

การพัฒนาโปรแกรมสำหรับการทำนายอวัยวะจากภาพทางการแพทย์ด้วยปัญญาประดิษฐ์
ประเภทการเรียนรู้เชิงลึก

The development program by using Deep learning for organ prediction



โครงการเล่มนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมสำหรับการทำนายอวัยวะจากภาพทางการแพทย์ด้วยปัญญาประดิษฐ์
ประเภทการเรียนรู้เชิงลึก

The development program by using Deep learning for organ prediction



โครงการเล่มนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ	การพัฒนาโปรแกรมสำหรับการทำนายอวัยวะจากภาพทางการแพทย์ด้วยปัญญาประดิษฐ์ประเภทการเรียนรู้เชิงลึก
นักศึกษา	นายจิรายุส ลีกิจวิวัฒน์ 59010221 นายมงคล ธิติวิรัตน์ 59011086
สาขาวิชา	วิศวกรรมชีวการแพทย์
อาจารย์ที่ปรึกษา	รศ.ดร. สุพันธ์ ตั้งจิตกุลมั่น
ปีการศึกษา	2562

บทคัดย่อ

การพัฒนาโปรแกรมสำหรับการจัดการภาพทางการแพทย์ด้วยปัญญาประดิษฐ์ประเภทการเรียนรู้เชิงลึก มีจุดประสงค์ในการจัดทำเพื่อพัฒนาโปรแกรมสำหรับจัดการภาพทางการแพทย์ (Auto segmentation) ที่เหมาะกับคนไทยและมีราคาไม่แพง เนื่องจากในปัจจุบัน การแยกภาพทางการแพทย์ (Segmentation) ต้องทำผ่านผู้เชี่ยวชาญซึ่งใช้เวลาในการทำ segmentation ต่อคนไข้ 1 คนใช้เวลาประมาณ 3-4 วัน ซึ่ง ณ ปัจจุบันมี software ที่จำหน่ายสำหรับทำกระบวนการแยกอวัยวะอัตโนมัติ (Auto segmentation)จำหน่ายและมีราคาแพง แต่ไม่สามารถนำมาใช้กับข้อมูลของผู้ป่วยที่เป็นคนไทยได้

โดยมีหลักการการทำงานของโปรแกรมคือ นำภาพทางการแพทย์ที่ได้จากเครื่อง CT scan , MRI ซึ่งจะเป็นไฟล์ DICOM จากนั้นนำภาพมาผ่านกระบวนการ Image processing เพื่อทำการแปลงภาพให้สามารถนำไปทำนายอวัยวะที่มีอยู่เพื่อใช้ในการตรวจสอบปัญญาประดิษฐ์ว่าสิ่งที่ทำการสอนไปนั้นมีความถูกต้อง แม่นยำและใกล้เคียงกับผลลัพธ์ที่ต้องการ เพื่อนำไปเข้าสู่กระบวนการ segmentation ไปใช้ปัญญาประดิษฐ์ (AI) โดยใช้การเรียนรู้เชิงลึก (Deep learning) เพื่อให้ได้โปรแกรม Auto segmentation ต่อไปในภายหน้า

Project	The development program by using Deep learning for organ prediction	
Student	Mr. Jirayuts Leekitviwat	Student ID. 59010221
	Mr. Mongkol Thitivirut	Student ID. 59011086
Program	Biomedical Engineering	
Project Advisor	Associate prof. Dr. Supan Tungjitkusonmun	
Academic Year	2019	

Abstract

The Medical image processing development by Deep learning, which is part of Artificial Intelligence, proposes for medical image auto segmentation that suitable for Thai people health management. Due to present medical segmentation must be done by medical expert which takes long time for the analyzing by time processing for one patient is around 3-4 days. Nowadays, there are new software which can be used in medical image auto segmentation with expensive price and unsuitable for Thai people's anatomy.

This developed program algorithm will get medical image, which is obtained from Computerized Tomography Scan (CT scan) or Magnetic Resonance Imaging (MRI), then image processing for image transform will be processed and get them ready for the segmentation. Afterward, the segmented data will be further processed in training by deep learning which resulted for auto segmentation program.

กิตติกรรมประกาศ

งานวิจัยการพัฒนาโปรแกรมสำหรับการจัดการภาพทางการแพทย์ด้วยปัญญาประดิษฐ์ ประเภทการเรียนรู้เชิงลึกฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์ ด้วยความกรุณาอย่างยิ่งจาก ผศ.ดร. สุพันธ์ ตั้งจิตกุศลมั่น, รศ.ดร. ภัทรพงษ์ ผาสุขกิจ ที่ได้สละเวลาอันมีค่าแก่คณะผู้วิจัยให้คำปรึกษา แนะนำ ตลอดจน ตรวจทานแก้ไขข้อบกพร่องด้วยความเอาใจใส่เป็นอย่างยิ่ง จนงานวิจัยฉบับนี้สำเร็จสมบูรณ์ ลุล่วงได้ด้วยดีคณะผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้จากใจจริง ขอขอบคุณ รศ.ดร. ชูชาติ ปิณฑวิรุจน์ ที่ได้กรุณาให้คำแนะนำช่วยเหลือ ตรวจแก้ไข และปรับปรุงงานวิจัยฉบับนี้ จนเสร็จสมบูรณ์ รวมทั้งขอขอบคุณข้อมูลอันมีค่าจาก GitHub ,YouTube และ google ที่ทำให้ งานวิจัยนี้สามารถดำเนินไปข้างหน้าได้ และสุดท้ายนี้ขออุทิศความดีที่มีในการศึกษาวิจัยนี้แต่บิดามารดา ครอบครัวของคณะผู้วิจัย ซึ่งสนับสนุนในทุกๆด้าน และกำลังใจจากมิตรแท้ทุกท่าน

จิรายุส ลีกิจวิวัฒน์

มงคล ธิติวิรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	XI
สารบัญรูปภาพ	XII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของงานวิจัย	1
1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนและเวลาในการศึกษาการจัดทำโครงการ	2
1.5 แผนการดำเนินงาน	3
1.6 โครงสร้างวิทยานิพนธ์	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 Hardware	5
2.2 Software	6
2.2.1 python	6
2.2.2 การเรียกใช้ module ของ python	7
2.2.2.1 การเรียกใช้มอดูล	7
2.2.2.2 การลบมอดูล	8
2.2.2.3 การละชื่อมอดูล	9
2.2.2.4 การย่อชื่อมอดูลและฟังก์ชัน	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2.2.5 คลาสที่มากับมอดูล	11
2.3 Neuron Network	12
2.3.1 โครงข่ายประสาทเทียม (Artificial Neuron Network : ANN)	12
2.3.1.1 องค์ประกอบของโครงข่ายประสาทเทียม	12
2.3.1.2 เพอร์เซปตรอน (Perceptron)	14
2.3.1.3 การเรียนรู้แบบมีผู้ฝึกสอน (learning with supervisor)	15
2.3.2 โครงข่ายประสาทคอนโวลูชันเต็มรูป (Fully Convolutional Neural Network- FCNN)	18
2.3.2.1 Upsampling Layer	19
2.3.2.2 Skip Connections	19
2.4 deep learning	20
2.4.1 deep learning	20
2.4.2 กระบวนการทำงานของ deep learning	21
2.4.3 Classification of Neural Networks	24
2.4.4 ชนิดของโครงข่าย deep learning (Types of Deep Learning Networks)	24
2.4.4.1 โครงข่ายประสาทแบบป้อนไปหน้า (<u>Feed-forward neural networks</u>)	25
2.4.4.2 โครงข่ายแบบวนซ้ำ (Recurrent neural networks : RNN)	25
2.4.4.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks :CNN)	26
2.4.4.4 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.5 ความสำคัญของ Deep learning	28
2.4.6 ข้อจำกัดของ deep learning (Limitation of deep learning)	28
2.4.7 10 สถาปัตยกรรม Deep learning ชั้นสูง	29
2.4.7.1 AlexNet	29
2.4.7.2 VGG Net VGG Network	30
2.4.7.3 GoogleNet	31
2.4.7.4 ResNet	33
2.4.7.5 ResNeXt	35
2.4.7.6 RCNN (Region Based CNN)	35
2.4.7.7 YOLO (You Only Look Once)	36
2.4.7.8 SqueezeNet	37
2.4.7.9 SegNet	39
2.4.7.10 GAN (Generative Adversarial Network)	39
2.5 เครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า Magnetic Resonance Imaging (MRI)	40
2.5.1 ข้อดีของการตรวจ MRI	40
2.5.2 วัตถุประสงค์ของการตรวจ MRI	41
2.5.3 ข้อควรระวังในการตรวจ MRI	41
2.6 การตรวจ CT scan (Computerized Tomography)	42

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VII

สารบัญ (ต่อ)

	หน้า
2.6.2 ความแตกต่างระหว่าง x-ray และ CT scan	43
2.6.3 ประโยชน์ของ CT scan	43
2.7 การแปลงภาพ	44
2.7.1 Histogram Equalization	44
2.7.2 Type of image	47
2.7.2.1 Binary image	47
2.7.2.2 Grayscale Image	47
2.7.2.3 RGB Image	48
2.7.2.4 Indexed Image	48
2.7.3 Morphological	49
2.7.2.1 การขยาย (Dilation)	49
2.7.2.2 การกร่อนขนาด (Erosion)	49
2.7.4 REGION-OF-INTEREST (ROI)	50
2.7.5 Softmax Function	50
2.7.6 Dropout	51
2.8 ค่าสัมประสิทธิ์ความคล้ายโซเรนเซน-ไดซ์ (Sørensen-Dice coefficient similarity)	52
2.9 Entropy และ Uncertainty ของระบบใน Deep Learning AI	53
2.9.1 ความหมายของ Entropy	53
2.9.2 การนำ Entropy ไปใช้กับ Deep learning	53
2.9.3 Cross Entropy	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VIII

สารบัญ (ต่อ)

	หน้า
2.10 ชนิดของภาพ	54
2.10.1 DICOM file image	54
2.10.1.1 ข้อสังเกต DICOM file	54
2.10.2 PNG image	55
บทที่ 3 ระเบียบวิธีวิจัย	56
3.1 การพัฒนาโปรแกรมเพื่อใช้ในการแยกอวัยวะโดยใช้ปัญญาประดิษฐ์ ชนิดการเรียนรู้เชิงลึก	56
3.2 การออกแบบการพัฒนาปัญญาประดิษฐ์เพื่อใช้ในการแยกอวัยวะ	56
3.2.1 การแปลงภาพจากภาพไดค้อมเป็นภาพพิง	57
3.2.2 การแยกส่วนอวัยวะ	59
3.2.3 การจัดการแบ่งข้อมูลภาพเพื่อใช้ในการเทรน	60
3.2.4 การออกแบบวิธีการทดลองและการ นำเข้ามอดูล	61
3.2.5 การตั้งค่าข้อมูลภาพเพื่อให้เหมาะสมกับอุปกรณ์ที่ใช้ในการสอนปัญญา ประดิษฐ์	63
3.2.6 การรับค่าข้อมูลและนำไปใช้ในรูปของเมทริกซ์	64
3.2.7 ชุดคำสั่งทำการย่อขนาดภาพการสอนให้ตรงตามตัวแปรที่ตั้งค่าไว้ และรับข้อมูลภาพเข้า	65
3.2.8 สร้าง training Image and Mask generator	66
3.2.9 สร้าง the validation Image and Mask generator	66
3.2.10 สร้าง training and validation generator that generate masks and images	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2.11 สร้าง class สำหรับการหาค่าความแม่นยำของการสอนและทดสอบ	67
3.2.12 การสร้างส่วนการสร้างรูปแบบสถาปัตยกรรม U-Net	67
3.2.13 การกำหนดค่าของโมเดล	70
3.2.14 ตั้งค่าการสอนปัญญาประดิษฐ์	70
3.2.15 ออกแบบชุดคำสั่งการทดสอบและคาดเดาผลลัพธ์ของปัญญาประดิษฐ์	71
3.2.16 ออกแบบชุดคำสั่งในการบีบอัดข้อมูลโดยไม่สูญเสียข้อมูลภาพ	72
3.2.17 ชุดคำสั่งสรุปความแม่นยำและการสูญเสียข้อมูลระหว่างสอน	73
บทที่ 4 ผลการทดลอง	74
4.1 ผลการทดลองเมื่อทำการสอนปัญญาประดิษฐ์โดยคาดเดาการแยกไตขวา	74
4.1.1 ผลการทดลองเมื่อทำการฝึกในครั้งแรก	74
4.1.2 ผลการทดลองเมื่อทำการฝึกในครั้งที่สอง	74
4.1.3 ผลการทดลองเมื่อทำการฝึกในครั้งที่สาม	75
4.2 ผลการทดลองเมื่อทำการสอนปัญญาประดิษฐ์โดยคาดเดาการแยกกระเพาะปัสสาวะ	77
4.2.1 ผลการทดลองเมื่อทำการฝึกในครั้งแรก	77
4.2.2 ผลการทดลองเมื่อทำการฝึกในครั้งที่สอง	77
4.2.3 ผลการทดลองเมื่อทำการฝึกในครั้งที่สาม	78
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	80
5.1 สรุปผลการทดลอง	80
5.2 ปัญหาที่พบ	80
5.3 แนวทางการแก้ไข	80

บรรณานุกรม

82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก	85
ภาคผนวก ก	86
ภาคผนวก ข	93



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 1/2562	2
ตารางที่ 1.2 ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 2/2562	3
ตารางที่ 2.1 ฟังก์ชันการแปลงแบบต่างๆ	14
ตารางที่ 2.2 แสดงพื้นที่ในการจัดเก็บ และ Resolution ของ Digital image	55



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
ภาพที่ 2.1 ภาพประกอบการทำงานระหว่าง CPU กับ GPU	5
ภาพที่ 2.2 สัญลักษณ์ของโปรแกรม python	6
ภาพที่ 2.3 โครงสร้างระบบประสาท	12
ภาพที่ 2.4 โครงสร้างของเพอร์เซปตรอน	15
ภาพที่ 2.5 ผลลัพธ์ที่ได้จากการสอนแบบ Classification ที่ไม่ซับซ้อน	16
ภาพที่ 2.6 ผลลัพธ์ที่ได้จากการสอนแบบ Classification ที่ซับซ้อน	16
ภาพที่ 2.7 ผลลัพธ์ที่ได้จากการสอนแบบ Regression	16
ภาพที่ 2.8 การประยุกต์แก้ปัญหของ Supervised Learning	17
ภาพที่ 2.9 Traditional Programming	17
ภาพที่ 2.10 แสดงกระบวนการเทรน เพื่อให้ได้ model ที่ต้องการ	18
ภาพที่ 2.11 การนำโมเดลที่ต้องการมาประยุกต์ใช้กับโปรแกรม	18
ภาพที่ 2.12 ภาพจำลองโครงข่ายประสาทคอนโวลูชันเต็มรูป	19
ภาพที่ 2.13 Deconvolution ในชั้น Upsampling Layer	19
ภาพที่ 2.14 Skip Connections เพื่อ up sampling ในชั้นที่ต่ำกว่าของแบบจำลอง	20
ภาพที่ 2.15 hidden layer	21
ภาพที่ 2.16 กระบวนการทำงานของ deep learning	22
ภาพที่ 2.17 การเรียนรู้ของ deep learning เฟสแรก	23
ภาพที่ 2.18 การเรียนรู้ของ deep learning เฟสสอง	23
ภาพที่ 2.19 ชนิดของโครงข่าย deep learning	24

ภาพที่ 2.20 โครงข่ายแบบวนซ้ำ 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
ภาพที่ 2.21 โครงข่ายประสาทแบบคอนโวลูชัน	27
ภาพที่ 2.22 สถาปัตยกรรมแบบ Alexnet	29
ภาพที่ 2.23 สถาปัตยกรรมแบบ VGG Net VGG Network	30
ภาพที่ 2.24 สถาปัตยกรรมแบบ VGG Net VGG Network	31
ภาพที่ 2.25 สถาปัตยกรรมแบบ GoogleNet	31
ภาพที่ 2.26 การทำงานของ GoogleNet	32
ภาพที่ 2.27 สถาปัตยกรรมแบบ ResNet	33
ภาพที่ 2.28 การทำงานของ ResNet	34
ภาพที่ 2.29 สถาปัตยกรรมแบบ ResNeXt	35
ภาพที่ 2.30 สถาปัตยกรรมแบบ RCNN	35
ภาพที่ 2.31 การทำงานของ RCNN	36
ภาพที่ 2.32 การทำงานของ YOLO	36
ภาพที่ 2.33 สถาปัตยกรรมแบบ YOLO	37
ภาพที่ 2.34 สถาปัตยกรรมแบบ SqueezeNet	37
ภาพที่ 2.35 การทำงานของ SqueezeNet	38
ภาพที่ 2.36 สถาปัตยกรรมแบบ SegNet	39
ภาพที่ 2.37 สถาปัตยกรรมแบบ GAN	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
ภาพที่ 2.38 เครื่อง MRI	40
ภาพที่ 2.39 เครื่อง CT scan	42
ภาพที่ 2.40 ตัวอย่างภาพก่อนทำ Histogram Equalization	44
ภาพที่ 2.41 ตัวอย่างกราฟก่อนทำ Histogram Equalization	44
ภาพที่ 2.42 ตัวอย่างภาพหลังทำ Histogram Equalization	45
ภาพที่ 2.43 ตัวอย่างกราฟหลังทำ Histogram Equalization	45
ภาพที่ 2.44 กราฟ <i>Cumulative Histogram</i> ก่อนทำ Histogram Equalization	46
ภาพที่ 2.45 กราฟ <i>Cumulative Histogram</i> หลังทำ Histogram Equalization	46
ภาพที่ 2.46 ภาพแบบ Binary หรือ ภาพขาว-ดำ	47
ภาพที่ 2.47 ระดับเฉดสีเทา	47
ภาพที่ 2.48 ภาพแบบ RGB	48
ภาพที่ 2.49 ภาพแบบ Indexed	48
ภาพที่ 2.50 การขยาย	49
ภาพที่ 2.51 การกร่อนขนาด	50
ภาพที่ 2.52 (a) neural network ที่ไม่ใช้ dropout (b) neural network ที่ใช้ dropout	51
ภาพที่ 2.53 กราฟเปรียบเทียบ error ระหว่าง neural network ที่ไม่มีการใช้ dropout และ มีการใช้ dropout	55
ภาพที่ 3.1 รูปแบบโครงสร้างของ U-Net	57
ภาพที่ 3.2 ชุดคำสั่งที่ใช้ในการแปลงภาพไดคอมเป็นพิง	58

สารบัญรูปภาพ (ต่อ)

	หน้า
ภาพที่ 3.3 ชุดคำสั่งสร้างภาพแยกอวัยวะในโปรแกรมแมทแลป	59
ภาพที่ 3.4 การแบ่งข้อมูลออกเป็นสัดส่วน	60
ภาพที่ 3.5 การตั้งโมดูลที่เป็นต้องใช้ออกมา	62
ภาพที่ 3.6 กำหนดค่าเพื่อเข้าสู่กระบวนการก่อนประมวลผลภาพ	63
ภาพที่ 3.7 การรับค่าของภาพในรูปแบบของเมทริกซ์และเก็บไว้ในตัวแปร	64
ภาพที่ 3.8 ชุดคำสั่งทำการย่อขนาดภาพการสอนให้ตรงตามตัวแปรที่ตั้งค่าไว้และรับข้อมูลภาพเข้ามา	65
ภาพที่ 3.9 ชุดคำสั่งสร้าง training Image and Mask generator	66
ภาพที่ 3.10 ชุดคำสั่งสร้าง the validation Image and Mask generator	66
ภาพที่ 3.11 ชุดคำสั่งสร้าง training and validation generator that generate masks and images	67
ภาพที่ 3.12 ชุดคำสั่งสร้าง classสำหรับการหาค่าความแม่นยำของการสอนและทดสอบ	67
ภาพที่ 3.13 ชุดคำสั่งการสร้างรูปแบบสถาปัตยกรรม U-Net	69
ภาพที่ 3.14 ชุดคำสั่งกำหนดค่าของโมเดล	70
ภาพที่ 3.15 ตั้งค่าการสอนปัญญาประดิษฐ์	70
ภาพที่ 3.16 ชุดคำสั่งการทดสอบและคาดเดาผลลัพธ์ของปัญญาประดิษฐ์	71
ภาพที่ 3.17 ชุดคำสั่งในการบีบอัดข้อมูลโดยไม่สูญเสียข้อมูลภาพ	72
ภาพที่ 3.18 ชุดคำสั่งสรุปความแม่นยำและการสูญเสียข้อมูลระหว่างสอน	73

ภาพที่ 4.1 แสดงค่าสถานะของการสอนในรอบแรก	74
--	----

สารบัญรูปภาพ (ต่อ)

	หน้า
ภาพที่ 4.2 แสดงค่าสถานะของการสอนในรอบที่สอง	74
ภาพที่ 4.3 แสดงค่าสถานะของการสอนในรอบที่สาม	75
ภาพที่ 4.4 รูปที่ต้นแบบที่ใช้ในการคาดเดา	75
ภาพที่ 4.5 รูปที่ปัญญาประดิษฐ์คาดเดา	75
ภาพที่ 4.6 รูปที่ต้นแบบที่ใช้ในการคาดเดา	76
ภาพที่ 4.7 รูปที่ปัญญาประดิษฐ์คาดเดา	76
ภาพที่ 4.8 การแสดงความแม่นยำของโมเดลนี้	76
ภาพที่ 4.9 การแสดงการสูญเสียของการสอนปัญญาประดิษฐ์	77
ภาพที่ 4.10 แสดงค่าสถานะของการสอนในรอบแรก	77
ภาพที่ 4.11 แสดงค่าสถานะของการสอนในรอบสอง	77
ภาพที่ 4.12 แสดงค่าสถานะของการสอนในรอบสาม	78
ภาพที่ 4.13 ภาพที่ใช้ในการทำนาย	78
ภาพที่ 4.14 ปัญญาประดิษฐ์ทำนาย	78
ภาพที่ 4.15 ภาพที่ใช้ในการทำนาย	79
ภาพที่ 4.16 ปัญญาประดิษฐ์ทำนาย	79
ภาพที่ 4.17 กราฟสรุปความแม่นยำ	79
ภาพที่ 4.18 กราฟสรุปความสูญเสีย	79

บทที่ 1

บทนำ

ในบทนี้กล่าวถึงความเป็นมาและความสำคัญ วัตถุประสงค์ของการศึกษา ขอบเขตของโครงการ ขั้นตอนและเวลาในการศึกษาจัดทำรายงาน แผนการดำเนินงาน รวมถึงโครงสร้างวิทยานิพนธ์เป็นต้น

1.1 ที่มาและความสำคัญ

เนื่องจากทางแพทยรังสีรักษาของโรงพยาบาลศิริราชต้องการจัดการภาพทางการแพทย์ เพื่อใช้ในการรักษาด้วยเทคนิครังสีรักษาซึ่งข้อมูลที่ได้มาจะอยู่ในรูปภาพทางการแพทย์ เช่น ภาพจากเครื่องเอกซเรย์คอมพิวเตอร์ (Computerized Tomography Scan; CT Scan) หรือ เครื่องตรวจด้วยแม่เหล็กไฟฟ้า (Magnetic Resonance Imaging; MRI) ซึ่งในปัจจุบันการแบ่งส่วนภาพทางการแพทย์ (Segmentation) ต้องอาศัยผู้เชี่ยวชาญซึ่งใช้เวลาในการทำโดยใช้เวลาการแบ่งส่วนต่อผู้ป่วย 1 คน เป็นเวลาประมาณ 3-4 วัน ในปัจจุบันมีซอฟต์แวร์สำหรับดำเนินการแบ่งส่วนอวัยวะอัตโนมัติ (Auto organ segmentation) แต่มีราคาแพงอีกทั้งไม่สามารถนำมาใช้กับข้อมูลของผู้ป่วยที่เป็นคนไทยได้ เพราะฉะนั้นแพทย์จึงต้องทำการแบ่งส่วนภาพด้วยตนเอง ซึ่งในกระบวนการรักษามีความจำเป็นอย่างมากก่อนจะทำการรักษาด้วยรังสี

ด้วยเหตุนี้ทีมวิจัยจึงมีความสนใจจะพัฒนาโปรแกรมสำหรับจัดการแบ่งส่วนภาพทางการแพทย์ (Medical image segmentation) ด้วยปัญญาประดิษฐ์ (AI) โดยใช้การเรียนรู้เชิงลึก (Deep learning) มาประยุกต์ใช้ให้มีโปรแกรมทำการแบ่งส่วนภาพที่เหมาะสมกับคนไทยและมีราคาไม่แพงเพื่อการตรวจสอบความถูกต้องก่อนนำเข้าสู่กระบวนการแบ่งส่วนภาพ จำเป็นต้องทำการทำนายโมเดลของอวัยวะก่อนว่ามีความถูกต้องกับวัตถุประสงค์หรือไม่

1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. ออกแบบและสร้างโปรแกรมเพื่อใช้ในการแบ่งส่วนภาพทางการแพทย์เพื่อใช้ในการแยกภาพอวัยวะของผู้ป่วยออกจากกัน
2. ใช้ปัญญาประดิษฐ์เรียนรู้ภาพที่ได้จากกระบวนการแยกภาพเพื่อให้ตัวโปรแกรมสามารถทำการแบ่งส่วนภาพแบบอัตโนมัติเองได้
3. ได้โปรแกรมที่สามารถทำการแบ่งส่วนภาพของอวัยวะแบบอัตโนมัติมีเหมาะสมกับคนไทยและมีความแม่นยำสูง
4. สามารถนำไปใช้เพื่อการศึกษาและนำไปต่อยอดในด้านงานของกระบวนการรูปภาพดิจิทัล

ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการ

1. โปรแกรมที่พัฒนาขึ้นมาสามารถทำการแบ่งส่วนอวัยวะแต่ละประเภทออกจากกันได้
2. ปัญญาประดิษฐ์สามารถเรียนรู้และทำนายโมเดลของอวัยวะที่ผ่านการตัดแยกภาพออกมา
3. ได้โปรแกรมที่มีความแม่นยำสูง

1.4 ขั้นตอนและเวลาในการศึกษาการจัดทำโครงการ

ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 1/2562

1/2562					
ขั้นตอนการดำเนินงาน	ระยะเวลา				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. เก็บข้อมูลภาพผู้ป่วย					
2. ศึกษากระบวนการแบ่งส่วนภาพ					
3. นำภาพผู้ป่วยมาเข้ากระบวนการแบ่งส่วนภาพ					
4. ศึกษารูปแบบการเรียนรู้เชิงลึก					
5. ออกแบบอัลกอริทึมที่จะใช้จริง					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.2 ขั้นตอนและระยะเวลาในการดำเนินงานของภาคเรียนที่ 2/2562

2/2562					
ขั้นตอนการดำเนินงาน	ระยะเวลา				
	ม.ค.	ก.พ.	มี.ค.	เม.ษ.	พ.ค.
1. ใช้โปรแกรมที่ออกแบบไว้ในการทดลอง					
2. เก็บข้อมูลที่ได้จากการทดลอง					
3. เก็บผลและแก้ไขให้ได้อัลกอริทึมที่ดีที่สุด					
4. ทำการทดลองและเก็บผลใหม่ให้โปรแกรมมีความแม่นยำสูง					
5. สรุปผล					

1.5 แผนการดำเนินงาน

1. รับข้อมูลภาพทางการแพทย์
2. ศึกษาเทคนิคการทำแบ่งส่วนภาพแบบแมนนวล
3. พัฒนาอัลกอริทึมสำหรับตัวโปรแกรม
4. ทดสอบและเก็บผลการทดลอง
5. ลองนำไปใช้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 โครงสร้างวิทยานิพนธ์

ผลการศึกษาและวิจัยได้นำมาเสนอในวิทยานิพนธ์ฉบับนี้โดยทำการแบ่งเนื้อหาเป็นบทต่างๆ ดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญ วัตถุประสงค์ของการศึกษา ขอบเขตของโครงการ ขั้นตอนและเวลาในการศึกษาจัดทำโรงงาน แผนการดำเนินงาน รวมถึงโครงสร้างวิทยานิพนธ์เป็นต้น

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

บทที่ 3 หลักการทำงานและการออกแบบ

บทที่ 4 วิธีการและผลการทดลอง

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ สรุปผลการทดลองและข้อเสนอแนะ ผลสรุปภาพรวมของการทดลองและวิจัยที่ได้จากบทที่ 4



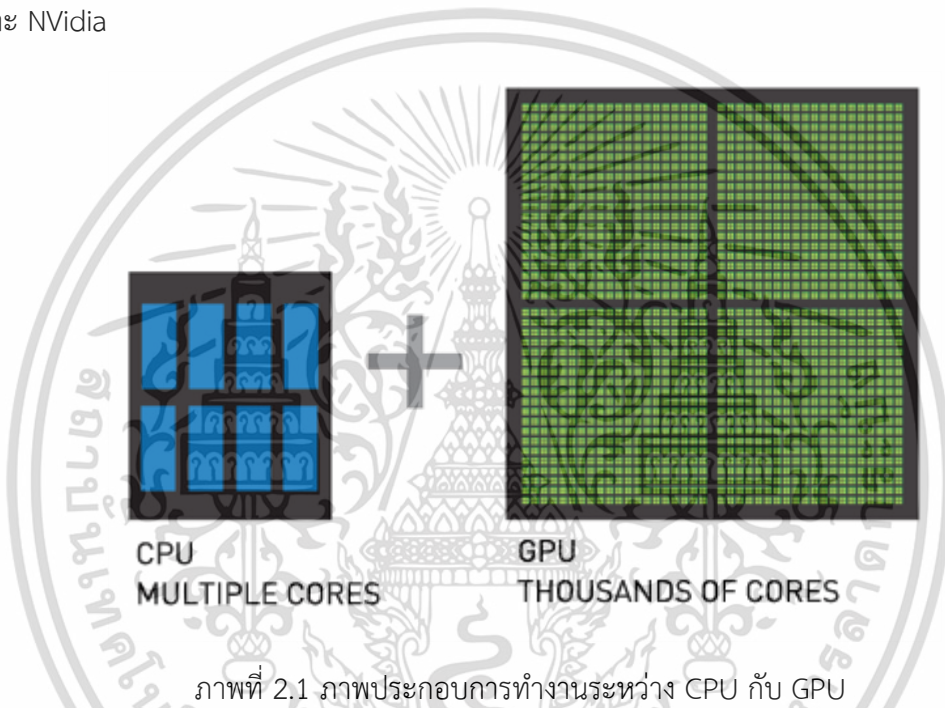
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 Hardware

GPU [1] หรือบางครั้งก็เรียกว่า VPU (Visual Processing Unit) คือหน่วยประมวลผลด้านกราฟิก 3 มิติ เราสามารถที่จะเห็น GPU ได้ในการดจอแยกที่เสียบอยู่ในเครื่องคอมพิวเตอร์หรือติดตั้งมาบนเมนบอร์ด ซึ่ง GPU ถือกำเนิดมาจากการพัฒนากราฟิกสามมิติ โดยบริษัท AMD/ATI และ NVidia



ทำหน้าที่เหมือนกับ CPU ในสมัยก่อนนั้นการประมวลผลทางด้านกราฟิกจะต้องใช้ทรัพยากรจาก CPU เป็นหลัก ทำให้การทำงานของ CPU นั้นหนักขึ้น แต่เมื่อมี GPU การประมวลผลทางด้านกราฟิกก็จะหมดก็จะไม่ไปประมวลผลที่ CPU แต่จะมาประมวลผลที่ GPU แทน เลยทำให้การทำงานของ CPU นั้นทำงานน้อยลง ทำให้ความร้อนภายในเครื่องลดลง

ประโยชน์ของ GPU ที่เห็นได้ชัด คือ การแสดงผลกราฟิกด้าน 3 มิติ นั้นมีความสวยงามมากขึ้น และการใช้ GPU จะเป็นการแบ่งเบาภาระการทำงานของของ CPU ทำให้ CPU มีอายุการใช้งานที่มากขึ้น

GPU ช่วยให้อุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ สามารถที่จะพัฒนาด้านกราฟิกได้มากขึ้น เช่น โทรศัพท์มือถือสมาร์ทโฟนในปัจจุบันมีการใช้ GPU ในการประมวลผลด้านกราฟิก 3 มิติ แม้กระทั่งเครื่องเล่นเกมก็มีการใช้งาน GPU ในการประมวลผลเพื่อให้ภาพออกมาสมจริงมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Software

2.2.1 python

Python [2] คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถทำงานอ่านภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่ง โดยมีสัญลักษณ์แสดงในภาพที่ 2.2



ภาพที่ 2.2 สัญลักษณ์ของโปรแกรม python

ภาษาตัว นี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้นและใช้งานได้ครอบคลุมกับทุกลักษณะงานไวยากรณ์ของภาษา Python ภาษา Python นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจว่าจะให้เป็นภาษาที่อ่านง่าย มันถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อนโดยมักจะใช้คำในภาษาอังกฤษในขณะที่ภาษาอื่นใช้เครื่องหมายวรรคตอน นอกจากนี้ Python มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษา C และ Pascal Python Interpreter Python interpreter นั้นเป็นตัวแปรภาษาของภาษา Python เพื่อให้สามารถรันโค้ด Python ได้ ซึ่งได้มากับไลบรารีมาตรฐานที่สามารถใช้งานได้ฟรี ซึ่งดาวน์โหลดได้ที่ <https://www.python.org/> เป็นโปรแกรมแบบ source และ binary สำหรับแพลตฟอร์มที่ได้รับความนิยม นอกจากนี้ Interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดของภาษา Python ลงไปและเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที

Python Interpreter นั้นยังสามารถนำเพิ่มความสามารถกับฟังก์ชันใหม่ที่ถูกพัฒนามาจากภาษา C และ C++ Python นั้นเหมาะสำหรับเป็นภาษาในการสร้าง Extension และแอปพลิเคชันที่ปรับแต่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การเรียกใช้มอดูล ของ Python [3]

2.2.2.1 การเรียกใช้มอดูล

คำสั่งที่ใช้ในการเรียกใช้มอดูลก็คือ import โดยพิมพ์ import แล้วตามด้วยชื่อมอดูลที่ต้องการเรียกใช้ เช่น เรียกใช้มอดูลที่ชื่อ math ซึ่งเป็นมอดูลที่เก็บฟังก์ชันที่เกี่ยวข้องกับการคำนวณทางคณิตศาสตร์

```
import math
```

เพียงเท่านี้ก็จะสามารถใช้ฟังก์ชันภายในมอดูล math ได้แล้ว ยกตัวอย่างเช่นฟังก์ชันชื่อ sin ซึ่งใช้คำนวณค่าไซน์ เวลาที่ใช้ก็ให้พิมพ์ชื่อมอดูลแล้วจึงตามด้วยชื่อฟังก์ชันนั้นโดยมีจุดคั่น

```
math.sin(1.57) # ได้ 0.9999996829318346
```

