB

จากที่ได้กล่าวมาทั้งหมดเนื่องจากงานนี้เป็นงานที่ใช้ชุดข้อมูลซึ่งถูก labels ด้วยอาการที่เกิดขึ้นในรูปนั้นเพียงแค่นั้น ทำให้พิจารณาได้ยากกว่าอาการมีโอกาสจากตรงไหนบ้าง ผู้จัดทำจึงได้เพิ่มการ

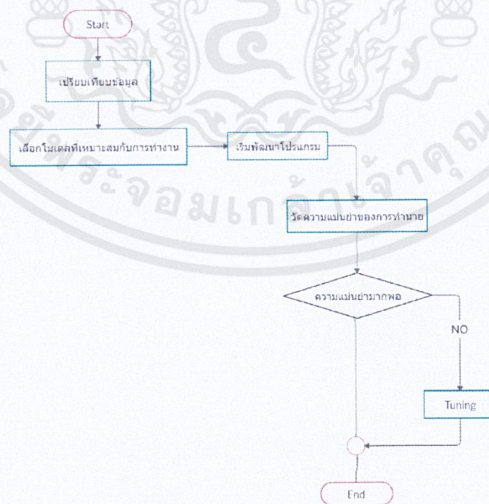
visualize ข้อมูลผลลัพธ์ด้วย activation map (ดึงเอา feature จาก convolution ชั้นสุดท้ายออกมาพลอตเป็น heatmap)

3.1.4.2 โมเดลสำหรับการแยกส่วนภาพทางการแพทย์สามมิติ

อย่างที่ได้อธิบายไปแล้วข้างต้นว่าชุดข้อมูลที่น่าสนใจมีสองปัญหาหลักด้วยกัน คือ class imbalance และชุดข้อมูลมีขนาดเล็กดังนั้นการศึกษโมเดลที่จะนำมาใช้ในงานนี้จำเป็นต้องมีความสามารถในการรับมือกับปัญหาพวกนี้ได้ดีด้วย ซึ่งในความเป็นจริงแล้วการแยกส่วนมีโมเดลที่หลากหลาย เช่น Mask-RCNN ที่สามารถทำในแบบ Semantic Segmentation ได้แต่ในภาพทางการแพทย์ที่เป็นแบบ grayscale และชุดข้อมูลที่มีเพียงแค่สองคลาส คือ 0 (ไม่ใช่เนื้องอก) และ 1 (เป็นเนื้องอก) ดังนั้นโมเดลที่มีความเหมาะสมมากที่สุดจะเป็น โมเดล U-Net เนื่องจากการที่มันสามารถจัดการกับปัญหาทั้งสองแบบได้ดีเมื่อมีการทำ data augmentation เยอะๆ ซึ่งส่วนสำคัญที่สุดก็คือด้านขวาของตัวโมเดล หรือก็คือส่วน up-convolution ที่ได้มีการทำ concatenation features จากส่วน down convolution ซึ่งเป็นลักษณะของ Res-CNN

3.1.5 ผังการทำงานของโปรแกรมปัญญาประดิษฐ์

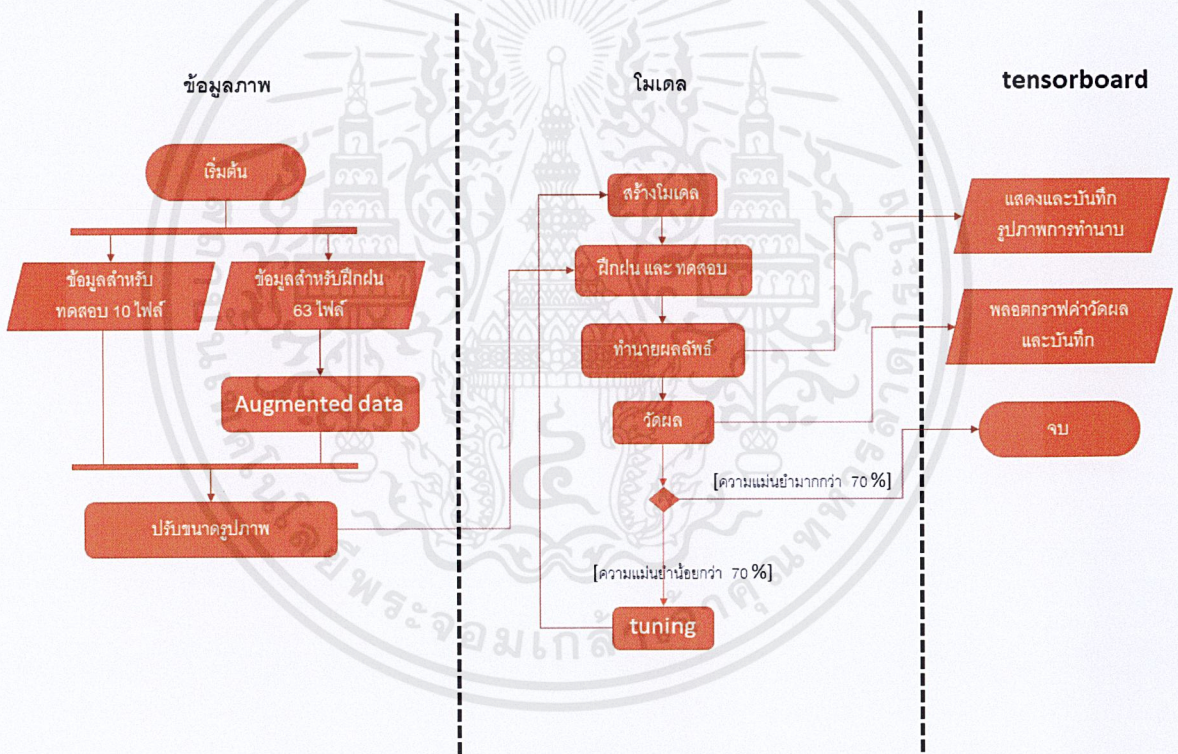
3.1.5.1 แผนผังการทำงานของเครื่องเรียนรู้ด้วยเครื่อง 2 มิติ



รูปที่ 24 การทำงานโดยรวมของโปรแกรมให้เกิดการเรียนรู้ด้วยเครื่อง

รูปที่ 24 คือการอธิบายอย่างกว้างของการตัดสินใจในแต่ละขั้นตอนสำหรับการทำงานในการเรียนรู้ด้วยเครื่อง เริ่มแรกสุดอย่างทีอธิบายไปแล้วว่าชุดข้อมูลแต่ละชุดจะมีความแตกต่างกันเล็กน้อย ดังนั้นจึงต้องมีการเปรียบเทียบเพื่อตัดสินใจว่าชุดข้อมูลใดที่มีความเหมาะสมกับทรัพยากรและเครื่องมือ หลังจากจากนั้นจึงค้นหาโมเดลที่มีความเหมาะสมกับงาน เช่น FCN ที่เหมาะกับการแยกส่วนเหมือนกับ U-Net แต่ด้วยความที่ชุดข้อมูลมี class imbalance สูงทำ U-Net มีความเหมาะสมมากกว่า หลังจากนั้นเป็นการพัฒนาโปรแกรมสำหรับการฝึกฝนโมเดลตั้งแต่การเตรียมความพร้อมข้อมูล (pre-processing) และผลลัพธ์ที่ได้จากการฝึกฝนของตัวโมเดลและส่วนสุดท้ายคือพิจารณาผลลัพธ์ที่ได้ว่าควรพัฒนาเพิ่มขึ้นไปอีกหรือไม่

