

การจำแนกภาวะถดถอยทางสมองด้วยภาพสแกนสมองโดยใช้การเรียนรู้เชิงลึก

MILD COGNITIVE IMPAIRMENT CLASSIFICATION FROM

MRI SCAN USING DEEP LEARNING



การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์

ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2565

KMITL-2022-SC-M-017-109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MILD COGNITIVE IMPAIRMENT CLASSIFICATION FROM
MRI SCAN USING DEEP LEARNING



WISITSAK POOMIN

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN
DATA SCIENCE AND ANALYTICS

KMITL-DIGITAL ANALYTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2022

KMITL-2022-SC-M-017-109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2022

SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อการค้นคว้าอิสระ	การจำแนกภาวะถดถอยทางสมองด้วยภาพสแกนสมองโดยใช้ การเรียนรู้เชิงลึก
ชื่อนักศึกษา	นายวิสิฐศักดิ์ พุ่มอินทร์
รหัสนักศึกษา	63605077
ปริญญา	วิทยาศาสตร์มหาบัณฑิต (วิทยาการข้อมูลและการวิเคราะห์) ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
พ.ศ.	2565
อาจารย์ที่ปรึกษาการค้นคว้าอิสระ	ผู้ช่วยศาสตราจารย์ ดร.ปัทมา เจริญพร

บทคัดย่อ

งานวิจัยนี้นำเสนอการใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning) โดยการใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network : CNN) เพื่อช่วยรังสีแพทย์นำไปประกอบการตัดสินใจในการจำแนกโรคของผู้ป่วยทั้งหมด 4 ประเภทคือ 1. คนปกติ (CN), 2. ผู้ป่วยภาวะถดถอยทางสมองระยะเริ่มต้น (Early-MCI), 3. ผู้ป่วยภาวะถดถอยทางสมองระยะสุดท้าย (Late-MCI) และผู้ป่วยโรคสมองเสื่อม (AD) จากการนำสถาปัตยกรรมโมเดล อินเซปชันวีที (InceptionV3) และ อินเซปชันเรสเนทวีทู (InceptionResnetV2) มาใช้ในการจำแนก สำหรับตัวอย่างภาพเอ็มอาร์ไอถูกประมวลโดยการปลอกกะโหลก แล้วเปลี่ยนจากภาพ 3 มิติ เป็น 2 มิติ และทำการเพิ่มจำนวนข้อมูล (Data Augmentation) จนได้ชุดข้อมูลที่มีรูปทั้งหมด 6400 รูป และเนื่องจากชุดข้อมูลเป็นข้อมูลไม่สมดุล (Imbalance Data) จึงทำการเพิ่มปริมาณข้อมูล (Over Sampling) เพื่อลดการไม่สมดุลของข้อมูล ส่งผลให้สถาปัตยกรรมโมเดลสามารถทำงานได้แม่นยำและมีประสิทธิภาพมากขึ้น และสถาปัตยกรรมโมเดลจะถูกทดสอบวัดประสิทธิภาพด้วยมาตรวัดต่างๆ โดยสถาปัตยกรรมโมเดลที่ให้ค่าประสิทธิภาพ และความแม่นยำ คือสถาปัตยกรรมโมเดล อินเซปชันเรสเนทวีทู (InceptionResnetV2) ที่มีค่าคะแนนความแม่นยำสมดุล (Balance Accuracy Score) อยู่ที่ 87.80% และค่าสัมประสิทธิ์ความสัมพันธ์ของแมททิว (Matthew's Correlation Coefficient) อยู่ที่ 83.77%

คำสำคัญ: การจำแนก, การเรียนรู้เชิงลึก, โครงข่ายประสาทเทียมแบบคอนโวลูชัน

Independent Study Title	Mild Cognitive Impairment Classification from MRI Scan Using Deep Learning
Students Name	Mr. Wisitsak Poomin
Student ID	63605077
Degree	Master of Science (Data Science and Analytics) KMITL-Digital Analytics and Intelligence Center
Year	2022
Independent Study Advisor	Asst. Prof. Dr. Pattama Charoenporn

Abstract

This research presents the experiments of deep learning techniques in classification by using Convolutional Neural Network (CNN) to assist radiologists in making decisions with brain Magnetic Resonance Imaging (MRI) scan for classifying the disease of all 4 types of patients: 1. Healthy (CN), 2. Patients with early state of Mid Cognitive Impairment (Early-MCI), 3. Patients with late state of Mild Cognitive Impairment (Late-MCI) and Patients with Alzheimer's Disease (AD) by using the InceptionV3 model and InceptionRestnetV2 model. For Image Processing with MRI Scan were Skull Striping, convert 3D images to 2D image and increase sampling of image by Data Augmentation until the datasets has 6400 images. And since the datasets is the imbalance data so we do over sampling with Synthetic Minority Over-sampling Technique (SMOTE) to reduce the imbalance of data. As a result, the model architectures have improved more accuracy and efficiency. The model architectures were benchmarked with the following metrics: Accuracy, Area Under Curve, F1-Score, Confusion Matrix, Balance Accuracy Score and Matthew's Correlation Coefficient. Finally, the best model architecture is InceptionRestnetV2 with Balance Accuracy Score is 87.80 % and Matthew's Correlation Coefficient is 83.77 %

Keywords: Classification, Deep learning, Convolutional Neural Network

กิตติกรรมประกาศ

งานวิจัยเรื่องการจำแนกผู้ป่วยโรคภาวะถดถอยทางสมองจากภาพเอ็มอาร์ไอสมองโดยใช้วิธีการเรียนรู้เชิงลึกนี้สามารถสำเร็จลุล่วงไปด้วยดี ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ.ดร.ปัทมา เจริญพร ที่ได้ให้คำชี้แนะแนวทางการแก้ไขปัญหา ความรู้เพิ่มเติมเกี่ยวกับทางการแพทย์และการประยุกต์ใช้ความรู้ให้เกิดประโยชน์สูงสุดในการพัฒนาปรับปรุงงานวิจัยนี้

ขอขอบพระคุณองค์กร ADNI ที่ได้รวบรวมข้อมูลเอ็มอาร์ไอสมองของผู้ป่วยโรคภาวะถดถอยทางสมองพร้อมคำอธิบายและการปรับปรุงภาพเบื้องต้นให้นักวิจัยจากทั่วโลก

สุดท้ายนี้ ขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งเป็นอีกหนึ่งปัจจัยสำคัญที่ทำให้ได้พบกับคณาจารย์ที่มีศักยภาพ ในการช่วยพัฒนาทักษะและมอบความรู้แก่นักศึกษา



วิสิฐศักดิ์ พุ่มอินทร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ญ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 โรคอัลไซเมอร์ (Alzheimer’s Disease: AD)	3
2.2 ภาวะถดถอยทางสมองหรือภาวะเอ็มซีไอ (Mild Cognitive Impairment: MCI)	4
2.3 การตรวจสมองด้วยคลื่นสนามแม่เหล็ก (Magnetic Resonance Imaging: MRI Brain)	4
2.3.1 T1-Weighted MR Image (T1-MRI)	5
2.3.2 T2-Weighted MR Image (T2-MRI)	5
2.3.3 Proton-Density MR Image (PD-MRI)	5
2.4 การประมวลผลภาพ (Image Processing)	6
2.4.1 การตัดแบ่งภาพ (Image Segmentation)	6
2.4.2 การซ้อนทับภาพ (Image Registration)	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.3 การสร้างภาพ 3 มิติ (3D Image Reconstruction)	8
2.5 การปรับค่าสีที่ไม่สม่ำเสมอ (Intensity Inhomogeneity Correction)	8
2.5.1 B1 Algorithm	8
2.5.1.1 หาจุดที่เหมาะสมของว็อกเซล	9
2.5.1.2 หาค่าความไวของ B1	9
2.5.1.3 การปรับปรุงและรวบรวมภาพสนามแม่เหล็ก	9
2.5.2 N3 Algorithm	10
2.5.2.1 สร้างรูปแบบการกระจายแบบไม่สม่ำเสมอ	10
2.5.2.2 การปรับค่าสีที่ไม่สม่ำเสมอ	10
2.6 การหาค่าจุดแบ่งแยกระหว่างสีขาวและดำ (Thresholding)	11
2.7 การเรียนรู้เชิงลึก (Deep Learning)	12
2.7.1 การเพิ่มประสิทธิภาพและความเร็วอย่างเหมาะสม (Optimizer)	14
2.7.1.1 Momentum Optimizer	14
2.7.1.2 RMSProp	14
2.7.1.3 Adam Optimizer	15
2.7.2 การเกิด Overfitting และ Underfitting ของข้อมูล	15
2.7.3 Data Augmentation	15
2.7.4 Regularization	16
2.7.4.1 Dropout Regularization	16
2.7.4.2 Early Stopping	16
2.7.4.3 Batch Normalization	17
2.7.5 Model Validation	17
2.7.5.1 K-Fold Cross Validation	18
2.7.5.2 Stratified K-Fold Cross-Validation	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.7.5.3 Hold-Out Validation	18
2.7.5.4 Leave-P-Out Cross-Validation	18
2.8 โครงข่ายประสาทเทียมแบบคอนโวลูชันนอล (Convolutional Neural Network: CNN)	19
2.8.1 Convolutional Layer	19
2.8.2 ReLu Layer	20
2.8.3 Pooling Layer	21
2.8.4 Fully connected layer	21
2.9 งานวิจัยที่เกี่ยวข้อง	22
2.9.1 Different multivariate techniques for automated classification of MRI data in Alzheimer's disease and mild cognitive impairment	22
2.9.2 MRI Simulation-Based Evaluation of Image-Processing and Classification Methods	22
2.9.3 Automated Classification of Alzheimer's Disease and Mild Cognitive Impairment using A Single MRI and Deep Neural Networks	23
2.9.4 Combining MRI and CSF Measures for Classification of Alzheimer's Disease and Prediction of Mild Cognitive Impairment Conversion	23
2.9.5 Detection of Alzheimer's Disease on Brain MRI using Inception V3 Network	23
2.9.6 Deep-learning approach with convolutional neural network for classification of maximum intensity projections of dynamic contrast-enhanced breast magnetic resonance imaging	24