จะเห็นว่าชื่อฟังก์ชันในที่นี้อยู่ต่อจากจุด ดูแล้วคล้ายกับเมธอด แต่ความจริงแล้วไม่ใช่ ปกติจุด . เป็นตัวบอกถึงความสัมพันธ์แบบไล่เรียงหน่วยใหญ่ไปหาหน่วยเล็กจุดในที่นี้แค่บอกว่าฟังก์ชันนี้เป็นฟังก์ชันในสังกัดของมอดูลไหนเท่านั้น อย่างไรก็ตามจริงๆ แล้วมอดูลก็ถือเป็นออบเจกต์ชนิดหนึ่ง ดังนั้นอาจมองว่าฟังก์ชันเป็นเมธอดของมอดูลก็ได้เช่นกัน เพราะเมธอดก็คือฟังก์ชันที่สังกัดอยู่กับออบเจกต์คลาสต่าง ๆ นั้นเอง คล้ายกับการใช้ภาษาซีมาก่อน อาจคุ้นเคยกับคำว่า include คำสั่ง import ในภาษาไพธอนนี้เทียบเท่าได้กับ include ในภาษาซี คือเป็นการเรียกใช้ชุดคำสั่งที่จะมาใช้งานในโปรแกรม เพียงแต่ในภาษา ไพธอนคำสั่งพื้นฐานต่างๆ สามารถใช้ได้ทันทีโดยไม่ต้องเรียกใช้ก่อน ผิดกับภาษาซีซึ่งมักจะต้องเริ่มด้วยการเรียกใช้ stdio.h และ conio.h เป็นประจำ ดังนั้นทำให้การใช้งานในระดับพื้นฐานอาจไม่จำเป็นต้อง import อะไรเลย แต่พอต้องการใช้งานในระดับเฉพาะทางขึ้นมาหน่อยจึงจะเรียกใช้หากลองเปรียบเทียบกับภาษาอื่น ๆ ก็เห็นทั้งที่ใช้ include และ import เช่นใน php ใช้ include ใน javascript ใช้ import ความหมายเหมือนกันเพียงแต่ในแต่ละภาษาใช้ต่างกันเท่านั้น ในภาษาไพธอนมอดูลก็ถือว่าเป็นออบเจกต์ชนิดหนึ่ง คือออบเจกต์ชนิดมอดูล ลองใช้ฟังก์ชัน type เพื่อหาชนิดดูได้

```
print(type(math)) # ได้ <class 'module'>
```

ส่วนสิ่งที่มากับมอดูลนั้นอาจเป็นออบเจกต์ชนิดใดก็ได้มีได้หลายอย่าง ชนิดอาจแตกต่างกันออกไป โดยส่วนใหญ่แล้วจะเป็นฟังก์ชัน เช่น

```
print(type(math.sin)) # <class 'builtin_function_or_method'>
```

เนื่องจาก math เป็นมอดูลภายในตัว ดังนั้นฟังก์ชันที่อยู่ในนี้ก็เลยมีชนิดเป็นฟังก์ชันหรือเมธอดภายในตัว แต่หากเป็นฟังก์ชันในมอดูลชนิดอื่นก็อาจมีชนิดที่ต่างกันออกไป ซึ่งก็แล้วแต่ผู้สร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอดูลนั้นจะกำหนดนอกจากฟังก์ชันแล้วมอดูลก็อาจประกอบด้วยข้อมูลหรือออบเจกต์ที่เป็นค่าตัวเลข หรือสายอักขระก็ได้ เช่น

```
print(math.pi) # ได้ 3.141592653589793
```

pi นี้คือค่า π เป็นแค่ค่าคงที่ตัวหนึ่ง หากหาชนิดก็จะได้ว่าเป็น float

2.2.2.2 การลบมอดูล

หากไม่ต้องการใช้มอดูลไหนแล้วก็อาจลบทิ้งโดยใช้คำสั่ง del จากนั้นก็จะไม่สามารถใช้มอดูลนั้นได้อีกจนกว่าจะเรียกใช้ใหม่

```
del math
math.sin(1.57) # ได้ NameError: name 'math' is not defined
```

หรืออาจลบแค่ฟังก์ชันในมอดูลนั้นตัวเดียว เช่น

```
del math.sin
```

แบบนี้ก็จะใช้ได้แค่ math.sin ส่วนฟังก์ชันอื่นเช่น math.cos ก็ยังใช้ได้อยู่

2.2.2.3 การละชื่อมอดูล

โดยทั่วไปฟังก์ชันที่อยู่ภายในมอดูลที่เรียกเข้ามาด้วยคำสั่ง import ธรรมดานั้นเวลาจะใช้จะต้องมีชื่อมอดูลนำหน้าเสมอ

แต่ก็มีวิธีที่จะทำให้ไม่ต้องใส่ชื่อมอดูลนำหน้าก็ได้ ซึ่งทำได้โดยใช้คำสั่ง from เช่น

```
from math import *
```

เท่านี้ก็สามารถใช้ฟังก์ชันทั้งหมดในมอดูล math ได้โดยไม่ต้องมี math. นำหน้าแล้ว เช่นพิมพ์แค่ sin(1.57) ง่ายๆสั้นๆ ดอกจันทน์ * ในที่นี้มีความหมายว่าเรียกใช้ทุกสิ่งทุกอย่างที่อยู่ภายในมอดูลนั้นออกมา ให้หมด แต่ว่าถ้าหากอยากเรียกใช้แค่บางอันก็ทำได้โดยแค่เปลี่ยน * เป็นชื่อของสิ่งที่ต้องการใช้ เช่น

```
from math import sin
```

ถ้าทำแบบนี้ก็จะใช้ได้แค่ฟังก์ชัน sin ส่วนฟังก์ชันอื่นจะยังใช้ไม่ได้ อย่างไรก็ตามการใช้ from กับ * นี้

โดยทั่วไปแล้วจะไม่ใช้ถ้าไม่แน่ใจว่าจะไม่มีผลกระทบอะไรเนื่องจากแต่ละมอดูลอาจมีฟังก์ชันที่ชื่อ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ไปยังเว็บไซต์อื่นโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมือนกัน ซึ่งจะทำให้เกิดการซ้อนทับกันได้ โดยฟังก์ชันจากมอดูลที่เรียกใช้ทีหลังจะไปทับของมอดูลตัวที่เรียกก่อน สำหรับผู้เริ่มฝึกหัด เพื่อความชัดเจนบางทีก็ควรจะเหลือชื่อมอดูลต้นทางเอาไว้ว่าจะดีต่อการเรียนรู้มากกว่าดังนั้นแนะนำว่าอย่าใช้formแต่ให้importโดยตรง

2.2.2.4 การย่อชื่อมอดูลและฟังก์ชัน

หากชื่อยาวๆก็สามารถย่อลงได้ บ่อยครั้งที่ชุดคำสั่งยาวๆมักถูกย่อชื่อลง การย่อชื่อสามารถทำได้โดยเขียน as ต่อท้ายตามด้วยชื่อที่ต้องการย่อ เช่น

```
import random as rd
```

จากนั้นเวลาเรียกใช้มอดูลนี้ก็พิมพ์แค่ชื่อที่ถูกย่อแล้วตามด้วยชื่อฟังก์ชัน เช่น

```
rd.uniform(0,100) # จะได้ค่าเลขสุ่มซึ่งอยู่ในช่วง 0 ถึง 100
```

มอดูล random นี้เป็นมอดูลที่ประกอบด้วยฟังก์ชันที่เกี่ยวข้องกับการสุ่ม ฟังก์ชัน uniform นั้นจะคืนค่าเป็นจำนวนจริงสุ่มภายในช่วงระหว่างอาร์กิวเมนต์2ตัวที่ใส่ลงไป สิ่งที่ย่อชื่อได้นั้นไม่เพียงแต่ชื่อมอดูลเท่านั้น แต่ชื่อฟังก์ชันภายในมอดูลก็สามารถย่อได้เช่นกัน เช่น

```
from random import uniform as unif
```

เพียงเท่านี้ก็สามารถใช้ random.uniform ได้โดยพิมพ์แค่ unif (0,100) ชื่อมอดูลที่ถูกเรียกขึ้นมาครั้งนี้สถานะเหมือนตัวแปรตัวหนึ่ง จะใช้ชื่อนี้เป็นตัวแปรทำอย่างอื่นอีกไม่ได้ ถ้าใช้จะถูกเขียนทับ เช่น

```
import math
```

```
math = 100
```

```
math.sin(1.57) # ได้ AttributeError: 'int' object has no attribute 'sin'
```

นั่นเพราะพอแทนค่าลงไปแบบนี้กลายเป็นว่าตัวแปรที่ชื่อ math ไปรับค่า 100 แล้วมันก็จะไม่ใช่เป็นมอดูลที่ชื่อ math อีกต่อไป กลายเป็นค่าจำนวนเต็มธรรมดา เราจะใช้มันไม่ได้อีกจนกว่าจะ import ใหม่หรือแม้แต่ว่าเวลาที่ใช้ชื่อย่อก็เช่นกันต้องระวังไม่ให้ชื่อย่อนั้นไปซ้ำกับชื่อตัวแปรที่ต้องการจะใช้ มอดูลย่อยภายในมอดูลหลัก ปกติแล้วภายในมอดูลหนึ่งจะประกอบไปด้วยฟังก์ชันมากมายอยู่บางทีฟังก์ชันภายในมอดูลนั่นเองก็อาจจะแบ่งเป็นหมวดหมู่ต่างๆซึ่งมีลักษณะ เฉพาะต่างกันออกไป เมื่อเป็นแบบนี้บางมอดูลจึงได้มีการแบ่งมอดูลออก เป็นส่วนย่อยๆลงไปอีก เพื่อความเป็นระเบียบมอดูลที่เป็นหน่วยย่อยลงไปอีกนั้นเรียกว่าซับมอดูล (submodule) ยกตัวอย่างเช่นมอดูลชื่อ os มอดูลนี้มีไว้สำหรับจัดการกับสิ่งที่เกี่ยวข้องกับระบบปฏิบัติการ เช่นฟังก์ชัน os.uname นั้นจะคืนค่าข้อมูลที่ใช้จำแนกบ่งบอกถึงตัวเครื่องนี้ os.getcwd จะคืนค่าชื่อของโฟลเดอร์ที่ใช้ทำงานอยู่ (ที่อยู่ของไฟล์ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รัน) แต่ว่าภายในมอดูล os นี้ก็ยังประกอบไปด้วยมอดูลย่อยที่ชื่อว่า path อยู่ ภายในนี้มีฟังก์ชันที่เกี่ยวข้องกับพาธ (เส้นทางที่บ่งชี้ตำแหน่งของไฟล์หรือส่วนประกอบต่างๆ) เช่น os.path.abspath เป็นฟังก์ชันที่จะคืนค่าพาธสมบูรณ์ของไฟล์ตัวอย่างการใช้มอดูล os

```
import os
print(os.uname())
print(os.getcwd())
print(os.path.abspath('untitled0.py'))
```

จะเห็นว่าเมื่อเรียกใช้มอดูล os โดย import ใดๆเวลาที่ใช้ os.path.abspath จะต้องพิมพ์ยาวได้ตั้งแต่ชื่อมอดูลหลัก ตามด้วยชื่อมอดูลย่อย แล้วจึงจะเป็นชื่อฟังก์ชัน อย่างไรก็ตามสามารถย่อได้โดยใช้ from

```
from os import path
```

แบบนี้ก็จะสามารถละ os. ด้านหน้าไปได้ path.abspath หรือย่อกว่านั้นคือละชื่อมอดูลย่อยไปด้วย

```
from os.path import *
```

หรือ

```
from os.path import abspath
```

แบบนี้ก็จะใช้ฟังก์ชัน abspath ได้โดยพิมพ์แค่ชื่อ abspath มอดูลย่อยก็สามารถทำการย่อชื่อได้ เช่น

```
from os import path as op
```

หรือ

```
import os.path as op
```

สองแบบนี้มีค่าเท่ากัน คือเวลาเรียกใช้ abspath จะต้องพิมพ์เป็น op.abspath จะเห็นได้ว่าการเรียกใช้มอดูลมีทางเลือกอยู่หลากหลาย มีความยืดหยุ่น ถือเป็นความสะดวกอย่างหนึ่งของภาษาไพธอน เพียงแต่อาจจะต้องเสียเวลาทำความเข้าใจกันสักเล็กน้อย อนึ่ง มอดูลย่อยนั้นก็อาจจะมีมอดูลย่อยแยกเล็กลงไปอีก ซึ่งยิ่งย่อก็จะยิ่งคั่นด้วยจุด . ไปเรื่อยๆ อาจมีย่อลงไปอีกก็ขึ้นก็ได้ แล้วแต่คนออกแบบมอดูลจะออกแบบมา

2.2.2.5 คลาสที่มากับมอดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่ได้มาจากการเรียกใช้มอดูลนั้นไม่ได้มีเพียงฟังก์ชันใหม่หรือตัวแปรใหม่ เท่านั้น แต่ยังสามารถใช้ออบเจ็กต์ชนิดใหม่ได้ด้วยหากในมอดูลนั้นมีการจำกัดความคลาสใหม่เฉพาะขึ้นมาเรื่องของการสร้างคลาสนั้นจะพูดถึงในบทหลังจาก นี้ไปอีกเพราะเป็นเรื่องที่ซับซ้อน แต่ในบทนี้จะพูดถึงคร่าวๆเกี่ยวกับการใช้ออบเจ็กต์จากคลาสที่ถูกสร้างภายใน มอดูลที่เรียกใช้ ในที่นี้ขอยกตัวอย่างมอดูล fractions ซึ่งเป็นมอดูลที่ใช้จัดการเลขเศษส่วน การทำงานของมอดูลนี้ก็คือนิยามอบเจ็กต์ (หรือก็คือข้อมูล) ชนิดใหม่ขึ้นมาชนิดหนึ่ง ซึ่งก็คือ "เลขเศษส่วน" แนวคิดของมอดูลนี้ก็คือ ปกติแล้วจะเห็นว่าตัวเลขจำนวนจริงต่างๆที่ถูกคำนวณภายในโปรแกรมนั้นจะอยู่ใน รูปของเลขทศนิยม แม้จะเป็นตัวเลขที่เขียนเศษส่วนแล้วดูเข้าใจง่ายกว่าก็ตาม เช่น $1/3$ ก็จะกลายเป็น 0.3333333333333333 ซึ่งค่าที่ได้ก็จะเป็นเพียงค่าประมาณเท่านั้น หากนำมาคำนวณก็อาจเกิดความคลาดเคลื่อนที่เล็กน้อยเพื่อรองรับความ จำเป็นในการใช้เลขในรูปของเศษส่วนซึ่งอาจเกิดขึ้นในบางสถานการณ์ มอดูล fractions จึงได้นิยามข้อมูลชนิดใหม่ ซึ่งสามารถเก็บตัวเลขให้อยู่ในรูปเศษส่วนได้ ตัวอย่างการใช้

```
import fractions
a = fractions.Fraction(1,3)
print(a) # ได้ 1/3
print(type(a)) # ได้ชนิดของข้อมูลเป็น <class 'fractions.Fraction'>
```

ในที่นี้จะเห็นว่าตัวแปร a กลายเป็นอบเจ็กต์ชนิดใหม่ ซึ่งก็คือชนิด fractions.Fraction ซึ่งอาจจะดูยาวไปสักหน่อยเพราะต้องขึ้นต้นด้วยชื่อมอดูลก่อน ตรงส่วน a = fractions.Fraction(1,3) นั้นเป็นการประกาศตัวแปรและใส่ค่าให้ โดยเวลาสร้างข้อมูลชนิดนี้ทำได้หลายวิธี แต่ในที่นี้ใช้วิธีที่ง่ายที่สุดก็คือใส่อาร์กิวเมนต์เป็นเลขเศษตามด้วยเลข ส่วน มองดูแล้วอาจรู้สึกว่ายากต้องพิมพ์ตัวยาวขนาดนี้เพื่อแค่ให้ได้จำนวนเศษส่วนมาตัวหนึ่ง อย่างไรก็ตามสามารถย่อได้ เช่น

```
from fractions import Fraction as fr
a = fr(1,3)
```

แบบนี้จะดูง่ายขึ้น เพียงแต่เวลาถึงชื่อชนิดข้อมูลโดย print(type(a)) ก็ยังจะได้ผลเป็น <class 'fractions.Fraction'> เหมือนเดิม เพราะในที่นี้คำว่า fr แค่มาทแทน fractions.Fraction เพื่อความสะดวกเท่านั้น ข้อมูลเลขเศษส่วนเวลาที่สั่ง print จะแสดงผลออกมาเป็นเลขเศษส่วน คือเป็นเลขเศษคั่นด้วย / แล้วตามด้วยเลขส่วน นี่เป็นตัวอย่างคร่าวๆของการสร้างและใช้ออบเจ็กต์ชนิดใหม่ ที่จริงแล้วยังมีอบเจ็กต์อีกจำนวนมากมายหลายชนิดซึ่งจะต้องได้เจอต่อไปอีก

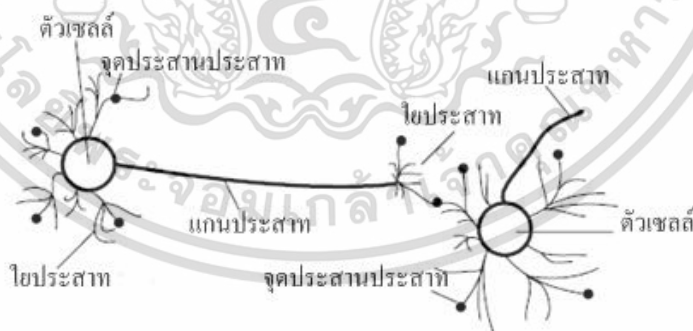
2.3 Neuron Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 โครงข่ายประสาทเทียม (Artificial Neuron Network : ANN) เป็นศาสตร์แขนงหนึ่งของทางด้านปัญญาประดิษฐ์ (Artificial Intelligence : AI) มีรูปแบบโครงสร้างและการทำงานของกรรมผลผลเหมือนกับสมองของสิ่งมีชีวิตซึ่งมีปรับเปลี่ยนตัวเองต่อการตอบสนองของอินพุตตามกฎของการเรียนรู้ (learning rule) หลังจากที่โครงข่ายได้เรียนรู้สิ่งที่ต้องการแล้ว โครงข่ายนั้นจะสามารถทำงานที่กำหนดไว้ได้โครงข่ายประสาทเทียมได้ถูกพัฒนาคิดค้นจากการทำงานของสมองมนุษย์โดยสมองมนุษย์ประกอบไปด้วยหน่วยประมวลผลเรียกว่า นิวรอน (เซลล์ประสาท หรือ neuron) จำนวนนิวรอนในสมองมนุษย์มีอยู่ประมาณและมีการเชื่อมต่อกันอย่างมากมายสมองมนุษย์จึงสามารถกล่าวได้ว่าเป็นคอมพิวเตอร์ที่มีการปรับตัวเอง (adaptive) ไม่เป็นเชิงเส้น (nonlinear) และทำงานแบบขนาน (parallel) ในการดูแลจัดการการทำงานร่วมกันของนิวรอนในสมองการคำนวณเชิงนิวรอนเป็นการคำนวณที่เลียนแบบมาจากการทำงานของสมองมนุษย์นั่นเอง

2.3.1.1 องค์ประกอบของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม[4]ได้ถูกพัฒนาขึ้นโดยอาศัย หลักการทำงานของสมองมนุษย์ ซึ่งสมองประกอบด้วยหน่วยประมวลผลพื้นฐานที่เรียกว่า นิวรอน (เซลล์ประสาท หรือ neuron) ภายในสมองประกอบด้วยนิวรอนจำนวนมหาศาล (ประมาณ 10^{11}) และมีจุดต่อจำนวนโครงข่ายประสาทประกอบขึ้นด้วยส่วนสำคัญ 3 ส่วน คือ โยประสาท (dendrite) ตัวเซลล์ (soma) และแกนประสาท (axon) ในแต่ละโครงข่ายประสาทจะเชื่อมต่อกันโดยจุดประสานประสาท (synapse) ซึ่งสามารถเปลี่ยนค่าความต้านทานได้ตามสัญญาณที่ส่งระหว่างกันของเซลล์ประสาท การส่งสัญญาณระหว่างเซลล์ประสาททำได้โดยการถ่ายทอดสารประกอบโซเดียมและโพแทสเซียม ภาพโครงร่างของระบบประสาทแสดงในภาพที่ 2.3



ภาพที่ 2.3 โครงสร้างระบบประสาท

การประมวลผลต่าง ๆ เกิดขึ้นในหน่วยประมวลผลย่อย เรียกว่า โหนด (node) ซึ่งโหนดเป็นการจำลองลักษณะการทำงานมาจากเซลล์การส่งสัญญาณ ระหว่างโหนดที่เชื่อมต่อกัน จำลองมาจากการเชื่อมต่อของโยประสาท และแกนประสาทในระบบประสาทของสมองมนุษย์ ภายในโหนด จะมีฟังก์ชันกำหนดสัญญาณส่งออกที่เรียกว่า ฟังก์ชันการแปลง (transfer function) ซึ่งทำหน้าที่

เปรียบเสมือนกระบวนการทำงานในเซลล์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงในรูปแบบที่ 2 โครงข่ายประสาทเทียมประกอบด้วย 5 องค์ประกอบ ดังนี้

1. ข้อมูลอินพุต (input) เป็นข้อมูลที่เป็นตัวเลข หากเป็นข้อมูลเชิงคุณภาพ ต้องแปลงให้อยู่ในรูปเชิงปริมาณที่โครงข่ายประสาทเทียมยอมรับได้
2. ข้อมูลเอาต์พุต (output) คือ ผลลัพธ์ที่เกิดขึ้นจริง จากกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม
3. ค่าน้ำหนัก (weights) คือ เป็นตัวบ่งบอกถึงความสำคัญของตัวแปรขาเข้าถูกนำมาใช้เพื่อการถ่วงน้ำหนักค่าของผลรวมเชิงเส้นข้อมูลขาเข้าที่เชื่อมต่อกับเซลล์ประสาทก่อนที่จะนำไปผ่านฟังก์ชันกระตุ้น ซึ่งค่าน้ำหนักนี้ จะถูกปรับเปลี่ยนค่าไปตามการเรียนรู้ของแบบจำลองเพื่อให้ได้ผลลัพธ์สุดท้ายของแบบจำลองมีความใกล้เคียงกับคำตอบที่ต้องการมากที่สุด
4. ฟังก์ชันผลรวม (Summation function: S) เป็นผลรวมของข้อมูลป้อนเข้า (p_i) และค่าน้ำหนัก (w_i) ดังสมการ (1)

$$S = \sum_{i=1}^n p_i w_i \quad (1)$$

5. ฟังก์ชันการแปลง (transfer function) ฟังก์ชันการแปลง เป็นส่วนที่ทำหน้าที่รวมค่าเชิงตัวเลขจากเอาต์พุตของนิวรอน แล้วทำการตัดสินใจว่าจะส่งสัญญาณเอาต์พุตออกไปในรูปใด ฟังก์ชันการแปลงสามารถเป็นได้ทั้งแบบเชิงเส้นหรือไม่เป็นเชิงเส้น การเลือกใช้ฟังก์ชันการแปลงจะขึ้นอยู่กับลักษณะของระบบ ที่นำเอาโครงข่ายประสาทเทียมไปประยุกต์ใช้ ฟังก์ชันการแปลงมีอยู่หลายรูปแบบที่ใช้งานทั่วไปมีรายละเอียดตามตารางที่ 2.1

ชื่อฟังก์ชัน	สมการ ความสัมพันธ์	MATLAB ฟังก์ชัน	สัญลักษณ์	กราฟความสัมพันธ์
ฮาร์ดลิมิต (hard limit)	$y = 0$ ถ้า $n < 0$ $y = 1$ ถ้า $n \geq 0$	hardlim		
ฮาร์ดลิมิตแบบ สมมาตร (symmetrical hard limit)	$y = -1$ ถ้า $n < 0$ $y = +1$ ถ้า $n \geq 0$	hardlims		
เส้นตรง (linear)	$y = n$	purelin		
เส้นตรงบวก (positive linear)	$y = 0$ ถ้า $n < 0$ $y = n$ ถ้า $n \geq 0$	poslin		
ซิกมอยแบบ ลอการิทึม (log- sigmoid)	$y = \frac{1}{1+e^{-n}}$	logsig		
ซิกมอยแบบเส้น สัมผัส ไฮเปอร์ โบลาร์ (hyperbolic tangent sigmoid)	$y = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	tansig		

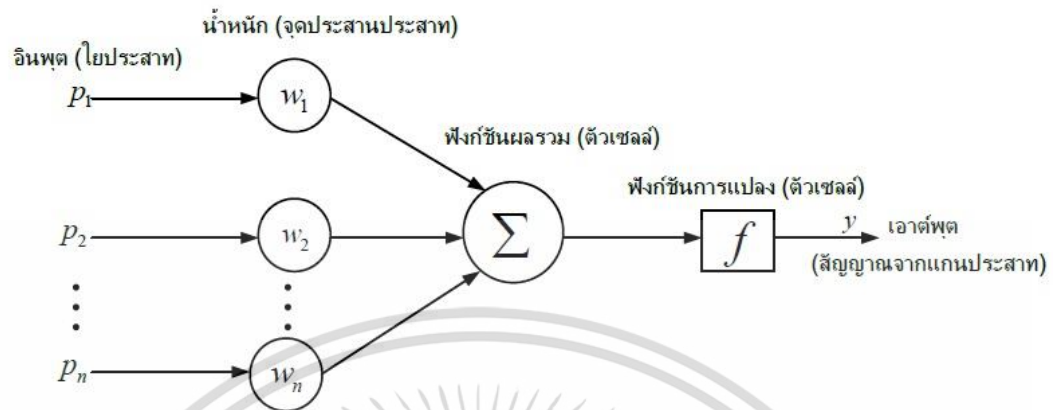
ตารางที่ 2.1 ฟังก์ชันการแปลงแบบต่างๆ

2.3.1.2 เพอร์เซปตรอน (Perceptron) [5]

เป็นหน่วยย่อยการเรียนรู้ของโครงข่ายประสาทเทียม ลักษณะดังภาพที่ 2.4 ประกอบด้วยเวกเตอร์ข้อมูลขาเข้า ($x_1, x_2, x_3, \dots, x_n$) ซึ่งจะถูกนำมาหาผลรวมเชิงเส้นแบบถ่วงน้ำหนักคูณด้วย $w_{1j}, w_{2j}, w_{3j}, \dots, w_{nj}$ ตามลำดับ และเพิ่มด้วยค่าเบี่ยงเบน (j) ก่อนที่จะนำค่าทั้งหมด (net_j) เข้าสู่ฟังก์ชันกระตุ้น $f(x)$ เพื่อสร้างข้อมูลขาออกของเซลล์ประสาท j ซึ่งจะถูกนำไปคำนวณหาค่าความคลาดเคลื่อน (error) ในแบบจำลองเมื่อเทียบกับคำตอบที่ต้องการค่าความคลาดเคลื่อนที่ได้นี้จะนำมาย้อนคำนวณเพื่อหาค่าน้ำหนักและค่าเบี่ยงเบนของแบบจำลองให้ได้ผลลัพธ์ที่ดีขึ้น ด้วยวิธีที่เรียกว่า backpropagation การทำงานเพื่อปรับค่าน้ำหนักและค่าเบี่ยงเบนของแบบจำลองจะทำงานไปเรื่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ๆ จนกว่าค่าความคลาดเคลื่อนจะน้อยมาก ๆ หรือค่าความคลาดเคลื่อนมีการเปลี่ยนแปลงค่าที่น้อยมาก ๆ แบบจำลองก็จะหยุดการปรับค่าน้ำหนักและค่าเบี่ยงเบน



ภาพที่ 2.4 โครงสร้างของเพอร์เซปตรอน

2.3.1.3 การเรียนรู้แบบมีผู้ฝึกสอน (learning with supervisor)

Supervised Learning[6] หรือการเรียนรู้แบบมีผู้ฝึกสอน เป็นศาสตร์แขนงหนึ่งใน AI หรือปัญญาประดิษฐ์ ภายใต้หัวข้อ Machine Learning ที่กำลังเป็นที่นิยมในการศึกษาและวิจัยกันในปัจจุบัน เนื่องจากทำได้ง่าย ต้นทุนต่ำ เพียงใช้คอมพิวเตอร์เครื่องเดียวก็สามารถศึกษาและทำงานจนเห็นผลได้เลยจริงๆ แล้วศาสตร์แขนงนี้มีมานานมากแล้วตั้งแต่ปี 1959 ถูกเสนอโดย Arthur Samuel แต่ด้วยเทคโนโลยีหรือระบบประมวลผลในตอนนั้นยังล้าสมัยอยู่ทำให้ยังไม่เป็นที่นิยมผิดกับในปัจจุบัน

Supervised Learning หรือการเรียนรู้แบบมีผู้ฝึกสอนนั้น คือการทำให้คอมพิวเตอร์สามารถหาคำตอบของปัญหาได้ด้วยตัวเอง หลังจากเรียนรู้จากชุดข้อมูลตัวอย่างไปแล้วระยะหนึ่ง หากจะให้เปรียบเทียบก็เหมือนกับการสอนเด็ก ลองนึกภาพว่าเราซื้อภาพสัตว์ให้เด็กที่ไม่เคยเห็นดู แล้วบอกว่า สัตว์ตัวไหนคือแมว ตัวไหนไม่ใช่แมว ซึ่ไป 2-3 วัน ให้เด็กได้เจอสัตว์หลายๆ ประเภท จนเด็กเริ่มเข้าใจ วันที่ 4-5 เราอาจจะลองเอาแมวตัวที่เด็กไม่เคยเห็นมาให้ดูสัก 10 ตัว รวมกับสัตว์อื่นๆ อีกจำนวนหนึ่ง โดยคราวนี้เราไม่บอกว่าสัตว์ตัวไหนคือแมว ตัวไหนไม่ใช่แมว ถ้าเด็กตอบถูกก็แปลว่าการสอนของเรามีประสิทธิภาพในทำนองเดียวกัน หากเราสอนเด็กไปเลยว่า สัตว์ที่เด็กเห็นนั้นเป็น แมว หมา หรือหมู เด็กก็อาจจะตอบได้มากกว่าแค่ แมว หรือไม่ใช่แมว วิธีนี้อาจจะต้องใช้กระบวนการสอนที่มีความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซับซ้อนมากขึ้นไปอีก เราเรียกรูปวิธีการสอนเด็กทั้ง 2 แบบนี้ว่า Classification ซึ่งจะได้ผลลัพธ์ตามภาพที่ 2.5 และ ภาพที่ 2.6

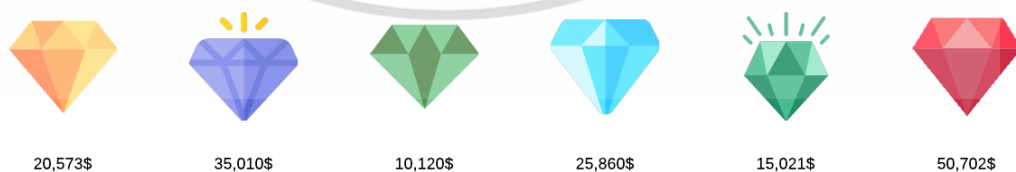


ภาพที่ 2.5 ผลลัพธ์ที่ได้จากการสอนแบบ Classification ที่ไม่ซับซ้อน



ภาพที่ 2.6 ผลลัพธ์ที่ได้จากการสอนแบบ Classification ที่ซับซ้อน

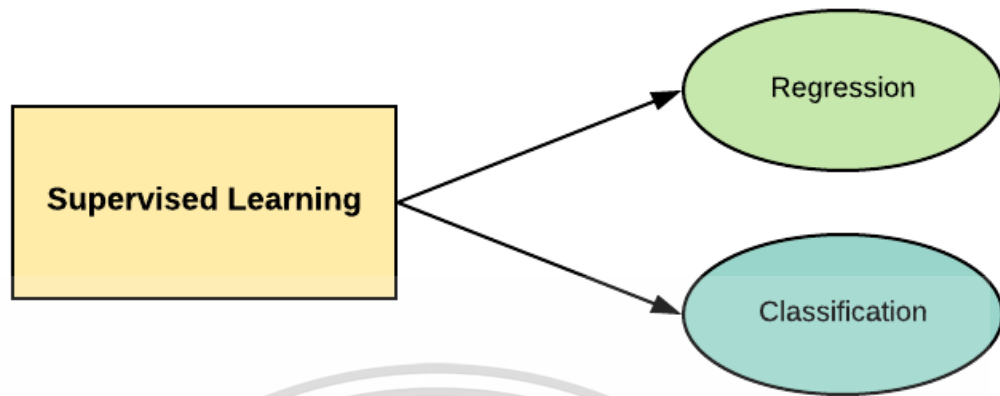
ถัดมาเรียกเด็กอีกคนมาสอนเรื่องราคาเพชร (diamond) เราหยิบเพชรอันหนึ่ง ขนาด 2 กระรัต สีเหลือง ระดับความสะอาด VS2 แล้วบอกเด็กว่า ราคา 2 ล้านบาท หยิบอีกเม็ดขนาด 3 กระรัต สีฟ้า ระดับความสะอาด VS1 แล้วบอกเด็ก 3 ล้านบาท ทำแบบนี้ไปหลาย ๆ เม็ดจนเด็กเกิดโมเดล หรือ กระบวนการคิด(logic) ในการคาดเดาราคาของเพชรขึ้นในหัว จนสวมหยิบเพชรเม็ดใหม่ขึ้นมา ก็อาจให้เด็กคาดเดาราคาได้เลย เราเรียกกระบวนการสอนเด็กแบบนี้ว่า Regression โดยแสดงผลลัพธ์การสอนได้ทั้งภาพที่ 2.7 และ ภาพที่ 2.8



ภาพที่ 2.7 ผลลัพธ์ที่ได้จากการสอนแบบ Regression

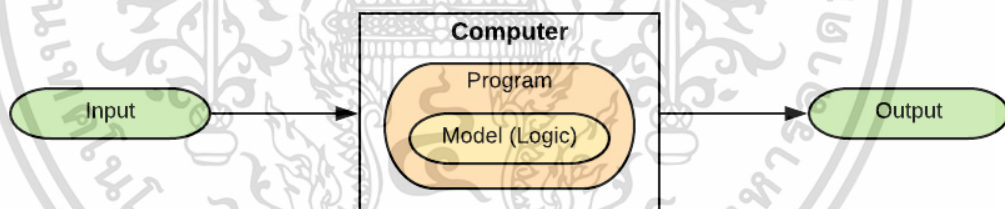
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการ Supervised Learning สามารถนำไปประยุกต์ใช้แก้ปัญหาได้ 2 รูปแบบ



ภาพที่ 2.8 การประยุกต์แก้ปัญหาของ Supervised Learning

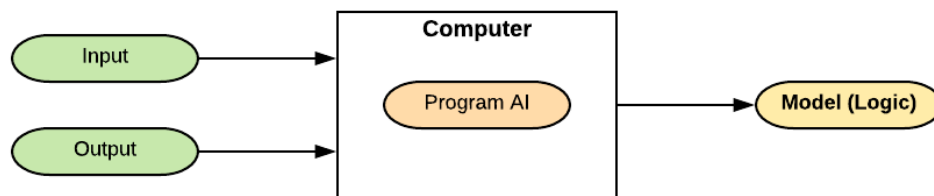
การเขียนโปรแกรมแบบดั้งเดิม เราจะนำ logic หรือ model ที่เราคิดขึ้นมา ใช้เขียนโปรแกรมเพื่อให้ได้ output จาก input ที่รับเข้ามา หากนำไปเปรียบเทียบกับตัวอย่างการสอนเด็ก ข้างต้น input คือภาพสัตว์ชนิดต่าง ๆ ส่วน output คือคำตอบว่าภาพที่รับเข้าไปเป็นภาพสัตว์ชนิดอะไร การเขียนโปรแกรมแบบนี้แทบเป็นไปได้ เนื่องจากความซับซ้อนของ model หรือ logic ที่เราต้องเป็นคนคิดขึ้นมาใช้ในโปรแกรม เพื่อแยกแยะภาพสัตว์ชนิดต่าง ๆ ดังภาพที่ 2.9