3.1.5.2 แผนผังการทำงานของเครื่องเรียนรู้ด้วยเครื่อง 3 มิติ



รูปที่ 25 ผังโปรแกรมการเรียนรู้ด้วยเครื่องกับภาพ 3 มิติ

จากรูปที่ 25 เป็นแผนภาพการทำงานของตัวโปรแกรมโดยละเอียดตั้งแต่ส่วนของ ข้อมูลส่วนแรกสุดคือการเตรียมความพร้อม จากข้อมูลทั้งหมด 73 ไฟล์จะแบ่งออกเป็นข้อมูลสำหรับการฝึกฝนจำนวน 63 ไฟล์ และข้อมูลสำหรับทดสอบจำนวน 10 ไฟล์ แล้วทำการเพิ่มจำนวนข้อมูล (augmented data) เพื่อให้เป็นไปตามการใช้งานที่มีประสิทธิภาพของโมเดล U-Net และเพื่อข้อมูลสำหรับการฝึกฝนที่มากขึ้น

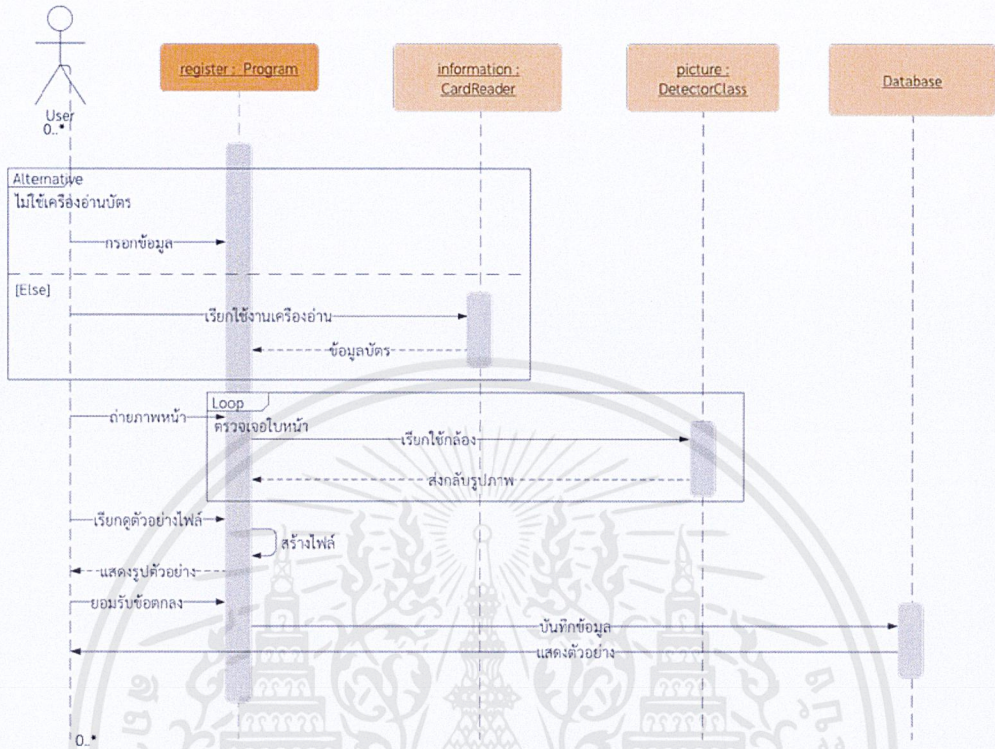
จากเดิมที่มีเพียงแค่เล็กน้อย ทั้งนี้ในส่วนสุดท้ายของ Data จะต้องมีการปรับขนาดรูปภาพ (resize) เสมอ เพื่อให้ส่วนสำคัญที่ได้ออกมาเป็นไปในทิศทางเดียวกันทั้งหมด ในส่วนของโมเดลเริ่มแรกสุดจากการสร้างโมเดล โดยในการทำงานของการแบ่งส่วนเนื้องอกบริเวณปอดจากรูปภาพ 3 มิติ โมเดลที่ใช้ คือ U-Net และ dice loss สำหรับการเรียนรู้ย้อนกลับผ่าน Adam optimizer เมื่อทำการฝึกฝนและทดสอบโมเดลไปพร้อมๆ กันในส่วนนี้โปรแกรมจะแปลง tensors มาเป็นรูปภาพเพื่อบันทึกลงไป Tensorboard และพลอตค่า loss และ learning rate เมื่อได้ตรวจสอบผลลัพธ์ทั้งหมดแล้วจะเป็นการพิจารณาเพื่อปรับแก้ไขบางส่วนเพื่อให้โมเดลสามารถที่จะเรียนรู้ได้มากขึ้นทำนายผลได้แม่นยำมากขึ้น โดยการทำให้ Hyperparameter tuning และ fine-tuning

3.2 การดำเนินงานในส่วนของการซอฟต์แวร์ลงทะเบียนและใช้งานโมเดล

3.2.1 ผังการทำงานของโปรแกรมสำหรับจัดการไฟล์

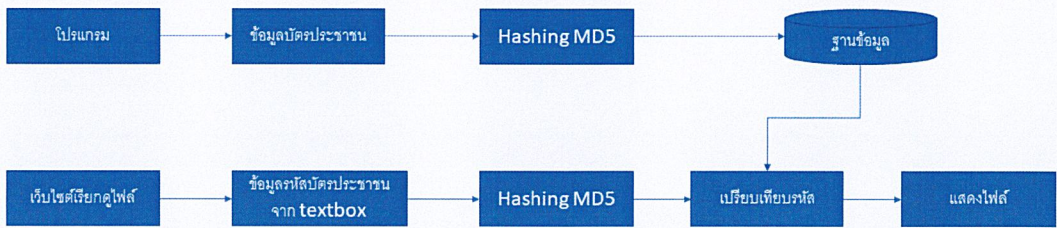
ในรูปที่ 26 คือลำดับการทำงานของโปรแกรมที่ใช้ในการทำงานของโปรแกรมสำหรับจัดการไฟล์ เอกสารสำหรับคำยินยอม เมื่อเริ่มต้นโปรแกรมขึ้นมาเราจะพบกับหน้าต่างของผู้ใช้งานที่มีการเก็บ ชื่อและนามสกุลผู้ป่วย รหัสผู้ป่วยของโรงพยาบาลและ รหัสบัตรประชาชนของผู้ป่วย โดยข้อมูลในส่วนนี้ได้มีการเขียนโปรแกรมให้ติดต่อกับเครื่องสแกนบัตรประชาชนไว้แล้วทำให้สามารถที่จะดึงข้อมูลเข้ามาจากบัตรประชาชนได้เลยโดยไม่จำเป็นต้องกรอกเอง เมื่อข้อมูลทั้งหมดถูกรวบรวมไปแล้วตัวบัตรประชาชน ในส่วนของการถ่ายภาพจะมีการเก็บรูปภาพบัตรประชาชนเพื่อใช้ในการเปรียบเทียบเลขบัตรประชาชนบนบัตรกับที่กรอกลงมาในตัวโปรแกรมด้วย เมื่อการทำงานทั้งหมดเสร็จสิ้นปุ่มสำหรับเรียกดูตัวอย่างไฟล์จะเปิดให้กดได้โดยมันจะแสดงรูปภาพตัวอย่างให้ดูบริเวณด้านข้างของ User interface ส่วนสุดท้ายคือการกดยืนยันยอมรับข้อตกลงทั้งหมดก่อนที่จะมีการอัปโหลดลงสู่ฐานข้อมูลได้