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การดำเนินงานวิจัย	25
3.1 ความเป็นมาและความสำคัญ	25
3.2 แนวทางการดำเนินงานวิจัย	25
3.3 การทำความเข้าใจกับข้อมูล (Data Understanding)	26
3.4 การเตรียมข้อมูล (Data Preparation)	27
3.4.1 การลอกกะโหลก (Skull Stripping)	27
3.4.2 การแปลงภาพจาก 3 มิติ เป็น 2 มิติ	30
3.4.3 การเพิ่มปริมาณข้อมูล (Data Augmentation)	31
3.5 การสำรวจข้อมูล (Exploratory Data Analysis)	32
3.6 การสร้างรูปแบบการทำนาย (Modeling)	34
3.7 การประเมินผล (Evaluation)	39
บทที่ 4 ผลการทดลองและอภิปรายผล	40
4.1 ผลการทดลองสถาปัตยกรรม InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	40
4.2 ผลการทดลองสถาปัตยกรรม InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	43
4.3 ผลการทดลองสถาปัตยกรรม InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	46
4.4 ผลการทดลองสถาปัตยกรรม InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	49
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	53
5.1 สรุปผลการดำเนินงานวิจัย	53
5.2 ข้อเสนอแนะ	54
5.2.1 การใช้ชุดข้อมูลที่เป็น 3 มิติในการจำแนก	54

5.2.2 การใช้สีพื้นที่ความร้อน (Heat Map)	54
บรรณานุกรม	55



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงจำนวนรูปภาพ	32
3.2 แสดงจำนวนรูปภาพหลังใช้คำสั่ง SMOTE	33
3.3 ค่า Hyperparameters ของ InceptionV3 และ InceptionRestnetV2	34
4.1 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	42
4.2 การวัดประสิทธิภาพของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	43
4.3 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	45
4.4 การวัดประสิทธิภาพของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	46
4.5 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	48
4.6 การวัดประสิทธิภาพของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	49
4.7 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	51
4.8 การวัดประสิทธิภาพของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	52
4.9 การเปรียบเทียบประสิทธิภาพสถาปัตยกรรมก่อนและการทำ Over Sampling	52

สารบัญรูป

รูปที่	หน้า
2.1 Axial Plane	5
2.2 Coronal Plane	5
2.3 Sagittal Plane	5
2.4 T1-MRI	5
2.5 T2-MRI	5
2.6 PD-MRI	5
2.7 การเรียนเชิงลึก (Deep Learning)	12
2.8 กราฟแสดงค่าระหว่างน้ำหนัก (Weights) และค่าความผิดพลาด (Cost)	13
2.9 ภาพแสดงการเปรียบเทียบก่อน-หลัง Dropout Regularization	16
2.10 กราฟแสดงจุดที่เหมาะสมของการหยุดการเรียนรู้ ของความสัมพันธ์ระหว่างความผิดพลาด และ รอบของการเรียนรู้	16
2.11 กราฟของฟังก์ชันการกระตุ้นแบบ ReLu และ Leaky ReLu	20
2.12 การลดขนาด Feature Map ด้วย Max Pooling	21
2.13 ขั้นตอนการทำงานของโครงข่ายประสาทเทียมแบบคอนโวลูชันนอล	21
3.1 คำสั่งในการแยกไฟล์แต่ละประเภทของผู้ป่วย	26
3.2 คำสั่งการติดตั้ง FSL Brain Extraction Tools	27
3.3 คำสั่งการปลอกกะโหลก	28
3.4 สมอังก่อนทำการปลอกกะโหลก	29
3.5 สมอังกหลังทำการปลอกกะโหลก	29
3.6 คำสั่งการแปลงรูปภาพจาก 3 มิติ เป็น 2 มิติ	30
3.7 ตัวอย่างรูปภาพที่แปลงจาก 3 มิติ เป็น 2 มิติ	30
3.8 คำสั่งการเพิ่มปริมาณข้อมูล (Data Augmentation)	31
3.9 คำสั่งการแสดงผลข้อมูลจากการทำเพิ่มปริมาณข้อมูล	31
3.10 คำสั่งการใช้ SMOTE	33

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 คำสั่งการใช้งาน InceptionV3	34
3.11 คำสั่งการใช้งาน InceptionV3 (ต่อ)	35
3.12 สถาปัตยกรรมโมเดลของ InceptionV3	36
3.13 คำสั่งการใช้งาน InceptionRestnetV2	37
3.14 สถาปัตยกรรมโมเดลของ InceptionRestnetV2	38
3.15 ตัวอย่างการวัดประสิทธิภาพโมเดล	39
4.1 กราฟ Accuracy ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	40
4.2 กราฟ AUC ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	41
4.3 กราฟ Loss ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	41
4.4 กราฟ Confusion Matrix ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling	42
4.5 กราฟ Accuracy ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	43
4.6 กราฟ AUC ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	44
4.7 กราฟ Loss ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	44
4.8 กราฟ Confusion Matrix ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling	45
4.9 กราฟ Accuracy ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	46
4.10 กราฟ AUC ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	47
4.11 กราฟ Loss ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	47
4.12 กราฟ Confusion Matrix ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling	48
4.13 กราฟ Accuracy ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	49

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.14 กราฟ AUC ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	50
4.15 กราฟ Loss ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	50
4.16 กราฟ Confusion Matrix ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling	51



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ภาวะความถดถอยทางสมอง (Mild Cognitive Impairment : MCI) คือ ระยะที่อยู่ระหว่างการเปลี่ยนแปลงของคนปกติและผู้ป่วยโรคสมองเสื่อม (Alzheimer's Disease : AD) ซึ่งผู้ป่วย MCI จะเป็นอาการเริ่มแรกของ ผู้ป่วย AD โดยมีสาเหตุมาจากการเสื่อมสภาพของเซลล์ประสาทในสมอง โดยผู้ป่วยที่เป็น MCI จะมีปัญหาด้านความจำ และสุขภาพจิต ซึ่งจำเป็นต้องได้รับการรักษาและการวินิจฉัยที่ถูกต้อง

การตรวจสมองด้วยคลื่นสนามแม่เหล็ก (Magnetic Resonance Imaging : MRI Brain) เป็นการตรวจโดยอาศัยคลื่นแม่เหล็กไฟฟ้าช่วยในการสร้างภาพ สามารถจำแนกเนื้อเยื่อที่แตกต่างกันได้หลายแบบ สามารถตรวจความผิดปกติในระยะเริ่มแรกได้ ใช้ในการวินิจฉัยโรคได้ เช่น เนื้องอกในสมอง และ Chronic Subdural Hematomas เป็นต้น และยังช่วยให้รังสีแพทย์ทราบว่ามีสมองส่วนไหนเหี่ยวฝ่อ นอกจากนี้การใช้การเรียนรู้เชิงลึกและการประมวลผลภาพจะช่วยให้ทำนายได้ว่าผู้ป่วย MCI รายใดจะเป็น AD