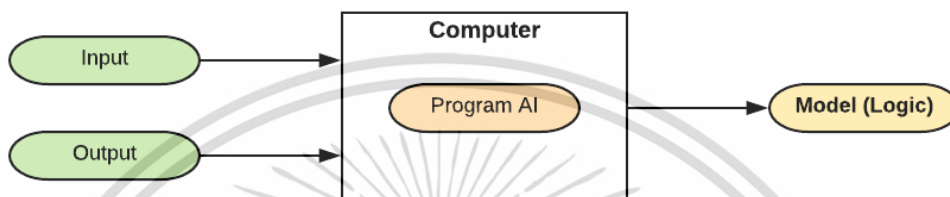
ภาพที่ 2.9 Traditional Programming

การใช้ Supervised Learning มีรูปแบบการเขียนโปรแกรมที่แตกต่างออกไป ในช่วงแรก เราเขียนโปรแกรมให้คอมพิวเตอร์สร้าง model หรือ logic ของโปรแกรมขึ้นมาเองจาก input (ภาพสัตว์) และ output (คำตอบ) ที่เราต้องการเช่นเดียวกับการสอนเด็ก จากนั้นเราจึงนำ model มาใช้ ดังนั้นยิ่งเรามี input และ output ที่มีความหลากหลายและจำนวนมากเท่าไร มี“โอกาส”ได้ model ที่มีความแม่นยำมากขึ้น กระบวนการสร้าง model แบบนี้เราเรียกว่าการ “ฝึก (train)” ซึ่งสามารถกินเวลาได้ตั้งแต่หลักวินาทีจนถึงหลายๆ วัน แล้วแต่ความซับซ้อนของโจทย์ที่เราต้องการแก้ และพลังในการประมวลผลของเครื่องคอมพิวเตอร์ที่เราใช้เทรนดังภาพที่ 2.10 และ ภาพที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.10 แสดงกระบวนการเทรน เพื่อให้ได้ model ที่ต้องการ



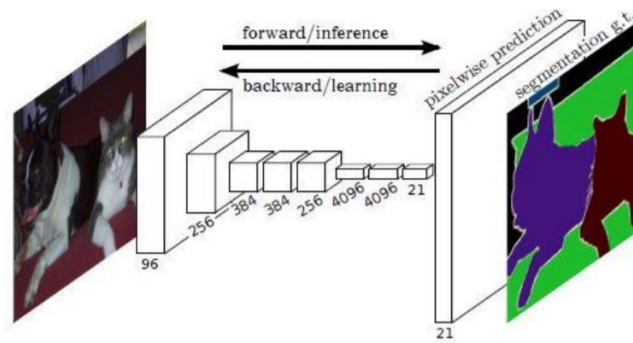
ภาพที่ 2.11 การนำโมเดลที่ต้องการมาประยุกต์ใช้กับโปรแกรม

จะเห็นว่ากระบวนการทำ Supervised Learning นั้นมีความซับซ้อนมากกว่าการเขียนโปรแกรมแบบดั้งเดิม แต่ข้อดีของมันก็คือสามารถทำสิ่งที่เป็นไปได้ให้เป็นไปได้ หากนี้ย้อนกลับไปเมื่อ 10–20 ปีก่อน การให้คอมพิวเตอร์มีความฉลาดพอที่จะแยกแยะสิ่งของได้ยังเป็นเหมือนดั่งความฝัน แต่ปัจจุบันนี้เราสามารถสร้าง model แบบที่ว่ามันขึ้นมาได้เอง เนื่องจากเทคโนโลยีต่าง ๆ มีความทันสมัยมากขึ้น ปัญหาที่เคยยากและไม่เคยแก้ได้ ก็สามารถถูกแก้ด้วยการใช้ทรัพยากรที่สมเหตุสมผลมากขึ้น

2.3.2 โครงข่ายประสาทคอนโวลูชันเต็มรูป (Fully Convolutional Neural Network-FCNN)

FCNN [7],[25] ถูกสร้างขึ้นด้วยเทคนิค Convolutional Network และ Deconvolutional Network ความแตกต่างระหว่าง CNN กับ FCNN คือ การใช้เทคนิค CNN นั้นจะใช้การเข้ารหัสภาพโดยคาดหวังให้ขั้นตอนสุดท้ายสามารถจำแนกรูปภาพออกมาได้เป็นชื่อภาพ แต่เป้าหมายของ FCNN คือการที่จะจำแนกในแต่ละพิกเซลหรือที่เรียกว่าการแบ่งส่วนตามความหมายจะต้องอาศัยการถอดรหัส (Decoding) ในชั้น Feature Map ของ CNN ซึ่งเป็นเทคนิคที่เรียกว่า Deconvolution ในปี ค.ศ. 2015 Long et al. ได้เผยแพร่ผลงานนำเสนอวิธีการ FCNN โดยได้เพิ่มชั้น Upsampling Layer เข้าไปแทนชั้นของ Fully connected layer และใช้ภาพ Segmentation Ground-Truth เป็นข้อมูลสอนของแบบจำลอง ดังภาพที่ 2.12

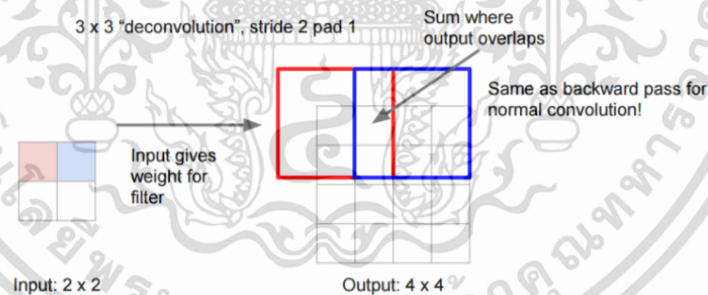
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.12 ภาพจำลองโครงข่ายประสาทคอนโวลูชันเต็มรูป

2.3.2.1 Upsampling Layer

Upsampling Layer [8] คือ ชั้นที่ใช้เทคนิค Deconvolution เพื่อขยายขนาดของ Feature Map ให้ใหญ่ขึ้นเท่ากับรูปภาพดั้งเดิม โดยใช้ข้อมูลขาเข้ามาสร้างเป็น Filter และนำไปพบกับข้อมูลขาออก โดยให้นำค่าบริเวณส่วนที่ทับซ้อนกันของ Filter ที่ทับลงไปบนข้อมูลขาออกมารวมกัน ดังภาพที่ 2.13 ผลลัพธ์สุดท้ายเมื่อผ่านชั้น Upsampling Layer จะมีขนาดด้านกว้างและด้านยาวเท่ากับรูปภาพดั้งเดิม แต่จะมีจำนวน Channels เท่ากับจำนวนคลาสในการจำแนก ซึ่งจะใช้ Softmax Function ในการจำแนกคลาสในแต่ละพิกเซลภายหลัง



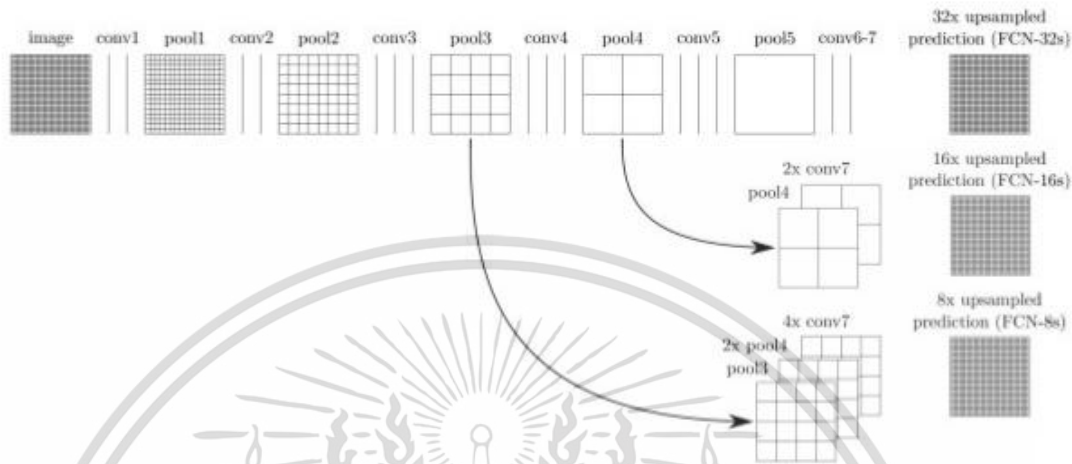
ภาพที่ 2.13 Deconvolution ในชั้น Upsampling Layer

2.3.2.2 Skip Connections [9]

เพื่อเพิ่มความแม่นยำและแก้ปัญหาการขาดหายของ feature map เมื่อผ่านแบบจำลองในชั้นที่อยู่ลึกเนื่องจากการทำ pooling จะส่งผลให้ข้อมูลนั้นละเอียดมากขึ้นการทำ upsampling ในชั้นตอนสุดท้าย จึงได้ผลลัพธ์ที่ผ่านการเรียนรู้เชิงลึก แต่สามารถถอดรหัสออกมาเป็นพิกเซลได้แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หยาบ การทำ skip connections คือ การนำเอาชั้น pooling ที่ต่ำกว่าชั้นสุดท้ายมาทำ upsampling ผลลัพธ์ที่ได้จึงสามารถถอดรหัสออกมาได้แบบละเอียดกว่าช่วยให้การจำแนกขอบในรูปภาพดีขึ้น ดังภาพที่ 2.14



ภาพที่ 2.14 Skip Connections เพื่อ up sampling ในชั้นที่ต่ำกว่าของแบบจำลอง

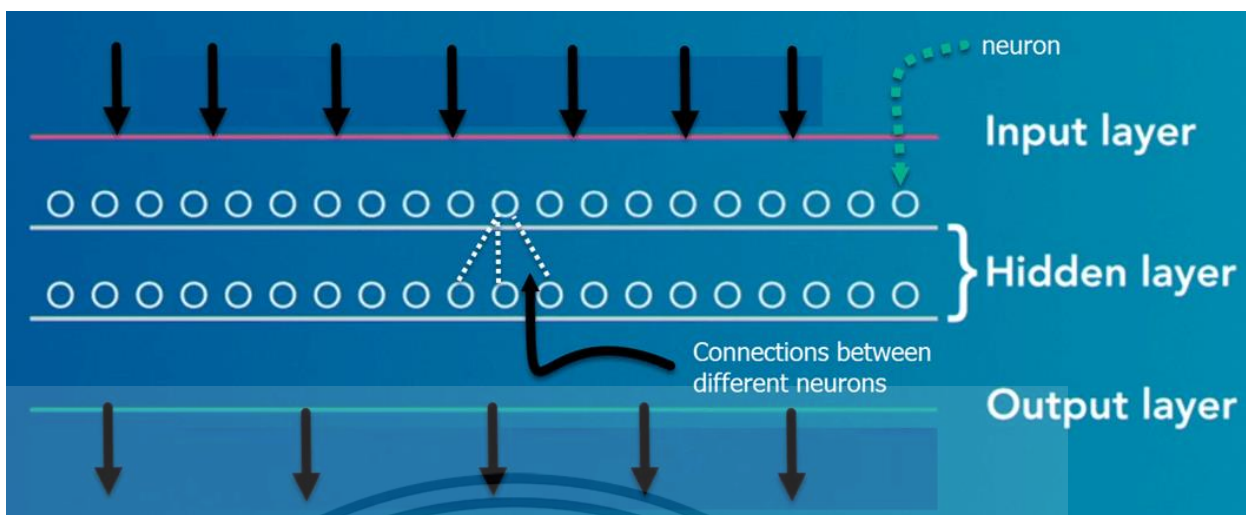
2.4 deep learning

2.4.1 deep learning [10] คือ ซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท (neurons) ในสมองมนุษย์ ถือเป็นซับเซตของ machine learning

Algorithm ของ deep learning ถูกสร้างขึ้นจากการนำเอา neural network หลายๆ layer มาต่อกัน โดย layer แรกสุดจะทำหน้าที่ในการรับข้อมูล (Input layer) layer สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา

(Output layer) ส่วน layer ระหว่าง layer แรกสุด และ layer สุดท้าย จะถูกเรียกว่า Hidden layer

คำว่า deep learning มีที่มาจากการใช้ layer ของ neural network หลายอันๆมาต่อกัน (มี Hidden layer มากกว่า 2 layer ก็ถือเป็น deep learning แล้ว) เนื่องจาก layer เหล่านี้เป็นโครงสร้างที่ถูกจัดเก็บแบบเป็นกองซ้อน (stack) จึงเปรียบได้ว่า layer ที่จำนวนเยอะๆ ก็จะทำให้มีโครงสร้างที่ลึก(deep) ยิ่งขึ้นนั่นเอง โดยแสดงในภาพที่ 2.15



ภาพที่ 2.15 hidden layer

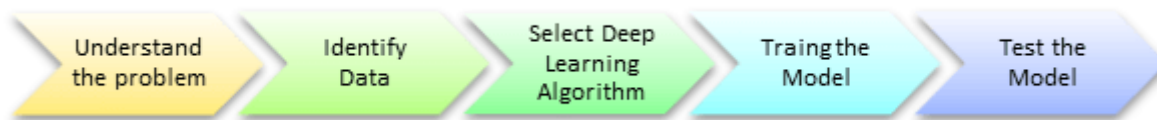
โดย Hidden layer ของแต่ละ layer จะเปรียบเสมือนว่าประกอบด้วย เซลล์ประสาท (neural) จำนวนมาก ซึ่งมีหน้าที่ในการประมวลผล รับข้อมูลจาก layer ที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยัง layer ที่อยู่ต่ำกว่า, ข้อดีของการส่งข้อมูลแบบนี้ก็คือ layer แต่ละ layer สามารถที่จะมี ค่าถ่วงน้ำหนัก (weight), ค่าความเอนเอียงของข้อมูล (bias) และ วิธีการประมวลผลทางคณิตศาสตร์ (activation function) ที่เป็นอิสระต่อกันได้

ยิ่งเราป้อนข้อมูลให้กับ model มากเท่าไร, layer แต่ละ layer ก็จะสามารถสกัด feature ที่มีความซับซ้อนมากยิ่งขึ้น

2.4.2 กระบวนการทำงานของ deep learning [23]

Model ที่ใช้ deep learning ให้ความแม่นยำ (accuracy) ที่สูงในหลายๆปัญหา ตั้งแต่การตรวจจับวัตถุ (object detection) ไปจนถึงการรู้จำเสียงพูด (speech recognition) โดยที่เราไม่จำเป็นต้องให้ความรู้พื้นฐานใดๆกับมันไว้ล่วงหน้าเลย เพียงแค่ให้ข้อมูลตัวอย่าง (input data) มันก็จะทำการเรียนรู้จากข้อมูลและสังเคราะห์เป็นองค์ความรู้ออกมาได้อย่างอัตโนมัติ อาทิเช่น การใช้ deep learning ในวงการเกม เราไม่จำเป็นต้องสอนว่าเล่นยังไง แค่ให้เรียนรู้จากผู้เล่นที่เก่งๆเป็นจำนวนมาก ก็สามารถเรียนรู้วิธีการเล่นเกมได้อย่างอัตโนมัติ โดยมีกระบวนการทำงานแสดงในภาพที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 กระบวนการทำงานของ deep learning

เพื่อที่จะเข้าใจหลักการของ deep learning มากยิ่งขึ้น , ให้คุณจินตนาการถึงครอบครัวที่มี พ่อแม่ และ ลูกน้อยวัยละอ่อน , ลูกน้อยคนนั้นมักจะชี้สิ่งต่างๆและพูดขึ้นพูดว่า " นี่คือเจ้าแมว " ซึ่งไม่ว่าจะชี้ไปที่ ทีวี ตู้เย็น หรือ สุนัข ก็มักจะพูดประโยคนี้ออกมา ในฐานะพ่อแม่ ทุกครั้งที่ลูกกำลังคิดผิดก็ควรมีการบอกกล่าว " ไม่ใช่ นั่นไม่ใช่แมว " แต่หากสิ่งที่ลูกชี้ขึ้นนั้นเป็นแมวจริงๆ บอกลูกว่า "ถูกต้อง นั่นแหละแมว " เมื่อทำเช่นนี้ไปเรื่อย ๆ เด็กคนนั้นก็จะสามารถแยกแยะได้เองโดยอัตโนมัติว่าสิ่งไหนคือแมว สิ่งไหนไม่ใช่แมว โดยที่อาจไม่ทันได้สังเกตด้วยซ้ำว่าใช้เราใช้อะไรเป็นเกณฑ์แยกแยะว่าสิ่งเป็นแมวหรือไม่ใช่แมว ซึ่งมันก็อาจจะเป็น ทาง , สี , ขนาดของมัน หรืออะไรที่ซับซ้อนกว่านั้นก็ไม่ได้ (สิ่งเหล่านี้คือตัวอย่างของ feature ที่นำมาใช้เป็นเกณฑ์แยกแยะของ model ในการแยกแยะว่าเป็นแมวหรือไม่) , neural network ก็มีหลักการทำงานเช่นเดียวกันกับตัวอย่างข้างต้น โดยหากมี layer ที่เยอะเท่าไร ก็จะสามารถสังเคราะห์ feature ที่มีความซับซ้อนได้มากยิ่งขึ้น ตัวอย่างเช่น neural network ที่มี 4 layer ย่อมสามารถสังเคราะห์ feature และ ทำงานได้ซับซ้อนมากกว่า neural network ที่มีเพียง 2 layer

การเรียนรู้เกิดขึ้นจาก 2 เฟส

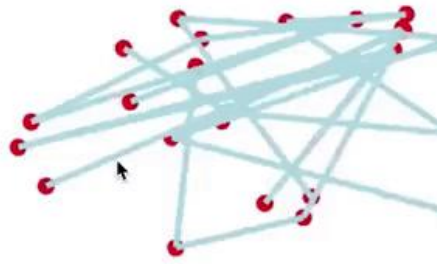
เฟสแรกคือ การประยุกต์ใช้ การแปลงแบบไม่เป็นเชิงเส้น (nonlinear transformation) กับ ข้อมูลที่ได้รับ (input) ได้ผลลัพธ์ (output) ออกมาอยู่ในรูปของโมเดลทางสถิติ (statistical model)

เฟสที่สอง คือ การนำโมเดลมาผ่านวิธีการทางคณิตศาสตร์ อย่าง derivative หรือ การดิฟ นั้นเอง

โดยทั้ง 2 เฟสนี้จะถูกทำซ้ำแล้วซ้ำเล่าจนกว่าจะได้มาซึ่งโมเดลที่ความแม่นยำ (accuracy) ในระดับที่น่าพึงพอใจ ซึ่ง การทำซ้ำๆ ของ 2 เฟสนี้ มีชื่อว่า iteration

ยกตัวอย่างการใช้งาน deep learning , ภาพข้างล่าง คือ โมเดลที่พยายามจะเรียนรู้ว่าจะเดินได้อย่างไร ซึ่งใช้ระยะเวลาในการ train เพียงแค่ 10 นาที ผลลัพธ์ที่ได้คือ โมเดลยังไม่สามารถที่จะสังเคราะห์ท่าเดินออกมาได้ มันเลยแสดงผลออกมาแบบมั่วๆ ดังแสดงในภาพที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.17 การเรียนรู้ของ deep learning เฟสแรก

หลังจาก 48 ชั่วโมงของการ train , อาจเรียกได้ว่าเจ้า model ตัวนี้เป็นนักเต้นต้วยงเลยก็ได้ ซึ่งสามารถเห็นได้ในภาพที่ 2.18 การเรียนรู้ของ deep learning เฟสสอง



ภาพที่ 2.18 การเรียนรู้ของ deep learning เฟสสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

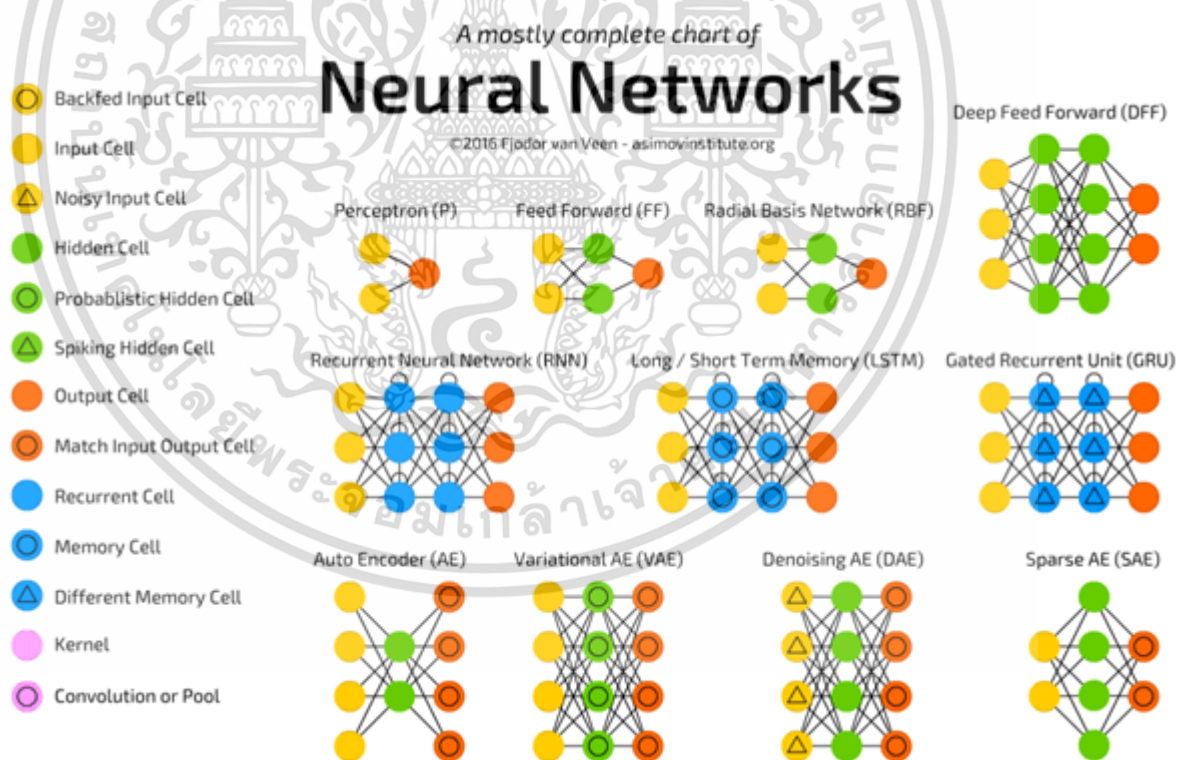
2.4.3 Classification of Neural Networks [11]

1) โครงข่ายประสาทแบบตื้น (Shallow neural network) : มี Hidden layer แค่ 1 layer (แต่ยังมี input layer และ output layer ประกอบเหมือนเดิม)

2) โครงข่ายประสาทแบบลึก (Deep neural network) : มี Hidden layer มากกว่า 1 layer , ยกตัวอย่างเช่น LeNet model ของ Google ที่ใช้สำหรับ การรับรู้ภาพ (image recognition) มี hidden layer มากถึง 22 layer

ทุกวันนี้ เราได้นำ deep learning มาประยุกต์ใช้อย่างแพร่หลาย เช่น รถยนต์ไร้คนขับ (driverless car) สมาร์ทโฟน search engine ของgoogle เครื่องจับเท็จ (Fraud detection) โทรท์สน์ และ อื่นๆอีกมากมาย

2.4.4 ชนิดของโครงข่าย deep learning (Types of Deep Learning Networks) โดยมี ชนิดของโครงข่ายทั้งหมด แสดงใน ภาพที่ 2.19



ภาพที่ 2.19 ชนิดของโครงข่าย deep learning

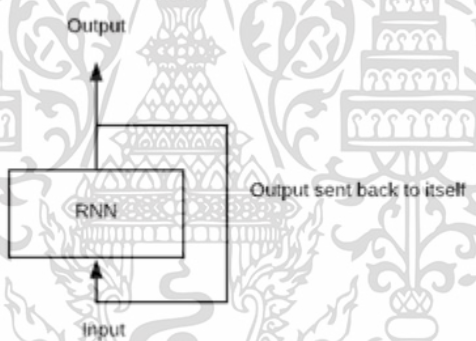
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.1 โครงข่ายประสาทแบบป้อนไปหน้า (Feed-forward neural networks)

Feed-forward neural networks ถือเป็นโมเดลที่มีโครงสร้างที่เรียบง่ายที่สุด เพราะว่า การดำเนินการของข้อมูลจะเป็นไปในทิศทางเดียว ก็คือ รับข้อมูลจาก input layer แล้วส่งต่อไปยัง hidden layer เลือดยๆ จนกระทั่งถึง output layer ก็จะหยุด (สังเกตได้ว่าจะไม่มีวงวน(loop) เกิดขึ้นเลย)

2.4.4.2 โครงข่ายแบบวนซ้ำ (Recurrent neural networks : RNN)

Recurrent neural networks คือ neural networks หลายเลเยอร์ที่สามารถเก็บ (store)ข้อมูล(information)ไว้ที่ node จึงทำให้มันสามารถรับข้อมูลเป็นแบบลำดับ (data sequences) และให้ผลลัพธ์ออกเป็นลำดับของข้อมูลได้ อธิบายอย่างง่ายๆ RNN ก็คือ neural network เชื่อมต่อกันหลายๆอันและยังสามารถต่อกันเป็นวงวน(loop)ได้นั่นเอง ดังแสดงในภาพที่ 2.20 เพราะฉะนั้น RNN จึงเหมาะสมในการประมวลผลข้อมูลที่เป็นลำดับอย่างมาก



ภาพที่ 2.20 โครงข่ายแบบวนซ้ำ

neural ใน RNN จะได้รับสัญญาณบอกว่าจุดไหนเป็นจุดเริ่มต้นของการรับข้อมูล

หลังจากที่ model ได้รับคำว่า "Do" เป็น input มันจะทำการสร้างเวกเตอร์ของตัวเลขที่ใช้แทนคำๆ นั้น ซึ่ง เวกเตอร์ นี้เปรียบเสมือน memory ของ model RNN

model จะทำรับ input จากคำต่อไปตามลำดับ " you" ,"want" ,"a"

โดยเมื่อ model ได้รับ input สุดท้ายของลำดับ ("a") , neural network จะส่งคำที่น่าจะสามารถเข้ากับประโยคได้(output) ออกมาพร้อมกับค่าความน่าจะเป็นของแต่ละคำ เช่น ถ้าหาก train model ออกมาดี มันอาจจะส่งคำว่า "café" หรือ "drink" หรือ "burger" เป็น output ที่มีค่าน่าจะเป็นสูงๆนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโดยทั่วไปของ RNN

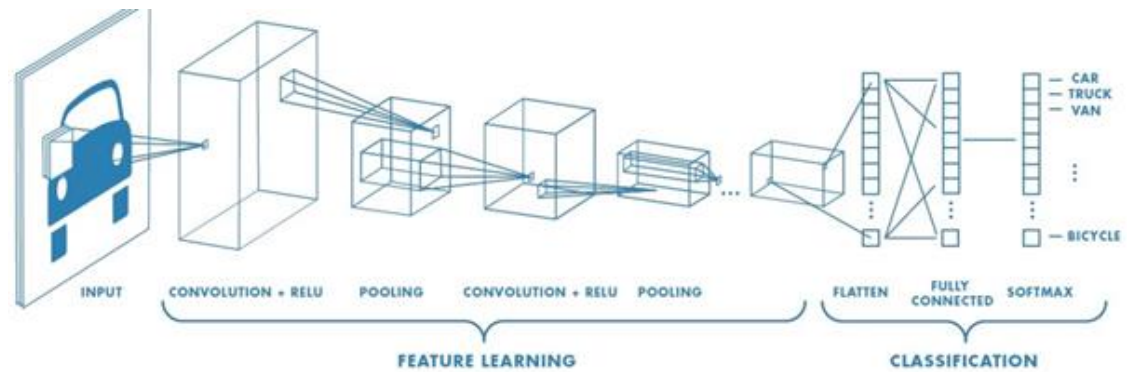
- ช่วยเทรดเดอร์ในการวิเคราะห์หลักทรัพย์ต่างๆ (Help securities traders to generate analytic reports)
- ตรวจสอบความผิดปกติในสัญญาการเงินต่างๆ (Detect abnormalities in the contract of financial statement)
- ตรวจสอบการปลอมแปลงบัตรเครดิต (Detect fraudulent credit-card transaction)
- เขียนคำบรรยายให้รูปภาพ (Provide a caption for images)
- สร้าง chatbot ที่มีประสิทธิภาพ (Power chatbots)
- การใช้งานในงานประเภทที่ต้องอาศัยเวลาทำกับข้อมูลอย่างตัวเนื่อง (time-series data or sequences) เช่น การอัดเสียง (audio record)

2.4.4.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks :CNN)

CNN เป็นเทคนิคหนึ่งของการเรียนรู้เชิงลึกที่ถูกพัฒนาขึ้นเพื่อวัตถุประสงค์ในการทำงานการเรียนรู้ด้วยภาพเนื่องจากลักษณะของภาพในคอมพิวเตอร์ประกอบด้วยพิกเซลหลายๆจุดมาต่อกันการตีความจากภาพด้วยวิธีการ CNN จึงใช้วิธีเปลี่ยนพิกเซลของรูปภาพให้อยู่ในรูปแบบการเข้ารหัส (encoding) ตัวเลขโดยวิธีการหนึ่งที่ยอมรับคือใช้ความเข้มสี RGB แทนข้อมูลในแต่ละพิกเซล ซึ่งจะมีค่าได้ตั้งแต่ 0-255 การเข้ารหัสจะประกอบด้วยชั้น convolution layer และชั้น pooling layer ซ้ำๆกันหลายชั้นซึ่งจะเรียกส่วนนี้ใน CNN ว่า convolutional base แล้วจึงมีชั้นสุดท้ายเป็น Fully Connected Layer ที่รวมข้อมูลทั้งหมดเข้าด้วยกัน ด้วยวิธีการนี้จะสามารถจำแนกประเภท (classify) ภาพได้โดยใช้ข้อมูลขาเข้า (input data) เป็นรูปภาพ และให้การทำนายผลลัพธ์ (output predictions) จากแบบจำลองเป็นความน่าจะเป็นที่รูปภาพจะตรงกับฉลาก (label) กำกับภาพดังกล่าว

ที่ 2.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.21 โครงข่ายประสาทแบบคอนโวลูชัน

CNN นั้น มักจะถูกใช้เพื่อการสกัด feature จากข้อมูลประเภทที่ไม่ค่อยเป็นระเบียบหรือไม่ได้มีโครงสร้างเป็นรูปแบบเฉพาะตัว (unstructured data) อย่างเช่น รูปภาพ (image) เป็นต้น

ตัวอย่างการใช้งาน CNN : การคิดคำบรรยายให้รูปภาพ (predict an image caption)

สมมติ CNN รับรูปภาพ input เป็นรูปแมว , รูปภาพเหล่านี้จะถูกจัดเก็บในรูปแบบของ pixel ซึ่งโดยทั่วไปแล้ว จะใช้ 1 layer ในการเก็บข้อมูลในรูปแบบเซตสีขาวดำ (greyscale) และจะใช้อีก 3 layer ในการเก็บข้อมูลในรูปแบบเซตสีต่างๆ

ระหว่างที่ model ทำการเรียนรู้(learn) (สกัด feature นั้นแหละ) ที่ hidden layer , model จะทำการหา feature ที่มีความเฉพาะเจาะจงต่อข้อมูล input ที่ได้รับ อย่างเช่น ในกรณีนี้อาจได้เป็น feature ที่ตรวจสอบหางของแมว เป็นต้น

เมื่อ model ทำการเรียนรู้(learn)เสร็จแล้ว มันจะสามารถให้คำตอบแก่แต่ละรูปภาพเป็นความน่าจะเป็น และหากรูปภาพนั้นๆมีความน่าจะเป็นแบบใดสูงที่สุด model จะตอบเป็นสิ่งนั้น เช่น หากโมเดลวิเคราะห์แล้วว่า รูปภาพนั้นมีความน่าจะเป็นว่าจะเป็นแมวสูง model ก็จะทำให้คำตอบว่ารูปภาพนั้นคือภาพแมวนั่นเอง

2.4.4.4 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

Reinforcement learning จัดเป็นรูปแบบหนึ่งของ machine learning โดยใช้รูปแบบของการให้รางวัล (reward) และ ลงโทษ (punishments) จากการเรียนรู้แบบลองผิดลองถูก (trial and error) Google's DeepMind ก็ใช้ Reinforcement Learning ในการ train model ในการเล่นหมากล้อมจนชนะแชมป์โลกมาแล้ว นอกจากนี้ยังมีการนำ Reinforcement Learning มาใช้ในวงการเกมเพื่อ train ให้ bot มีความฉลาดมากยิ่งขึ้น โดยมีการใช้ algorithm ที่น่าสนใจได้แก่ Q-learning ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deep Q network , State-Action-Reward-State-Action (SARSA) และ Deep Deterministic Policy Gradient (DDPG)

2.4.5 ความสำคัญของ Deep learning

Deep learning ถือเป็นเครื่องมือที่ทรงพลังในการทำนาย(prediction)ผลลัพธ์ต่างๆ อีกทั้งยังสามารถหา pattern หรือ สังเคราะห์ข้อมูล ได้เหนือกว่าองค์ความรู้เดิมที่มีอยู่ (unsupervised learning)

Big data ก็เปรียบเสมือนเชื้อเพลิงของ deep learning การผสมผสานระหว่างทั้งสองอย่างเรียกได้ว่าสามารถทำให้มนุษยชาติก้าวไปขึ้นอีกขั้น ไม่ว่าจะในด้านของ ผลผลิต การขาย การบริหาร และ นวัตกรรม

อีกทั้ง deep learning ยังทำงานได้มีประสิทธิภาพได้ดีกว่าวิธีการดั้งเดิม กล่าวคือ deep learning algorithm ให้ความแม่นยำมากกว่า machine learning algorithm ในด้านของจำแนกภาพ (image classification) มากถึง 41% ด้านของการรู้จำใบหน้า(face recognition)มากถึง 27% และการรู้จำเสียง (voice recognition) มากถึง 25%

2.4.6 ข้อจำกัดของ deep learning (Limitation of deep learning)

1) ต้องการระบุชื่อหรือจัดประเภทข้อมูล (Data labeling)

ปัจจุบัน AI ส่วนมากอาศัยการ train โดยใช้หลักการของ supervise learning ซึ่งนั่นก็หมายความว่ามนุษย์จะต้องตั้งชื่อและจัดประเภทของข้อมูล(data labeling) ด้วยตัวเองก่อนจะนำไป train (Data labeling professionals : ถูกคาดว่าจะจะเป็นอาชีพที่ popular ในอนาคต เมื่อเข้าสู่ยุคของ AI) ซึ่งข้อมูลเหล่านี้ก็มีจำนวนมาก จึงไม่แปลกที่อาจเกิดความผิดขึ้นได้ ตัวอย่างเช่น รถยนต์ขับเคลื่อนอัตโนมัติ (self - driving car) จำงคนหลายร้อยคนเพื่อที่จะ labeling วิดีโอที่จะใช้ในการ train ระบบ