Register Software



รูปที่ 26 ลำดับการทำงานของซอฟต์แวร์ลงทะเบียน

โดยในการทำงานของตัวโปรแกรมที่อธิบายข้างต้นเป็นการทำงานของตัวซอฟต์แวร์ผ่าน Java application ซึ่งเมื่อต้องการอัปเดตหรือแก้ไขเพียงเล็กน้อยจะทำได้ยากจึงได้มีการนำตัวแอปพลิเคชันทั้งหมดลงไปในเว็บไซต์เฟรมเวิร์ก Webswing ทำให้ตัว Java application สามารถทำงานผ่านเว็บเบราว์เซอร์ได้เพื่อแก้ไขปัญหาเรื่องความยุ่งยากในการดูแลรักษา เมื่อมีการทำงานของโปรแกรมผ่านเบราว์เซอร์ในฝั่งของการสร้างไฟล์จึงมีการพัฒนาเว็บไซต์สำหรับการเรียกดูไฟล์ การทำงานของเว็บเพจนี้คือรับข้อมูลสำหรับการค้นหาไฟล์มาเป็น ชื่อ หรือ รหัสบัตรประชาชนของผู้ป่วยซึ่งในการทำงานของโปรแกรม ก่อนจะเก็บรหัสบัตรประชาชนจะมีการเข้ารหัสด้วยแฮชฟังก์ชัน MD5 เมื่อมีการเรียกดูไฟล์จึงต้องการแฮชซึ่งเพื่อเปรียบเทียบรหัสที่เก็บไว้ในฐานข้อมูลก่อนเพื่อที่จะได้นำไฟล์ที่ตรงกับที่ต้องการออกมาได้ดังรูปที่ 27



รูปที่ 27 ผังการทำงานของข้อมูลที่มีการเข้ารหัส

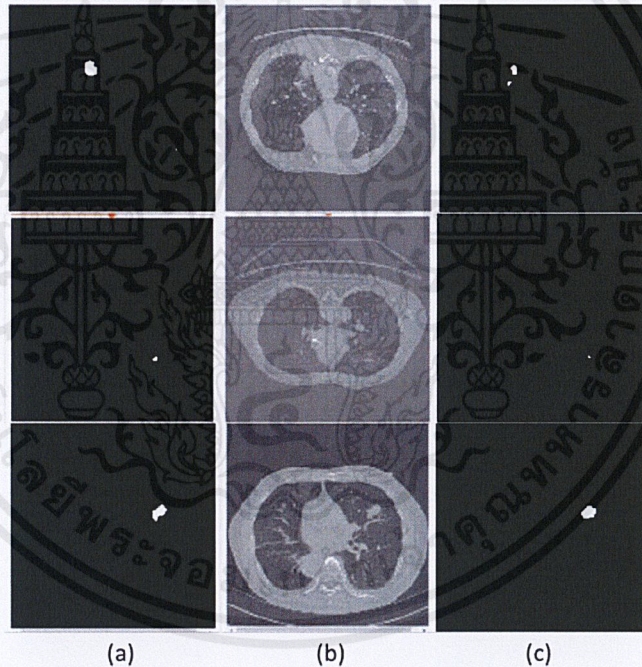


บทที่ 4

ผลลัพธ์การทำงาน

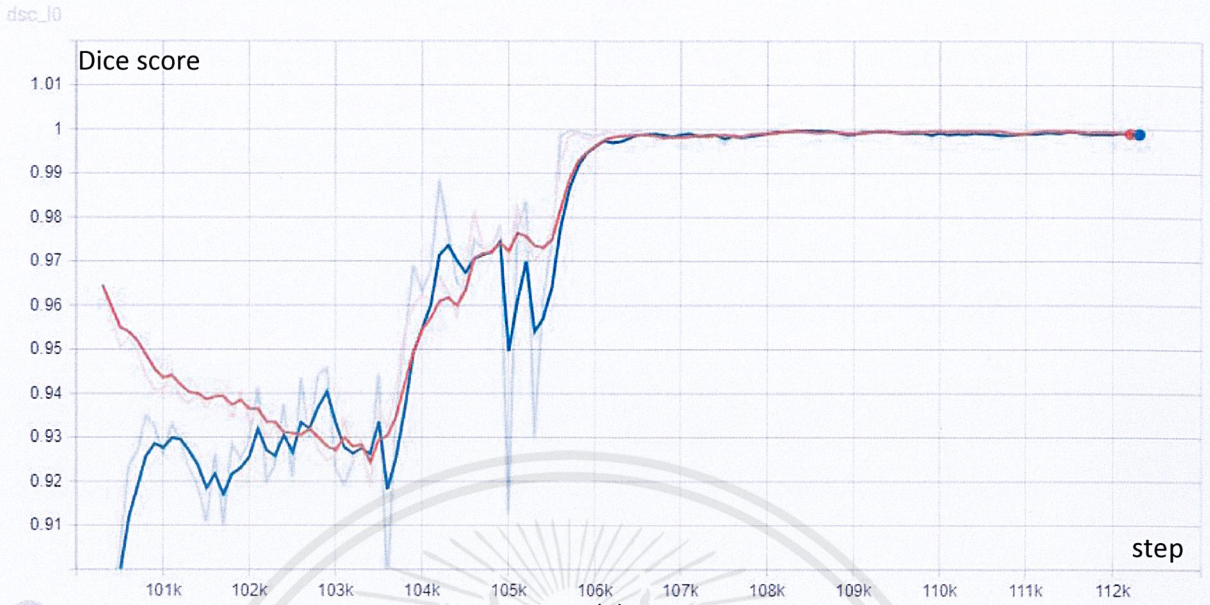
4.1 ผลลัพธ์ของการเรียนรู้ด้วยเครื่องกับการแยกส่วน

ภาพผลลัพธ์จากการทำนายด้วยภาพ 3 มิติโดยในรูปด้านล่าง รูปที่ 28(a) เป็นภาพ mask ที่เป็นค่าความจริง (ground truth) รูปที่ 28(b) เป็นภาพ input ที่ใช้เพื่อให้โมเดลวิเคราะห์และทำนายออกมาว่าส่วนไหนของรูปรูปที่เป็นเนื้องอก รูปที่ 28(c) เป็นผลลัพธ์จากการทำนายของโมเดล (predictions) ในกราฟด้านสุดท้ายเป็น dice score (F1 score) เป็นตัวเลขที่ใช้บ่งบอกถึงความเหมือนกันของขอบเขตผลลัพธ์จากการทำนาย และค่าความเป็นจริง