ในประเทศไทยมีแพทย์เฉพาะทางด้านรังสีแพทย์และ MRI (Magnetic Resonance Imaging) ด้านสมองค่อนข้างน้อย ส่งผลให้ผู้ที่มีความชำนาญด้านการอ่านภาพถ่ายสมองส่วนใหญ่จะมีอายุค่อนข้างมาก โดยส่วนใหญ่ผู้ที่อายุ 40 ปีขึ้นไปจะเริ่มมีอาการความผิดปกติของสายตาหรือสายตาเริ่มเสื่อมภาพ ส่งผลให้ผู้ที่มีความชำนาญการด้านการอ่านภาพถ่ายสมองจำเป็นต้องมีสายตาที่ดีเพื่อการวินิจฉัยที่แม่นยำ ดังนั้นงานวิจัยนี้จะศึกษาเรื่องการใช้ความรู้ของเครื่องมาใช้ในการประมวลผลภาพถ่ายสมองด้วยเทคนิคการประมวลผลภาพ (Image Processing) และการเรียนรู้เชิงลึก (Deep Learning) เพื่อใช้ความรู้และความชำนาญของแพทย์เฉพาะทางด้านรังสีแพทย์ ที่มากประสบการณ์ มาสร้างเป็นความรู้ของเครื่อง (Machine Learning) เพื่อเป็นเครื่องมือช่วยในการตัดสินใจ สำหรับรังสีแพทย์ที่สายตาเริ่มเสื่อมสภาพ และรังสีแพทย์ที่ยังขาดประสบการณ์ ให้สามารถวินิจฉัยให้แม่นยำมากขึ้น

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อศึกษาการวินิจฉัยของภาวะถดถอยทางสมองจาก MRI
2. เพื่อศึกษาเปรียบเทียบความแม่นยำก่อนและหลังเพิ่มปริมาณข้อมูล
3. เพื่อเปรียบเทียบประสิทธิภาพของสถาปัตยกรรมโมเดล

1.3 ขอบเขตของงานวิจัย

1. ชุดข้อมูลในการทดลองงานวิจัยประกอบไปด้วย ภาพสแกนสมองของผู้ป่วยโรคสมองเสื่อม ผู้ป่วยภาวะถดถอยทางสมองระยะเริ่มต้น ผู้ป่วยภาวะถดถอยทางสมองระยะสุดท้าย และผู้ที่มีภาวะปกติ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเป็นเครื่องมือให้รังสีแพทย์ใช้ในการประกอบการตัดสินใจในการวินิจฉัย
2. ได้เปรียบเทียบประสิทธิภาพของโมเดลเพื่อศึกษาต่อยอดการทดลองในอนาคต

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาวิธีการปรับปรุงภาพสแกนสมอง
2. ศึกษาวิธีการจำแนกข้อมูลด้วยรูปภาพโดยวิธีการเรียนรู้เชิงลึก
3. รวบรวมข้อสงสัยและปรึกษาอาจารย์แพทย์
4. เก็บข้อมูลภาพสแกนสมอง สำหรับการฝึกสอน
5. ทำการทดลองปรับปรุงภาพสแกนสมอง
6. ทำการทดลองการจำแนกข้อมูลด้วยวิธีการเรียนรู้เชิงลึก
7. ประเมินผลและปรับปรุงแก้ไขโมเดลการเรียนรู้เชิงลึก
8. วิเคราะห์และสรุปผลการทดลอง
9. จัดทำเอกสารประกอบปัญหาพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องซึ่งประกอบไปด้วย ภาวะถดถอยทางสมองหรือภาวะเอ็มซีไอ (Mild Cognitive Impairment : MCI), โรคอัลไซเมอร์ (Alzheimer's Disease : AD), การตรวจร่างกายด้วยคลื่นแม่เหล็กไฟฟ้า (MRI), การประมวลผลภาพ (Image Processing), การกำจัดสัญญาณรบกวน (Denoising), การปรับค่าสีที่ไม่สม่ำเสมอ (Intensity Inhomogeneity Correction), การปรับภาพเป็นสีขาวดำ (Thresholding), การปรับเปลี่ยนรูปร่าง (Morphological Operation), การซ้อนทับภาพ (Registration), การจำแนกประเภท (Classification), การเรียนรู้เชิงลึก (Deep Learning), โครงข่ายประสาทเทียมแบบคอนโวลูชันนอล (Convolutional Neural Network), และ งานวิจัยที่เกี่ยวข้อง

2.1 โรคอัลไซเมอร์ (Alzheimer's Disease : AD)

โรคอัลไซเมอร์หรือโรคสมองเสื่อมเกิดจากความผิดปกติของเซลล์สมองทำลาย มีหลายสาเหตุ เช่น การเสื่อมที่เกิดขึ้นตามวัย พันธุกรรม อุบัติเหตุทางสมอง โรคภาวะถดถอยทางสมองโรคการติดเชื้อของสมอง โรคทางกายที่มีผลกระทบต่อเซลล์สมอง เป็นต้น ปัจจุบันไม่ได้เริ่มเกิดในผู้ป่วยอายุ 65 ปี เพราะจากการศึกษาพบว่าการเกิดโรคจะเริ่มตั้งแต่ช่วงอายุ 40 – 65 ปี ซึ่งในเมืองไทยตอนนี้มีสถิติผู้ป่วยด้วยโรคอัลไซเมอร์ถึงเกือบ 2 ล้านคน

โรคอัลไซเมอร์มีระยะเวลาก่อนโรคนาน 15 – 20 ปีกว่าจะมีอาการสมองเสื่อมชัดเจน การแสดงอาการของโรคจะเป็นไปอย่างช้า ๆ เริ่มต้นจากไม่มีความผิดปกติเรื่องความจำ และเริ่มมีอาการความจำถดถอย ซึ่งการที่ผู้ป่วยเริ่มมีอาการความจำถดถอยเกิดขึ้นเป็นผลมาจากการสะสมของสารเบต้าอามีลอยด์ที่ทำลายเซลล์สมองมาแล้ว 10 – 15 ปี ต่อมาผู้ป่วยจึงจะมีอาการมากขึ้นเรื่อย ๆ จนมีอาการสมองเสื่อมชัดเจน ทำให้การวินิจฉัยและรักษาทำได้ไม่ทันทั่วถึง ซึ่งส่วนหนึ่งมาจากความเพิกเฉยคิดว่าผิดปกติทางความจำเล็กน้อยไม่ได้เป็นอาการเริ่มต้นของโรคสมองเสื่อมและเข้าใจผิดที่คิดว่าโรคอัลไซเมอร์เป็นโรคที่เกิดในผู้สูงอายุ ไม่สามารถป้องกันหรือรักษาให้หายขาดได้และต้องทานยาอย่างต่อเนื่องไปตลอดชีวิต

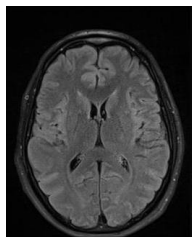
2.2 ภาวะถดถอยทางสมองหรือภาวะเอ็มซีไอ (Mild Cognitive Impairment: MCI)

ภาวะถดถอยทางสมอง หรือ Mild Cognitive impairment (MCI) เป็นภาวะสูญเสียความสามารถของสมองในการจำ มีสาเหตุมาจากการเสื่อมของเซลล์ประสาทในสมองซึ่งแตกต่างจากการเสื่อมในภาวะปกติตามอายุขัย ปัจจัยที่ก่อให้เกิดการเสื่อมของเซลล์ประสาทที่มากกว่าภาวะปกติ ได้แก่ อุบัติเหตุที่เกี่ยวข้องกับสมอง เนื้องอก การขาดวิตามินและสารอาหาร การติดเชื้อ หรือการใช้ยาบางชนิดที่มากเกินไป

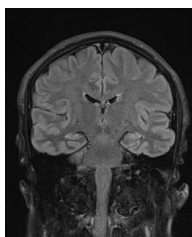
ภาวะถดถอยทางสมอง เป็นอาการเบื้องต้นก่อนจะลุกลามกลายเป็นโรคสมองเสื่อมชนิดอัลไซเมอร์ (Alzheimer's Disease : AD) โดยถ้าสามารถสังเกตภาวะ MCI ได้ก่อนก็จะช่วยชะลอหรือป้องกันการเกิดโรคสมองเสื่อมชนิดอัลไซเมอร์ (Alzheimer's Disease : AD) ได้ ดังนั้นการเฝ้าระวังอาการเบื้องต้นจึงเป็นการช่วยชะลอการเกิดโรค รวมถึงลดความรุนแรงของการเป็นโรคสมองเสื่อมชนิดอัลไซเมอร์ได้

2.3 การตรวจสมองด้วยคลื่นสนามแม่เหล็ก (Magnetic Resonance Imaging: MRI Brain)