2) ต้องการชุดข้อมูลจำนวนมาก (Obtain huge training datasets)

deep learning นั้นต้องข้อมูลจำนวนมากในการ train ข้อมูล ยกตัวเช่น ต้องการข้อมูลจำนวน 1000 ตัวอย่าง เพื่อจะทำให้ได้โมเดล classification ที่มีประสิทธิภาพ และในบางกรณีก็ต้องการมากกว่า 1 ล้านตัวอย่างเพื่อที่จะทำให้ model ทำงานได้อย่างมีประสิทธิภาพใกล้เคียงกับมนุษย์ , เน้นอนว่า

deep learning นั้นเป็นที่นิยมในบริษัทเทคโนโลยีชั้นนำทั่วโลก ซึ่งพวกเขาใช้ big data ที่สะสมข้อมูล เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีเนื้อที่ไม่น้อยกว่า petabytes (1 ล้าน gigabytes) นั้นเป็นเหตุผลว่าทำไม บริษัทเทคโนโลยีชั้นนำจึงสามารถสร้าง deep learning model ที่มีประสิทธิภาพ และ มีความแม่นยำสูง

3) ปัญหาในการอธิบาย (Explain a problem)

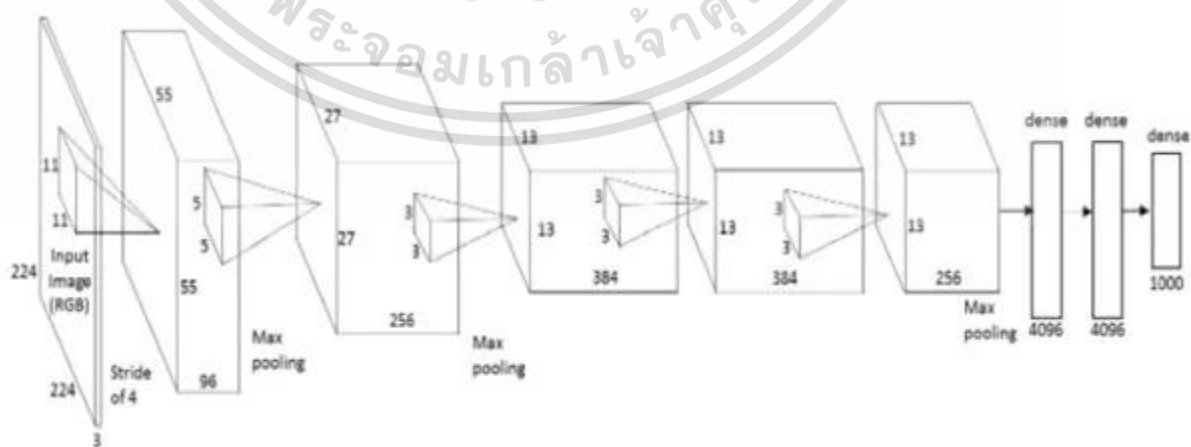
ไม่ใช่เรื่องง่ายที่มนุษย์เราจะอธิบาย AI model ที่ซับซ้อนได้ ยกตัวอย่างเช่น คำถามว่าทำไม model ถึงเกิดการตัดสินใจแบบนั้น (why a particular decision was obtained ?) นี่ก็เป็นอีกเหตุผลหนึ่งที่ทำให้ AI ถึงได้รับการยอมรับช้านัก ก็เพราะบางอย่างมันยากที่มนุษย์เราจะสามารถอธิบายนั่นเอง

2.4.7 10 สถาปัตยกรรม Deep learning ชั้นสูง [12]

2.4.7.1 AlexNet

Alexnet คือสถาปัตยกรรมแรกที่จะนำมาเสนอโดยเป็นผู้บุกเบิกใน Deep learning มันค่อนข้างจะง่ายและยังเป็น เครื่องมือสถาปัตยกรรมที่ทรงพลัง ซึ่งช่วยในการปูทางในการทำวิจัย Deep learning จนมาถึง ณ ตอนนี

Alexnet เหมือนจะเป็น สถาปัตยกรรมง่าย ๆ ซึ่งมีการ convolution และ pooling layer อยู่บนสุดและตามด้วยการ Fully connected layer โดยมีภาพแสดงหลักการทำงานในภาพที่ 2.14 Alexnet คือ สถาปัตยกรรมที่พื้นฐานที่สุดและถูกค้นพบในปี 1980 โดยจะทำการแยก model ตามมาตราส่วน และใช้ GPU สำหรับการฝึก ในปี 1980 CPU ถูกใช้สำหรับการฝึก neural network ในขณะที่ Alexnet มีความเร็วในการฝึกมากกว่า 10 เท่า ถึงแม้ว่าจะค่อนข้างล้าสมัยแล้วในปัจจุบัน Alexnet ก็ยังคงใช้เพื่อเป็นจุดเริ่มต้นของการประยุกต์ทำ deep neural network ในการประมวลผลทางภาพหรือ การจดจำคำพูดทั้งหมด โดยแสดงภาพสถาปัตยกรรมดังภาพที่ 2.22



ภาพที่ 2.22 สถาปัตยกรรมแบบ Alexnet

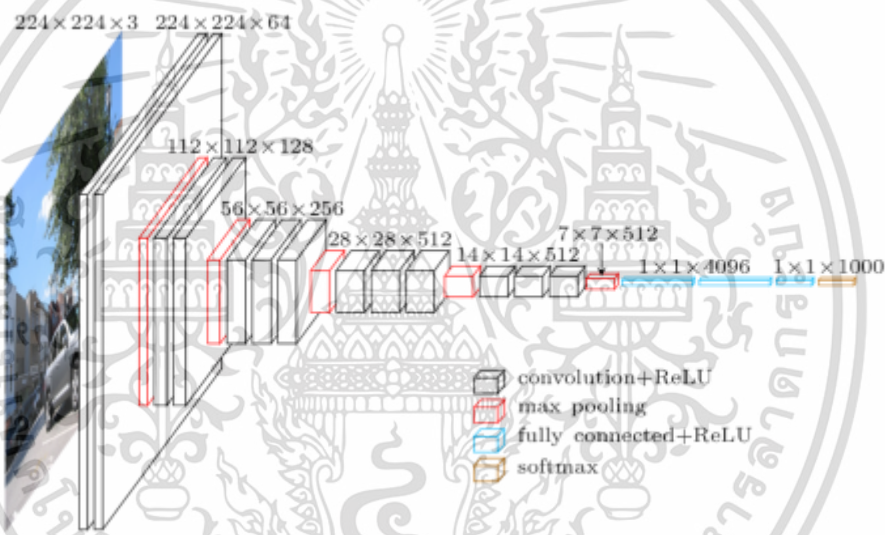
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.7.2 VGG Net VGG Network ถูกแนะนำโดยกลุ่มนักวิจัยภาพเสมือนจริงที่ oxford โดยเครือข่ายนี้มีรูปร่างคล้ายกับทรงพีระมิด ที่ซึ่งด้านล่างสุดที่ใกล้กับภาพนั้นจะกว้างใน ขณะที่บนสุดของเลเยอร์จะเล็ก โดยแสดงหลักการทำงานได้ดังภาพที่ 2.23 และ ภาพที่ 2.24

ข้อได้เปรียบของ VGG คือ เป็นสถาปัตยกรรมที่ดีสำหรับ benchmark เมื่อทำงานที่แยกกัน

อีกทั้งระบบเครือข่ายการ pre trained สำหรับการ VGG สามารถหาได้ทั่วไปบนอินเทอร์เน็ต

ในทางกลับกัน ข้อเสียของ VGG คือ เวลาที่ใช้ในการฝึกนั้นค่อนข้างช้าถึงจะทำงานบน GPU ก็ต้องใช้ เวลาเป็นอาทิตย์



ภาพที่ 2.23 สถาปัตยกรรมแบบ VGG Net VGG Network

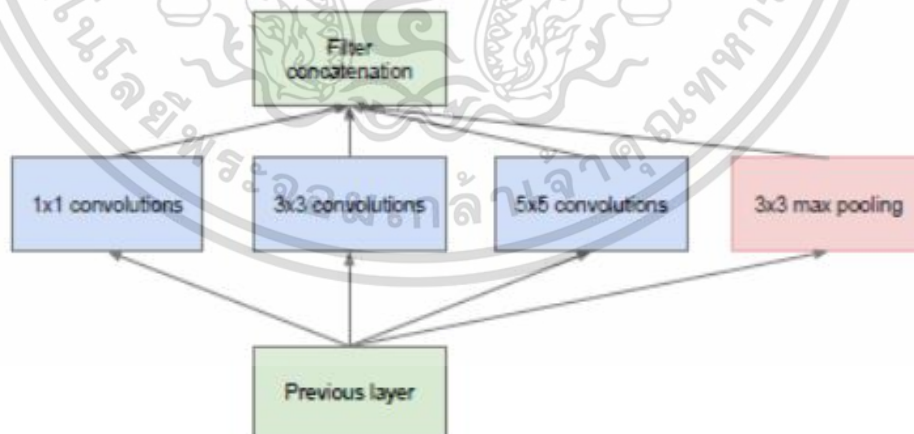
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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					

ภาพที่ 2.24 สถาปัตยกรรมแบบ VGG Net VGG Network

2.4.7.3 GoogleNet

Google Net เป็น class ของ สถาปัตยกรรมที่ถูกออกแบบมาโดยนักวิจัยที่ Google และยังคงชนะ ImageNet2014 อีกทั้งยังเป็น Model ที่ทรงพลัง มี สถาปัตยกรรมแสดงการทำงาน ในภาพที่ 2.25

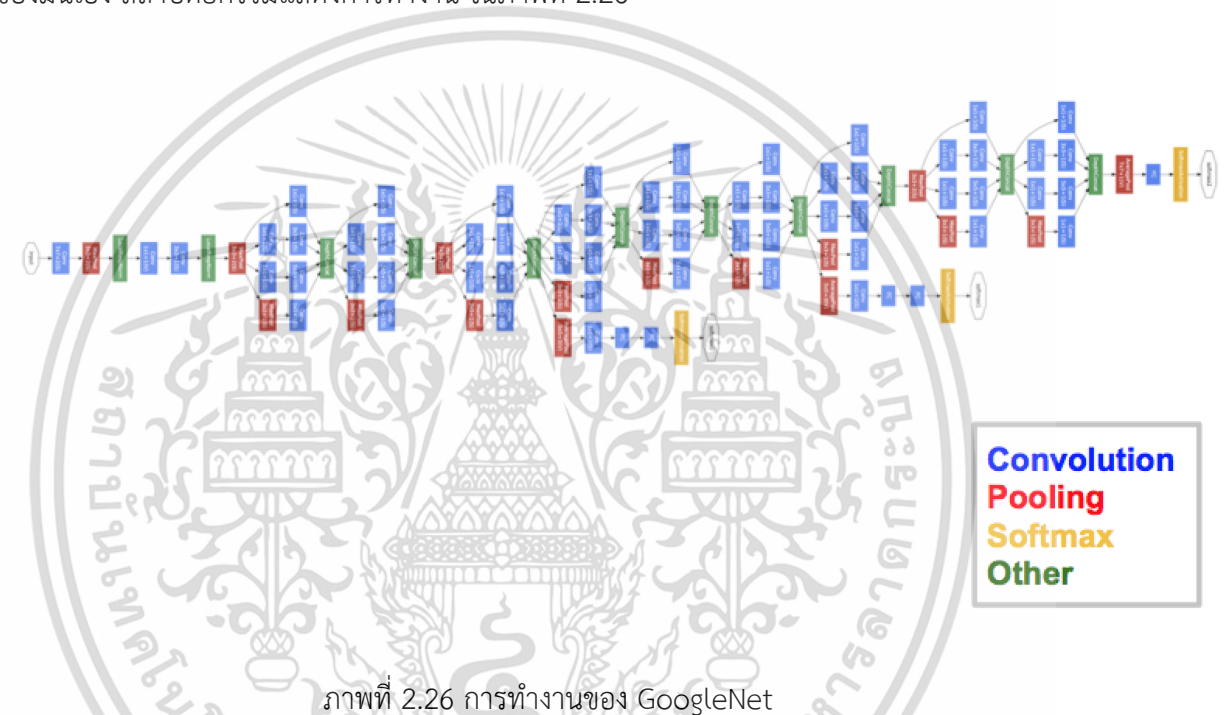


ภาพที่ 2.25 สถาปัตยกรรมแบบ GoogleNet

ในสถาปัตยกรรมนี้จะค่อนข้างลึกเพราะมีถึง 22 เลเยอร์เมื่อเทียบกับ VGG ที่มีเพียง 19 เลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามที่เราเห็นมาก่อนหน้า มีการเปลี่ยนแปลงไปมากกว่าที่เราเห็นสถาปัตยกรรมแบบ sequential ที่เห็นก่อนหน้านี้ ใน 1 เลเยอร์ มี feature extractors หลายชนิด ซึ่งสามารถช่วยให้ประสิทธิภาพของเครือข่ายดีอย่างเห็นได้ชัด อีกทั้งเครือข่ายที่ฝึกตัวมันเองได้ตามที่เราเลือกให้มันปฏิบัติ มันสามารถเลือกการ convolute input หรือทำการ pooling ได้โดยตรง สถาปัตยกรรมสุดท้ายมีการเก็บโมดูลเริ่มแรกซ้อนทับกับโมดูลอื่น ๆ อยู่มากมาย ถึงการฝึกจะแตกต่างกันใน GoogleNet ส่วนใหญ่เลเยอร์ชั้นบนสุดมี output เลเยอร์ของมันเอง โดยความแตกต่างนี้จะช่วยให้โมเดลมาบรรจบกันเร็วขึ้น และมีการฝึกที่เกี่ยวข้องกันโดยจะทำการฝึกแบบพาราเรลสำหรับเลเยอร์ของมันเอง สถาปัตยกรรมแสดงการทำงาน ในภาพที่ 2.26



ภาพที่ 2.26 การทำงานของ GoogleNet

ข้อได้เปรียบของ GoogleNet

-GoogleNet เร็วกว่า VGG

-ขนาดข้อมูล pre-trained ของ GoogleNet มีขนาดเล็กกว่า VGG โดย VGG จะมีมากกว่า 500MB ในขณะที่ GoogleNet มีเพียง 96 MB

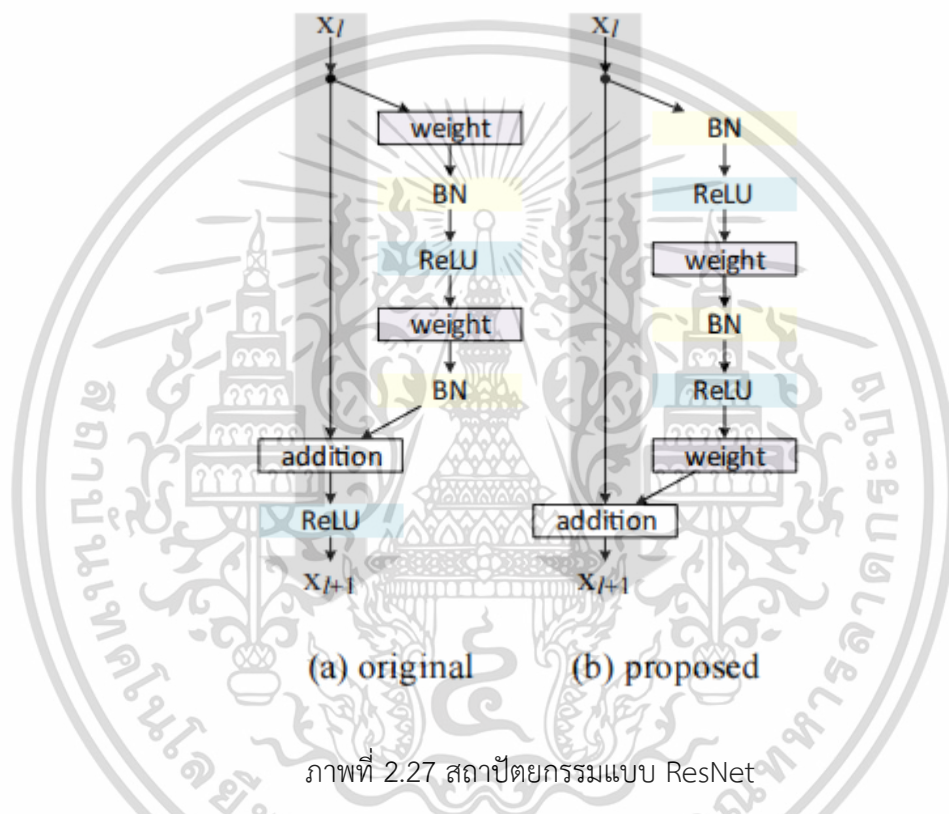
ข้อเสียของ GoogleNet

-ไม่ค่อยเห็นผลได้ชัดเพราะมีการเปลี่ยนแปลงสถาปัตยกรรมในอนาคตซึ่งจะทำให้โมเดลนั้นทำงานได้ดีขึ้น มีการเปลี่ยนแปลงอย่างหนึ่งที่เรียกว่า Xception Network ที่ซึ่งเกี่ยวข้องกับการจำกัดความแตกต่างของโมเดลเริ่มต้นให้มีมากขึ้นโดยตามหลักทฤษฎีตอนนี้มีแบบไม่จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.7.4 ResNet

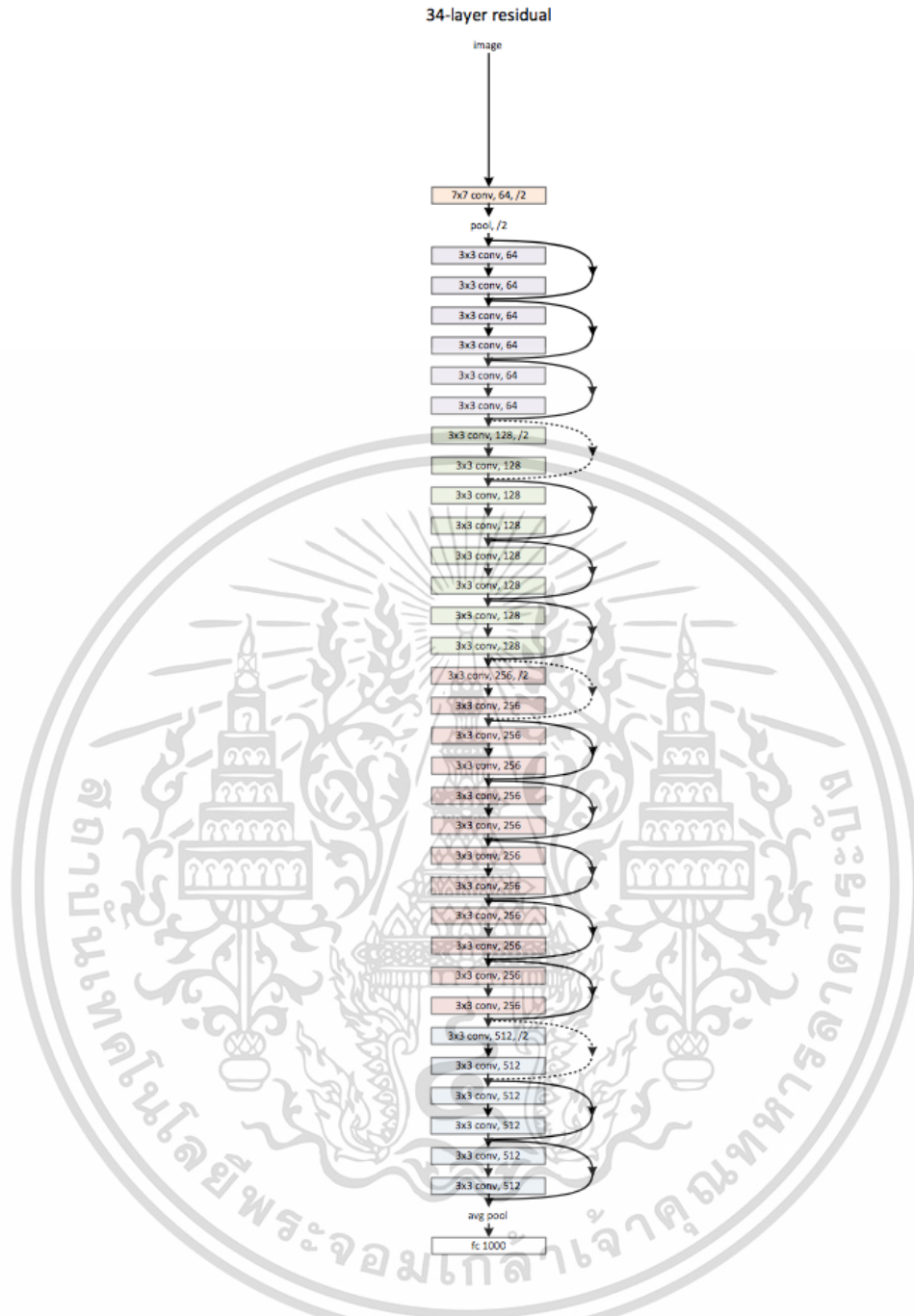
ResNet คือหนึ่งในสถาปัตยกรรมที่เปรียบได้กับความลึกของ สถาปัตยกรรม Deep learning สถาปัตยกรรมแสดงการทำงาน ในภาพที่ 2.27 จริง ๆ Residual Networks มาจากโมดูลล่าสุดหลายๆ ตัวตัวมาอยู่รวมกัน ซึ่งเป็นพื้นฐานของ บล็อกของสถาปัตยกรรม ResNet โดย Residual module มีสองทางเลือกหากไม่ทำตามที่ถูกตั้งตั้งแต่แรกสามารถข้ามขั้นตอนบางขั้นตอนได้ในปัจจุบันนี้จะคล้ายกับ GoogleNet Residual module เหล่านี้สามารถซ้อนทับกับโมดูลอื่น ๆ ได้



ภาพที่ 2.27 สถาปัตยกรรมแบบ ResNet

ResNet ใช้พื้นฐานของ SGD แทนที่ fancy adaptive learning technique ซึ่งจะทำการฝึกของโมเดลเป็นไปได้อย่างครบถ้วน อีกทั้งยังเปลี่ยนกระบวนการ preprocessing ของ input โดยแบ่ง input ออกและนำมาติดต่อกันและทำการป้อนเข้าสู่ระบบ ซึ่งสามารถแสดงหลักการทำงานได้ดังภาพที่ 2.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



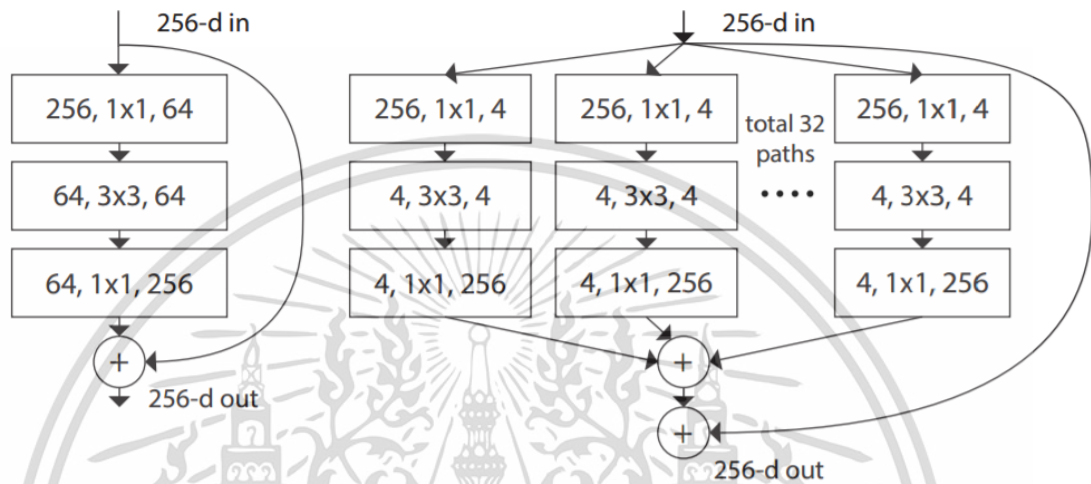
ภาพที่ 2.28 การทำงานของ ResNet

ข้อได้เปรียบของ ResNet มีมากมายอีกทั้ง Residual เลเยอร์นับพันใช้เพื่อทำการสร้างเครือข่ายและฝึก นี่ค่อนข้างแตกต่างจาก sequential network ทั่ว ๆ ไปที่ซึ่งมีการลดการเพิ่มประสิทธิภาพลงสัมพันธ์กับจำนวนเลเยอร์ที่มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4.5 ResNeXt

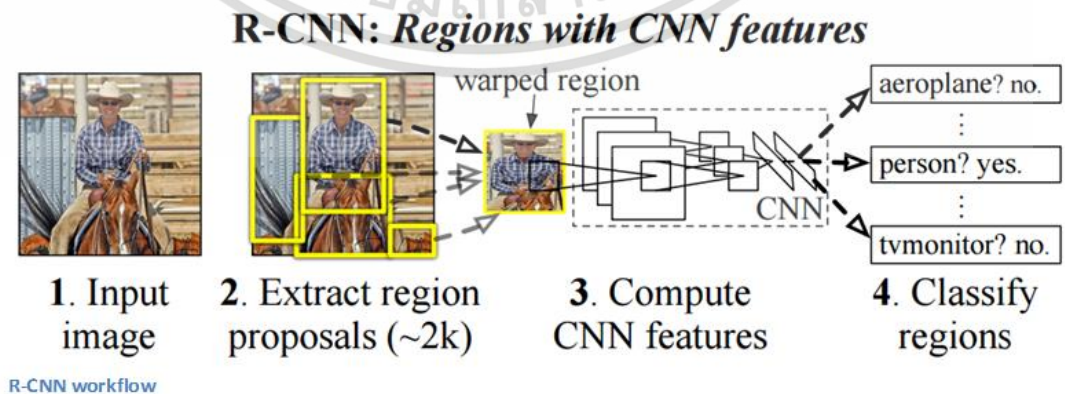
ResNeXt เป็นเทคนิคที่ปัจจุบันใช้สำหรับการทำ object recognition จะทำการสร้างคอนเซ็ปต์เริ่มแรกและนำ ResNet มาพัฒนาสถาปัตยกรรมใหม่ ภาพที่ 2.29 สำหรับสรุปการทำงาน residual module ของ ResNeXt



ภาพที่ 2.29 สถาปัตยกรรมแบบ ResNeXt

2.4.7.6 RCNN (Region Based CNN)

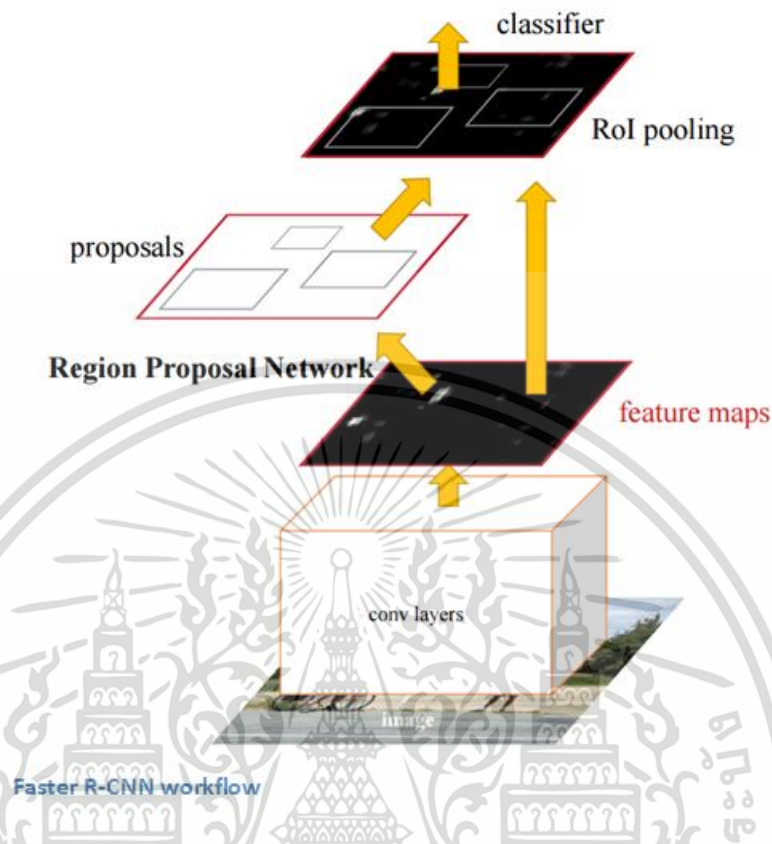
Region Based CNN architecture เรียกว่ามีอิทธิพลมากที่สุดสำหรับสถาปัตยกรรมของ deep learning แสดงในภาพที่ 2.30 สามารถนำไปประยุกต์ได้ การทำ object detection ได้ เพื่อแก้ปัญหา การ detection โดยมีโครงสร้างหลักการทำงาน แสดงในภาพที่ 2.31 โดย RCNN จะทำการวาดขอบเขตของกล่องที่วัตถุในรูปจากนั้นจะจดจำวัตถุนั้น ๆ



ภาพที่ 2.30 สถาปัตยกรรมแบบ RCNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

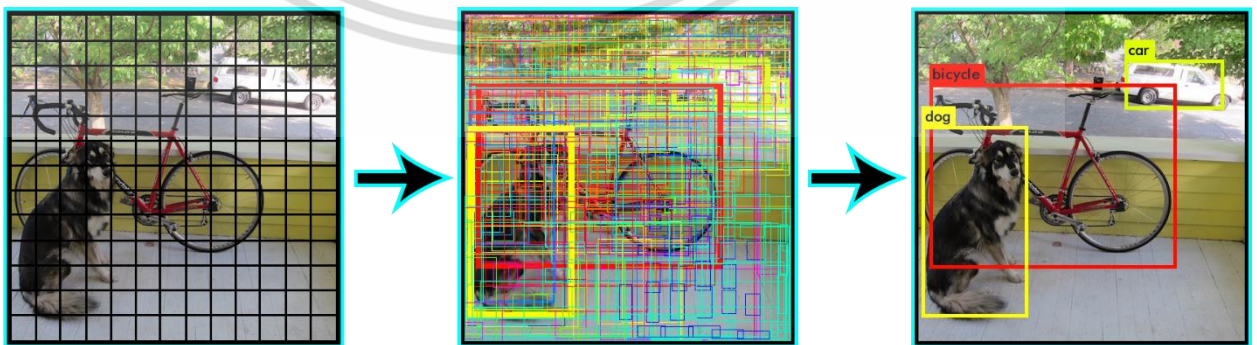
The structure of RCNN is as follows:



ภาพที่ 2.31 การทำงานของ RCNN

2.4.7.7 YOLO (You Only Look Once)

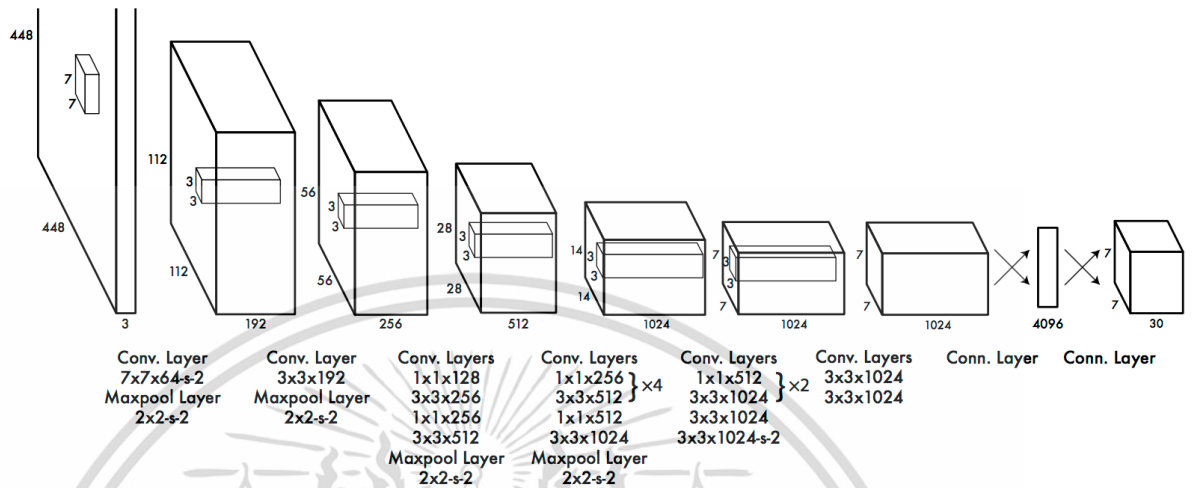
YOLO คือระบบที่สร้างบน deep learning สำหรับการแก้ปัญหา การตรวจพบวัตถุตั้งรูปที่เห็นด้านล่าง ขั้นแรกจะแบ่งรูปออกด้วยขอบเขตจากนั้นทำการจดจำอัลกอริทึมที่มควคู่ไปกับการตรวจสอบวัตถุเป็น class จากนั้นจะทำการรวมกล่องทั้งหมดต่อกัน ดังแสดงในภาพที่ 2.32



ภาพที่ 2.32 การทำงานของ YOLO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

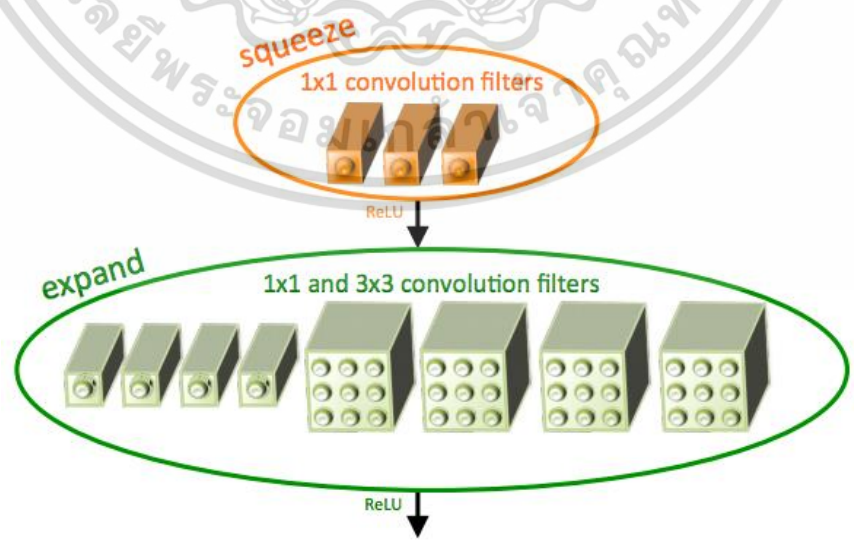
ถึงแม้จะให้ประสิทธิภาพที่ลดลงกว่า RCNN counterpart มันยังคงมีข้อได้เปรียบสำหรับการแก้ปัญหาแบบ real time ได้ และมีสถาปัตยกรรมดังแสดงในภาพ 2.33