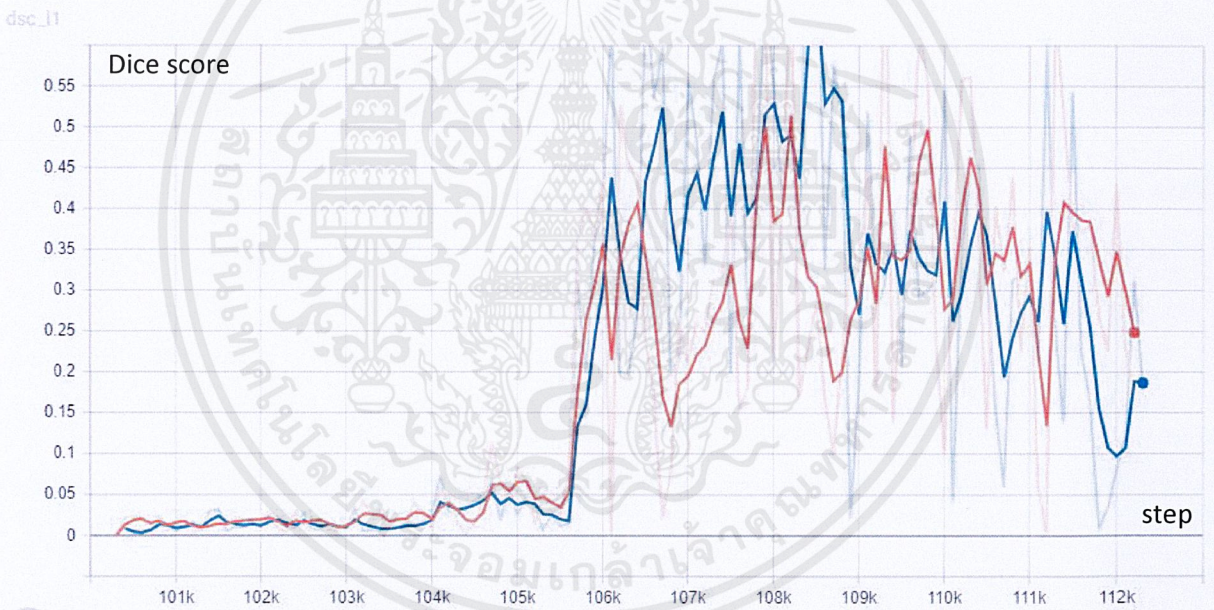


รูปที่ 28 รูปภาพผลลัพธ์จากการแบ่งส่วน

(a):ภาพแยกส่วนตามจริง (b):ภาพอินพุต (c):ภาพแยกส่วนจากการทำนาย



(a)



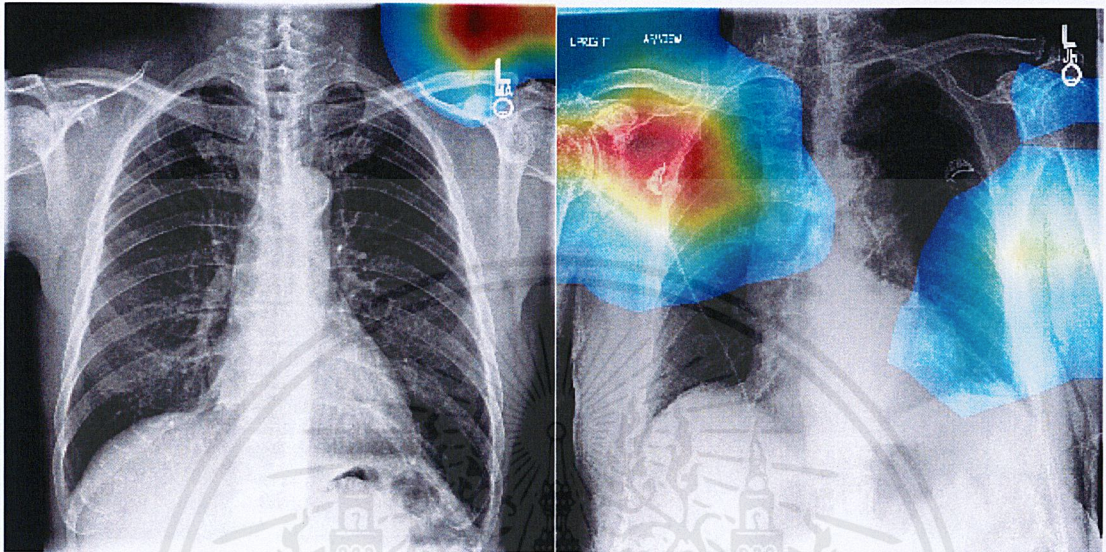
(b)

รูปที่ 29 กราฟ dice score ของการแยกส่วนภาพ 3 มิติในแต่ละ step

ผลลัพธ์ของการทำงานการแยกส่วนเนื้องอกหลังจากการใช้เทคนิค transfer learning เห็นได้ว่าผลลัพธ์โดยรวมหลังทำการยกเลิก freeze ตัวโมเดลความค่า dice score ของคลาส 0 รูปที่ 29(a) จะอยู่ที่เกือบ 100% ในขณะที่คลาส 1 รูปที่ 29(b) จะกระจายอย่างไม่มีนัยสำคัญแสดงถึงตำแหน่งที่ถูกตัดแต่

ขอบของรูปร่างคลาดเคลื่อนอยู่บ้างโดยผลลัพธ์ (output) ทั้งหมดจะมีขนาดเล็กกว่าค่าความจริง (ground truth) ดังแสดงในรูปที่ 28