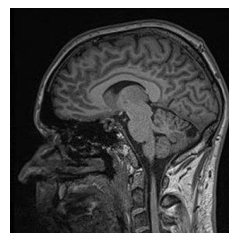
การตรวจสมองด้วยคลื่นสนามแม่เหล็ก (Magnetic Resonance Imaging : MRI Brain) คือ การตรวจสมอง โดยใช้เครื่องตรวจไอออนถ่ายคลื่นความถี่วิทยุไปยังผู้ป่วยที่อยู่ในสนามแม่เหล็กไฟฟ้าแรงสูงและให้พลังงาน เหล่านี้สะท้อนกลับมา จากนั้นนำข้อมูลไปประมวลผลและสร้างเป็นภาพแสดงรายละเอียดอวัยวะภายในร่างกายแบบ 3 มิติ ประกอบไปด้วยแนวขวาง (Axial Plane) แนวยาว (Coronal Plane) และ แนวเฉียง (Sagittal Plane) ซึ่งจัดเป็นภาพที่มีความละเอียดสูง จึงเป็นที่นิยมใช้ในการตรวจดูรายละเอียดสมอง ไขสันหลัง หัวใจ เนื้องอก เป็นต้น



รูปที่ 2.1 Axial Plane



รูปที่ 2.2 Coronal Plane



รูปที่ 2.3 Sagittal Plane

หมายเหตุ. จาก <https://www.suyogadhav.com/projects/2019/01/19/brainy/>

ภาพสแกนสมองสำหรับการวินิจฉัยโรคสมองเสื่อมสามารถแบ่งออกได้เป็น 3 ประเภทหลัก คือ

2.3.1 T1-Weighted MR Image (T1-MRI)

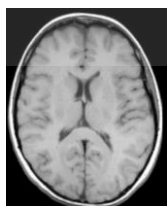
T1-MRI คือ ภาพสแกนสมองที่สามารถแบ่งขอบเขตของเนื้อเยื่อได้อย่างชัดเจน โดยน้ำจะมีสีดำและไขมันมีสีขาว ซึ่งจัดเป็นประเภทที่นิยมใช้ในการตรวจจ้อวัยวะของสมองอย่างละเอียด รวมถึงการเปลี่ยนแปลงของหลอดเลือดที่อาจเกิดการฉีกขาด

2.3.2 T2-Weighted MR Image (T2-MRI)

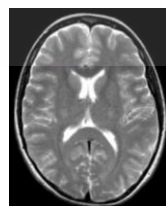
T2-MRI คือ ภาพสแกนสมองที่มีสีตรงกันข้ามกับ T1-MRI โดยน้ำจะมีสีขาวและไขมันมีสีดำ ซึ่ง จัดเป็นประเภทที่นิยมใช้ในการตรวจความผิดปกติของปริมาณน้ำในสมอง

2.3.3 Proton-Density MR Image (PD-MRI)

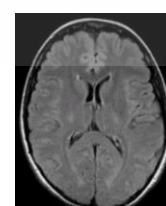
PD-MRI คือ ภาพสแกนสมองที่ นิยมใช้เมื่อต้องการความแตกต่างของสี (Contrast)ระหว่างส่วนที่เป็นสีเทา (Gray Matter) กับส่วนที่เป็นสีขาว (White Matter) สูงและ ต้องการความแตกต่างของสี (Contrast) ระหว่างส่วนที่เป็นเนื้อสมองกับน้ำไขมันสันหลังดำ โดย ทั้งน้ำและไขมันจะมีสีขาวทั้งคู่



รูปที่ 2.4 T1-MRI



รูปที่ 2.5 T2-MRI



รูปที่ 2.6 PD-MRI

หมายเหตุ. จาก <https://sites.google.com/site/postgraduatetraining/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ(Image Processing) เป็นกระบวนการจัดการและวิเคราะห์รูปภาพให้เป็นข้อมูลในแบบดิจิทัล โดยใช้คอมพิวเตอร์ในการประมวลผลโดยวิธีการต่าง ๆ เพื่อให้ได้ภาพที่มีคุณสมบัติตามความต้องการทั้งในเชิงคุณภาพและปริมาณ มีหลากหลายรูปแบบ เช่น การปรับปรุงคุณภาพของภาพ (Image Enhancement) การปรับเปลี่ยนหรือแปลงรูปภาพทั้งขนาดและรูปร่าง (Image Transformation) การกรองภาพหรือการกำจัดสัญญาณรบกวนออกจากภาพ (Image Filters) การซ้อนทับภาพ (Image Registration) การคืนสภาพของภาพ (Image Restoration) การตัดแบ่งภาพหรือคัดเลือกส่วนที่ต้องการและการหาขอบภาพในวัตถุ (Image Segmentation and Edge Detection) การบีบอัดภาพ (Image Compression) การสร้างภาพ 3 มิติ (3D Image Reconstruction) เป็นต้น

การประมวลผลภาพทางการแพทย์ (Medical Image Processing) ซึ่งมีประโยชน์หลากหลายด้าน แพทย์ผู้เชี่ยวชาญสามารถวิเคราะห์ภาพจำนวนมากได้อย่างรวดเร็ว และเป็นการเพิ่มประสิทธิภาพในการวินิจฉัยโรคได้ดีขึ้น เช่น บริเวณของการเกิดมะเร็ง จะเห็นภาพของมะเร็งที่ชัดเจนยิ่งขึ้น ทำให้การตรวจวิเคราะห์เป็นไปอย่างรวดเร็วและมีคุณภาพ นอกจากนี้ยังมีการใช้ในระบบการรักษาด้วยเครื่องเอ็มอาร์ไอ (Magnetic Resonance Imaging : MRI) ซึ่งสามารถถ่ายภาพส่วนที่เป็นเนื้อเยื่อได้ ส่งผลให้ง่ายต่อการนำไปวิเคราะห์ และสามารถลดระยะเวลาและเพิ่มความแม่นยำในการวินิจฉัยโรคสำหรับแพทย์รังสี นอกจากนี้สามารถนำเทคนิคการประมวลผลภาพ (Image Processing Techniques) ร่วมกับการเรียนรู้เชิงลึก (Deep Learning) จะส่งผลให้การทำงานมีความแม่นยำมากขึ้น

2.4.1 การตัดแบ่งภาพ (Image Segmentation)

เป็นวิธีการแบ่งส่วนใดส่วนหนึ่งของภาพที่เราสนใจออกมาจากภาพที่เราต้องการ ซึ่งการแบ่งส่วนภาพนี้ โดยส่วนใหญ่แล้วจะเป็นขั้นตอนเบื้องต้นและสำคัญอย่างมากของการประมวลผลภาพทางการแพทย์ เนื่องจากภาพทางการแพทย์ที่ได้จากเครื่องถ่ายภาพแบบต่าง ๆ นั้น โดยปกติมักจะมีองค์ประกอบอื่น ๆ ที่อยู่ใกล้เคียงกับอวัยวะที่ทำถ่ายภาพมา เช่น เนื้อเยื่อ กระดูก อวัยวะข้างเคียง หรือแม้กระทั่งสัญญาณรบกวน (Noise) ที่ขึ้นในขณะที่ถ่ายภาพ ด้วยเหตุนี้ การวิเคราะห์เฉพาะอวัยวะที่ต้องการจึงจำเป็นต้องใช้การแบ่งส่วนภาพมาทำหน้าที่ตัดแยกส่วนที่เราต้องการออกมา ตัวอย่างเช่น การแบ่งส่วนเนื้อสมองจากภาพสมอง การแบ่งส่วนภาพหัวใจห้องล่างซ้ายจากภาพหัวใจ MRI การแบ่งส่วนเฉพาะเส้น

โลหิต การแบ่งส่วนข้อกระดูกสันหลังจากภาพลำกระดูกสันหลัง หรือ การแบ่งส่วนของทารกจากภาพอัลตราซาวด์ เป็นต้น การแบ่งส่วนภาพทางการแพทย์มีทั้งการแบ่งส่วนภาพแบบ 2 มิติ และ 3 มิติ ขึ้นอยู่กับความจำเป็นและวัตถุประสงค์ของการนำไปวิเคราะห์ โดยวิธีการแบ่งส่วนภาพที่กำลังได้รับความนิยมในงานวิจัยเกี่ยวกับภาพทางการแพทย์ ได้แก่ แอ็กทีฟคอนทัวร์ (Active Contour) และ แอ็กทีฟเซอร์เฟซ (Active Surface) เป็นต้น

2.4.2 การซ้อนทับภาพ (Image Registration)