ภาพที่ 2.33 สถาปัตยกรรมแบบ YOLO

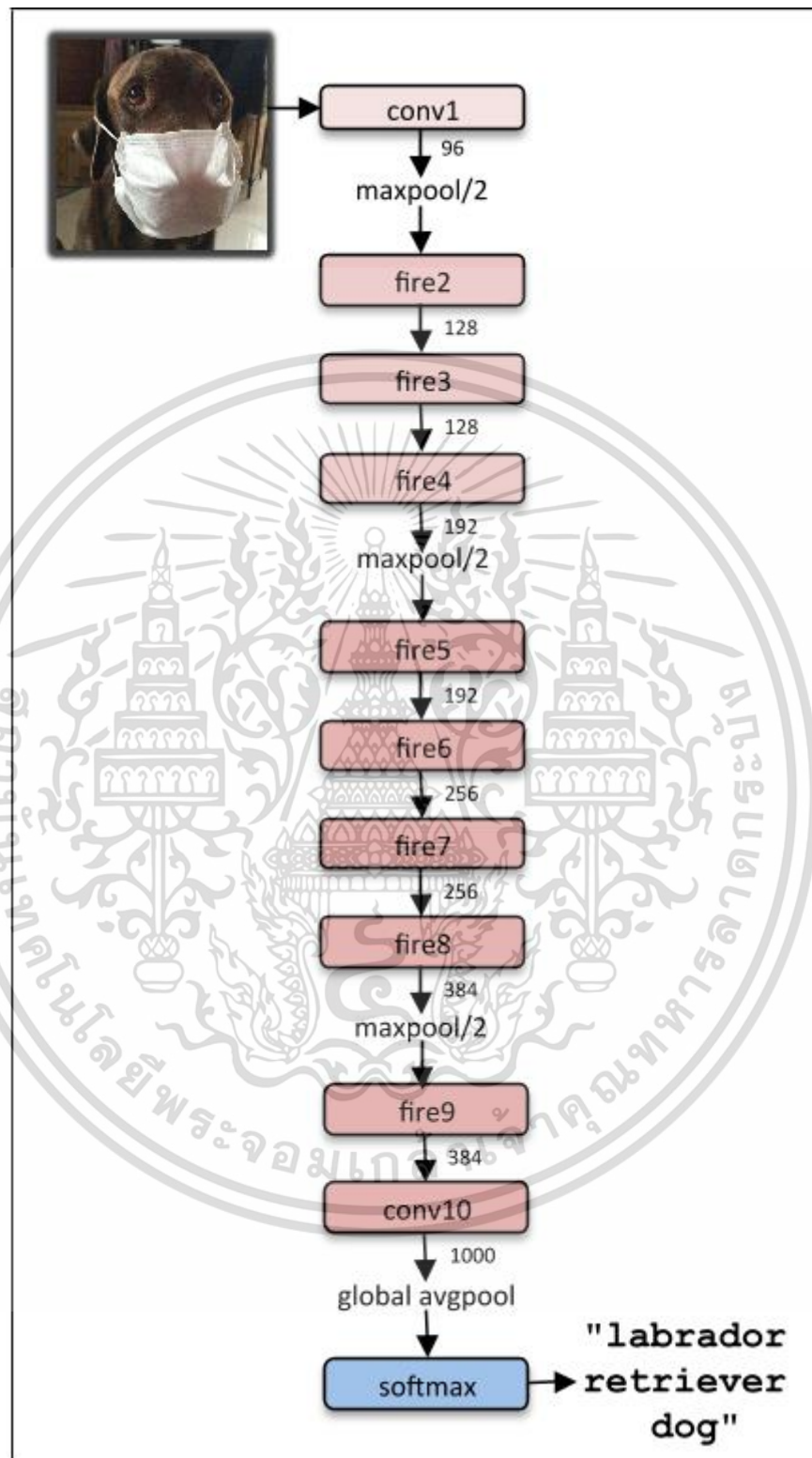
2.4.7.8 SqueezeNet

SqueezeNet คือสถาปัตยกรรมหนึ่งที่มีประสิทธิภาพที่มีประโยชน์อย่างมากในช่วงของ คลื่นความถี่ต่ำ ที่ใช้ใน แพลตฟอร์มมือถือ โดยสถาปัตยกรรมนี้สามารถแสดงได้ดังภาพมรา 2.34 ใช้พื้นที่เพียง4.9mb ในทางกลับกัน ในช่วงแรกใช้ประมาณ 100mb และมีหลักการทํางานแสดงได้ดังภาพที่ 2.35



ภาพที่ 2.34 สถาปัตยกรรมแบบ SqueezeNet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

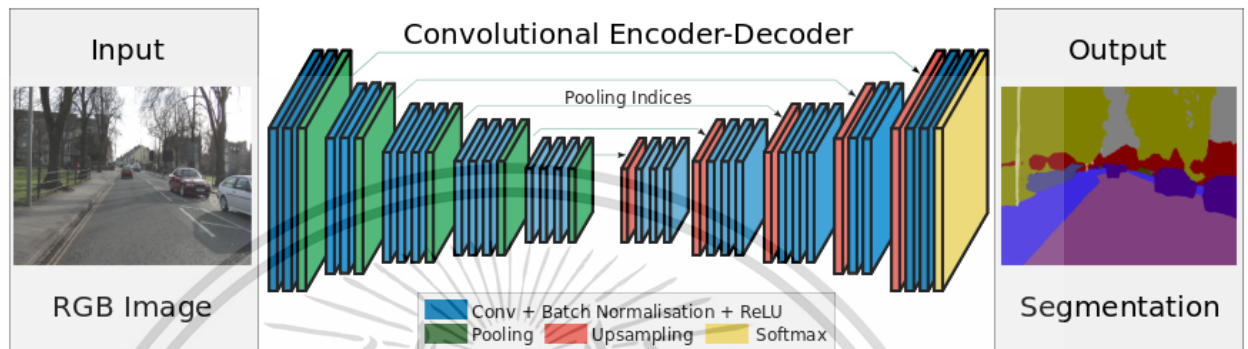


ภาพที่ 2.35 การทำงานของ SqueezeNet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.7.9 SegNet

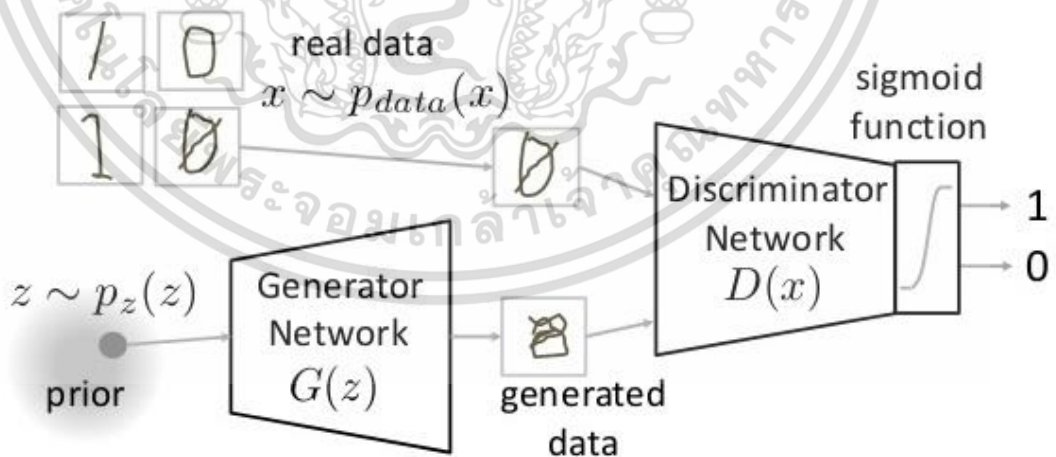
SegNet คือสถาปัตยกรรมของ deep learning เพื่อใช้ในการแก้ปัญหาการ segmentation มีสถาปัตยกรรมแสดงการทำงาน ในภาพที่ 2.36 ภาพมันมาจากลำดับของ processing layer ตามด้วยการเชื่อมกันของเซ็ทของ pixelwise ที่ตรงกันโดยใช้เป็นรหัสเพื่อใช้ในการ classification



ภาพที่ 2.36 สถาปัตยกรรมแบบ SegNet

2.4.7.10 GAN (Generative Adversarial Network)

GAN เป็นเครือข่ายระบบประสาทสายพันธ์ใหม่ มีสถาปัตยกรรมแสดงการทำงาน ในภาพที่ 2.37 ซึ่งระบบเครือข่ายจะทำการสร้างรูปขึ้นมาใหม่แต่ไม่ได้ใช้สำหรับการนำไปเทรนด้าเซ็ท แต่คือสามารถใช้เป็นด้าเซ็ทได้



ภาพที่ 2.37 สถาปัตยกรรมแบบ GAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า Magnetic Resonance Imaging (MRI)

เครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า [13] เป็นเครื่องมือที่ใช้สำหรับตรวจอวัยวะภายในร่างกายดังแสดงได้ในภาพที่ 2.38 โดยอาศัยหลักการของคลื่นแม่เหล็กไฟฟ้าในการสร้างภาพ ซึ่งสามารถให้รายละเอียดภาพที่คมชัด และสามารถตรวจหาความผิดปกติของอวัยวะต่าง ๆ ได้อย่างดี การตรวจด้วยวิธีนี้จำเป็นต้องใช้เวลาประมาณ 30 นาที ถึง 1 ชั่วโมง ขึ้นอยู่กับอวัยวะที่ต้องการตรวจ



ภาพที่ 2.38 เครื่อง MRI

2.5.1 ข้อดีของการตรวจ MRI

1. การตรวจด้วยวิธีนี้สามารถจำแนกคุณสมบัติของเนื้อเยื่อที่แตกต่างกันได้หลายแบบ และตรวจหาสิ่งผิดปกติในระยะแรกได้
2. ตรวจได้ทุกระนาบโดยไม่ต้องเคลื่อนย้ายหรือเปลี่ยนท่าผู้ป่วย
3. สามารถตรวจเนื้อเยื่ออ่อนที่อยู่ภายในกระดูกได้
4. ไม่มีรังสีเอ็กซ์ที่เป็นอันตรายแก่ร่างกาย ทำให้สามารถตรวจในผู้ป่วยที่ตั้งครรภ์ในช่วง 6 เดือน - 9 เดือนได้หากมีข้อบ่งชี้การส่งตรวจที่เหมาะสม โดยพยายามหลีกเลี่ยงการตรวจในระยะ 3 เดือนแรกของการตั้งครรภ์
5. สามารถทำการตรวจได้แม้เป็นโรคไตวายโดยไม่จำเป็นต้องฉีดสารทึบรังสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. โอกาสแพ้สารที่ใช้ในการตรวจ (Gadolinium) น้อยมาก เมื่อเทียบกับสารทึบรังสีที่ใช้ในการตรวจเครื่องเอกซเรย์คอมพิวเตอร์

2.5.2 วัตถุประสงค์ของการตรวจ MRI

1. ตรวจหาความผิดปกติของสมอง ได้แก่ สมองขาดเลือด, เนื้องอก, สาเหตุการชัก, การอักเสบติดเชื้อของเนื้อสมองและเยื่อหุ้มสมอง
2. ตรวจหาความผิดปกติของระบบกระดูกสันหลังและไขสันหลัง ได้แก่ หมอนรองกระดูกเคลื่อน, เนื้องอกไขสันหลัง, การติดเชื้อ, บาดเจ็บไขสันหลัง
3. ตรวจหาความผิดปกติของหลอดเลือดในสมองและลำตัว โดยไม่ต้องฉีดสารทึบรังสี
4. ตรวจหาความผิดปกติของเนื้อเยื่อกล้ามเนื้อ กระดูกส่วนต่างๆ รวมทั้งการฉีกขาดของกล้ามเนื้อ เส้นเอ็นบริเวณข้อกระดูก เช่น ข้อเข่า
5. ตรวจหาความผิดปกติที่เกิดขึ้นกับอวัยวะในอุ้งเชิงกราน เช่น มดลูก ต่อมลูกหมาก และกระเพาะปัสสาวะ
6. ตรวจหาความผิดปกติต่างๆ บริเวณทรวงอก หัวใจ ช่องท้อง ท้องและเต้านมสตรี
7. การตรวจพิเศษอื่นทาง MRI อื่นๆ เช่น MR Perfusion หรือ MR spectroscopy

2.5.3 ข้อควรระวังในการตรวจ MRI

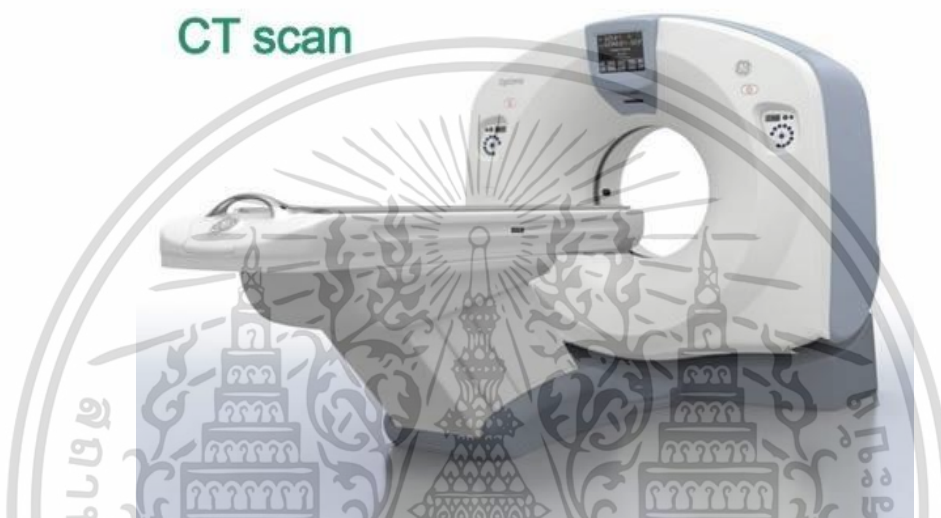
เนื่องจากเครื่องตรวจ MRI มีสนามแม่เหล็กแรงสูงตลอดเวลา ทำให้มีผลต่อการทำงานของอุปกรณ์ที่มีส่วนผสมของโลหะทั้งที่อยู่ร่างกาย หรือที่ติดมากับผู้ป่วย ดังนั้น ผู้ป่วยต้องแจ้งให้เจ้าหน้าที่ผู้เกี่ยวข้องทราบ ทั้งแพทย์ พยาบาล และผู้ดูแล ในกรณีต่อไปนี้

1. การผ่าตัดใส่เครื่องกระตุ้นการเต้นของหัวใจ (Cardiac pacemaker)
2. การผ่าตัดติดคลิปอุดหลอดเลือดในเส้นเลือดโป่งพอง (Aneurysm clips)
3. ผู้ป่วยที่ต้องเปลี่ยนลิ้นหัวใจเทียมชนิดโลหะ
4. การผ่าตัดใส่อวัยวะเทียมภายในหู (Ear implant)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. มีโลหะต่างๆ อยู่ในร่างกาย เช่น ข้อเทียมต่างๆ โลหะตามกระดูก กระสุนปืน เป็นต้น
6. การตั้งครรภ์โดยเฉพาะในระยะ 3 เดือนแรก
7. มีสิ่งแปลกปลอมที่เป็นโลหะติดอยู่ที่ตา
8. กลัวที่แคบ หรือไม่สามารถนอนราบในอุโมงค์ตรวจได้

2.6 การตรวจ CT scan (Computerized Tomography) หรือเอกซเรย์คอมพิวเตอร์



ภาพที่ 2.39 เครื่อง CT scan

เครื่อง CT scan ดังแสดงในภาพที่ 2.39 เป็นการตรวจหาความผิดปกติอวัยวะต่างๆ ในร่างกายด้วยลำแสงเอกซเรย์ โดยฉายลำแสงเอกซเรย์ผ่านอวัยวะที่ต้องการตรวจในแนวตัดขวาง และให้คอมพิวเตอร์สร้างภาพ ภาพที่ได้จึงเป็นภาพตัดขวางส่วนที่ต้องการตรวจอย่างละเอียด โดยมีข้อบ่งชี้ของการตรวจดังนี้

1. ตรวจหาเนื้องอกในอวัยวะต่างๆ รวมทั้งตำแหน่งและขนาดของเนื้องอก
2. ตรวจหาการแพร่กระจายของเนื้องอกไปยังต่อมน้ำเหลืองที่อยู่ใกล้เคียง
3. ตรวจดูการคั่งของเลือดในสมอง ช่องท้อง และอุ้งเชิงกราน
4. ตรวจหาความผิดปกติของหลอดเลือด เช่น เส้นเลือดโป่งพอง เส้นเลือดอุดตัน เป็นต้น
5. ตรวจหาความผิดปกติของกระดูก และข้อต่อต่างๆ เช่น การหัก การหลุด และการอักเสบ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 การตรวจ CT scan มี 4 ระบบ คือ

1. ระบบสมอง ได้แก่ การตรวจเอกซเรย์คอมพิวเตอร์ของสมอง ต่อมใต้สมอง ตา ต่อมน้ำลาย และคอ เป็นต้น ในการตรวจนี้ จะต้องฉีดสารทึบรังสีเข้าทางหลอดเลือดดำ เพื่อช่วยให้เห็นพยาธิสภาพของโรคชัดเจนขึ้น

2. ระบบช่องท้องและทรวงอก ได้แก่ การตรวจเอกซเรย์คอมพิวเตอร์ภายในช่องท้องและอุ้งเชิงกราน การตรวจระบบนี้ ผู้ป่วยต้องดื่มสารทึบรังสี/น้ำเปล่า และ/หรือ สวนสารทึบรังสี/น้ำเปล่า เข้าทางทวารหนัก เพื่อแยกลำไส้ออกจากเนื้อเยื่ออื่นๆ ของช่องท้อง และ ในผู้ป่วยหญิงอาจต้องใส่ผ้าอนามัยชนิดสอดภายในช่องคลอด เพื่อแยกช่องคลอดออกจากเนื้อเยื่ออื่นๆ ทั้งนี้ขึ้นอยู่กับดุลยพินิจของรังสีแพทย์ นอกจากนี้ยังจำเป็นต้องฉีดสารทึบรังสีเข้าทางหลอดเลือดดำเพื่อช่วยให้เห็นพยาธิสภาพของโรคชัดเจนขึ้น

3. ระบบกระดูก กล้ามเนื้อ ข้อต่อและกระดูกสันหลัง ซึ่งมักใช้ในการวินิจฉัยโรคเนื้องอกของกล้ามเนื้อ กระดูก หรือการอักเสบของข้อต่อต่าง ๆ และลักษณะทางกายวิภาคของกระดูกสันหลัง โดยสามารถให้การวินิจฉัยโรคกระดูกได้ดีกว่าการตรวจเอกซเรย์ทั่วไป

4. ระบบหลอดเลือด ได้แก่ การตรวจเอกซเรย์คอมพิวเตอร์ของหลอดเลือดสมอง หลอดเลือดหัวใจ หลอดเลือดแดงใหญ่ หลอดเลือดแดงไต และหลอดเลือดแดงที่ขา เป็นต้นในการตรวจนี้จำเป็นต้องฉีดสารทึบรังสีเข้าทางหลอดเลือดดำ

2.6.2 ความแตกต่างระหว่าง x-ray และ CT scan

ความแตกต่างระหว่างการเอกซเรย์ธรรมดา กับเอกซเรย์คอมพิวเตอร์ คือเอกซเรย์ธรรมดา จะให้ภาพการตรวจเป็นภาพ 2 มิติ คือกว้าง และยาว ไม่สามารถบอกความลึกได้ และจะให้ภาพเป็นภาพรวมของทั้งอวัยวะ ดังนั้นจึงเป็นข้อจำกัดของเอกซเรย์ธรรมดา เมื่อเปรียบเทียบกับเอกซเรย์คอมพิวเตอร์ซึ่งใช้เทคโนโลยีในการตรวจที่ซับซ้อนกว่าเอกซเรย์ธรรมดามาก ซึ่งจะให้ภาพเป็น 3 มิติ และยังสามารถฉายภาพอวัยวะออกเป็นแผ่นบาง ๆ ในภาพตัดขวางได้หลายสิบแผ่น จึงช่วยให้แพทย์อ่านความผิดปกติของอวัยวะนั้น ๆ ได้ละเอียดและแม่นยำกว่า

2.6.3 ประโยชน์ของ CT scan

ด้วยความสามารถในการสร้างภาพที่มีส่วนประกอบของเนื้อเยื่อ กระดูก และระบบหลอดเลือดอยู่รวมกัน จึงทำให้ CT scan มีความสำคัญในการวินิจฉัยโรค และให้รายละเอียดแก่แพทย์ได้เกือบทุกส่วนของร่างกาย ตั้งแต่ศีรษะจรดปลายเท้า ไม่ว่าจะเป็นโรคเกี่ยวกับหัวใจและหลอดเลือด โรค

กระดูก การได้รับอุบัติเหตุ และการแพร่กระจายของโรคมะเร็ง
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ซึ่งการเขียนเพื่อการศึกษาก็เช่นนี้ เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การแปลงภาพ

2.7.1 Histogram Equalization [14]

คือการปรับรูปภาพให้มีการกระจายตัวของความเข้มแสงอย่างสม่ำเสมอ ซึ่งจะทำให้กราฟความถี่สะสม (cumulative histogram) เพิ่มขึ้นเป็นเส้นตรง ซึ่ง

ภาพที่ 2.40 แสดงถึง ตัวอย่างภาพก่อนทำ Histogram Equalization

ภาพที่ 2.41 แสดงถึง ตัวอย่างกราฟก่อนทำ Histogram Equalization

ภาพที่ 2.42 แสดงถึง ตัวอย่างภาพหลังทำ Histogram Equalization

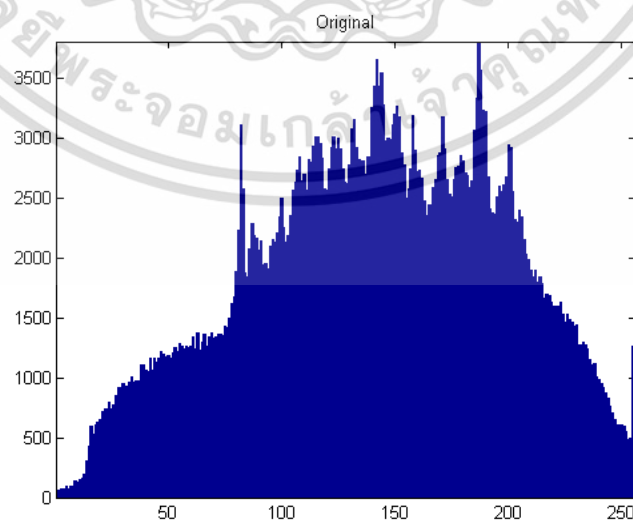
ภาพที่ 2.43 แสดงถึง ตัวอย่างกราฟ *Cumulative Histogram* ก่อนทำ Histogram Equalization

ภาพที่ 2.44 แสดงถึง ตัวอย่างกราฟหลังทำ Histogram Equalization

ภาพที่ 2.45 แสดงถึง ตัวอย่างกราฟ *Cumulative Histogram* หลังทำ Histogram Equalization



ภาพที่ 2.40 ตัวอย่างภาพก่อนทำ Histogram Equalization

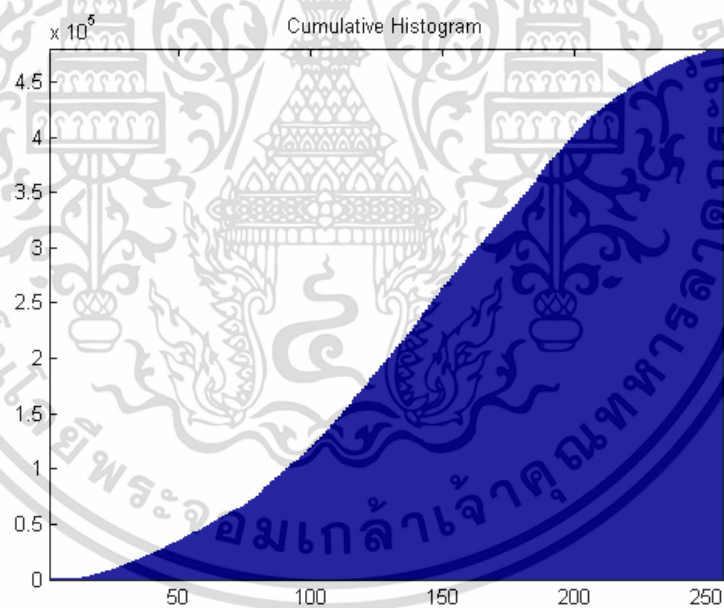


ภาพที่ 2.41 ตัวอย่างกราฟก่อนทำ Histogram Equalization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

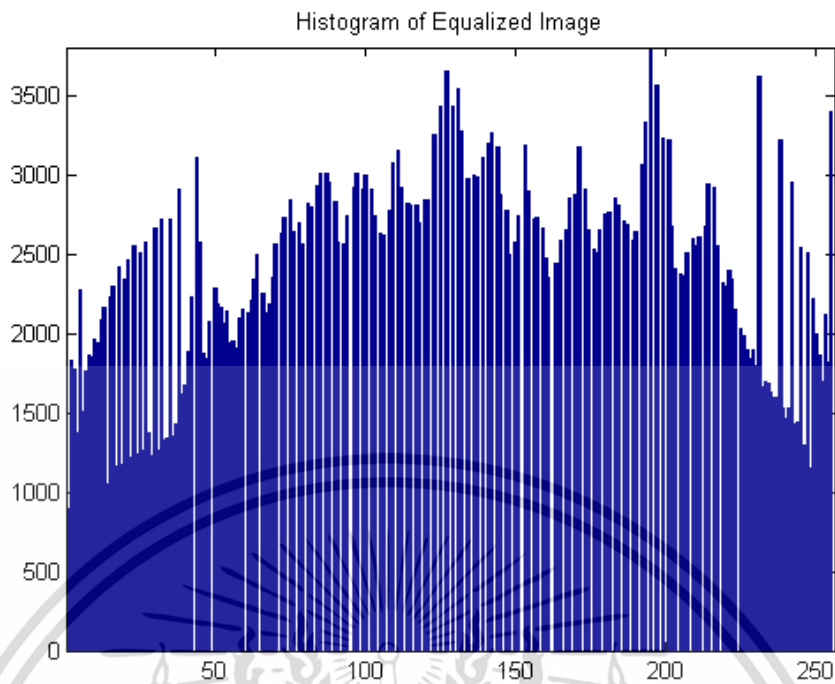


ภาพที่ 2.42 ตัวอย่างภาพหลังทำ Histogram Equalization

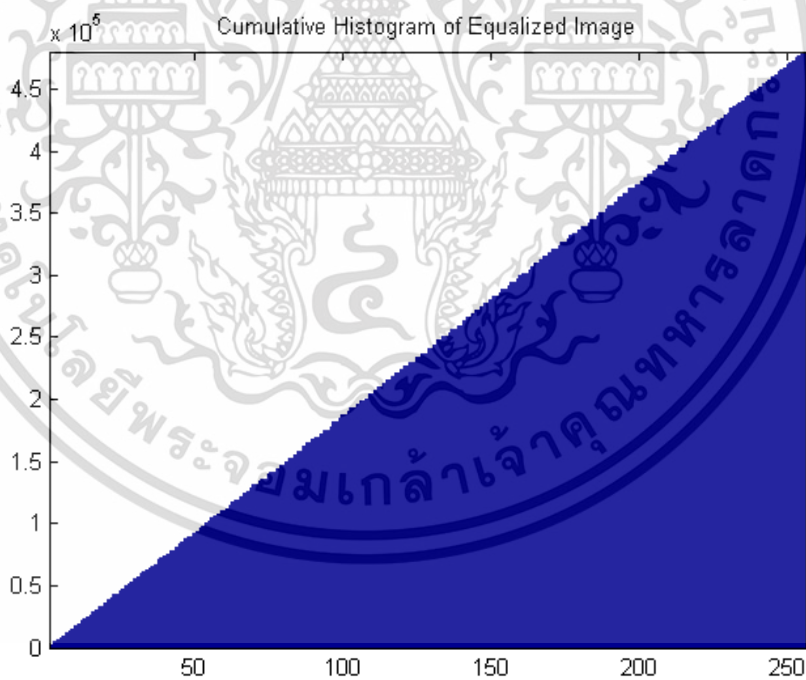


ภาพที่ 2.43 ตัวอย่างภาพหลังทำ Histogram Equalization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.44 ตัวอย่างกราฟ *Cumulative Histogram* ก่อนทำ Histogram Equalization



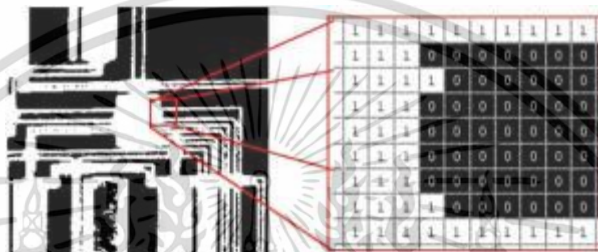
ภาพที่ 2.45 ตัวอย่างกราฟ *Cumulative Histogram* หลังทำ Histogram Equalization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 Type of image [15]

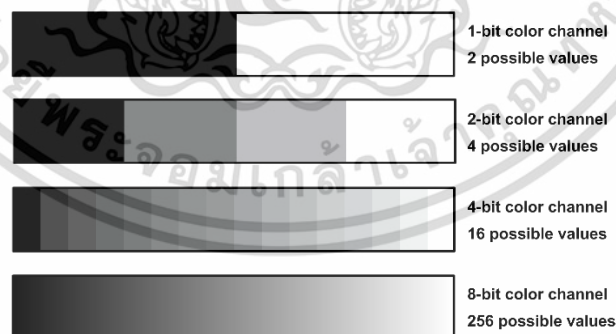
ภาพดิจิทัลที่ได้จะมีรูปแบบการเก็บเป็นเมทริกซ์ ซึ่งจะมีการจัดเก็บภาพแต่ละชนิดต่างกัน ขึ้นอยู่กับระบบสีของภาพดังกล่าว โดยแบ่งชนิดของภาพได้ดังนี้

2.7.2.1 Binary image ภาพไบนารีคือภาพที่ในแต่ละตำแหน่งของภาพมีเพียงแค่ 2 สถานะ คือ 0 กับ 1 กล่าวคือ สีดำเมื่อมีสถานะเป็น 0 และสีขาวเมื่อมีสถานะเป็น 1 โดยการแปลงภาพระดับสีเทาให้กลายเป็นภาพไบนารี จะต้องกำหนดค่าขีดแบ่ง (Threshold) เพื่อกำหนดค่าระดับความเข้มสีเทาที่จะถูกแปลงกลายเป็นสีขาวหรือสีดำดังภาพที่ 2.46



ภาพที่ 2.46 ภาพแบบ Binary หรือ ภาพขาว-ดำ

2.7.2.2 Grayscale Image ภาพระดับสีเทาคือภาพที่สร้างมาจากค่าของความเข้มสีจากแต่ละจุด โดยความเป็นไปได้ของ ภาพระดับสีเทาในภาพทั้งหมดจะขึ้นอยู่กับจำนวนบิตที่ใช้ ดังแสดงในภาพที่ 2.47 ซึ่งมักนิยมใช้ระดับสีเทา 8 บิตที่มี ระดับสีเทาทั้งหมด 256 ระดับ โดยนิยมระบุเป็นตัวเลขทศนิยมระหว่างช่วง 0-1 หรือจำนวนเต็ม 0-255 โดยให้ 0 แทนสีขาวและ 255 แทนสีดำ



ภาพที่ 2.47 ระดับเฉดสีเทา

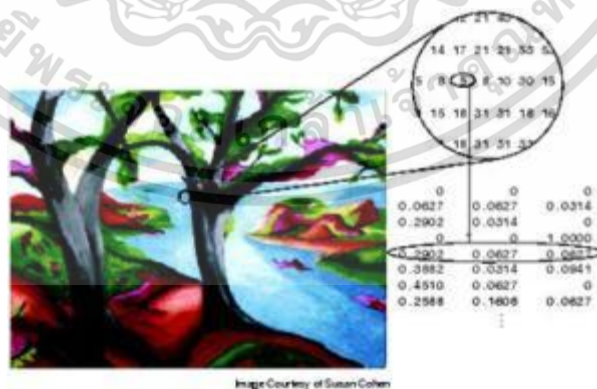
2.7.2.3 RGB Image หรือ Tricolor Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด $m \times n \times 3$ โดยที่ m คือความยาว และ n คือความกว้างของภาพในหน่วยพิกเซล ส่วนมิติสุดท้ายนั้น ในแต่ละมิติจะเก็บค่าสีแยกกัน คือสีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) ดังแสดงในภาพที่ 2.48

0.2235	0.1294	Blue	0.4199	0.2902	0.2902	0.2902	0.2902
0.5804	0.2902	0.0627	0.2902	0.2902	0.2902	0.2235	0.2588
0.5176	0.1922	0.0627	Green	0.1922	0.2588	0.2588	0.2588
0.5176	0.1294	0.1608	0.1294	0.1294	0.2588	0.2588	0.2588
0.5176	0.1608	0.0627	0.1608	0.1922	0.2588	0.2588	0.2588
0.5498	0.2235	0.5498	Red	0.7412	0.7765	0.7765	0.7412
0.5498	0.3882	0.5176	0.5804	0.5804	0.7765	0.7765	1.0000
0.5498	0.2588	0.2902	0.2588	0.2235	0.4824	0.2235	0.2235
0.5498	0.2235	0.1608	0.2588	0.2588	0.1608	0.2588	0.2588
0.5498	0.1608	0.2588	0.2588	0.2588	0.2588	0.2588	0.2588



ภาพที่ 2.48 ภาพแบบ RGB

2.7.2.4 Indexed Image เป็นรูปที่มีรูปแบบการเก็บแบบ indexed คือ ภาพประเภทนี้จะเก็บค่าสีเป็น indexed และในแต่ละช่องอาร์เรย์ จะเก็บตำแหน่งของสีใน indexed นั้นๆ ไว้ ดังแสดงในภาพที่ 2.49



ภาพที่ 2.49 ภาพแบบ Indexed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

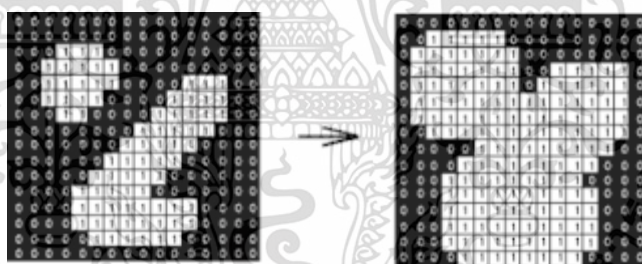
2.7.3 Morphological [16]

การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ (MORPHOLOGICAL PROCESSING)

การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ คือการประมวลผลของภาพทางด้านโครงสร้าง โดยเกี่ยวกับการแยกส่วนประกอบของภาพออกเพื่อใช้ในการแสดงรูปร่างในเมทริกซ์จะประกอบไปด้วยค่าระดับขาว-ดำ 2 ค่า คือ 0 และ 1

2.7.2.1 การขยาย (Dilation)

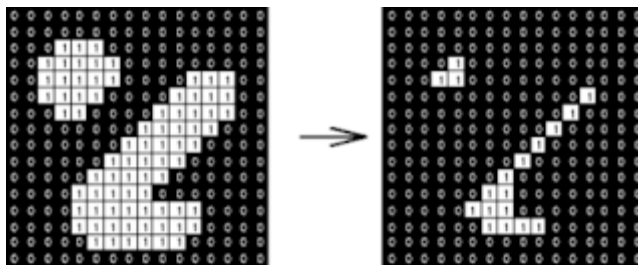
การขยาย (Dilation) จะพิจารณาข้อมูลภาพซึ่งเป็นภาพขาว-ดำ เป็นการขยายภาพให้ใหญ่ขึ้น เพื่อเพิ่มสีให้กับวัตถุที่แสดงผลในขั้นตอนสุดท้ายซึ่งการขยายวัตถุจะทำได้โดยการกำหนดส่วนประกอบ โครงสร้าง (Structuring element) และนำส่วนประกอบโครงสร้างไปกราดบนข้อมูลภาพตามลำดับตลอดทั้งภาพโดยเมื่อจุดเริ่มต้นของส่วนประกอบโครงสร้างหรือจุดกำเนิดตรงกับตำแหน่งข้อมูลภาพที่เท่ากับ 1 จะทำการยูเนียนส่วนประกอบโครงสร้าง เข้ากับข้อมูลภาพ ดังภาพที่ 2.50