4.2 ผลลัพธ์ของการทำงานในการจำแนกประเภทของอาการในปอด จากภาพเอกซเรย์ 2 มิติ

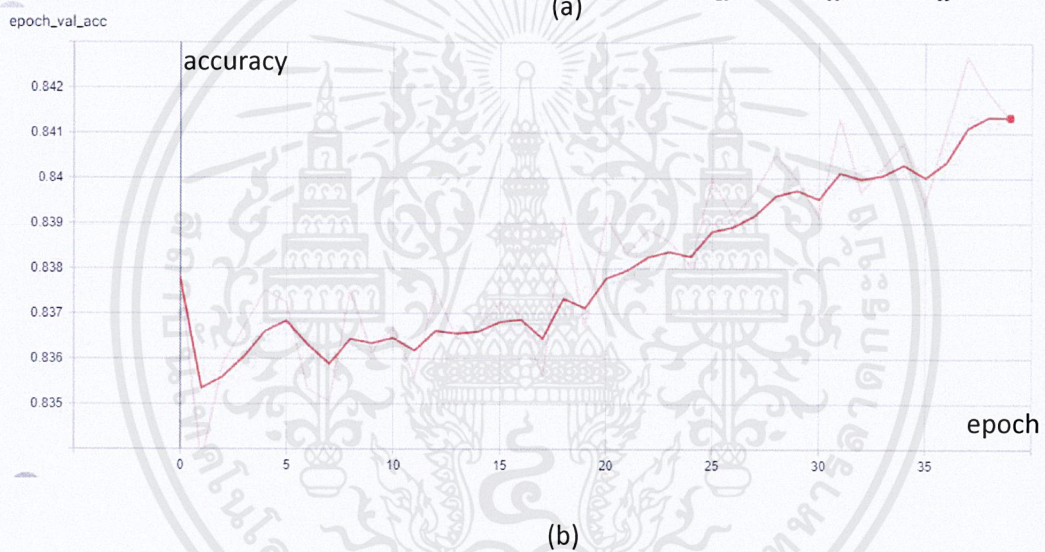
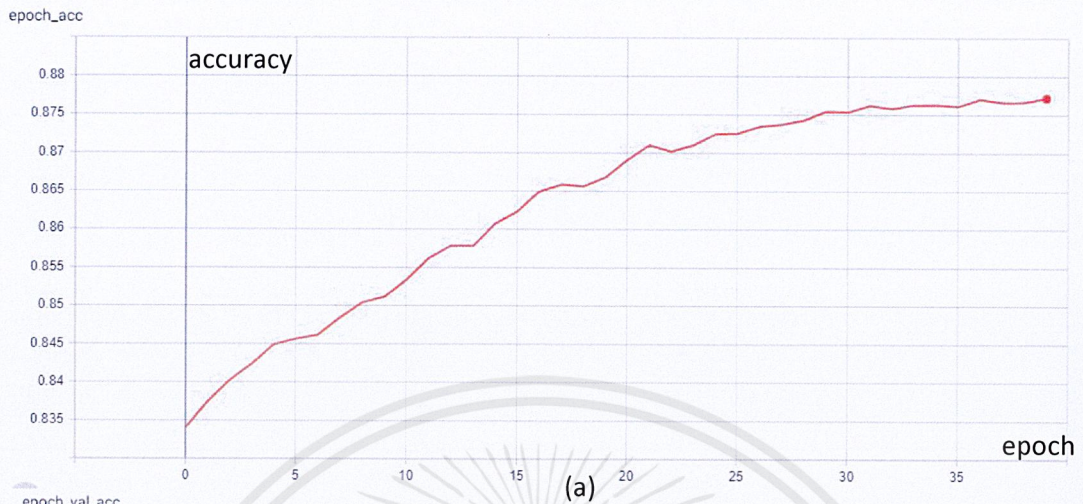


(a) (b)
รูปที่ 30 ภาพของ convolution activation map

จากรูปที่ 30 แสดงถึงผลลัพธ์หลังจากการใช้เทคนิค Class Activation Map (CAM) ซึ่งเป็นการนำภาพจาก convolution layer สุดท้ายในเน็ตเวิร์คออกมาทำการพลอตซ้อนทับกับรูปภาพอินพุต ดังนั้นเราจะสามารถรับรู้ได้ถึงจุดที่โมเดลให้ความสนใจรูปภาพจนนำไปสู่การตัดสินใจของผลลัพธ์นั้นๆ ได้อย่างไรก็ตามแต่ผลลัพธ์นี้ยังไม่สามารถนำไปถึงขั้นที่ใช้ตัดสินใจได้ว่าโมเดลจะทำนายผลลัพธ์ออกมาผิดหรือถูกเพราะในบางครั้งผลลัพธ์ของ CAM ก็ไม่แสดงออกมาเลยในทางกลับกันผลลัพธ์จากการจัดกลุ่มกับถูกต้อง

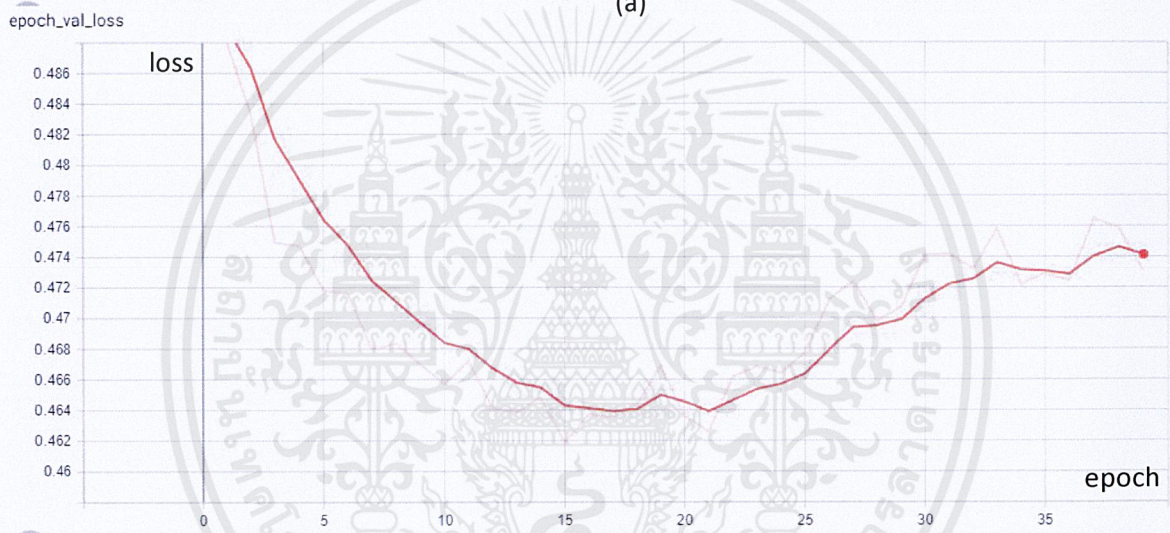
ในส่วนของรูปที่ 30(a) จะแสดงถึง CAM บนคลาส No findings จะเห็นได้ว่าโมเดลให้ความสนใจไปบนตำแหน่งที่ไม่มีนัยสำคัญใดๆ รูปที่ 30(b) เป็น CAM ของผลลัพธ์ในคลาส Support devices ซึ่งโมเดลให้ความสนใจไปที่บริเวณของอุปกรณ์ที่ติดอยู่บริเวณช่วงอก

จากรูปที่ 31 คือผลลัพธ์ที่ได้จากการทำนายความผิดปกติที่เกิดขึ้นจากรูปเอกซเรย์ 2 มิติความแม่นยำโดยรวมจากทุกคลาสขณะฝึกฝนอยู่ที่ 87% รูปที่ 31(a) สำหรับการวัดผลในส่วนของ การฝึกฝนตัวโมเดลในขณะที่ความแม่นยำที่เกิดขึ้นจากส่วนของการทดสอบจะอยู่ที่ 84% รูปที่ 31(b)



รูปที่ 31 ความแม่นยำการทำนายในแต่ละ epoch

ค่า loss สำหรับโมเดล DenseNet เป็นดังที่แสดงในรูปที่ 32(a) เป็น loss ที่เกิดขึ้นระหว่างการฝึกฝนส่วนด้านล่าง คือค่า loss ที่เกิดขึ้นระหว่างการทดสอบโดยรูปทั้งสองนี้มีความสำคัญสำหรับการพิจารณา overfitting จากรูปภาพจะสังเกตเห็นได้ว่าค่าความผิดพลาด (loss) ที่เกิดขึ้นในตอนทดสอบในรูปที่ 32(b) กลับมีค่าเพิ่มขึ้นมาในช่วงการทดสอบประมาณ epoch ที่ 25 หรือก็คือโมเดลเริ่มที่จะเกิดการ overfit แล้วนั่นเองทำให้การกำหนด early-stop จะเกิดขึ้นใน epoch ที่ 35