การซ้อนทับภาพ (Image Registration) เป็นวิธีการนำข้อมูลของสองภาพหรือมากกว่ามารวมกันเพื่อให้เกิดภาพใหม่ที่มีข้อมูลภาพสมบูรณ์มากขึ้น โดยภาพใหม่ที่ได้นี้ จะเป็นการรวมตัวกันของข้อมูลหรือรายละเอียดในแต่ละภาพที่นำมาผสมผสานกัน มีวัตถุประสงค์เพื่อให้ได้ภาพที่มีรายละเอียดและข้อมูลที่เพียงพอสำหรับการนำไปใช้งาน หรือการนำภาพไปวิเคราะห์ โดยส่วนใหญ่แล้วภาพที่จะนำมาซ้อนทับกันนั้น อาจเป็นภาพถ่ายของอวัยวะเดียวกัน ที่ถ่ายต่างเวลากัน ต่างมุมมองกัน หรือ ใช้เทคนิคในการถ่ายภาพที่แตกต่างกัน เป็นต้น และการนำวิธีการซ้อนทับภาพมาใช้กับภาพทางการแพทย์ มีประโยชน์ในหลาย ๆ ด้าน ตัวอย่างเช่น การตรวจติดตาม หรือหาความผิดปกติของอวัยวะต่าง ๆ ทำได้โดยการนำภาพถ่ายของอวัยวะที่ต้องการตรวจ ที่ได้ถ่ายไว้ในอดีต มาทำการซ้อนทับกับภาพถ่ายของอวัยวะเดียวกันที่ถ่ายไว้ในปัจจุบัน โดยทำให้ตำแหน่งของอวัยวะต่าง ๆ ของทั้งสองภาพตรงกัน ซึ่งการทำในลักษณะนี้ จะทำให้เห็นถึงความเปลี่ยนแปลงที่เกิดขึ้นของอวัยวะนั้น ว่ามีการเปลี่ยนแปลงไปอย่างไรบ้างในช่วงเวลานั้น มีแนวโน้มที่จะเป็นอย่างไรต่อไป มีสิ่งผิดปกติเกิดขึ้นหรือไม่ มีอวัยวะที่โตขึ้นผิดปกติหรือไม่ เป็นต้น การนำภาพทางการแพทย์ที่ใช้เทคนิคในการถ่ายภาพแตกต่างกัน มาทำการซ้อนทับภาพ เป็นอีกหนึ่งประโยชน์ของวิธีการนี้ เนื่องจากภาพทางการแพทย์ที่ถ่ายโดยใช้เทคนิคการถ่ายภาพเพียงแบบเดียว อาจจะทำให้ได้ข้อมูลไม่ครบถ้วนตามที่ต้องการ จึงจำเป็นต้องใช้เทคนิคการถ่ายภาพหลาย ๆ แบบ เพื่อให้ได้ข้อมูล รายละเอียดของอวัยวะ หรือองค์ประกอบรอบข้างอื่น ๆ ของอวัยวะนั้น ๆ เพิ่มมากขึ้น ซึ่งจะเป็ประโยชน์อย่างมากในการนำภาพไปวิเคราะห์

2.4.3 การสร้างภาพ 3 มิติ (3D Image Reconstruction)

การสร้างภาพ 3 มิติ (3D Image Reconstruction) คือการวิเคราะห์ภาพทางการแพทย์ โดยใช้ภาพ 3 มิติ กำลังได้รับความนิยมอย่างมากในปัจจุบัน เนื่องจากภาพ 3 มิติ สามารถแสดงให้เห็นถึงภาพรวมหรือรายละเอียดในมุมมองต่าง ๆ ของอวัยวะได้ จึงมีประโยชน์อย่างมาก ในการวิเคราะห์ภาพทางการแพทย์ โดยอวัยวะหรือส่วนของร่างกายที่ได้มีการวิเคราะห์ในรูปแบบ 3 มิติ ตัวอย่างเช่น สมอง หัวใจ กระดูก ฟัน และขากรรไกร เป็นต้น

ภาพ 3 มิติสำหรับภาพทางการแพทย์นั้น มักสร้างมาจากภาพ 2 มิติหลาย ๆ ภาพ ทำได้ โดยการนำภาพเหล่านั้น มาผ่านกระบวนการประมวลผลภาพ เช่น การแบ่งส่วนภาพ เป็นต้น เพื่อให้ได้รายละเอียด ส่วนประกอบต่าง ๆ หรือข้อมูลที่จำเป็นของอวัยวะที่ต้องการ จากนั้น นำมาประกอบกันเพื่อขึ้นรูปเป็นภาพ 3 มิติ ซึ่งภาพ 3 มิติที่ได้นี้ จะมีลักษณะหรือรูปร่างที่ เหมือนกับอวัยวะจริงเพียงใด ขึ้นอยู่กับข้อมูลของภาพ 2 มิติที่นำมาประมวลผล ถ้าภาพ 2 มิติที่ ได้จากเครื่องถ่ายภาพมีภาพจำนวนมากเพียงพอ ถ่ายในทุกส่วนตัดอย่างละเอียด หรือ ได้ถ่ายไว้ในหลายมุมมอง ก็ยิ่งทำให้ภาพ 3 มิติที่ได้ใกล้เคียงกับความเป็นจริงมากขึ้น

2.5 การปรับค่าสีที่ไม่สม่ำเสมอ (Intensity Inhomogeneity Correction)

การปรับค่าสีที่ไม่สม่ำเสมอ (Intensity Inhomogeneity Correction) คือ การแก้ไขปัญหา พิกเซลที่เป็นสัญญาณรบกวนประเภทที่มีค่าสีกลมกลืนคล้ายกับพิกเซลรอบข้างสูง (Bias Field) จนคล้าย กับการแรงแสง ผลให้เป็นปัญหาที่มีหลายชื่อเรียก เช่น RF inhomogeneity Shading artifact หรือ B0 intensity non- uniformity เป็นต้น โดยปัญหาค่าสีที่ไม่สม่ำเสมอเหล่านี้จะส่งผลกระทบต่อ ประสิทธิภาพของการประมวลผลภาพ

2.5.1 B1 Algorithm

B1 Algorithm คือ อัลกอริทึมที่ใช้ในการปรับค่าความไม่สม่ำเสมอของ B1 ซึ่งมีสาเหตุ มาจากความคลาดเคลื่อนในขั้นตอนการรับส่งสัญญาณของขดลวด โดยการใช้อัลกอริทึม B1 นี้มี จุดประสงค์เพื่อเพิ่มอัตราส่วนระหว่างสัญญาณต่อเสียงรบกวนให้มากที่สุด (SNR) ซึ่งมีลำดับ การทำงาน ดังนี้

2.5.1.1 หาจุดที่เหมาะสมของว็อกเซล

ว็อกเซลที่เหมาะสม คือ ว็อกเซลที่มีค่าความถี่ของสัญญาณที่เพียงพอ ซึ่งความถี่ดังกล่าวสามารถหาได้จากค่า Z-score ของภาพสนามแม่เหล็กตั้งต้น โดยมีวิธีการคำนวณเบื้องหลัง คือ การใช้กระบวนการคอนโวลูชัน ซึ่งเป็นการกำหนดเคอร์เนลขนาด 5×5 และนำเคอร์เนลไปเปรียบเทียบกับภาพสนามแม่เหล็กที่ละส่วนจนครบทั้งภาพ โดยถ้าหากว่าค่า Z-score มีค่าต่ำกว่า 1.0 จะถือว่าเป็นส่วนที่ไม่เหมาะสม เพราะมีความเหมือนกับสัญญาณรบกวนมากเกินไป

2.5.1.2 หาค่าความไวของ B1

ค่าความไวของ B1 เกิดจากการคำนวณผลรวมของสัญญาณทจริง $t(x)$ ซึ่งเดิมแล้วเป็นสัญญาณที่ผสมกันระหว่างความไวของ B1 ($b(x)$) และ สัญญาณรบกวน ($n(x)$) โดยเขียนมีรูปแบบสมการ ดังนี้

$$r_c(x) = t(x)b_c(x) + n_c(x) \quad (2.1)$$

นอกจากนี้สามารถปรับรูปแบบสมการ เพื่อให้ง่ายและเหมาะสมต่อการคำนวณมากขึ้น โดยใช้ การหาจุดที่เหมาะสมของว็อกเซล เนื่องจากอัลกอริทึม B1 จะผ่านขั้นตอนการหาจุดที่เหมาะสมของว็อกเซลก่อนเสมอ ส่งผลให้สามารถจัดรูปแบบของสมการใหม่ได้ ดังนี้

$$f(r_c(x)) = B(x) + e(x) \quad (2.2)$$

2.5.1.3 การปรับปรุงและรวบรวมภาพสนามแม่เหล็ก

การหาค่าสัญญาณจริง $t(x)$ เกิดจากการนำว็อกเซลแต่ละว็อกเซลมาหาค่าเฉลี่ยแบบถ่วงน้ำหนัก โดยสามารถเขียนรูปแบบสมการได้ดังนี้

$$\hat{t}(x) = \sum_c \omega_c(x)\alpha_c(x)r_c(x) \quad (2.3)$$

2.5.2 N3 Algorithm

N3 Algorithm คือ อัลกอริทึมที่นิยมใช้ในการปรับค่าสีที่ไม่สม่ำเสมอ ด้วยวิธีการแบบทำซ้ำเพื่อจำแนกแหว่งพิกเซลที่เป็นสัญญาณรบกวน (Bias Field) กับ พิกเซลที่เป็นค่าสีของเนื้อสมองจริง (True Tissue) โดยมีแนวคิดการทำงานดังนี้