ภาพที่ 2.50 การขยาย

2.7.2.2 การกร่อนขนาด (Erosion)

การกร่อนขนาด (Erosion) เป็นการกร่อนขนาดบริเวณขอบของวัตถุ ซึ่งการกร่อนมีวิธีคล้ายกับการขยายคือ สร้างส่วนประกอบโครงสร้างขึ้นมาแล้วนำไปกราดตามข้อมูลภาพ โดยจะเลื่อนไปทุกตำแหน่งเปรียบเทียบกับข้อมูลภาพ ถ้าข้อมูลมีค่าเหมือนกับส่วนประกอบโครงสร้างจะทำการกำหนดค่าข้อมูลภาพที่ตรงกับตำแหน่งที่ตรงกับจุดเริ่มต้นหรือ จุดกำเนิดของส่วนประกอบโครงสร้างให้เท่ากับ 1 ดังแสดงในภาพที่ 2.51



ภาพที่ 2.51 การกร่อนขนาด

2.7.4 REGION-OF-INTEREST (ROI)

Region-of-interest (ROI) คือบริเวณที่เราสนใจ อาจจะเป็นบริเวณใดภายในภาพก็ได้ โดยการตีกรอบล้อมรอบบริเวณที่สนใจ ด้วยวงกลม กรอบสี่เหลี่ยม หรือกรอบรูปเหลี่ยมใดๆ เพื่อนำภาพเฉพาะส่วนดังกล่าวมาประมวลผล หรือเปลี่ยนแปลงภาพตามต้องการโดยไม่มีผลกระทบต่อส่วนอื่นๆ ซึ่งใน 1 ภาพ สามารถกำหนดได้หลายๆ ROI เมื่อกำหนดตำแหน่งต่างๆ แล้วจะสร้าง Mask ที่เป็น Binary Mask สำหรับใช้กำหนดขอบเขตที่จะมีการเปลี่ยนแปลงภายใน รูปภาพนั้นๆ โดยให้ค่าส่วนที่สนใจเป็น 1 หรือสีขาว และให้ส่วนอื่นๆ เป็น 0 หรือสีดำ

2.7.5 Softmax Function

Softmax Function [17] หรือ SoftArgMax Function หรือ Normalized Exponential Function คือ ฟังก์ชันที่รับ Input เป็น Vector ของ Logit จำนวนจริง แล้ว Normalize ออกมาเป็นความน่าจะเป็น Probability ที่ผลรวมเท่ากับ 1 ซึ่งจะมีสมการการดำเนินการตามสมการที่(2)

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i=1, \dots, K \text{ and } z = (z_1, \dots, z_K) \in \mathbb{R}^k \quad (2)$$

ประโยชน์ของ Softmax Function

Softmax มักถูกนำไปไว้ Layer สุดท้าย ของ Neural Network เพื่อให้ Output ออกมาเป็น Probability ไปคำนวณ Negative Log Likelihood เป็น Cross Entropy Loss เช่น ในงาน Single Class Classification

Softmax ถูกนำไปใช้บ่อย ในงาน Classification จนถึงขนาดมีคนเรียกว่า Softmax Classifier หรือ Softmax Loss

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสียของ Softmax Function

เหมาะกับการใช้งานที่คาดหวัง Output ที่ถูกต้องอันเดียวเท่านั้น (ใกล้เคียง Max Function)

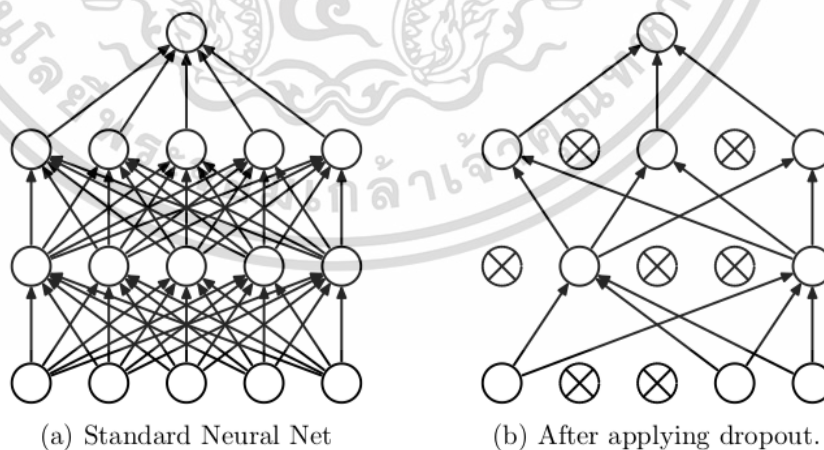
เนื่องจาก ตัวหารต้อง Sum รวมทั้งหมดทุก Item ทุกครั้ง จึงทำให้มีปัญหาเรื่อง Performance ถ้ามีจำนวน Item มาก ๆ เช่น Output เป็น 1 ใน 500,000 คำใน Dictionary ในงาน NLP

2.7.6 Dropout

โมเดล Deep Neural Network มักจะมีปัญหา Overfit เมื่อเทรนกับ Dataset ที่มีขนาดเล็ก แต่สามารถแก้ปัญหา Overfit โดยใช้วิธีการ Ensembles สร้างหลาย ๆ โมเดลแล้วเอา Output [18] มาเฉลี่ยกัน แต่ทำแบบนี้ค่อนข้างสิ้นเปลืองทรัพยากร เวลา และต้องคอย Maintain หลายโมเดล dropout function จึงเกิดขึ้นมาเพื่อใช้ในการแก้ปัญหาดังกล่าว

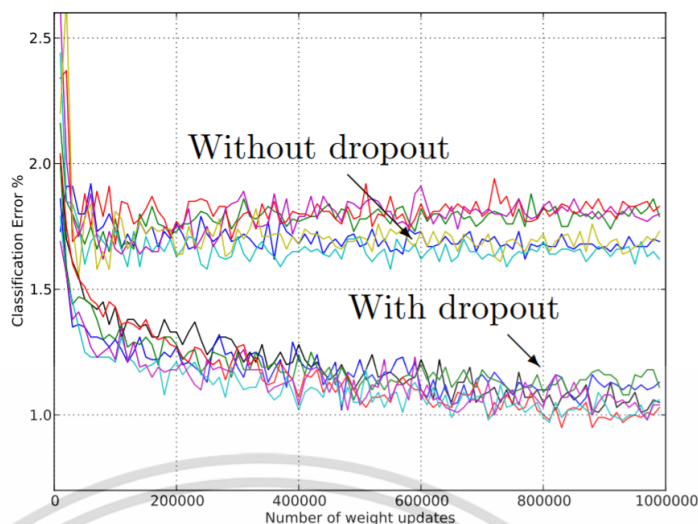
Dropout สามารถใช้เพียงโมเดลเดียว มาจำลองเป็นหลาย ๆ โมเดลได้ โดยการสุ่มถอดบาง Node ออก ในระหว่างการเทรน วิธีนี้เรียกว่า Dropout เป็นวิธีที่ฉลาด ประหยัดทั้งเวลา และทรัพยากร และที่สำคัญยังไม่ต้อง Maintain หลาย ๆ โมเดล

Dropout ถือเป็นวิธี Regularization แบบหนึ่ง ช่วยลดการจำข้อสอบ ลด Overfit และทำให้โมเดล Deep Neural Network ทุก ๆ สถาปัตยกรรม Generalization ดีขึ้นดังภาพที่ 2.52 และมีค่า error แสดงในภาพที่ 2.53



ภาพที่ 2.52 (a) neural network ที่ไม่ใช้ dropout (b) neural network ที่ใช้ dropout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.53 กราฟเปรียบเทียบ error ระหว่าง neural network ที่ไม่มีการใช้ dropout และ มีการใช้ dropout

2.8 ค่าสัมประสิทธิ์ความคล้ายโซเรนเซน-ไดซ์ (Sørensen-Dice coefficient similarity) [19]

(Dice, 1945; Sørensen, 1948) สถิติที่ใช้ในการเปรียบเทียบความคล้ายของสองตัวอย่าง แนวคิดพื้นฐาน ของค่าสัมประสิทธิ์ความคล้ายโซเรนเซน-ไดซ์นั้นคล้ายกับค่าสัมประสิทธิ์ความคล้ายแจ็กการ์ด (Jaccard coefficient similarity) ซึ่งถูกใช้อย่างแพร่หลายในงานค้นคืนสารสนเทศ กล่าวคือ สามารถคำนวณได้จากจำนวนสมาชิกที่ทั้งสองตัวอย่างมีร่วมกันต่อจำนวนสมาชิกทั้งหมด แต่หลักการคำนวณ ค่าสัมประสิทธิ์ความคล้ายโซเรนเซน-ไดซ์จะลดผลกระทบจากการมีสมาชิก ร่วมกันของกลุ่มตัวอย่าง ดังนั้นจึงสามารถพิจารณาค่าที่คำนวณได้ในฐานะค่าความคล้ายของข้อมูล ทั้งชุด ในงานขั้นนี้ได้ค่าสัมประสิทธิ์ความคล้ายโซเรนเซน-ไดซ์ในการวัดความคล้ายระหว่าง ข้อความในระดับคำ ทั้งนี้ กำหนดให้เอ็นแกรมของคำมีความยาว n , $S(A, n)$ แทนชุดของเอ็นแกรม ของ คำในข้อความ A , $S(B, n)$ แทนชุดของเอ็นแกรมของคำในข้อความ B , QSn ซึ่งแทนผลหารของ ความคล้าย (quotient of similarity) ระหว่างข้อความ A กับข้อความ B จะหาได้จากสมการที่ (3) ค่า QSn เป็นตัวเลขระหว่าง 0 ถึง 1 เช่นเดียวกันกับค่า $lcsnorm$ หากค่า QSn เท่ากับ 0 แปลว่า ข้อความ A และข้อความ B ไม่ปรากฏชุดของเอ็นแกรมร่วมกันเลย แต่หากค่า QSn เท่ากับ 1 แปลว่า ข้อความ A และข้อความ B เหมือนกันทุกประการ

$$QSn = \frac{2|S(A,n) \cap S(B,n)|}{|S(A,n)| + |S(B,n)|} \quad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 Entropy และ Uncertainty ของระบบใน Deep Learning AI [20]

2.9.1 ความหมายของ Entropy

Entropy [24] หรือ เอนโทรปี เป็นการวัดความไม่แน่นอน (uncertainty) ของระบบหรือความน่าจะเป็นที่จะเกิดสิ่งต่าง ๆ โดยมีสมการการคำนวณดังสมการที่ (4)

$$H(p) = - \sum_{i=1}^N p_i \log (p_i) \quad (4)$$

เมื่อ p คือความน่าจะเป็นที่จะเกิดเหตุการณ์ต่าง ๆ จากทั้งหมด N เหตุการณ์

2.9.2 การนำ Entropy ไปใช้กับ Deep learning

ตัวอย่างหนึ่งของการนำเอา Entropy ไปใช้งานก็คือ การที่เราไม่ไว้วางใจผลการคำนวณของโมเดลที่มี Entropy สูง แล้วส่งให้มนุษย์เป็นผู้ตัดสินใจแทน กระบวนการนี้เป็นหัวใจสำคัญของ AI Security เราควรรู้ว่าเมื่อไหร่ที่เราสามารถไว้วางใจ หรือเชื่อมั่นใจในการตัดสินใจของโมเดล และเมื่อไหร่ที่เราควรจะเข้าไปแทรกแซงการตัดสินใจนั้น แต่ Entropy เป็นเพียงวิธีการหนึ่งที่ใช้วัดความไม่แน่นอน แต่ในความเป็นจริงแล้วยังมีวิธีอีกมากมายที่ถูกเสนอในงานวิจัยต่าง ๆ

2.9.3 Cross Entropy

Cross Entropy เป็นอีกหนึ่งเครื่องมือสำคัญที่ใช้ใน Deep Learning โดยที่ Cross Entropy ไม่ได้ใช้วัดความวุ่นวาย แต่ใช้วัดความแตกต่างกันของการแจกแจงความน่าจะเป็น ภาพแต่ละภาพที่นำมาใช้สอนโมเดลจะมีค่าความน่าจะเป็นของการจัดอยู่ในสิ่งใดสิ่งหนึ่งอย่างชัดเจน 100% เช่น รูปภาพหมา ก็มีความเป็นหมา 100% จะไม่มีภาพที่มีความเป็นหมา 80% และแมว 20% อยู่ในชุดข้อมูลที่ใช้สอน (Training Set) เป้าหมายของการเรียนรู้โดยทั่วไปก็คือ ทำให้โมเดล Deep Learning ของเราสามารถที่จะให้คำตอบด้วยความน่าจะเป็นที่เหมือนกับความเป็นจริง (Ground Truth) มากที่สุด ซึ่งวิธีการที่ใช้ก็คือ เราจะค่อย ๆ วัดความต่างของการแจกแจงความน่าจะเป็นของข้อมูลจริง กับการแจกแจงความน่าจะเป็นของโมเดลเรา แล้วพยายามปรับค่าให้มีความต่างก็น้อยที่สุด โดยมีสมการการคำนวณดังสมการที่ (5)

$$H(p,q) = - \sum p \log q \quad (5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 ชนิดของภาพ

2.10.1 DICOM file image

Digital Imaging and Communications in Medicine (DICOM) [21] เป็นมาตรฐานที่กำหนด โดย National Electrical Manufacturers Association (NEMA) โดยมีจุดประสงค์ เพื่อการเผยแพร่ ภาพทางการแพทย์ เช่น CT scans, MRIs, CR, หรือ ultrasound และ ภาพทางการแพทย์ อื่น ๆ ในบทที่ 10 ของมาตรฐาน ได้อธิบายถึง รูปแบบของ File ที่ใช้ในการกำหนด ให้เป็นมาตรฐาน เดียวกัน เพื่อ ภาพจะสามารถ แลกเปลี่ยนกันได้ระหว่าง เครื่องมือทางการแพทย์ หรือ Software ทาง การแพทย์ ให้สามารถอ่านข้อมูลของคนไข้ ที่มาจากเครื่องมือต่างชนิด หรือต่างบริษัทได้

DICOM File จะประกอบด้วยข้อมูลในส่วน หัวของ File หรือที่เรียกว่า Header (โดย Header จะประกอบไปด้วยข้อมูลเกี่ยวกับคนไข้ ชื่อ นามสกุล รูปแบบ ของภาพ จำนวนภาพ ลักษณะของภาพ และอื่น ๆ)

DICOM เป็นมาตรฐานที่อุปกรณ์ทางการแพทย์ที่เป็น Digital ในรุ่นใหม่ ๆ ต้องสนับสนุน มาตรฐานนี้ หาก เครื่องมือใดที่ไม่มีความจำเป็นที่ต้องเกี่ยวข้องกับระบบ PACS ก็ไม่จำเป็นที่จะต้อง เป็น DICOM ก็ได้ แต่หากจะให้ดีเพื่อการพัฒนาในอนาคต ควรเลือกเครื่อง มือทางการแพทย์ให้ เป็น DICOM

2.10.1.1 ข้อสังเกต DICOM file

- DICOM files are not recognized by Windows® as image files ดังนั้น เราไม่สามารถเปิดไฟล์โดยใช้ software หรือ application ทั่วไปได้

- ในการจะเปิดดูไฟล์ จำเป็นต้องใช้ additional software package called “DICOM browser” เช่น 3D-Doctor Viewer v 4.0.070803, DCM View etc.

- รูปภาพที่แสดงจะประกอบไปด้วยจุดจำนวนมาก เรียกแต่ละจุดนี้ว่า “pixels” ซึ่งจำนวน pixels นี้ก็ขึ้นอยู่กับ ชนิดของ Digital medical image ที่เราใช้ เช่น ถ้าเป็นของ CT scan จะเป็น 512 x 512 pixels ขณะที่ของ MRI จะเป็น 256 x 256 pixels โดยพื้นที่ในการจัดเก็บ และ Resolution ของ Digital image ต่างๆ เป็นไปตามตารางที่ 2.2

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

ตารางที่ 2.2 แสดงพื้นที่ในการจัดเก็บ และ Resolution ของ Digital image

2.10.2 PNG image

png ย่อมาจาก Portable Network Graphics [27] เป็นรูปแบบรูปภาพที่พัฒนาขึ้นมาจากรูปแบบรูปภาพ GIF เพื่อแก้ปัญหาด้านสิทธิบัตรของภาพแบบ GIF โดยที่ ไฟล์ภาพแบบ PNG ไสเบรารีสำหรับ PNG คือ libpng ซึ่งเขียนด้วยภาษาซี ปัจจุบัน PNG สนับสนุนโดยเว็บเบราว์เซอร์เกือบทุกตัว

MIME type ของไฟล์รูปแบบ PNG คือ image/png ส่วนนี้เวลาเราเขียน php สามารถอ่านค่า MIME type ด้วย PHP ของเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

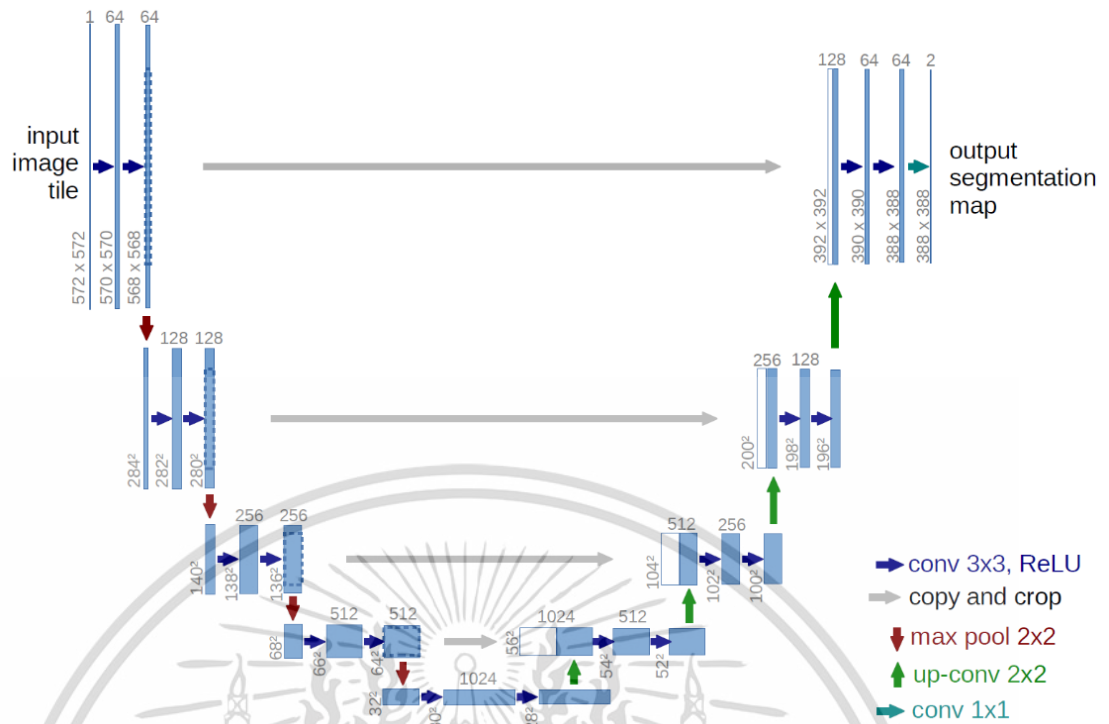
ระเบียบวิธีวิจัย

3.1 การพัฒนาโปรแกรมเพื่อใช้ในการแยกอวัยวะโดยใช้ปัญญาประดิษฐ์ชนิดการเรียนรู้เชิงลึก

การพัฒนาโปรแกรมเพื่อใช้ในการแยกอวัยวะโดยใช้ปัญญาประดิษฐ์ชนิดการเรียนรู้เชิงลึกสามารถทำการแยกภาพที่เป็นอวัยวะของผู้ป่วยโดยการนำภาพ เอ็กซ์เรย์มาจากนั้นนำปัญญาประดิษฐ์ที่พัฒนาขึ้นมาไปทำการตรวจสอบภาพนั้นและทำการทำนายผลของอวัยวะที่ต้องการออกมา ดังนั้นในวิทยานิพนธ์นี้จึงนำเสนอขั้นตอนการพัฒนาโปรแกรมเพื่อใช้ในการแยกอวัยวะโดยใช้ปัญญาประดิษฐ์ชนิดการเรียนรู้เชิงลึก

3.2 การออกแบบการพัฒนาปัญญาประดิษฐ์เพื่อใช้ในการแยกอวัยวะ

การออกแบบการพัฒนาปัญญาประดิษฐ์เพื่อใช้ในการแยกอวัยวะมีพื้นฐานมาจากผู้เข้าร่วมงานประกวดการแข่งขันด้านวิทยาศาสตร์คอมพิวเตอร์ที่มีชื่อว่า “Data Science Bowl 2018” เนื่องจากงานประกวดการแข่งขันนี้เป็นงานประกวดระดับสากลซึ่งมีเงินมากกว่า 1 ล้านดอลลาร์จึงทำให้ข้อมูลที่ให้โดยเจ้าของงานประกวด รวมถึงผู้เข้าประกวดมีความน่าเชื่อถือมากที่สุด ซึ่งความแตกต่างของระบบที่อ้างอิงและระบบที่ใช้คือ ชนิดของข้อมูลภาพอินพุตที่นำเข้าสู่ระบบ การออกแบบระบบสถาปัตยกรรมรวมถึงขั้นตอนการออกแบบปัญญาประดิษฐ์ โดยข้อมูลที่ผู้ทำงานวิจัยนี้ใช้เป็นข้อมูลที่ได้จากโรงพยาบาลศิริราชซึ่งเป็นภาพเอ็กซ์เรย์ตัดขวางช่องท้องส่วนล่างที่มีรูปของอวัยวะภายในของผู้ป่วยโดยผ่านการแก้ไขข้อมูลโดยแพทย์และผู้เชี่ยวชาญ เพื่อป้องกันการละเมิดสิทธิของผู้ป่วยรวมถึงได้รับการยอมรับเพื่อใช้ในงานวิจัยแล้ว โดยในข้อมูลภาพของผู้ป่วยแต่ละรายมีจำนวน 110-127 ภาพ มีผู้ป่วยจำนวน 10 ราย โดยผู้ทำการวิจัยนำข้อมูลเหล่านี้มาเข้าสู่โครงสร้าง U-Net ดังภาพที่ 3.1



ภาพที่ 3.1 รูปแบบโครงสร้างของ U-Net

3.2.1 การแปลงภาพจากภาพไคคอมเป็นภาพ png

จากภาพที่ 3.1 การที่จะนำรูปสู่ระบบของ U-Net ได้ต้องการนำข้อมูลภาพที่ได้ไปผ่านการประมวลผลภาพดิจิทัล (Image processing) โดยนำภาพที่ได้ไปเข้าสู่กระบวนการประมวลผลภาพก่อน (Preprocessing) เนื่องจากภาพที่ได้เป็นภาพไคคอม [21] นำมาแปลงเป็นภาพ png [22] ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import os

import pydicom as pd

import cv2

cwd = './RAW_DATA'

num_dicom = 0

num_dir = 0

dicom_files = []

for dirpath, dirnames, filenames in os.walk(cwd):

    for dirname in dirnames:

        num_dir += 1

    for file in filenames:

        if file.startswith('CT'):

            file_path = os.path.join(dirpath,file)

            dc = pd.read_file(file_path)

            x = dc.pixel_array

            index = dc.InstanceNumber

            number = dc.SeriesNumber

            x = cv2.normalize(x,None, 0, 255,cv2.NORM_MINMAX, cv2.CV_8U)

            write_path = dirpath.replace('RAW_DATA','RAW_DATA_images2')

            if not os.path.exists(write_path):

                os.makedirs(write_path)

            filename = os.path.join(write_path,'%s_im%.6i.png'%(number,index))

            cv2.imwrite(filename,x)

            num_dicom += 1 print(num_dir) print(num_dicom)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการศึกษา
 ภาพที่ 3.2 ชุดคำสั่งที่ใช้ในการแปลงภาพไดค้อมเป็นพิง
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การแยกส่วนอวัยวะ[16]

เมื่อได้ภาพของข้อมูลดังกล่าวแล้วนำมาเป็นส่วนประกอบหนึ่งของการสอนปัญญาประดิษฐ์ หากแต่ยังไม่เพียงพอเพราะยังขาดข้อมูลที่เป็น ภาพแยกของอวัยวะนั้น ๆ เพื่อใช้ในการสอน [26]ปัญญาประดิษฐ์โดยขั้นตอนการทำภาพแยกของอวัยวะจะใช้โปรแกรมแมทแลบโดยใช้หลักการของการปรับปรุงภาพดังนี้ รูปแบบบรรทัดฐาน รีเจียนออฟอินเทอร์เรส และ ความหนาแน่นของพิกเซลของภาพเมื่อผ่านกระบวนการประมวลผลภาพดิจิทัล ดังรูปที่ 3.3

```

close all
clear
clc
img = imread(' filename.png');
c = histeq(img);
d = imgaussfilt(img,1);
e = imbilatfilt(d,100);

imh = imadjust(e,[0.5,0.6],[0.2,0.9]);
imh = adapthisteq(e);
b = imbilatfilt(imh);
a = imcomplement(b);
img2= imcomplement(a);
for level = 0.3:0.01:0.8
    bw = im2bw(img2,level);
    label = bwlabel(bw);
    stats = regionprops(label,'solidity','Area');
    denisty = [stats.Solidity];
    area = [stats.Area];

    high_dense_area = denisty > 0.85 ; % 0.45colon//0.4pelvic girdle//0.85urinary//0.65 kidney
    max_area=max(area(high_dense_area));
    tumor_label = find(area==max_area);
    tumor = ismember(label,tumor_label);
    se = strel('square',5);
    tumor = imdilate(tumor,se);%%imdilate,imerode,imopen,imclose
    fig = figure;
    %subplot(1,3,1),imshow(img,[],title('Urinary'));
    subplot(1,1,1),imshow(tumor,[]);%title('UrinaryAlone');
    B = bwboundaries(tumor,'noholes');
    %subplot(1,3,3),imshow(img2);

    hold on
    for i=1:length(B)
        a = plot(B{i}(:,2),B{i}(:,1)); 'y','linewidth',2.5;
    end

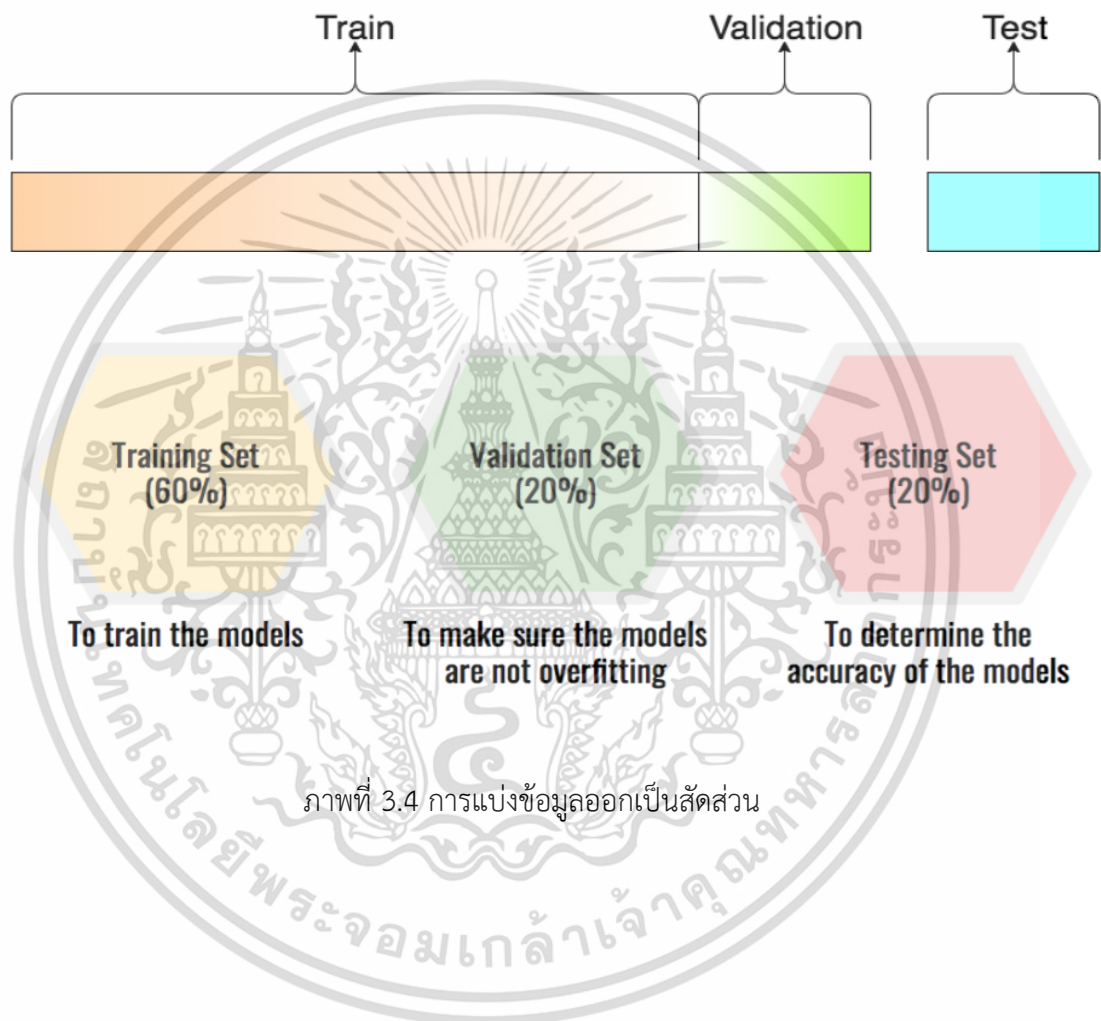
```

ภาพที่ 3.3 ชุดคำสั่งสร้างภาพแยกอวัยวะในโปรแกรมแมทแลบ [26]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การจัดการแบ่งข้อมูลภาพเพื่อใช้ในการเทรน

เมื่อได้ข้อมูลรูปภาพที่ต้องการออกมาแล้วทำการแบ่งข้อมูลทั้งหมดของภาพที่ผ่านการประมวลผลภาพดิจิทัลเป็น 4 ส่วน ดังนี้ 1. ภาพที่ใช้ในการสอน 2. ภาพที่ใช้ในการสอนชนิดผ่านโปรแกรมแมทแลป 3. ภาพที่ใช้ในการทดสอบ 4. ภาพที่ใช้ในการทดสอบชนิดผ่านโปรแกรมแมทแลป โดยสัดส่วนของภาพที่ใช้ในการสอนคิดเป็น 80% ของข้อมูลภาพทั้งหมดดังภาพที่ 3.4



ภาพที่ 3.4 การแบ่งข้อมูลออกเป็นสัดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การออกแบบวิธีการทดลองและการนำเข้ามาดู

เมื่อแบ่งสัดส่วนของข้อมูลที่ได้ออกมาทั้งหมดแล้ว ขั้นตอนต่อไปคือการออกแบบวิธีการทดลองเพื่อทำการสร้างปัญหาประดิษฐ์ โดยจะทำการนำเข้ามาข้อมูลจำเป็นที่จะต้องแปลงข้อมูลภาพให้เป็นเมทริกซ์แถวลำดับ 2 มิติที่พร้อมใช้งานในรูปแบบจำลองโดยทำการเข้ารหัสแต่ละพิกเซลเข้ากับคลาสของภาพ mask ซึ่งผลลัพธ์ที่ได้ออกมาจะเป็นชุดข้อมูลแถวลำดับ 2 มิติ โดยเป็นข้อมูลของภาพนำเข้าและภาพ mask การจัดการ ชุดข้อมูลสอนจะกระทำโดยฟังก์ชัน train Generator ซึ่งจะทำการ import module keras [28] ในการทำกระบวนการ Data augmentation โดยที่จะรับข้อมูลตัวแปร ในรูปแบบ dictionary ดังนี้

```
import os

import sys

import random

import warnings

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt
```

```

from tqdm import tqdm

from itertools import chain

from skimage.io import imread, imshow, imread_collection, concatenate_images

from skimage.transform import resize

from skimage.morphology import label

import cv2 as cv

from keras.models import Model, load_model
from keras.layers import Input
from keras.layers.core import Dropout, Lambda
from keras.preprocessing import image
from keras.layers.convolutional import Conv2D, Conv2DTranspose
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras import backend as K
import tensorflow as tf

from matplotlib import pyplot as plt

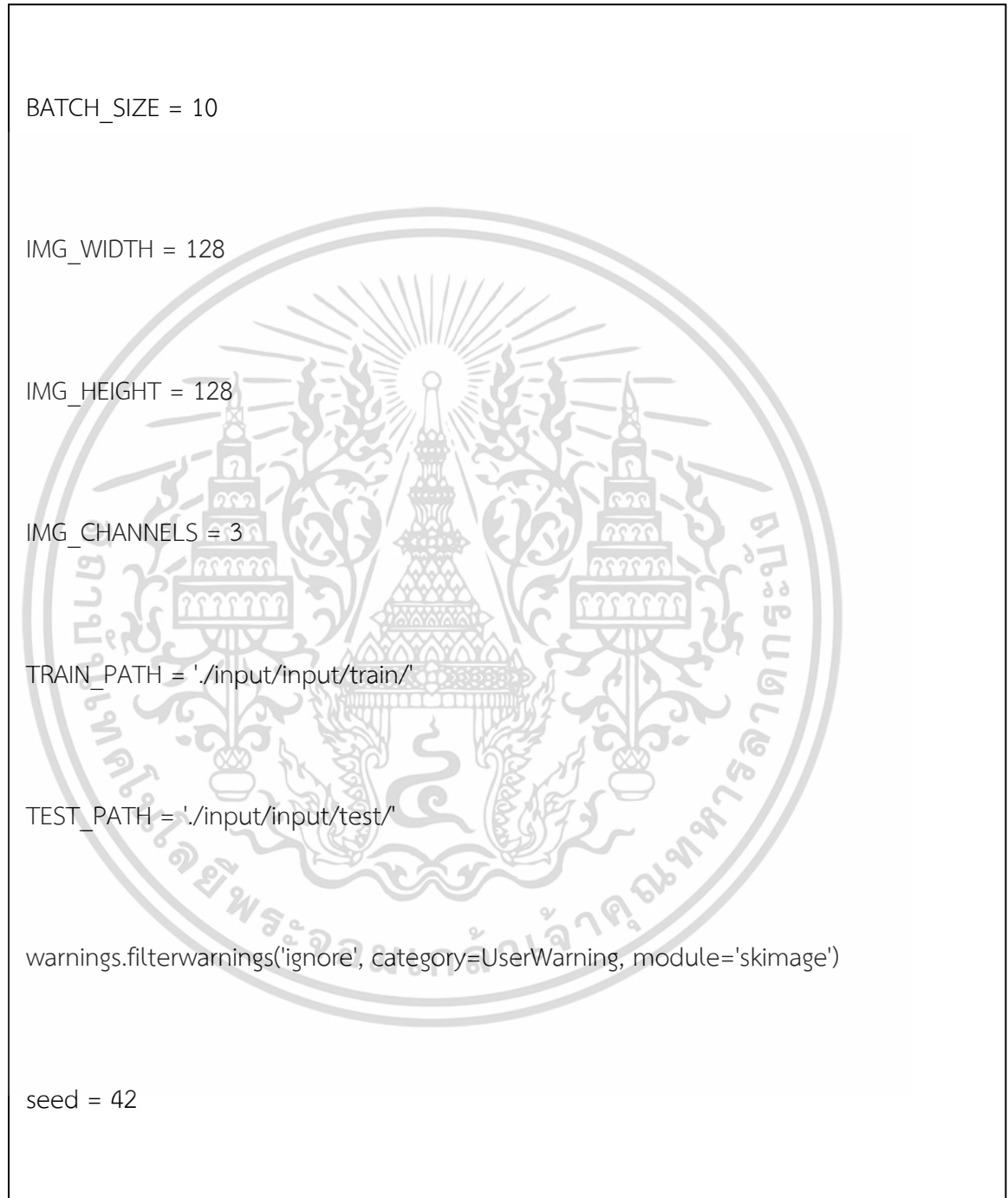
```

ภาพที่ 3.5 การตั้งโมดูลที่จำเป็นต้องใช้ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การตั้งค่าข้อมูลภาพเพื่อให้เหมาะสมกับอุปกรณ์ที่ใช้ในการสอนปัญญาประดิษฐ์

ทำการตั้งค่าข้อมูลภาพเพื่อให้เหมาะสมกับอุปกรณ์ที่ใช้ในการสอนปัญญาประดิษฐ์



ภาพที่ 3.6 กำหนดค่าเพื่อเข้าสู่กระบวนการก่อนประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 การรับค่าข้อมูลและนำไปใช้ในรูปของเมทริกซ์

ทำการรับค่าข้อมูลของภาพโดยแปลงเป็นรูปแบบของเมทริกซ์และนำไปใช้ในรูปของตัวแปร

```
# Get train and test IDs

train_ids = next(os.walk(TRAIN_PATH))[2]

test_ids = next(os.walk(TEST_PATH))[2]

np.random.seed(10)

train_ids[1]
test_ids[1]

for i, value_train in enumerate(train_ids) :
    train_ids[i] = train_ids[i][:4]

for i, value_test in enumerate(test_ids) :
    test_ids[i] = test_ids[i][:4]
```

ภาพที่ 3.7 การรับค่าของภาพในรูปแบบของเมทริกซ์และเก็บไว้ในตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 ชุดคำสั่งทำการย่อขนาดภาพการสอนให้ตรงตามตัวแปรที่ตั้งค่าไว้และรับข้อมูลภาพเข้า

```

X_train = np.zeros((len(train_ids), IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS),
dtype=np.uint8)
Y_train = np.zeros((len(train_ids), IMG_HEIGHT, IMG_WIDTH, 1), dtype=np.bool)
print('Getting and resizing train images and masks ... ')
sys.stdout.flush()
for n, id_ in tqdm(enumerate(train_ids), total=len(train_ids)):
    path = TRAIN_PATH + id_
    img = cv.imread(path + '.png')
    img = resize(img, (IMG_HEIGHT, IMG_WIDTH), mode='constant',
preserve_range=True)
    X_train[n] = img
    mask = np.zeros((IMG_HEIGHT, IMG_WIDTH, 1), dtype=np.bool)
    for mask_file in next(os.walk('input/' + 'train/' + 'label/'))[2]:
        mask_ = imread('input/' + 'train/' + 'label/' + mask_file)
        mask_ = np.expand_dims(resize(mask_,
(IMG_HEIGHT, IMG_WIDTH), mode='constant', preserve_range=True), axis=-1)
        mask = np.maximum(mask, mask_)
    Y_train[n] = mask
# Get and resize test images
X_test = np.zeros((len(test_ids), IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS),
dtype=np.uint8)
sizes_test = []
print('Getting and resizing test images ... ')
sys.stdout.flush()
for n, id_ in tqdm(enumerate(test_ids), total=len(test_ids)):
    path = TEST_PATH + id_
    img = cv.imread(path + '.png')
    sizes_test.append([img.shape[0], img.shape[1]])
    img = resize(img, (IMG_HEIGHT, IMG_WIDTH), mode='constant',
preserve_range=True)
    X_test[n] = img
print('Done!')