รูปที่ 32 ค่า loss ที่ได้จากการทำนายแต่ละ epoch

สำหรับผลลัพธ์ความแม่นยำของโมเดลหลังจากที่ทำการฝึกฝนเรียบร้อยแล้วและต้องการนำขึ้นใช้งาน โดยการพิจารณาแยกคลาสจะได้ความแม่นยำสำหรับคลาสจะเป็นดังรูปที่ 33 โดยที่คลาส Pleural Other จะมีความแม่นยำสูงที่สุดที่ 98.97% และความแม่นยำต่ำสุด คือคลาส Support Devices ที่ 55.59% ด้วยการพิจารณาจากข้อมูลโดยรวม 2612 ภาพ แถวบนสุดของรูปที่ 33 คือ จำนวนของรูปที่โมเดลทำนาย ถูกจากทั้งหมด 2612 ภาพ

รูปที่ 34 เป็นผลลัพธ์ความแม่นยำของโมเดลหลังจากที่ทำการฝึกฝนเรียบร้อยแล้วและต้องการนำขึ้นใช้งานโดยการพิจารณาแยกคลาสเฉพาะข้อมูลที่โมเดลไม่เคยเรียนรู้มาก่อนจำนวน 262 รูปภาพ จะได้ความแม่นยำสำหรับคลาส Pleural Other สูงที่สุดที่ 97.71% และความแม่นยำต่ำสุด คือคลาส Lung Opacity ที่ 53.44% แถวบนสุดของรูปที่ 34 คือจำนวนของรูปภาพที่ทำนายถูกจากทั้งหมด 262 ภาพ

[2538, 2527, 2348, 1667, 2539, 2102, 2504, 2565, 2332, 2430, 1771, 2585, 2543, 1452]
No Finding 97.17
Enlarged Cardiomediatinum 96.75
Cardiomegaly 89.89
Lung Opacity 63.82
Lung Lesion 97.21
Edema 80.47
Consolidation 95.87
Pneumonia 98.2
Atelectasis 89.28
Pneumothorax 93.03
Pleural Effusion 67.8
Pleural Other 98.97
Fracture 97.36
Support Devices 55.59

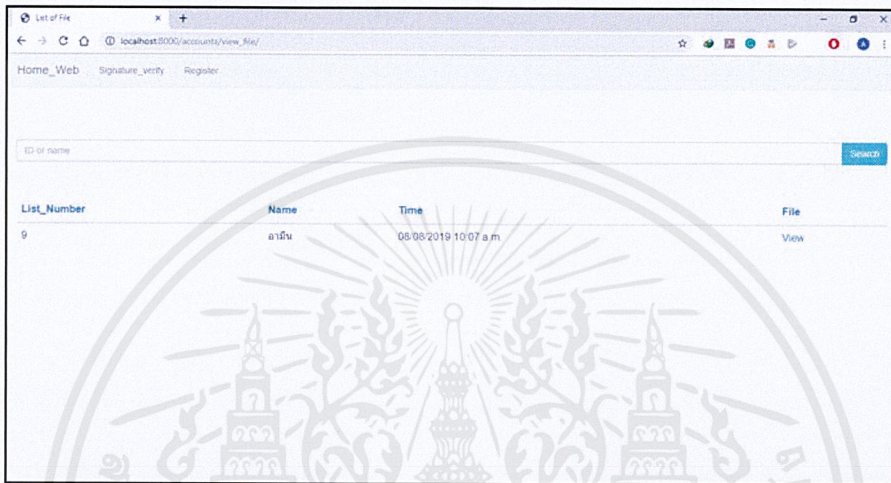
รูปที่ 33 ความแม่นยำจากการทำนายโดยรวม

[240, 241, 237, 140, 249, 222, 243, 251, 219, 226, 158, 256, 243, 154]
No Finding 91.6
Enlarged Cardiomediatinum 91.98
Cardiomegaly 90.46
Lung Opacity 53.44
Lung Lesion 95.04
Edema 84.73
Consolidation 92.75
Pneumonia 95.8
Atelectasis 83.59
Pneumothorax 86.26
Pleural Effusion 60.31
Pleural Other 97.71
Fracture 92.75
Support Devices 58.78

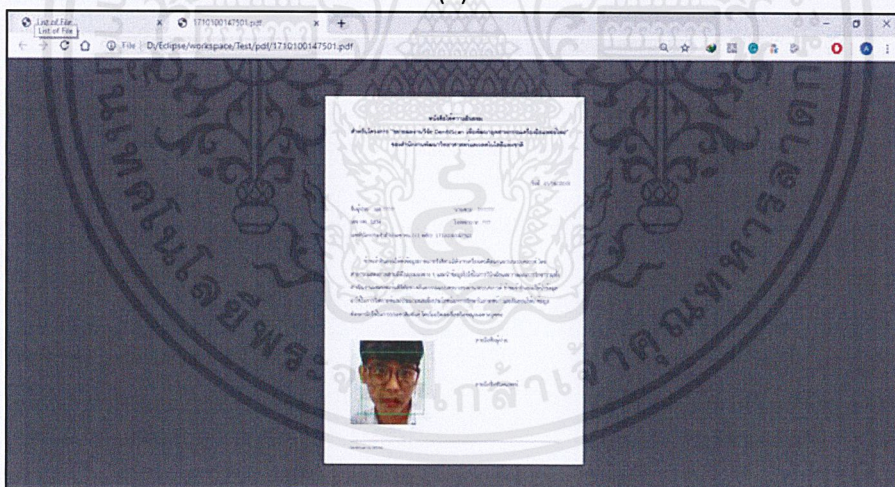
รูปที่ 34 ความแม่นยำของการทำนายจากโมเดลจากรูปที่ไม่เคยเห็นมาก่อน

4.3 ผลลัพธ์ของการใช้งานโปรแกรมสร้างไฟล์เงื่อนไขยอมรับ

ผลลัพธ์การทำงานของซอฟต์แวร์โดยรวมเป็นไปตามที่สร้างไว้ในแผนงานซึ่งเริ่มตั้งแต่การอ่านบัตรประชาชนไปตลอดจนถึงการบันทึกข้อมูลไฟล์เงื่อนไขยอมรับตาม database รูปที่ 35(a) คือหน้าเว็บไซต์ในการเรียกดูไฟล์ที่ผู้ป่วยได้ลงทะเบียนไปเรียบร้อยแล้ว รูปที่ 35(b) คือการเรียกดูไฟล์ลงทะเบียนของผู้ป่วยแบบ PDF ผ่านเว็บเบราว์เซอร์



(a)



(b)

รูปที่ 35 เว็บไซต์เรียกดูไฟล์เงื่อนไขยอมรับ และภาพของการเรียกดูไฟล์ PDF จากเพจเรียกไฟล์