2.5.2.1 สร้างรูปแบบการกระจายแบบไม่สม่ำเสมอ

รูปแบบการกระจายแบบไม่สม่ำเสมอ (Nonuniformity Model) คือ รูปแบบที่บ่งบอกถึงความไม่สม่ำเสมอของค่าสีพิกเซล โดยสามารถเขียนรูปแบบสมการได้ ดังนี้

$$v(x) = u(x)f(x) + n(x) \quad (2.4)$$

เมื่อ

v คือ ค่าของสัญญาณที่วัดได้ (Measured signal)

u คือ ค่าสัญญาณที่ต้องการ (True signal)

f คือ ค่าสีของพิกเซลที่กลมกลืนกับพิกเซลรอบข้าง (Bias field)

n คือ ค่าสัญญาณรบกวนเกาส์ (White gaussian noise)

x คือ ตำแหน่งของการกระจาย

2.5.2.2 การปรับค่าสีที่ไม่สม่ำเสมอ

ค่าสีที่ไม่สม่ำเสมอ (f) คือค่าสีของพิกเซลที่มีความกลมกลืนกับพิกเซลรอบข้างสูง และมีการเปลี่ยนแปลง ของค่าสีจากสีหนึ่งไปอีกสีหนึ่งอย่างช้าๆ ซึ่งเป็นค่าสีที่เราไม่ต้องการ ดังนั้นผลลัพธ์ที่ต้องการ คือ ภาพมีค่าสีที่ไม่สม่ำเสมอที่น้อยที่สุด (f) และมีสัญญาณที่ต้องการมากที่สุด (u) แต่ภาพตั้งต้นที่นำเข้ามานั้นยังไม่สามารถระบุขอบเขตของค่าสีที่ไม่สม่ำเสมอ (f) ได้ ส่งผลให้ไม่สามารถระบุพิกเซลที่ต้องการปรับ ให้เป็นพิกเซลที่ต้องการ(u)ได้ ซึ่งสามารถแก้ปัญหานี้ได้ด้วยการใช้คุณสมบัติของค่าสีที่ไม่สม่ำเสมอข้างต้น (f) จากนั้นทดลองหาค่าที่เป็นไปได้เพื่อหาค่าที่ทำให้ค่าสัญญาณที่เราต้องการมีค่าสูงที่สุด(u) ซึ่งส่งผลให้เกิดปัญหาคือ จำนวนค่าที่ใช้ในการทดลองมากเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงแก้ปัญหาดังกล่าวด้วยการปิดการกระจายตัวของสัญญาณที่วัดได้ทั้งหมด (v) ก่อน หลังจากนั้นทดลองหาค่าที่ทำให้ค่าสัญญาณที่เราต้องการมีค่าสูงที่สุด (u)

2.6 การหาค่าจุดแบ่งแยกระหว่างสีขาวและดำ (Thresholding)

การหาค่าจุดแบ่งแยกระหว่างสีขาวและดำ (Thresholding) เป็นการพิจารณาว่าจุดภาพใดควรเป็นจุดขาวหรือจุดดำ ซึ่งทำได้โดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่ง เรียกว่าค่าขีดแบ่ง ซึ่งเป็นค่าความเข้มแสงค่าหนึ่งที่ใช้แยกแยะประเภทของจุดภาพ เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุและพื้นหลัง ในขั้นตอนนี้ทำการตัดพื้นหลังออกจากพื้นหน้า โดยใช้วิธี Threshold ที่มีระดับความเข้มอยู่ระหว่างกลุ่มทั้งสองของ Histogram ซึ่งค่า Threshold ที่ได้จะอยู่ระหว่าง 0-255 เท่านั้น Threshold จะถูกนำไปเพื่อเปรียบเทียบค่าของแต่ละพิกเซล หากค่า $g(x, y)$ น้อยกว่า Threshold จุดพิกเซลนั้นจะถูกปรับให้เป็นสีดำหรือส่วนของวัตถุ และหากค่า $g(x, y)$ มากกว่าหรือเท่ากับ Threshold จุดพิกเซลนั้นจะถูกปรับให้เป็นสีขาว โดยมีรูปแบบสมการดังนี้

$$g(x, y) = \begin{cases} 1, & f(x, y) < T \\ 0, & f(x, y) \geq T \end{cases} \quad (2.5)$$

เมื่อ

$g(x, y)$ คือ พิกเซลใหม่

$f(x, y)$ คือ พิกเซลเดิม

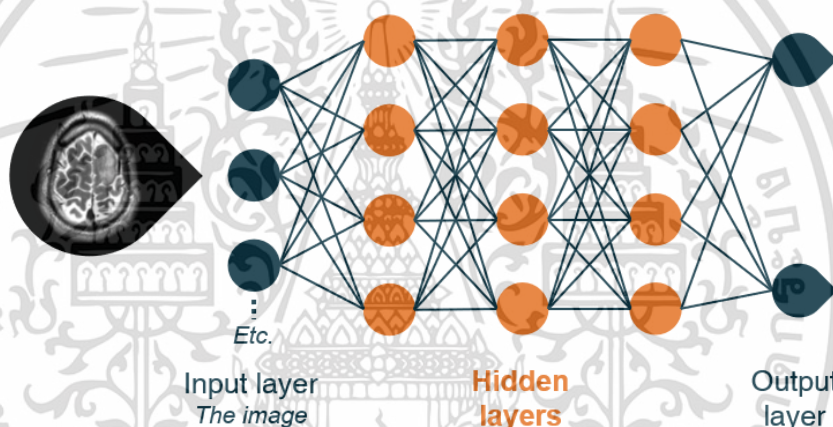
T คือ ค่าของจุดแบ่งแยก

1 คือ วัตถุที่สนใจ

2 คือ พื้นหลัง

2.7 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก (Deep Learning) เป็นส่วนหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นอัลกอริทึมที่ใช้สำหรับการเรียนรู้ที่สามารถทำให้เครื่องจักรสามารถตัดสินใจได้เช่นเดียวกับมนุษย์ โดยการเรียนรู้ของเครื่องเป็นการประยุกต์ใช้ความรู้ทางด้านสถิติ ในการวิเคราะห์ข้อมูลและสร้างแบบจำลองสำหรับทำนายผลลัพธ์จากข้อมูล โดยการเรียนรู้เชิงลึก (Deep Learning) จะประกอบไปด้วย 3 ชั้นหลัก คือ ชั้นป้อนข้อมูลเข้า (Input Layer) ชั้นกลางหรือชั้นซ่อน (Hidden Layer) และชั้นแสดงผลลัพธ์ (Output Layer) โดยชั้น Hidden Layer จะต้องมีตั้งแต่ 2 ชั้นขึ้นไปจึงถือว่าเป็น Deep Neural Network ดังรูปที่ 2.7



รูปที่ 2.7 การเรียนรู้เชิงลึก (Deep Learning)

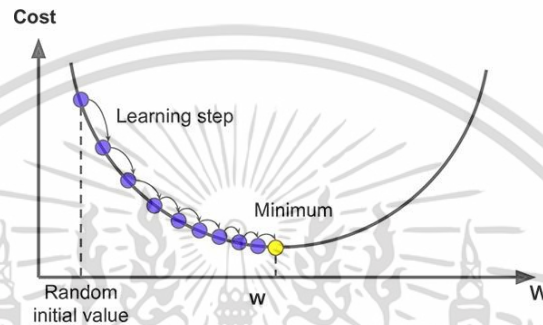
หมายเหตุ. จาก <https://www.quantib.com/blog/how-does-deep-learning-work-in-radiology>