```

ภาพที่ 3.8 ชุดคำสั่งทำการย่อขนาดภาพการสอนให้ตรงตามตัวแปรที่ตั้งค่าไว้และรับข้อมูลภาพเข้ามา
เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับโรงเรียนเพื่อการศึกษาเท่านั้น มิใช่อยู่ใต้เห็นแบบลิขสิทธิ์ในราคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8 สร้าง training Image and Mask generator

```

image_datagen = image.ImageDataGenerator(shear_range=0.5, rotation_range=50,
zoom_range=0.2, width_shift_range=0.2, height_shift_range=0.2, fill_mode='reflect')

mask_datagen = image.ImageDataGenerator(shear_range=0.5, rotation_range=50,
zoom_range=0.2, width_shift_range=0.2, height_shift_range=0.2, fill_mode='reflect')

# Keep the same seed for image and mask generators so they fit together

image_datagen.fit(X_train[:int(X_train.shape[0]*0.9)], augment=True, seed=seed)

mask_datagen.fit(Y_train[:int(Y_train.shape[0]*0.9)], augment=True, seed=seed)

x=image_datagen.flow(X_train[:int(X_train.shape[0]*0.9)],batch_size=BATCH_SIZE,shuffle=T
rue, seed=seed)

y=mask_datagen.flow(Y_train[:int(Y_train.shape[0]*0.9)],batch_size=BATCH_SIZE,shuffle=Tr
ue, seed=seed)

```

ภาพที่ 3.9 ชุดคำสั่งสร้าง training Image and Mask generator

3.2.9 สร้าง the validation Image and Mask generator

```

# Creating the validation Image and Mask generator

image_datagen_val = image.ImageDataGenerator()

mask_datagen_val = image.ImageDataGenerator()

image_datagen_val.fit(X_train[int(X_train.shape[0]*0.9):], augment=True, seed=seed)

mask_datagen_val.fit(Y_train[int(Y_train.shape[0]*0.9):], augment=True, seed=seed)

x_val=image_datagen_val.flow(X_train[int(X_train.shape[0]*0.9):],batch_size=BATCH_SIZE,s
huffle=True, seed=seed)

y_val=mask_datagen_val.flow(Y_train[int(Y_train.shape[0]*0.9):],batch_size=BATCH_SIZE,sh
uffle=True, seed=seed)

```

ภาพที่ 3.10 ชุดคำสั่งสร้าง the validation Image and Mask generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.10 สร้าง training and validation generator that generate masks and images

```
train_generator = zip(x, y)
val_generator = zip(x_val, y_val)
```

ภาพที่ 3.11 ชุดคำสั่งสร้าง training and validation generator for masks and images

3.2.11 สร้าง class สำหรับการหาค่าความแม่นยำของการสอนและทดสอบ

```
def mean_iou(y_true, y_pred):
    prec = []
    for t in np.arange(0.5, 1.0, 0.05):
        y_pred_ = tf.to_int32(y_pred > t)
        score, up_opt = tf.metrics.mean_iou(y_true, y_pred_, 2)
        K.get_session().run(tf.local_variables_initializer())
        with tf.control_dependencies([up_opt]):
            score = tf.identity(score)
        prec.append(score)
    return K.mean(K.stack(prec), axis=0)
```

ภาพที่ 3.12 ชุดคำสั่งสร้าง class สำหรับการหาค่าความแม่นยำของการสอนและทดสอบ

3.2.12 การสร้างการสร้ารูปแบบสถาปัตยกรรม U-Net

ชุดคำสั่งที่ 3.13 จะเป็นการสร้างส่วนของ contract path โดยจะอ้างอิงจาก ภาพที่ 3.10 เป็นต้นแบบอ้างอิงโดยทำการแบ่งแต่ละเลเยอร์ออกเป็น 10 เลเยอร์โดยทุกเลเยอร์จะใช้ activation function Elu เริ่มด้วย เลเยอร์ที่ 1 2 3 และ 4 ประกอบด้วยฟังก์ชันคอนโวลูชันในรูปแบบ 2 มิติ ฟังก์ชัน Dropout และฟังก์ชัน Maxpooling ต่อมาในเลเยอร์ 5 ประกอบด้วยฟังก์ชันคอนโวลูชันในรูปแบบ 2 มิติ และฟังก์ชัน Dropout สุดท้ายเลเยอร์ 6 7 8 และ 9 ประกอบด้วยฟังก์ชันคอนโวลูชันในรูปแบบ 2 มิติ ฟังก์ชัน Dropout ฟังก์ชัน คอนโวลูชัน 2 มิติ แบบทรานโพส และ ฟังก์ชัน คอนแคทที่เนท ในส่วนสุดท้ายของเอาพุท ใช้คอนโวลูชัน 2 มิติ และใช้ activation function ชนิด Sigmoid

```

# Build U-Net model
inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
s = Lambda(lambda x: x / 255) (inputs)
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (s)
c1 = Dropout(0.1) (c1)
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c1)
p1 = MaxPooling2D((2, 2)) (c1)
c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(p1)
c2 = Dropout(0.1) (c2)
c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c2)
p2 = MaxPooling2D((2, 2)) (c2)
c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(p2)
c3 = Dropout(0.2) (c3)
c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c3)
p3 = MaxPooling2D((2, 2)) (c3)
c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(p3)
c4 = Dropout(0.2) (c4)
c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(c4)
p4 = MaxPooling2D(pool_size=(2, 2)) (c4)
c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(p4)
c5 = Dropout(0.3) (c5)
c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(c5)
u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same') (c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')
(u6)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

u7 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same') (c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u7)
c7 = Dropout(0.2) (c7)
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c7)

u8 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u8)
c8 = Dropout(0.1) (c8)
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c8)
u9 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same') (c8)
u9 = concatenate([u9, c1], axis=3)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u9)
c9 = Dropout(0.1) (c9)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c9)

outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)

```

ภาพที่ 3.13 ชุดคำสั่งการสร้างรูปแบบสถาปัตยกรรม U-Net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.13 การกำหนดค่าของโมเดล

เมื่อสร้างรูปแบบจำลองเรียบร้อยแล้วจึงทำการเก็บโมเดลนี้โดยดั่งภาพที่ 3.14 ทำการเซฟโมเดลในชื่อว่า “1organ.h5” ในการรันครั้งต่อไปสามารถนำคำสั่ง model.save [28] ออกได้เลย

```
# Fit model
earlystopper = EarlyStopping(patience=3, verbose=1)
checkpointer = ModelCheckpoint('lkidtr.h5', verbose=1, save_best_only=True)
results = model.fit_generator(train_generator, validation_data=val_generator,
validation_steps=10, steps_per_epoch=250, epochs=3, callbacks=[earlystopper,
checkpointer])
```

ภาพที่ 3.14 ชุดคำสั่งกำหนดค่าของโมเดล

จากนั้นทำการตรวจสอบโมเดล ว่าเหมาะในการเริ่มการสอนหรือไม่และทำการโหลดโมเดลเก่ามาเพื่อทำการปรับเปลี่ยน ทำการกำหนดจำนวนรอบในการสอน อัตราการเรียนรู้ รวมถึงขนาดของข้อมูลที่ใช้ในการสอนในแต่ละครั้งต่อรอบ และพัฒนาค่าที่ได้ต่าง ๆ โดยทำการสอน ครั้งละ 250 สเต็ปต่อรอบ โดยทำทั้งหมด 3 รอบดั่งภาพที่ 3.15

3.2.14 ตั้งค่าการสอนปัญญาประดิษฐ์

```
model = Model(inputs=[inputs],outputs=[outputs])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=[mean_iou])
model.save('1organ.h5')
model.summary()
```

ภาพที่ 3.15 ตั้งค่าการสอนปัญญาประดิษฐ์

เมื่อทำการตั้งค่าเรียบร้อยแล้วจะใช้เวลาในการสอนปัญญาประดิษฐ์ โดยเวลาจะขึ้นอยู่กับอุปกรณ์ที่ใช้ จำนวนข้อมูล และ การตั้งค่าโมเดล จากนั้นนำมาทำการทดสอบตามชุดคำสั่งดั่งภาพที่ 3.16 โดยทำการให้ปัญญาประดิษฐ์ทำการคาดเดารูปแบบของอวัยวะอีกด้วย

3.2.15 ออกแบบชุดคำสั่งการทดสอบและคาดเดาผลลัพธ์ของปัญญาประดิษฐ์

```

# Threshold predictions
preds_train_t = (preds_train > 0.5).astype(np.uint8)
preds_val_t = (preds_val > 0.5).astype(np.uint8)
preds_test_t = (preds_test > 0.5).astype(np.uint8)
# Create list of upsampled test masks
preds_test_upsampled = []
for i in range(len(preds_test)):
    preds_test_upsampled.append(resize(np.squeeze(preds_test[i]),
                                       (sizes_test[i][0],
                                        sizes_test[i][1]),
                                       mode='constant',
                                       preserve_range=True))
# Perform a sanity check on some random training samples
ix = random.randint(0, len(preds_train_t))
imshow(X_train[ix])
plt.show()
imshow(np.squeeze(Y_train[ix]))

```

ภาพที่ 3.16 ชุดคำสั่งการทดสอบและคาดเดาผลลัพธ์ของปัญญาประดิษฐ์

3.2.16 ออกแบบชุดคำสั่งในการบีบอัดข้อมูลโดยไม่สูญเสียข้อมูลภาพ

เมื่อแสดงผลภาพที่ได้แล้ว การจัดเก็บข้อมูลให้อยู่ในโมเดลเพื่อนำมาใช้ในอนาคตจึงต้องทำ บีบอัดข้อมูลโดยไม่ทำให้สูญเสียข้อมูลของภาพเพื่อใช้ในการไม่ทำให้ความแม่นยำสูญเสียไปโดยใช้ ข้อมูลชุดคำสั่งดังภาพ 3.17

```
# Run-length encoding stolen from https://www.kaggle.com/rakhlin/fast-run-length-encoding-python
def rle_encoding(x):
    dots = np.where(x.T.flatten() == 1)[0]
    run_lengths = []
    prev = -2
    for b in dots:
        if (b>prev+1): run_lengths.extend((b + 1, 0))
        run_lengths[-1] += 1
        prev = b
    return run_lengths
def prob_to_rles(x, cutoff=0.5):
    lab_img = label(x > cutoff)
    for i in range(1, lab_img.max() + 1):
        yield rle_encoding(lab_img == i)
new_test_ids = []
rles = []
for n, id_ in enumerate(test_ids):
    rle = list(prob_to_rles(preds_test_upsampled[n]))
    rles.extend(rle)
    new_test_ids.extend([id_] * len(rle))
```

ภาพที่ 3.17 ชุดคำสั่งในการบีบอัดข้อมูลโดยไม่สูญเสียข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.17 ชุดคำสั่งสรุปความแม่นยำและการสูญเสียข้อมูลระหว่างสอน

เมื่อทำการแสดงการแสดงผลการทำนายของปัญญาประดิษฐ์แล้ว ทำการสรุปข้อมูลทั้งหมดที่ใช้ในการสอนในรูปแบบกราฟ การสูญเสียในการสอนและความแม่นยำในการทำนายดังภาพที่ 3.18

```
plt.show()
imshow(np.squeeze(preds_train_t[ix]))
plt.show()
# Perform a sanity check on some random validation samples
ix = random.randint(0, len(preds_val_t))
imshow(X_train[int(X_train.shape[0]*0.9):][ix])
plt.show()
imshow(np.squeeze(Y_train[int(Y_train.shape[0]*0.9):][ix]))
plt.show()
imshow(np.squeeze(preds_val_t[ix]))
plt.show()
plt.plot(results.history['mean_iou'])
plt.plot(results.history['val_mean_iou'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(results.history['loss'])
plt.plot(results.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

ภาพที่ 3.18 ชุดคำสั่งสรุปความแม่นยำและการสูญเสียข้อมูลระหว่างสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง หรือการวิเคราะห์ข้อมูล

ในส่วนของบทนี้จะอธิบายถึงผลลัพธ์ของการสอนปัญญาประดิษฐ์ โดยทำการทดลองครั้งละ 250 ระดับต่อรอบ และทำทั้งหมด 4 รอบ กับสองอวัยวะ ซึ่งเราได้ทำการทดสอบการคาดเดารูปร่างอวัยวะของผู้ป่วยออกมาสองชนิด ดังนี้ 1. กระเพาะปัสสาวะ โดยใช้ข้อมูลทั้งหมด 10 เคส ซึ่งแบ่งเป็นข้อมูลใช้สอน 8 เคส และข้อมูลที่ใช้ในการทดสอบ 2 เคส 2. ไต ได้ใช้ข้อมูลทั้งหมด 9 เคส เนื่องจากมีเคสที่ 6 ไม่สามารถทำการแยกออกมา เพราะ ภาพที่ได้รับนั้นไม่มีส่วนของไต จึงไม่สามารถนำเข้ามาทำการสอนได้ อาจจะทำให้เกิดข้อมูลที่เป็นขยะได้จึงแบ่งออกเป็น 9 เคส โดยใช้เคสที่ทำการสอนทั้งหมด 7 เคส และทำการทดสอบ 2 เคส

4.1 ผลการทดลองเมื่อทำการสอนปัญญาประดิษฐ์โดยคาดเดารูปร่างไตขวา

4.1.1 ผลการทดลองเมื่อทำการฝึกในครั้งแรก

```
loss: 0.6902 - mean_iou: 0.2100.  
.  
loss: 0.0520 - mean_iou: 0.7105 - val_loss: 0.0560 - val_mean_iou: 0.8098val_loss  
Epoch 00001: val_loss improved from inf to 0.05596
```

ภาพที่ 4.1 แสดงค่าสถานะของการสอนในรอบแรก

ผลการทดลองในการฝึกครั้งแรกจะเห็นว่าเริ่มต้นสอนมีการสูญเสีย 0.8199 และในครั้งสุดท้ายมีค่าความสูญเสียในการสอน 0.0520 ค่าความแม่นยำในการฝึกอยู่ที่ 0.7105

4.1.2 ผลการทดลองเมื่อทำการฝึกในครั้งที่สอง