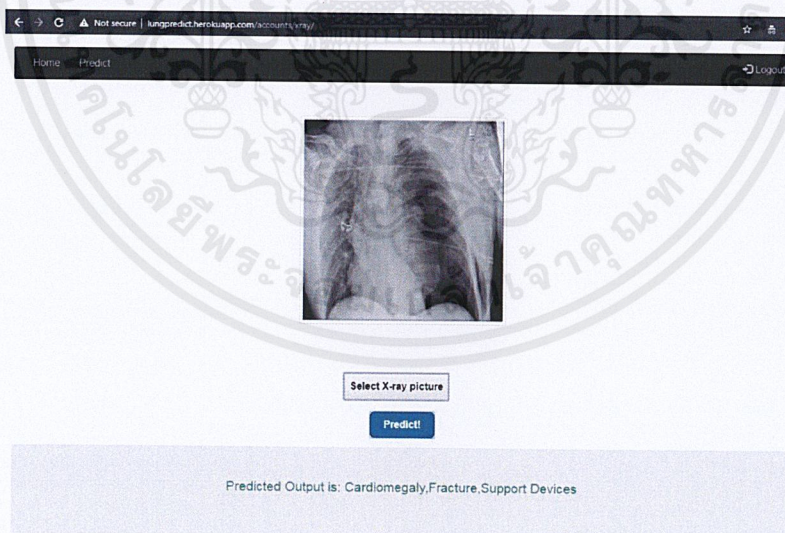
โดยผลการทำงานของ การตรวจจับใบหน้าได้ทำการทดลองความสามารถในการตรวจจับใบหน้าโดยปัจจัยหลักๆ ที่ให้ความสนใจ คือ ความสว่างโดยรอบตัวกล้อง และระยะห่างของกล้องกับใบหน้าที่จะตรวจจับได้ ผลลัพธ์ ดังรูปที่ 36 โดยที่ความสว่างของห้องทำงานโดยปกติจะอยู่ที่ 250-300 lx และความเร็วในการตรวจจับขึ้นอยู่กับความละเอียดของตัวกล้องด้วย ด้วยการทดลองผ่านกล้องที่มีความละเอียดสองระดับ คือ 720p และ 1080p ในกล้องที่มีความละเอียดต่ำกว่าจะสามารถทำการตรวจจับใบหน้าได้รวดเร็วกว่า

ระยะห่าง ความสว่าง	30 เซนติเมตร	60 เซนติเมตร	90 เซนติเมตร
15 – 20 lx	ไม่พอ	ไม่พอ	ไม่พอ
40 – 50 lx	พอ	ไม่พอ	ไม่พอ
80 – 90 lx	พอ	พอ	พอ

รูปที่ 36 ความสามารถในการทำงานของโปรแกรมส่วนถ่ายภาพหน้า

4.4 ผลลัพธ์การทำงานของตัวอย่างเว็บเพจเรียกใช้งานโมเดล DenseNet

หลักการการทำงานของรูปที่ 37 จะเป็นการทำงานของเว็บไซต์จะเป็นการเรียกใช้โมเดลที่ถูกจัดทำขึ้นเป็น API (Application Programming Interfaces) เมื่อมีการส่งรูปภาพเข้าไปประมวลผลโปรแกรมจะส่งค่าสุดท้ายกลับมาเป็น JSON response แสดงความผิดปกติที่เจอ ด้วยความที่เป็นโมเดลจากปัญญาประดิษฐ์ดังนั้นเมื่อ feature ที่ input เข้าไปมีลักษณะใกล้เคียงกับสิ่งใดมากที่สุดมันก็จะส่งผลลัพธ์ออกมาดังนั้น ผลลัพธ์จะได้ออกมาเสมอขึ้นกับว่ามันให้น้ำหนักกับรูปนั้นไปที่ผลลัพธ์คลาสใดบ้าง ในส่วนนี้สามารถดูเพิ่มเติมได้ในภาคผนวก ก



รูปที่ 37 ผลลัพธ์จากการเรียกใช้งาน API ทำนายภาพเอกซเรย์

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

5.1.1 สรุปผลการทำงานของซอฟต์แวร์ลงทะเลเบียน

การทำงานของซอฟต์แวร์ เป็นส่วนที่มีขึ้นเพื่อสร้างเอกสารที่ใช้ยืนยันตัวตนของบุคคลที่จะเข้ารับการถ่ายภาพ การทำงานทั้งหมดจะเป็นดังภาพซึ่งทำงานเป็นปกติทั้งหมดในกรณีที่มีความผิดพลาดขึ้นมา จะมีการแจ้งเตือนกลับมายังหน้าอินเตอร์เฟซให้ผู้ได้รับรู้

5.1.2 สรุปผลของการทำนายความผิดปกติจากภาพเอกซเรย์ 2 มิติด้วยปัญญาประดิษฐ์

การทำนายผลความผิดปกติทั้งหมด 12 กลุ่มอาการ และ 2 กลุ่ม คือ ไม่พบความผิดปกติ และ ผู้ป่วยมีการใช้อุปกรณ์สำหรับสนับสนุนการรักษาบริเวณทรวงอกโดยผลการทำนายโดยรวมจากทุกกลุ่มมีค่าความแม่นยำอยู่ที่ประมาณ 85%

5.1.3 สรุปผลของการตรวจหาเนื้องอกด้วยปัญญาประดิษฐ์จากภาพ CT-scan 3 มิติ

การทำนายหาตำแหน่งของเนื้องอกภายในบริเวณปอดจะมีผลการทำนายก่อนการใช้เทคนิค transfer learning ความแม่นยำจากการวัดผลแบบ Dice score จะไม่เกินทั้งหมด 40% แต่ผลลัพธ์หลังการทำ transfer learning ไปแล้ว dice score จะไม่คงที่แต่โดยรวมแล้วจะอยู่ที่มากกว่า 45% เมื่อพิจารณาทั้ง 2 คลาสโดยในการทำนายที่ได้ผลลัพธ์ที่ดีที่สุดจะอยู่ที่ประมาณ 82%

5.1.4 สรุปผลการเรียกใช้ API ปัญญาประดิษฐ์จากเว็บแอปพลิเคชัน

การทำงานของตัวเว็บแอปพลิเคชันสำหรับการทำนายความผิดปกติจากภาพเอกซเรย์บริเวณทรวงอก สามารถทำงานได้เป็นปกติโดยการส่งภาพผ่าน HTTP และส่งผลลัพธ์กลับมาแสดงในรูปของ JSON โดยการทำงานโดยรวมจะมีข้อเสียตอนที่ต้องใช้การโหลดตัวโมเดลซึ่งเป็นไฟล์ขนาดใหญ่ซึ่งอาจใช้เวลาานานกว่าปกติ

5.2 ข้อเสนอแนะ

เนื่องจากในงานวิจัยครั้งนี้ยังมีประสิทธิภาพในเรื่องของความแม่นยำไม่มากถึงในระดับที่สมควร ดังนั้นการพัฒนาต่อสำหรับส่วนของงานนี้ควรจะเน้นไปในเรื่องของความแม่นยำ และอาจจะใช้การใช้ชุดข้อมูลเพิ่มเติมเพื่อให้โมเดลได้เรียนรู้มากยิ่งขึ้น