หลักการทำงานจะเริ่มจากการป้อนข้อมูลจากด้านหน้าไปด้านหลัง (Feedforward Neural Network) ที่ Input Layer ส่งผลให้เกิดการหาจุดเด่นของข้อมูล (Pattern Recognition) ที่ Hidden Layer เพื่อหาผลลัพธ์ หลังจากนั้นจะแสดงผลลัพธ์ที่ Output Layer ในแต่ละรอบ (Epoch) ซึ่งการแสดงผลลัพธ์ในช่วงแรกจะมีค่าความแม่นยำ (Precision) ที่ต่ำ ส่งผลให้ต้องมีการป้อนข้อมูลและทำการเรียนรู้โดยป้อนค่าผลลัพธ์ที่ได้แบบย้อนกลับ (Backpropagation Neural Network) และปรับค่า Parameters เพื่อเพิ่มความแม่นยำและให้เหมาะสมกับเนื้องาน โดยจำนวนรอบ (Epoch) จะเกิดขึ้นหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งตามรอบที่กำหนดไว้ เพื่อให้เกิดความผิดพลาดที่น้อยที่สุด เมื่อเทรนโมเดลแล้วจะมีการประเมินผลโมเดล (Model Evaluation) เพื่อวัดประสิทธิภาพและความแม่นยำของโมเดล

การเรียนรู้ของการเรียนรู้เชิงลึกเกิดจากการปรับค่าน้ำหนักภายในโครงข่าย โดยคำนวณจากค่าความผิดพลาดและค่าความถูกต้องที่ตัวโมเดลทำนายผลลัพธ์ออกมา เป็นการปรับค่าน้ำหนักเพื่อให้ค่าความผิดพลาด (Cost) มีค่าน้อยที่สุดโดย ดังรูปที่ 2.8



รูปที่ 2.8 กราฟแสดงค่าระหว่างน้ำหนัก (Weights) และค่าความผิดพลาด (Cost)

หมายเหตุ. จาก <https://saugatbhattarai.com.np/what-is-gradient-descent-in-machine-learning/gradient-descent-2/>

การคำนวณ Gradient Descent Optimizer มีหลักการคือ หามุม Theta ของค่าความผิดพลาด หรือ Loss function หรือ Cost หรือ Error แล้วนำมาลบค่าของน้ำหนักล่าสุด เพื่อให้ค่าความผิดพลาดน้อยที่สุด มีรูปแบบสมการ ดังนี้

Weight Update

$$w = w - \eta \frac{\partial E}{\partial w} \quad (2.6)$$

Loss Function : Mean Squared Error

$$E = \frac{1}{n} \sum_k (y_k - a_k)^2 \quad (2.7)$$

Loss Function : Cross Entropy Error With Softmax Function

$$E = - \sum_k y_k \ln a_k \quad (2.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gradient Descent จะทำการปรับค่าน้ำหนักทุกครั้งหลังจากเทรนโมเดล 1 รอบ ก่อให้เกิดอัตราการเรียนรู้ (Learning Rate) ที่มากเกินไป ส่งผลให้การปรับให้ Loss Function มีค่าเข้าใกล้ศูนย์ช้าเกินไป และตัวโมเดลไม่สามารถปรับค่าน้ำหนักจน ค่า Loss Function หลุดออกจาก Local Minimum สามารถแก้ไขได้โดยใช้วิธี Stochastic Gradient Descent โดยจะคำนวณและปรับค่าน้ำหนักทุกๆการเทรนโมเดลหนึ่งตัวอย่าง ใช้ในกรณีที่ค่า และวิธี Mini Batch Gradient Descent จะทำการคำนวณและปรับค่าน้ำหนักทุกๆ n ตัวอย่าง

2.7.1 การเพิ่มประสิทธิภาพและความเร็วอย่างเหมาะสม (Optimizer)

นอกจาก Gradient Descent Optimizer ยังมีการใช้การเพิ่มประสิทธิภาพและความเร็วอย่างเหมาะสมด้วย Optimizer เช่น Momentum Optimizer , RMSProp และ Adam Optimizer

2.7.1.1 Momentum Optimizer

การเพิ่มประสิทธิภาพด้วย Momentum Optimizer จะไม่ใช่เป็นการเพิ่มความเร็้ว แต่จะเป็นการเร่งความเร็ว เพื่อจำลองกลไกแรงเสียดทานและป้องกันไม่ให้โมเมนตัมมากเกินไป ทำให้ Gradient Descent สามารถรักษาทิศทางการเคลื่อนที่ได้ดีขึ้น จนหลุดออกจาก Local Minimum ได้ง่ายขึ้น โดยมีรูปแบบสมการ ดังนี้

$$vt = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta) \quad (2.9)$$

$$w = w + v_t \quad (2.10)$$

2.7.1.2 RMSProp

การเพิ่มประสิทธิภาพด้วย RMSProp จะใช้หลักการของ Running Average เพื่อไม่ให้ Gradient Descent ถูกหารด้วยค่าศูนย์ โดยใช้ค่า E เป็นค่าที่ใช้ในการถ่วงน้ำหนักของ Gradient Descent เก่า และลดบทบาทของค่า Learning Rate โดยสามารถเขียนรูปแบบสมการ ได้ดังนี้

$$E[g^2]_t = \gamma E[g^2]_t + (1 - \gamma)g_t^2 \quad (2.11)$$

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (2.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1.3 Adam Optimizer

Adam ย่อมาจาก Adaptive Moment Estimation หรือคือความสามารถประมาณค่าการปรับเปลี่ยนของโมเมนต์ได้ ซึ่งเกิดจากการผสมผสานข้อดีของ Momentum Optimizer และ RMSProp คือ การปรับเปลี่ยน Learning Rate ตามความสำคัญของข้อมูลและการสะสมโมเมนต์ เพื่อให้ค่า Gradient มีทิศทางคงที่สำหรับการเคลื่อนที่ โดยมีรูปแบบสมการ ดังนี้

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.13)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.14)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.15)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.16)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (2.17)$$

2.7.2 การเกิด Overfitting และ Underfitting ของข้อมูล

Overfitting และ Underfitting คือการลำเอียงของข้อมูล ส่งผลให้โมเดลคำนวณผิดพลาด โดยสามารถเกิดได้จากหลายสาเหตุ เช่น ข้อมูลสูญหายบางส่วน (Missing Value) ข้อมูลผิดปกติ (Outlier Data) หรือความไม่สมดุลในข้อมูล (Imbalance Dataset) ส่งผลให้ประสิทธิภาพในการทำนายผลลดลง

2.7.3 Data Augmentation

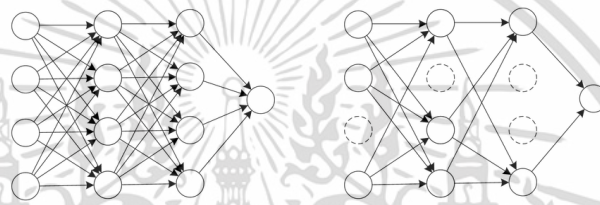
Data Augmentation คือ การเพิ่มจำนวนข้อมูลสำหรับการฝึกเพื่อให้มีความหลากหลายในการเรียนรู้มากขึ้น เช่น การปรับองศาภาพ การปรับขนาดภาพ การตัดภาพบางส่วน การปรับแสง เป็นต้น โดยความหมายของข้อมูลยังเหมือนเดิม แต่เพิ่มมิติในการเรียนรู้ที่เพิ่มขึ้นช่วยลดการเกิด Overfitting หรือ ความลำเอียงของข้อมูล (Bias)

2.7.4 Regularization

Regularization คือ กระบวนการลดค่าความผิดพลาดเพื่อป้องกันการไม่ให้โมเดลเกิดการ Overfitting หรือ Underfitting โดยมีเทคนิคการ Regularization ดังนี้

2.7.4.1 Dropout Regularization

Dropout Regularization คือ การลดการเรียนรู้ที่มากเกินไปของข้อมูล รวมไปถึงการลด Overfitting โดยปลดการเชื่อมต่อของ Node บางตัวออกแบบสุ่มขณะทำการเรียนรู้โมเดล

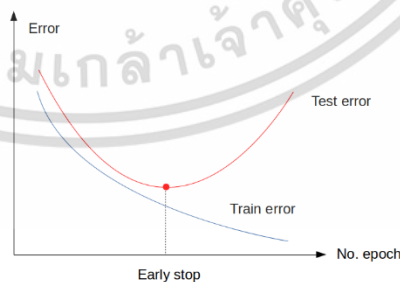


รูปที่ 2.9 ภาพแสดงการเปรียบเทียบก่อน-หลัง Dropout Regularization

หมายเหตุ. จาก <https://capablemachine.in/2020/08/20/regularization-in-neural-networks/>