```
loss: 0.0262 - mean_iou: 0.8101  
.  
loss: 0.0233 - mean_iou: 0.8358 - val_loss: 0.0372 - val_mean_iou: 0.8550  
Epoch 00002: val_loss improved from 0.05596 to 0.03723
```

ภาพที่ 4.2 แสดงค่าสถานะของการสอนในรอบที่สอง

ผลการทดลองในการฝึกรอบสองจะเห็นว่าเริ่มต้นเทรนมีความสูญเสีย 0.0262 และในครั้งสุดท้ายมีค่าความสูญเสีย 0.0233 ค่าความแม่นยำในการฝึกอยู่ที่ 0.8358

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ผลการทดลองเมื่อทำการฝึกในครั้งที่สาม

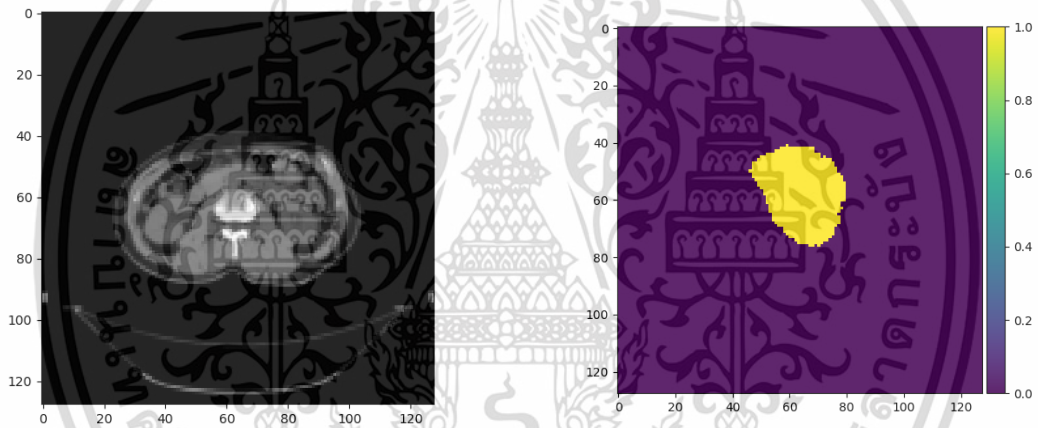
loss: 0.0250 - mean_iou: 0.8551

loss: 0.0178 - mean_iou: 0.8676 - val_loss: 0.0280 - val_mean_iou: 0.8780

Epoch 00003: val_loss improved from 0.03723 to 0.02796

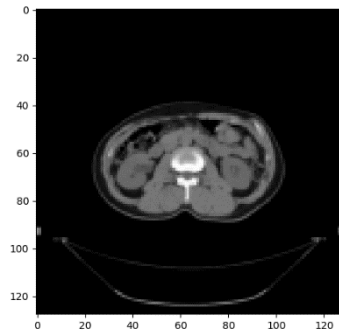
ภาพที่ 4.3 แสดงค่าสถานะของการสอนในรอบที่สาม

ผลการทดลองในการฝึกกรอบสามจะเห็นได้ว่าเริ่มต้นการสอนมีความสูญเสีย 0.0250 และในครั้งสุดท้ายมีค่าความสูญเสีย 0.0178 ค่าความแม่นยำในการฝึกอยู่ที่ 0.8676

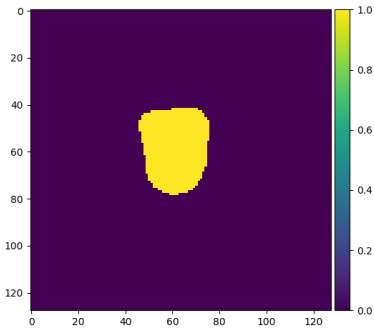


ภาพที่ 4.4 รูปที่ต้นแบบที่ใช้ในการคาดเดา ภาพที่ 4.5 รูปที่ปัญญาประดิษฐ์คาดเดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

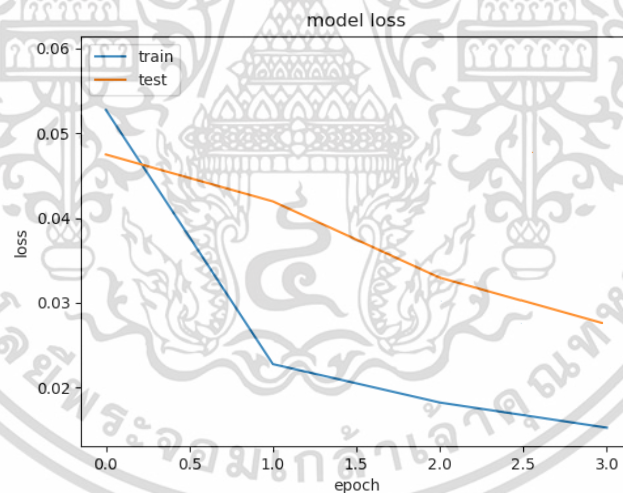


ภาพที่ 4.6 รูปที่ต้นแบบที่ใช้ในการคาดเดา



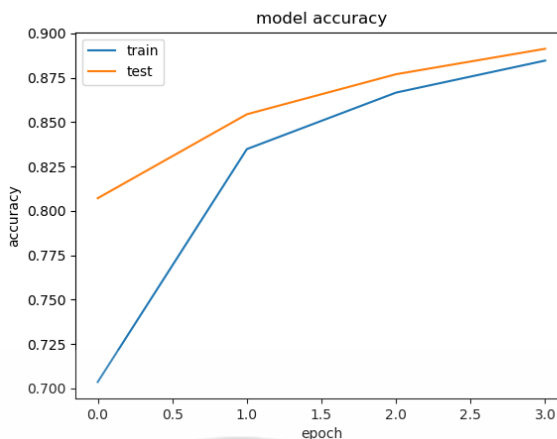
ภาพที่ 4.7 รูปที่ปัญหาประติษฐ์คาดเดา

ในการแสดงผลรูปของการฝึกในอวัยวะแรกใช้การเปรียบเทียบจากค่าสัมประสิทธิ์ความคล้ายโชนเซน-ไดซ์ โดยในการเริ่มสอนค่า จะเริ่มที่เข้าใกล้ 0 และเพิ่มมากขึ้นเข้าใกล้ค่า 0.9 ดังภาพที่ 46 โดยในแนวแกนนอนเป็นจำนวนรอบ และ แนวแกนตั้งเป็นค่าความแม่นยำ และแสดงค่าการการสูญเสียในการสอนปัญหาประติษฐ์ดังภาพที่ 47 โดยในแนวแกนนอนเป็นจำนวนรอบ และ แนวแกนตั้งเป็นค่าความสูญเสีย



ภาพที่ 4.8 การแสดงความแม่นยำของโมเดลนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.9 การแสดงการสูญเสียของการสอนปัญญาประดิษฐ์

4.2 ผลการทดลองเมื่อทำการสอนปัญญาประดิษฐ์โดยคาดเดารูปทรงกระเพาะปัสสาวะ

4.2.1 ผลการทดลองเมื่อทำการฝึกในครั้งแรก

loss: 0.6527 - mean_iou: 0.2193

loss: 0.0712 - mean_iou: 0.6534 - val_loss: 0.0548 - val_mean_iou: 0.7395

Epoch 00001: val_loss improved from inf to 0.05475

ภาพที่ 4.10 แสดงค่าสถานะของการสอนในรอบแรก

ผลการทดลองในการฝึกครั้งแรกจะเห็นว่าเริ่มต้นสอนมีการสูญเสีย 0.6527 และในครั้งสุดท้ายมีค่าความสูญเสียในการสอน 0.0712 ค่าความแม่นยำในการฝึกอยู่ที่ 0.6534

4.2.2 ผลการทดลองเมื่อทำการฝึกในครั้งที่สอง

loss: 0.0336 - mean_iou: 0.7405

loss: 0.0379 - mean_iou: 0.7677 - val_loss: 0.0451 - val_mean_iou: 0.7861

Epoch 00002: val_loss improved from 0.05475 to 0.04505

ภาพที่ 4.11 แสดงค่าสถานะของการสอนในรอบสอง

ผลการทดลองในการฝึกครั้งแรกจะเห็นว่าเริ่มต้นสอนมีการสูญเสีย 0.0336 และในครั้งสุดท้ายมีค่า

ความสูญเสียในการสอน 0.0379 ค่าความแม่นยำในการฝึกอยู่ที่ 0.7677

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดลองเมื่อทำการฝึกในครั้งที่สาม

loss: 0.0346 - mean_iou: 0.7863

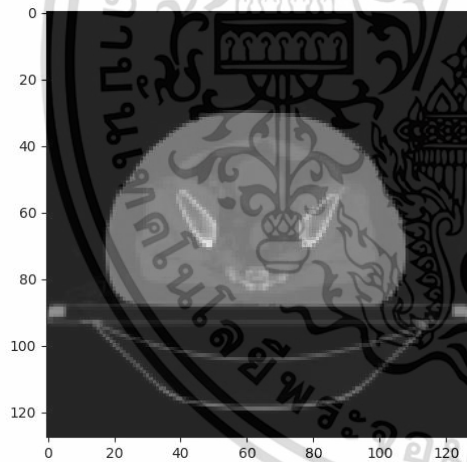
loss: 0.0350 - mean_iou: 0.7967 - val_loss: 0.0491 - val_mean_iou: 0.8046

Epoch 00003: val_loss did not improve from 0.04505

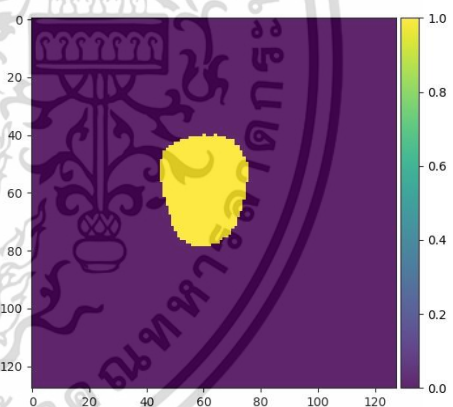
ภาพที่ 4.12 แสดงค่าสถานะของการสอนในรอบสาม

ผลการทดลองในการฝึกกรอบสามจะเห็นได้ว่าเริ่มต้นการสอนมีความสูญเสีย 0.0346 และในครั้งสุดท้ายมีค่าความสูญเสีย 0.0350 ค่าความแม่นยำในการฝึกอยู่ที่ 0.7967

ในการแสดงผลสรุปของการฝึกในอวัยวะที่สองใช้การเปรียบเทียบจากค่าสัมประสิทธิ์ความคล้ายโซเรนเซน-ไดซ์ โดยในการเริ่มสอนค่า จะเริ่มที่เข้าใกล้ 0 และเพิ่มมากขึ้นเข้าใกล้ค่า 0.9 ดังภาพที่ 46 โดยในแนวแกนนอนเป็นจำนวนรอบ และ แนวแกนตั้งเป็นค่าความแม่นยำ และแสดงค่าการการสูญเสียในการสอนปัญญาประดิษฐ์ดังภาพที่ 47 โดยในแนวแกนนอนเป็นจำนวนรอบ และ แนวแกนตั้งเป็นค่าความสูญเสีย

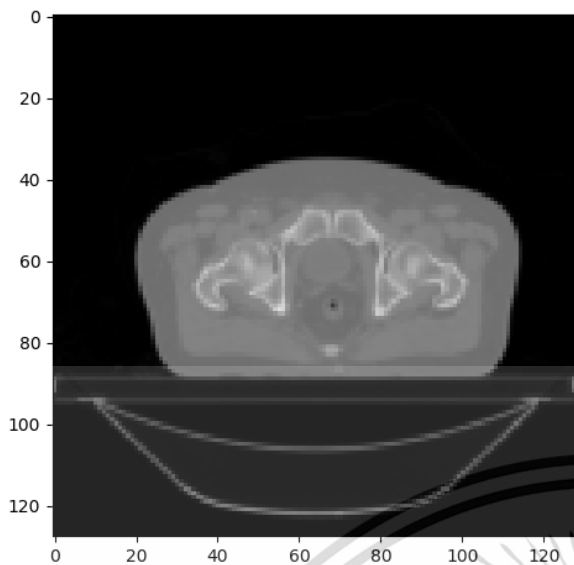


ภาพที่ 4.13 ภาพที่ใช้ในการทำนาย

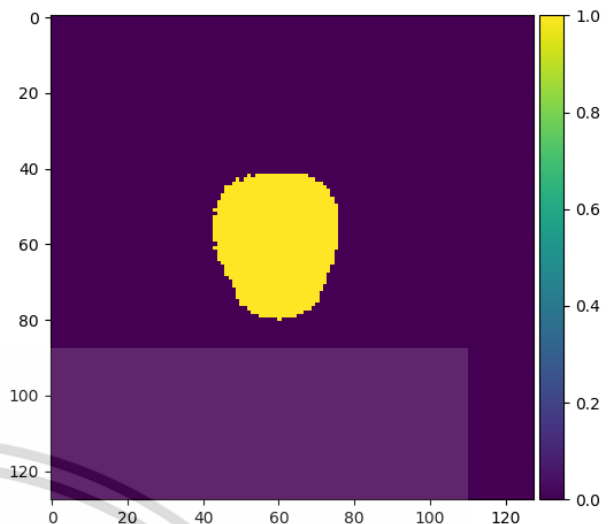


ภาพที่ 4.14 ปัญญาประดิษฐ์ทำนาย

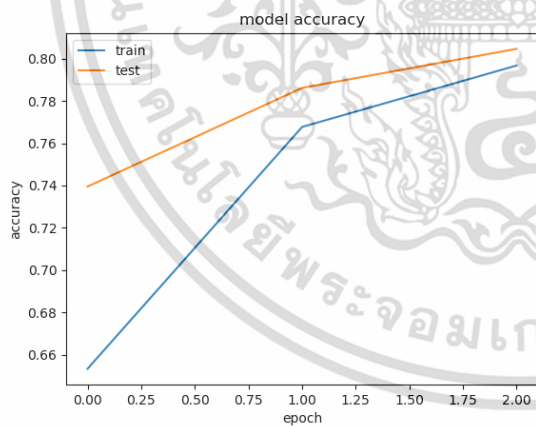
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



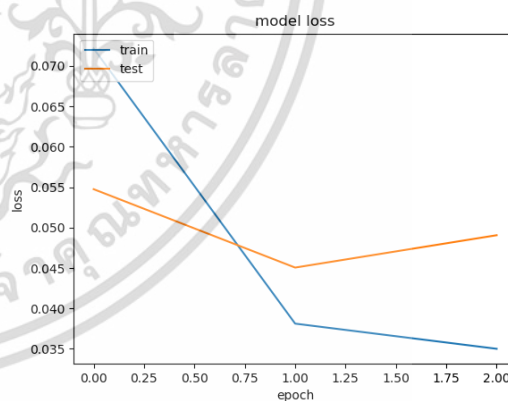
ภาพที่ 4.15 ภาพที่ใช้ในการทำงาน



ภาพที่ 4.16 ปัญญาประดิษฐ์ทำนาย



ภาพที่ 4.17 กราฟสรุปความแม่นยำ



ภาพที่ 4.18 กราฟสรุปความสูญเสีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

ในการทำการฝึกช่วงแรกในอวัยวะของไตขวา มีการพัฒนาการลดลงของค่า loss ลงเรื่อย ๆ จนค่า loss จะไม่สามารถลดลงได้อีก ในขณะเดียวกัน ค่า mean_iou หรือค่าความแม่นยำ มีการเพิ่มมากขึ้นเรื่อย ๆ จนเริ่มหยุดการเพิ่มขึ้น ความชันของกราฟจะค่อย ๆ ลดลง จนเป็นเส้นตรง ส่งผลให้การทำการฝึกมากขึ้นทำให้ผลลัพธ์ของความแม่นยำออกมาได้ดีเรื่อย ๆ และการคาดเดาของปัญญาประดิษฐ์พัฒนามากขึ้นมีความใกล้เคียงกับข้อมูลที่ใส่เข้าไปมากขึ้น โดยผลลัพธ์ในด้านของการทำนายมีความน่าพึงพอใจอย่างมาก และเราสามารถนำโมเดลที่เราทำการเก็บไว้ไปใช้ในการต่อยอดในงานอื่น ๆ ได้อีกมากมายในงานเชิง Data science และ Big data ได้อีกด้วย

5.2 ปัญหาที่พบ

1. ในขณะที่ทำการสอนปัญญาประดิษฐ์โดยมีข้อมูลที่มีอยู่จำกัดการออกแบบระบบสถาปัตยกรรมของการสอนปัญญาประดิษฐ์เป็นสิ่งจำเป็นเนื่องจากหากทำการฝึกโดยจำนวนการฝึกน้อยเกินไป ทำให้เกิดข้อผิดพลาดที่โมเดลจะเรียนรู้ไม่เพียงพอหรือเรียกว่า Underfit แต่ถ้าทำการสอนมากจนเกินไปจะทำให้เกิดสิ่งที่เรียกว่า Overfit ทำให้ไม่สามารถสอนโมเดลได้อีกและไม่สามารถใช้การโมเดลนั้นได้
2. การทำการสอนในแต่ละรอบเนื่องจากชุดข้อมูลที่เรามีจำนวนขนาดหนึ่งซึ่งส่งผลต่อการใช้เวลาในการฝึกปัญญาประดิษฐ์
3. การทำนายของปัญญาประดิษฐ์ยังมีความแม่นยำที่ไม่ถูกต้อง 100 % เนื่องจากการนำไปใช้ในทางการแพทย์ความแม่นยำขั้นต่ำที่ต้องใช้ต้องไม่ต่ำกว่า 97 % แต่จากการทดลองนั้น คาดเดาได้เพียง 89 % และคาดเดา กระเพาะปัสสาวะได้เพียง

5.3 แนวทางการแก้ไข

1. การออกแบบระบบของการฝึกปัญญาประดิษฐ์ จึงจำเป็นต้องออกแบบจำนวนที่ใช้ในการฝึกที่พอดี ถ้าหากเกิดการ underfit ให้ทำการเพิ่มจำนวนการฝึก แต่ถ้าหากเกิด overfit ต้องทำการเพิ่มจำนวนข้อมูลเข้าไปดังนั้นผลงานวิจัยนี้สามารถนำไปใช้กับระบบของ Big data เข้ามาเกี่ยวข้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเพิ่มความแม่นยำและความสามารถในการทำนายนั้นสามารถทำได้โดยการถ่ายภาพที่เป็นข้อมูลเข้าพุทให้ดีขึ้น มีการคัดกรองในสิ่งที่เป็นข้อมูลที่ไม่จำเป็นออกไปเพื่อลดการสูญเสียในการฝึก และยังทำให้ความแม่นยำเพิ่มมากขึ้นอีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] GPU (ออนไลน์).2019 ,November 2; Available from:
<https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73>
- [2] บริษัทเอโซซอฟต์จำกัด , Python (ออนไลน์).2019,September 20; Available from :
<https://www.aosoft.co.th/article/322/Python->
- [3] import python(ออนไลน์) .2020,January 22;
 Available from: <http://www.tohoho-web.com/python/module.html>
- [4] นาย วิทยา พรพิชฌพงษ์, โครงข่ายประสาทเทียม (Artificial Neural Networks - ANN) (ออนไลน์) .2019 ,November 4 ;Available from:
<https://www.gotoknow.org/posts/163433>
- [5] เพอร์เซปตรอน (Perceptron) (ออนไลน์). 2020,January 19;Available from:
<https://medium.com/odds-team/%E0%B8%AA%E0%B8%A3%E0%B8%B8%E0%B8%9B-deep-learning>
- [6] learning with supervisor (ออนไลน์). 2020,January 10;
 Available from: <https://medium.com/@every.phu/supervised-learning>
- [7] Fully Convolutional Neural Network– FCNN(ออนไลน์). 2020,January 12;
 Available from: <https://medium.com/@natthawatphongchit/>
- [8] Upsampling Layer (ออนไลน์). 2020,January 15; Available from:
<https://machinelearningmastery.com/upsampling-and-transpose-convolution-layers-for-generative-adversarial-networks/>
- [9]_Skip Connection(ออนไลน์). 2020,February 4; Available from:
<https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [10] deep learning (ออนไลน์). 2019,October 4; Available from:<https://www.thaiprogrammer.org/2018/12/deep-learning>
- [11] Web Developer Nextzy Technology Co., LTD., สรุปแนวคิด Neural Network (ออนไลน์).2019 ,August 22; Available from: <https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic>
- [12] type of deep learning architecture (ออนไลน์) .2019,September 10; Available from: <https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/>
- [13] โรงพยาบาลศิริราช , MRI, CT Scan, PET Scan (ออนไลน์).2019,August 20 ;Available from: <http://www.thaibreastcancer.com/969/>
- [14] ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพดิจิทัล (ออนไลน์).2019,October 7; Available from: <https://nextsoftwares.wordpress.com/2014/05/22/>
- [15] iBook engineering, การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ (ออนไลน์).2019, November 5; Available from: <http://ibookengineering.blogspot.com/2015/08/morphological-processing.html>
- [16] การแยกบริเวณรูปภาพ (Image Segmentation) (ออนไลน์).2019,October 3; Available from: <https://www.bloggang.com/mainblog.php?id=hin-kmitnb>
- [17] Softmax function (ออนไลน์) . 2019,September 10;Available from: <https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/>
- [18] dropout (ออนไลน์).2020,January 15;Available from: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>
- [19] Dice coefficient . (ออนไลน์) . 2020,February 9; Available from: <http://www.arts.chula.ac.th/~ling/thesis/2560PhD-Ling-Supawat.pdf>
- [20] Entropy ความวุ่นวาย และ Uncertainty ความไม่แน่นอนของระบบใน Deep Learning AI Available . 2020,February 9;from: [https://mc.ai/entropy-](https://mc.ai/entropy-%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%A7%E0%B8%B8%E0%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[B9%88%E0%B8%99%E0%B8%A7%E0%B8%B2%E0%B8%A2-%E0%B9%81%E0%B8%A5%E0%B8%B0-uncertainty-%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B9%84/?fbclid=IwAR2EPGK mXd_BLEFytRf3K1CDU-damwZq6-guqO-Enlexcp_-k9ea_WrVIEk](https://medium.com/@parinkittipongdaja/blefytrf3k1cdudamwzq6-guqo-enlexcp-k9ea-wrviek)

[21] dicom image (ออนไลน์) .2020,January 26;

Available from:<https://medium.com/@parinkittipongdaja/>

[22] PNG image (ออนไลน์) .2020,February 2;

Available from: [https://www.mindphp.com/developer/67-](https://www.mindphp.com/developer/67-%E0%B8%A3%E0%B8%B9%E0%B8%9B%E0%B8%A0%E0%B8%B2%E0%B8%9E/1953-png-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)

[.%E0%B8%A3%E0%B8%B9%E0%B8%9B%E0%B8%A0%E0%B8%B2%E0%B8%9E/1953-png-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html](https://www.mindphp.com/developer/67-%E0%B8%A3%E0%B8%B9%E0%B8%9B%E0%B8%A0%E0%B8%B2%E0%B8%9E/1953-png-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)

[23] Deep Learning with Python by Francois Chollet. 2020,January 19;

Available from: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>

[24] นายศุภวัจน์ แต่รุ่งเรือง. การตรวจเทียบภายนอกหากลักลอกในงานวิชาการโดยใช้แบบจำลอง ซัพพอร์ตเวกเตอร์แมชชีนและ การวัดค่าความละม้ายของข้อความ. หน้า112.

[25] นายสิโรตม มงคลธนาภรณ์. การจำแนกโครงสร้างจุลภาคไทเทเนียมด้วยโครงข่ายประสาทคอนโวลูชันเต็มรูป. หน้า21

[26] M. Thitivirut, J. Leekitviwat, C. Pathomsathit and P. Phasukkit, "Image Enhancement by using Triple Filter and Histogram Equalization for Organ Segmentation," *2019 12th Biomedical Engineering International Conference (BMEiCON)*, Ubon Ratchathani, Thailand, 2019, pp. 1-5.

[27] ความแตกต่างของ JPG / GIF / PNG(ออนไลน์) .2020,January 26;

Availablefrom: <https://natchaphak.wordpress.compng/>

[28] keras(ออนไลน์) .2020,January 26; Availablefrom:<https://keras.io/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Image enhancement by using triple filter and histogram equalization for organ segmentation

Mongkol Thitivirut
Faculty of Engineering, King
Mongkut's Institute of Technology
Ladkrabang Bangkok, Thailand
59011086@kmitl.ac.th

Jirayuts Leekitviwat
Faculty of Engineering, King
Mongkut's Institute of Technology
Ladkrabang Bangkok, Thailand
59010221@kmitl.ac.th

Carat Pathomsathit
Faculty of Engineering,
Mahidol University,
Nakhonpathom, Thailand
carat.pat@student.mahidol.ac.th

Pattarapong Phasukkit
Faculty of Engineering, King
Mongkut's Institute of
Technology Ladkrabang,
Bangkok, Thailand
pattarapong.ph@kmitl.ac.th

Abstract— Convolution filters are used for blurring and sharpening to modifying image when combined with the Histogram equalization for enhancing medical image by adjusting the contrast of the image. Then, the image will be optimized. So, use morphological for segmentation. This research has been designed the filtering in a part of the convolution filter by using the triple filter to adjust the threshold of morphological and normalization value. The results of this experiment are presented that the accuracy of segmentation is increasing and detecting the multiple organs. Nevertheless, we also found that each algorithm can be used for each organ.

Keywords—Histogram equalization, Normalization, Morphological, Convolutional filter.

1. INTRODUCTION

First process of the cancer treatment diagnosis, the physician takes the patient to gather the medical image by Ct scan, MRI or Cone-beam CT, which is used for planning to treatment through radiotherapy, which increases the efficiency of treatment and decreases the damage, which the patient gets. The physician must manual organelle segmentation by himself. Then the image reconstruction process to gathering the image into a 3D-model form. This process will take several times per case.

So, this research has been designed the process of image processing for auto segmentation in CT scan which can help to decrease time to organelle contouring and time for cancer treatment. Also, gain the efficiency to cure the patient.

The image preprocessing technique for preparing the data to segmentation by designing the experiment, first of all, Turns all image resources into grayscale, then use the grayscale images to the process of histogram equalization and filtering them into primary filter and take all images that filtered through the secondary filtering, by the primary filter must not

be the same filter with the secondary filter. Next, Getting the data through the secondary filter and adjust the contrast of images before taking them to the histogram equalization process again. Later, Use the bilateral filter as the tertiary filter and enhance images by adjusting the normalization value and inverting them into binary images, next, to the process of segmentation, the organelle by using the region of interest (ROI) for locating the specific organelle with the variable thresholding.

II. RELATED WORK

A. Convolution filter

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement. Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

1) 2D's convolution

Convolution involving one-dimensional signals is referred to as 1D convolution or just convolution. Otherwise, if the convolution is performed between two signals spanning along two mutually perpendicular dimensions, then it will be referred to as 2D convolution. This concept can be extended to involve multidimensional signals due to which we can have multidimensional convolution. In the digital domain, convolution is performed by multiplying and accumulating the instantaneous values of the overlapping samples corresponding to two input signals, one of which is flipped. This definition of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1D convolution is applicable even for 2D convolution except that, in the latter case, one of the inputs is flipped twice.

This kind of operation is extensively used in the field of digital image processing wherein the 2D matrix representing the image will be convolved with a comparatively smaller matrix called 2D kernel

2) Gaussian filter

Gaussian filter is considered as an ideal time domain filter. It is one of the non-uniform lowpass filter. The impulse response of such filters is a Gaussian function that is gaussian filters are those filters with Gaussian function. It has a minimum possible group delay. The main function of Gaussian filter is to minimize the low and high signals from distortion [12]. In an image handling, a gaussian obscure which is otherwise called gaussian smoothening is an aftereffect of obscuring of a picture by a gaussian capacity. This gaussian smoothening administrator is typically a 2-D convolutional administrator to obscure pictures and expel subtle elements and clamors. Salt and pepper are usually more challenging for gaussian filter.

3) Median

Median filtering is a nonlinear operation frequently utilized as a part of picture handling to decrease salt and pepper and spot noise [9]. This sort of separating keeps the sharpness of a picture edge while evacuating clamors. In this filter the neighboring pixels are ranked according to brightness and middle esteem turn into the new incentive for the local pixel. While sifting this filter does not move limits as can occur with traditional smoothening filters. Since the middle is less touchy than the intend to outrageous qualities that can be more viably evacuated. Median filter can be connected over and over again if essential since the edges are negligibly debased. Digitization commotions, for example, straight lines exhibit in the dominant part of procured mammogram images can be sifted utilizing a two-dimensional (2D) median Filtering approach in a 3-by-3 neighborhood association [10][11]. The fundamental inconveniences of median filters are it will make another flag point that does not exist in unique flag and which might be undesirable in some application. The median filter may remove both the noise and the fine detail. Anything relatively small compared to the size of the neighborhood will have only minimal effect on the value of the median, and will be filtered out that is, the median filter can't distinguish fine detail from noise.

4) Bilateral filter

Bilateral filtering was introduced by Tomasi and Manduchi [14] as an alternative to anisotropic diffusion [15]. It is a nonlinear filter which combines domain and range filtering where the output is a weighted average of the input image f . It extends the concept of Gaussian smoothening by weighting the filter coefficients with their corresponding relative pixel intensities. Pixels that are very difference in intensity from the center pixel are weighted as less influence even though they may be in close proximity to the central pixel [16].

B. Histogram equalization

Histogram equalization is a technique for adjusting the pixel values in an image to enhance the contrast by making those intensities more equal across the board

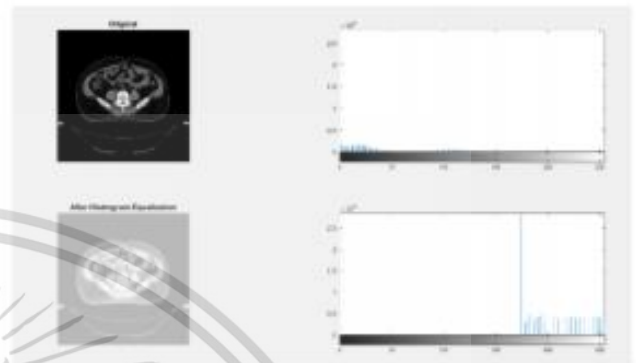


Figure 1. Histogram equalization

C. Morphological

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

1) *Dilation*: The dilation operation makes an object to grow by size. The extent to which it grows depends on the nature and shape of the structuring element. The dilation f an image A (set) by structuring element B is defined as

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (1)$$

If set B is reflected about its origin and shifted by z , then the dilation of A by B is the set of all displacements z such that B and A have at least one common element. Dilation, as said above, adds pixels to the boundary elements. The dilation process enlarges the number of pixels with value one (foreground) and shrinks the number of pixels with value zero (background).

2) *Erosion*: The erosion operation is complementing of the dilation operation in context with the operation effect. That is erosion operation causes object to lose its size. The erosion of an image A by structuring element B is defined as

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2)$$

The erosion of image A by structuring element B is the set of all points z such that the structuring element B is translated by z is a subset of the image. This operation results in loss of boundary pixels of the object. The erosion process enlarges the number of pixels with value zero (background) and shrinks the number of pixels with value one (foreground)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) *Opening*: The opening of an image is a combinational operation of erosion and dilation. The opening of an image A by structuring element B is defined as

$$A \circ B = (A \ominus B) \oplus B \quad (3)$$

The above definition gives the relationship between opening and erosion & dilation. It states that the opening operation is nothing but the erosion of an image by a structuring element and the resultant is dilated with the same structuring element.

4) *Closing*: The closing of an image is also a combinational operation of erosion and dilation. It differs from the opening operation in the sense of order of occurrence of erosion and dilation operations. The closing of an image A by structuring element B is defined as

$$A \cdot B = (A \oplus B) \ominus B \quad (4)$$

The relation between erosion & dilation with closing is given in the above mathematical statement. It shows that closing operation is the dilation of an image A by the structuring element B and the resultant is eroded with the same structuring element.

D. Segmentation

Segmentation is the process that dividing the data of image into shrink by the individual section, which is that area stand for object. Segmentation will be finished when the target object is divided completely, and the result of Segmentation process is measuring the last process of image analytical by computer. In this paper use thresholding method segmentation. Thresholding methods are the simplest methods for image segmentation. These methods divide the image pixels with respect to their intensity level. These methods are used over images having lighter objects than background. The selection of these methods can be manual or automatic (i.e. can be based on prior knowledge or information of image features. There are basically three types of thresholding:

1) *Global Thresholding*: This is done by using any appropriate threshold value T. This value of T will be constant for whole image. On the basis of T the output image can be obtained from original image as:

$$q(x, y) = \begin{cases} 1, & \text{if } p(x, y) > T \\ 0, & \text{if } p(x, y) \leq T \end{cases} \quad (5)$$

2) *Variable Thresholding*: In this type of thresholding, the value of T can vary over the image. This can further be of two types:

a) *Local Threshold*: In this the value of T depends upon the neighborhood of x and y.

b) *Adaptive Threshold*: The value of T is a function of x and y.

3) *Multiple Thresholding*: In this type of thresholding, there are multiple threshold values like T0 and T1. By using these output image can be computed as:

$$q(x, y) = \begin{cases} m, & \text{if } p(x, y) > T1 \\ n, & \text{if } p(x, y) \leq T1 \\ 0, & \text{if } p(x, y) \leq T0 \end{cases} \quad (6)$$

The values of thresholds can be computed with the help of the peaks of the image histograms. Simple algorithms can also be generated to compute these.[7]

E. Normalization

normalization is a process that changes the range of pixel intensity values. Applications include photographs with poor contrast due to glare, for example. Normalization is sometimes called contrast stretching or histogram stretching. In more general fields of data processing, such as digital signal processing, it is referred to as dynamic range expansion. Normalization transforms an n-dimensional grayscale image

$$I: \{X \subseteq \mathbb{R}^n\} \rightarrow \{Min, \dots, Max\} \quad (7)$$

with intensity values in the range (Min, Max), into a new image

$$I_N: \{X \subseteq \mathbb{R}^n\} \rightarrow \{newMin, \dots, newMax\} \quad (8)$$

The linear normalization of a grayscale digital image is performed according to the formula

$$I_N = (I - Min) \frac{newMax - newMin}{Max - Min} + newMin \quad (9)$$

III. EXPERIMENT

We use the Data which has an amount of Medical image 127 data that gathers from Computerized Tomography. Scan in range of lower abdomen to the upper leg. So, we designed the experiment. firstly, Turns all image resources into grayscale. then use the grayscale images to the process of ordinary histogram equalization. and filtering them into the primary filter with 4 kinds of filter, 2D's convolution, Gaussian Blur, Median and Bilateral filter. Then take all images that filtered to the secondary filter. The primary filter must not be the same filter with the secondary filter. Example, Primary filter use 2D's convolution then secondary filter use another filter that must not be 2D's convolution. Therefore, get all images through the secondary filter to adjust the contrast. Before, taking them to the ordinary histogram equalization process. Later, Use the bilateral filter as the tertiary filter and enhance images by adjusting the normalization. Inverting all images to binary images. Lastly, The process of the organ segmentation by thresholding method segmentation. and using the region of interest (ROI) detection to locate the organ. by adjusting the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

density and the area of the organ, which has the variable thresholding for contour the target organ. Workflow of the experiment in the figure 2.

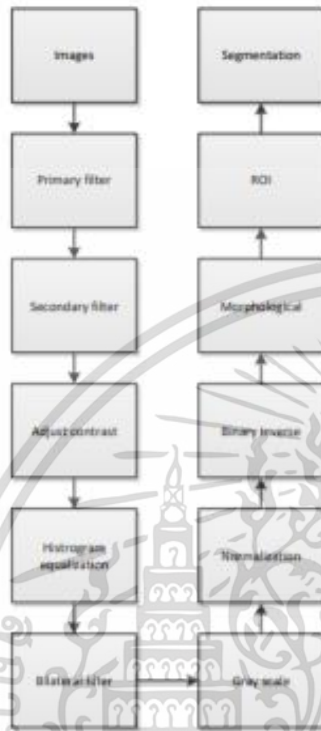


Figure 2. Workflow of the experiment.

IV. CONCLUSION

From the experiment, our process of image processing that we obtain the best optimization process, Firstly, Uses the Gaussian filter as the primary filter and Bilateral filter as the secondary filter. Then, Uses the Bilateral filter as the tertiary. According to, Soft tissue throughout the body will have varying water contents depending on cell types and the processes going on in and around those cells [2]. We found that the electron density affects the thresholding value of image and value of normalization. because of the water level in individual organs. So, we have the value of density and normalization of segmentation individual organelle following the table I. the result image that through the experiment following in figures 4, 5, 6, 7, 8. So, figures that are showed have 3 section. In the first of section is the original image, next is the mask of target organelle alone, last is the image that through the process and contour organelle. Accordingly, this research can be improved for future work which can be applied with deep learning work. To summarize the Accuracy of this process for organ segmentation is following this table II. However, this paper can be improved. And can be applied with deep learning as the pre-training data in the future.

TABLE I. The value of normalization and density individual organelle

Name	Normalization	Density
Skeleton	0.83	0.4
Kidneys	0.9	0.8
Urinary	0.91	0.85
Colon	0.89	0.45

a. From human physiology

Figure 3. Percentage of water in 4 type of organs



Figure 4. Urinary Bladder



Figure 5. Right Kidney



Figure 6. Left Kidney



Figure 7. Colon



Figure 8. Skeleton

TABLE II. ACCURACY OF EACH ORGANS SEGMENTATION

Name	Accuracy (%)
Skeleton	100
Kidneys	96
Urinary	96.4
Colon	94.6

a. From 1200 training image in each of 4 types organs.

Figure9. Accuracy of 4 types of organ segmentation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REFERENCES

- [1] S. Suhas and C. R. Venugopal, "MRI image preprocessing and noise removal technique using linear and nonlinear filters," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICECCOT), Mysuru, 2017, pp. 1-4 Imaging of the urinary tract: the role of CT and MRI, *Pediatr Nephrol*, 2011
- [2] Pethig, Ronald & Kell, Douglas. (1987). *The passive electrical properties of biological systems: Their significance in physiology, biophysics and biotechnology. Physics in medicine and biology.*
- [3] Wang Rui and Wang Guoyu, "Medical X-ray image enhancement method based on TV-homomorphic filter," 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, 2017, pp. 315-318.
- [4] Y. Jia, C. Rong, C. Wu and Y. Yang, "Research on the decomposition and fusion method for the infrared and visible images based on the guided image filtering and Gaussian filter," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2017, pp. 1797-1802.
- [5] Z. Yao, Z. Lai, C. Wang and W. Xia, "Brightness preserving and contrast limited bi-histogram equalization for image enhancement," 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, 2016, pp. 866-870.
- [6] H. Kaur and J. Rani, "MRI brain image enhancement using Histogram Equalization techniques," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2016, pp. 770-773.
- [7] *Medical Image Segmentation Methods, Algorithms, and Applications*, Alireza Norouzia, Mohd Shafry Mohd Rahima, Aynan Altameemb, Tajzila Sabae, Abdolvahab Ehsani Rada, Amjad Rehman & Mueen Uddin, 2014
- [8] S. Yadav and K. K. Singh, "Image x-sectionalzison techniques: A review," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2017, pp. 195199.
- [9] A. Rebbi, S. Abid and F. Fnaiech, "Fabric defect detection using local homogeneity and morphological image processing," 2016 International Image Processing, Applications and Systems (IPAS), Hammamet, 2016, pp. 1-5.
- [10] Rahimeh Rouhi, Mehdi Jafari, Shohreh Kazeri, Peiman Keshavarzian, "Benign and malignant breast tumors classification based on region growing and CNN segmentation," February 2015
- [11] T. Huang, G. Yang, G. Tang, "A fast two-dimensional median filtering algorithm", *Acoustics, Speech and Signal Processing*, IEEE Transactions on, vol. 27, 1979.
- [12] Armen Sahakyan, Hakop Sarukhanyan, "Segmentation of the Breast Region in Digital Mammograms and Detection of Masses", (IACSA) *International Journal of Advanced Computer Science and Applications*, Vol. 3, No.2, 2012.
- [13] Nithya. K. Aruna.A. Anandakumar.H. Anuradha.B."A Survey On Image Denoising Methodology On Mammogram Images" *International journal of scientific & technology research* volume 3, issue 11, november 2014.
- [14] Kshema, M. J. George and D. A. S. Dhas, "Preprocessing filters for mammogram images: A review," 2017 *Conference on Emerging Devices and Smart Systems (ICEDSS)*, Tiruchengode, 2017, pp. 1-7.
- [15] c. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proc. IEEE International Conference on Computer Vision*, Bombay, India, pp. 839-846, 1998.
- [16] P. Perona and J. Malik, "Scale-Space and Edge Detection using Anisotropic Diffusion," *IEEE Trans. PAMI*, Vol. 12, No.7, pp.629-639, 1990.
- [17] S. Srisak, "Bilateral filtering as a tool for image smoothing with edge preserving properties," 2014 *International Electrical Engineering Congress (iEECON)*, Chonburi, 2014, pp.1-4.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2018 Data Science Bowl

Find the nuclei in divergent images to advance medical discovery

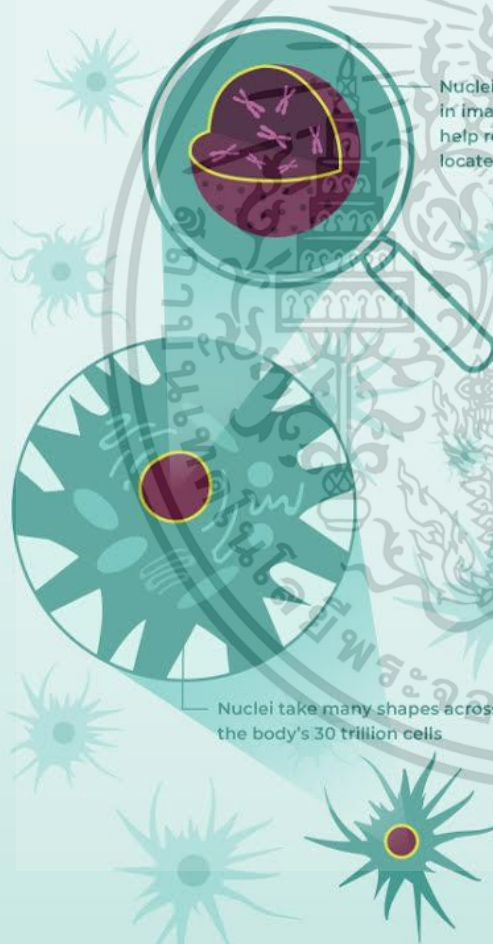
Spot Nuclei. Speed Cures.

The challenge: Create an algorithm to automate nucleus detection

40%
of all deaths are caused by illnesses like heart disease and cancer¹

75%
of rare diseases affect children²







30%
of affected children with rare diseases die before age 5²



Nuclei are distinctive in images and can help researchers locate cells

Nuclei take many shapes across the body's 30 trillion cells

Finding the nucleus helps to...

-  locate cells in varied conditions to enable faster cures
-  free biologists to focus on solutions
-  improve throughput for research and insight
-  reduce time-to-market for new drugs— currently 10 years
-  increase # of compounds for experiments
-  improve health and increase quality of life

SOURCES: ¹Heart disease and cancer causing 40% of all deaths – World Health Organization (WHO). 2015 – 56.4 million deaths. <http://www.who.int/mediacentre/factsheets/fs310/en/>
2015 – 17.7 deaths from cardiovascular diseases (CVDs). <http://www.who.int/mediacentre/factsheets/fs317/en/>
2015 – 8.8 million deaths from cancer. <http://www.who.int/mediacentre/factsheets/fs297/en/>
²Childhood diseases - 75% of rare diseases affect children; 30% die before age 5 – European Society of Paediatric Oncology (SIOP). <http://www.siope.eu/SIOPE-EU/English/SIOPE-EU/Advocacy-Activities/Rare-Diseases/page.aspx/148>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้ในการทำงานวิจัย แบ่งเป็นสองส่วนดังนี้

1. ด้านอุปกรณ์เครื่องมือที่ใช้ - CPU : Inter(R) Core(TM) i7-7700HQ , 2.80GHz (8 CPUs)
 - Ram: Patriot Viper ddr4 8 gb x 2
 - SSD : Sumsung 970 Evo 250 gb M.2 NVMe
 - GPU : Nvidia GTX 1060 (3 gb)
 - DirectX Version : DirectX 12
2. ด้านโปรแกรม
 - matlab
 - Python 3.7.4 64 bit
 - Visual Studio Code
 - Keras Tensorflow

วิธีการติดตั้ง Keras + Tensorflow (GPU version) บน windows 10

เครื่องมือที่ต้องใช้สำหรับการติดตั้ง Keras และ Tensorflow สำหรับ GPU

1. Visual Studio 2017 Community Edition ใช้สำหรับ Compile CUDA library โดย
ขณะที่เขียน นี้ CUDA อยู่ version 9 แต่เราจะลง version 8 เพราะ Tensorflow
version ใหม่ ยังไม่เสถียร
2. Anaconda (64-bit) with Python 3.6 เน้นไปที่ Python version 3 เพราะ version
2 แทบจะเลิกใช้งานแล้ว
3. CUDA V8 (64-bit) ใช้สำหรับ คำนวณบน GPU สำหรับการคูณ Matrix parallel
computing
4. cuDNN v6 สำหรับ CUDA 8.0 ใช้รัน Deep Neural Net (convolution neural
networks)
5. Keras 2 + Tensorflow-gpu
 - Keras เป็นส่วนที่ทำให้เขียน Deep Neural Network สะดวกขึ้นง่ายขึ้น เพราะไม่ต้อง
คำนวณจำนวน input output โดย keras จะเป็นตัวทำให้ และส่งให้ Back end เช่น
Theano, Tensorflow, or CNTK
6. Tensorflow-gpu (1.4) เป็น Backend สำหรับทำ Deep Neural Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Installation

Outline step:

1. Install Visual Studio 2017 Community Edition
2. Install Anaconda (64-bit) with Python 3.6
3. Install CUDA and CuDNN
4. Install Tensorflow-gpu + Keras

1. Installation Visual Studio 2017 Community Edition

ทำการ Download Visual Studio จากนั้นทำการติดตั้งแบบ Online จากนั้นก็ทำการติดตั้ง ตัวโปรแกรมจะทำการ Download package มาติดตั้งให้หลังจากติดตั้งเรียบร้อยแล้วให้ทำการ add path ของ Visual Studio 2017 ด้วย วิธีการคือ

- 1.1 เพื่อชื่อ VS14_PATH ในส่วนของ System variables ด้วย path “C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC” ซึ่งถ้าติดตั้งเสร็จแล้วมันจะมี ชื่อตัวแปรนี้ในระบบมาให้เราจะใช้ชื่อ VS14_PATH อ้างอิง path ยาวๆ จากนั้นทำการ Add path ดังกล่าวไปที่ระบบ คลิกที่ Path และเลือก Edit จากนั้นทำการ ใส่ Path ตามลงไปในส่วนสุดท้ายได้เลย

2. install Anaconda

ให้ทำการ Download Anaconda ลงมาที่เครื่อง Download Anaconda version ล่าสุดที่เป็น python 3.x สำหรับ Anaconda เป็น ตัวช่วยในการจัดการ python เนื่องจาก python มี library เยอะแยะมากเวลาทำการพัฒนาอาจจะใช้ library ไม่เหมือนกันคนละ version Anaconda สามารถจะลอง version ของ python ต่าง ๆ ได้กล่าวคือสามารถ สลับไปใช้งาน python 2 python 3 ได้สะดวก เลือกสลับลง library คนละversion ได้ อีกส่วนหนึ่งที่สำคัญ อย่าลืม Add path ให้กับ Anaconda ด้วย วิธีก็ตอนลงให้เลือกว่าทำการ Add path ให้กับ Anaconda ติดแล้วมันจะเป็นสีแดงๆ สามารถปล่อยไว้ได้เลย

2. 1 install package สำหรับ Anaconda

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้ว ทีนี้จะทำการติดตั้ง Package ให้กับ Anaconda อย่างที่บอกไป Anaconda สามารถสร้าง Environment ใหม่ขึ้นมาได้ ดังนั้นเราจะทำการสร้าง Environment ชื่อ Py3TF เอาไว้ติดตั้ง Tensorflow จะได้ไม่กระทบต่อ Environment หลักของเรา

2.1.1 เริ่มจาก เปิด console ขึ้นมา ทำการ เรียก python ก่อนเพื่อเช็คว่าเป็น version อะไร exit() ถ้าขึ้นแสดงว่าใช้งาน python ได้แล้ว ถ้าไม่ได้ลองลง Anaconda ใหม่ และอย่าลืม Add path ด้วยทำการสร้าง Environment ใหม่ ขึ้นมาโดยใช้คำสั่ง

```
conda create — name <environment-name> <python-packages>
```

```
conda create -name py36tf python=3.6.1 numpy pip scipy mkl
```

ดังนั้นจะสร้าง environment ชื่อ py36tf และมี packages ตามหลังเป็น python version 3.6.1 , numpy, pip, scipy และ mkl สร้าง environment ใหม่ให้กับ Anaconda เมื่อติดตั้งเสร็จแล้วจะกลับมาที่ command หลัก วิธีการ สลับไปใช้ Environment ที่เราสร้างก็ ทำการเรียกคำสั่ง activate py36tf เมื่อหน้าจอที่ทำการติดตั้งเสร็จเรียบร้อยแล้ว สามารถทำการ Activate py36tf ได้

3. Install CUDA and CuDNN

Download CUDA Toolkit ซึ่ง Version ที่ใช้ตอนนี้ จะเป็น version 8 ล่าสุดจะเป็น version 9 ซึ่ง version 9 tensorflow ยังไม่เสถียร เมื่อทำการติดตั้ง Finished แล้วให้ทำการตรวจสอบว่า CUDA ติดตั้งแล้วหรือยัง โดยปกติแล้ว CUDA จะทำการ ADD System variables ให้อัตโนมัติชื่อว่า CUDA_PATH ให้ทำการเช็คที่ Environment Variables หลังจาก ติดตั้งเสร็จ ทำการ Add path ของ CUDA ด้วยโดยใช้ ชื่อ CUDA_PATH ที่ Cuda_toolkit สร้างให้เป็นชื่อย่อของ Path CUDA ให้เลือก ที่ Path ในช่องของ User variable แล้วกด ok เมื่อติดตั้งเสร็จแล้วต่อไปจะทำการลง CuDNN เพื่อใช้สำหรับ ทำ Deep learning เมื่อ Download มาแล้วทำการแตกไฟล์ ไฟล์ที่มีจะประกอบไปด้วย Folder bin , include และ lib ให้ทำการ Copy 3 Folder นี้เอาไปวางไว้ที่ Path ที่ CUDA ติดตั้งอยู่ ถ้าจำที่อยู่ที่ CUDA ติดตั้งไม่ได้ให้ย้อนขึ้นไปดู CUDA_PATH ด้านบน

4. Install Tensorflow-gpu + Keras

ติดตั้ง Tensorflow-gpu และ Keras เมื่อพร้อม ให้ทำการเปลี่ยนไปใช้ Environment py36tf หากจำคำสั่งไม่ได้ พิมพ์ activate py36tf หน้า command prom ทำการติดตั้ง tensorflow-gpu โดยใช้คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pip install --upgrade tensorflow-gpu
```

เมื่อติดตั้งเสร็จเรียบร้อยแล้วให้ทำการติดตั้ง keras ต่อโดย ใช้คำสั่ง

```
pip install --upgrade keras
```

สำหรับการติดตั้งทั้งหมด หลังจากนั้นจะทำการ Test ว่าสามารถใช้งานได้จริง ๆ หรือไม่ ให้พิมพ์ คำสั่ง python เพื่อเข้าใช้งาน python สังเกตว่า Environment py36tf นั้นเป็น python version 3.6.1 ซึ่งคนละ version กับที่ download มา ดังนั้นสามารถใช้ python version ใดก็ได้โดยเพื่อทำการ สร้าง Environment ชื่อใหม่และกำหนด python version สามารถรันโค้ดตามคำสั่งนี้ได้เลยเพื่อทำการทดสอบ keras

```
from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

batch_size = 128
num_classes = 10
epochs = 12

# input image dimensions
img_rows, img_cols = 28, 28

# the data, shuffled and split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

if K.image_data_format() == 'channels_first':
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
input_shape = (1, img_rows, img_cols)
else:
    x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
    x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
    input_shape = (img_rows, img_cols, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                activation='relu',
                input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
optimizer=keras.optimizers.Adadelta(),  
metrics=['accuracy'])  
  
model.fit(x_train, y_train,  
        batch_size=batch_size,  
        epochs=epochs,  
        verbose=1,  
        validation_data=(x_test, y_test))  
score = model.evaluate(x_test, y_test, verbose=0)  
print('Test loss:', score[0])  
print('Test accuracy:', score[1])
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มทดสอบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้