เอกสารอ้างอิง

- [1] Gao Huang, Z. L., Kilian Q. Weinberger, and Laurens van der Maaten. *Densely connected convolutional networks*. ARXIV eprint, 2016.
- [2] Irvin, Jeremy, et al. *Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison*. 2019: arXiv preprint arXiv:1901.07031, n.d.
- [3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. *U-net: Convolutional networks for biomedical image segmentation*. International Conference on Medical image computing and computer-assisted intervention, 2015. Springer, Cham.
- [4] Simpson, Amber L., et al. *A large annotated medical image dataset for the development and evaluation of segmentation algorithms*. arXiv preprint arXiv:1902.09063, 2019.
- [5] Rajpurkar, Pranav, et al. *Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning*. arXiv preprint arXiv:1711.05225, 2017.
- [6] Athalye, Anish, Nicholas Carlini, and David Wagner. *Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples*. arXiv preprint arXiv:1802.00420, 2018.
- [7] Zech, John R., et al. *Confounding variables can degrade generalization performance of radiological deep learning models*. arXiv preprint arXiv:1807.00431, 2018.
- [8] [deeplearning]. *Retrieved from <https://medium.com/@athivvat/deep-learning>*. 2017.
- [9] Rajpurkar, Pranav, et al. *Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists*. PLoS medicine 15.11 : e1002686., 2018.

- [10] Baltruschat, Ivo M., et al. *Comparison of deep learning approaches for multi-label chest X-ray classification*. Scientific reports 9.1 (2019): 6381., 2019.
- [11] Kopelowitz, Evi, and Guy Englehard. *Lung Nodules Detection and Segmentation using 3D Mask-RCNN*. 2019.
- [12] Wang, Xintao, et al. *Esrgan: Enhanced super-resolution generative adversarial networks*. Proceedings of the European Conference on Computer Vision (ECCV), 2018.
- [13] Brockman, Greg, et al. *Openai gym*. arXiv preprint arXiv:1606.01540, 2016.
- [14] Chen, Tianqi, and Carlos Guestrin. *Xgboost: A scalable tree boosting system*. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
- [15] "boosting learning". [Online].
แหล่งที่มา *edureka.com*. 2017.
- [16] "instance and semantic segmentation". [Online]. แหล่งที่มา *analyticvidhya.com*. 2018.
- [17] Arreola, Luis, Gesem Gudiño, and Gerardo Flores. *Object recognition and tracking using Haar-like Features Cascade Classifiers: Application to a quad-rotor UAV*. arXiv preprint arXiv:1903.03947, 2019.
- [18] "MD5 circuit". [Online].
แหล่งที่มา <https://en.wikipedia.org/wiki/MD5>. 2016.
- [19] "PBKDF2 pic". [Online].
แหล่งที่มา <https://en.wikipedia.org/wiki/PBKDF2>. 2016.
- [20] "Wrong prediction". [Online].
แหล่งที่มา <http://myselph.de/neuralNet.html>. 2018.

[21] "JNI". [online].

แหล่งที่มา https://www.researchgate.net/figure/Java-Native-Interface-JNI-to-access-the-C-code_fig3_331872802. n.d.

[22] "homography". [online].

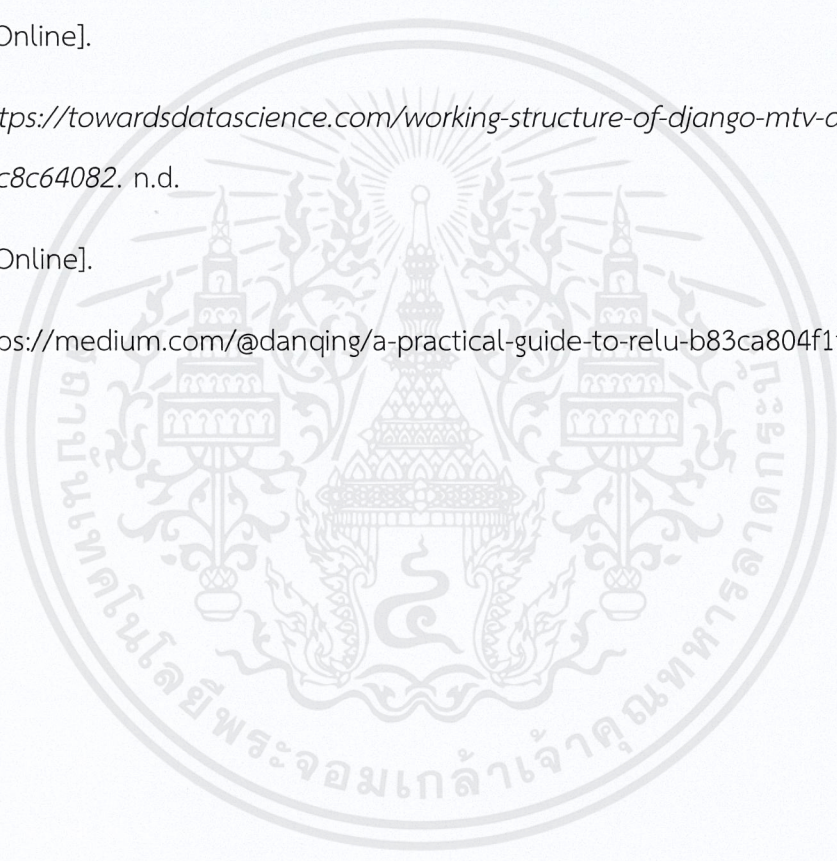
แหล่งที่มา <https://inst.eecs.berkeley.edu/~cs19426/fa15/upload/files/proj7B/cs194-av/>. n.d.

[23] "MTV". [Online].

แหล่งที่มา <https://towardsdatascience.com/working-structure-of-django-mtv-architecture-a741c8c64082>. n.d.

[24] "Relu". [Online].

แหล่งที่มา <https://medium.com/@dandqing/a-practical-guide-to-relu-b83ca804f1f7>



ภาคผนวก ก อธิบายส่วนของการทำงานที่ผิดพลาดของ Deep learning

จากรูปที่ 38 จะเห็นได้ว่าภาพอินพุตเป็นภาพของแผ่นที่โลกแต่โมเดลก็ยังให้ผลลัพธ์ไขเชิงความผิดพลาด บริเวณทรงอกได้สาเหตุของผลลัพธ์ที่เป็นแบบนี้มาจากสาเหตุที่โมเดลจำเป็นต้องมีผลลัพธ์ไปยังเอาต์พุต โหนดเสมอ

myselph.de Rigid Body Game Physics ▾ Hodgkin-Huxley Neural Net for MNIST

Neural Net for Handwritten Digit Recognition in JavaScript

Draw a digit in the box below and click the "recognize" button.

Display Preprocessing
 Scale Stroke Width

clear recognize

Home Predict Logout

Select X-ray picture

Predict!

Predicted Output is: Fracture

รูปที่ 38 ความผิดพลาดในการทำงานของโมเดล [20]