2.7.4.2 Early Stopping

Early Stopping คือ การหาจุดที่เหมาะสมของการเรียนรู้ของโมเดลไม่ให้เรียนรู้มากหรือน้อยเกินไป เพื่อป้องกันไม่ให้เกิดการ Overfitting และ Underfitting โดยจะหยุดการเรียนรู้เมื่อค่าของ Metrics เช่น Loss, Accuracy มีค่าคงที่ ส่งผลให้สามารถลดจำนวนรอบการเรียนรู้หรือ Epoch ได้



รูปที่ 2.10 กราฟแสดงจุดที่เหมาะสมของการหยุดการเรียนรู้ ของความสัมพันธ์ระหว่างความผิดพลาด และ รอบของการเรียนรู้

หมายเหตุ. จาก <https://guopai.github.io/ml-blog18.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4.3 Batch Normalization

Batch Normalization คือการปรับค่าของข้อมูลภายใน Hidden Layer Node ให้อยู่ในขอบเขตที่กำหนดก่อนทำการส่งต่อไปยัง Node โดยเทียบกับค่าเฉลี่ย และ ค่าเบี่ยงเบนมาตรฐาน ของทุก Activation ใน Layer ของ Batch เพื่อให้สามารถปรับค่า Activation เป็นค่าที่เหมาะสมได้ โดยมีรูปแบบสมการดังนี้

- Mini Batch Mean

$$\mu_B = \frac{1}{m_B} \sum_{i=1}^{m_B} x^{(i)} \quad (2.18)$$

- Mini Batch Variance

$$\sigma_B^2 = \frac{1}{m_B} \sum_{i=1}^{m_B} (x^{(i)} - \mu_B)^2 \quad (2.19)$$

- Normalize

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2.20)$$

- Scale and Shift

$$z^{(i)} = \gamma \otimes \hat{x}^{(i)} + \beta \quad (2.21)$$

2.7.5 Model Validation

Model Validation คือ การประเมินผลข้อมูลใหม่ที่โมเดลไม่เคยเรียนรู้มาก่อนเพื่อให้โมเดลสามารถทำนายผลลัพธ์ออกมาได้ โดยใช้เทคนิคการสุ่มตัวอย่างซ้ำ (Resampling) เนื่องจากการทดสอบโมเดลด้วยข้อมูลที่เรียนรู้มาแล้วไม่สามารถประเมินข้อมูลใหม่ที่ไม่เคยเจอมาก่อนได้ จะส่งผลให้โมเดลเกิด Overfitting โดยการทำให้ Model Validation มีดังนี้

2.7.5.1 K-Fold Cross Validation

K-Fold Cross Validation คือ การแบ่งข้อมูลออกเป็นจำนวน K ส่วน ในสัดส่วนที่เท่ากัน โดยจะทำการเรียนรู้โมเดล K รอบ และในแต่ละรอบจะป้อนข้อมูล $K-1$ สำหรับการเรียนรู้และส่วนที่เหลือสำหรับการวัดประสิทธิภาพ จะทำซ้ำจนกว่าข้อมูลทุกชุดที่แบ่งถูกนำไปเรียนรู้ การทำ K-Fold Cross validation จะไม่เหมาะกับข้อมูลที่ไม่สมดุล (Imbalance Data) เนื่องจาก บาง Fold จะมีจำนวนข้อมูลมากกว่าคลาสอื่น ส่งผลให้โมเดลทำนายเฉพาะคลาสที่มีข้อมูลมากกว่า

2.7.5.2 Stratified K-Fold Cross-Validation

Stratified K-Fold Cross-Validation เป็นแก้ไขปัญหของข้อมูลที่ไม่สมดุล (Imbalanced Data) โดยจะแบ่งข้อมูลในแต่ละคลาสให้เท่ากันในแต่ละ Fold ซึ่งจะช่วยแก้ไขข้อจำกัดของ K-Fold Cross Validation

2.7.5.3 Hold-Out Validation

Hold-Out Validation คือการแบ่งข้อมูล 2 ส่วนคือ ข้อมูลสำหรับการเรียนรู้ (Training Set) และ ข้อมูลสำหรับทดสอบ (Testing Set) โดยป้อนข้อมูลสำหรับการเรียนรู้ให้โมเดลเพื่อทำการเรียนรู้และคำนวณค่า loss เพื่อปรับค่าถ่วงน้ำหนัก ส่วนข้อมูลสำหรับทดสอบจะถูกป้อนเข้าโมเดลสำหรับการวัดผล โดยการทดสอบจะเป็น Hyperparameter สำหรับปรับโมเดลเพื่อหาสัดส่วนที่เหมาะสม การทำ Hold-Out Validation จะไม่เหมาะกับชุดข้อมูลที่มีขนาดเล็ก เนื่องจากจะทำให้โมเดลมีค่าแปรปรวนค่อนข้างสูง

2.7.5.4 Leave-P-Out Cross-Validation

Leave-P-Out Cross-Validation คือการแบ่งข้อมูลจากข้อมูลทั้งหมด n ตัว เป็น 2 ชุดข้อมูลคือ ข้อมูลสำหรับการเรียนรู้ $n-p$ ตัว และข้อมูลสำหรับทดสอบ p ตัว และจะทำการวนซ้ำจนครบทุกตัวของข้อมูล และคำนวณหาค่าเฉลี่ยของ Metrics ของโมเดลทุกตัว วิธีนี้จะใช้การประมวลผลของคอมพิวเตอร์ค่อนข้างสูง

2.8 โครงข่ายประสาทเทียมแบบคอนโวลูชันนอล (Convolutional Neural Network : CNN)

โดยทั่วไปแล้ว โครงสร้างของโครงข่ายประสาทเทียมประกอบด้วย ชั้นนำเข้า (Input Layer), ชั้นซ่อน (Hidden Layer) และชั้นแสดงผลลัพธ์ (Output Layer) หรือเรียกว่า Multi-Layer Perceptron (MLP) แต่โครงข่ายประสาทเทียมแบบคอนโวลูชันนอล ได้เพิ่มการคำนวณที่มีลักษณะแบบคอนโวลูชัน (Convolutional) มีวัตถุประสงค์เพื่อทำงานร่วมกับรูปภาพ จะมีการจำลองการมองเห็นของมนุษย์ที่เป็นพื้นที่ย่อย ๆ และนำกลุ่มของพื้นที่ย่อยมาวิเคราะห์ว่าภาพเป็นภาพของอะไร ผ่านการสกัด คุณลักษณะของรูปภาพ (Feature extraction) ส่วนต่างๆของรูปภาพ มักใช้กับการจำแนกรูปภาพ (Image classification) หลักการทำงานของ โครงข่ายประสาทเทียมแบบคอนโวลูชันนอล ประกอบไปด้วย Convolutional Layer , ReLu (Rectified Linear Unit) Layer , Pooling Layer และ Fully Connected Layer

2.8.1 Convolutional Layer

ชั้น Convolutional เป็นการคัดกรองคุณลักษณะของรูปภาพผ่านการคำนวณด้วยตัวกรอง (Filter) เพื่อสกัดคุณลักษณะ โดยใช้หลักการทำงานแบบ Sliding Window คือผ่านทุก Input Matrix ซึ่งผลลัพธ์จะได้เป็นแผนผังคุณลักษณะ (Feature Map) แต่หนึ่งคุณลักษณะจะเป็น Filter หนึ่งตัว ดังนั้นการใช้ Filter ที่แตกต่างกันจะส่งผลให้ Feature Map แตกต่างกันด้วย และขนาดของ Filter ที่เลือกใช้ ต้องเล็กกว่าขนาดของ Input Matrix

การคำนวณ Input Matrix จะถูกคำนวณแบบ 3 มิติ คือความกว้าง ความยาว และความลึก ซึ่งทั้ง 3 มิตินี้จะแสดงถึงช่องสี (Color Channel) ดังนั้นการเลือก Filter นั้นจะต้องครอบคลุมความลึกของ Input Matrix โดยการกำหนดค่า Stride หรือ ขนาดของการ Sliding ของ Filter

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k K[u, v] I[i - u, k - v] \quad (2.22)$$

$$G[i, j] = K * I \quad (2.23)$$

จากการคำนวณ Output Matrix จะมีขนาดเล็กลง ยิ่งจำนวนชั้น Convolutional มาก จะส่งผลให้ขนาดของภาพเล็กลงมากขึ้น ดังนั้นจึงมีวิธี Zero Padding เป็นการเติมค่าศูนย์ไปที่

Input Matrix เพื่อให้ Feature Map ที่ได้หลังจาก Convolutional มีขนาดเท่ากับ Input Matrix

2.8.2 ReLu Layer

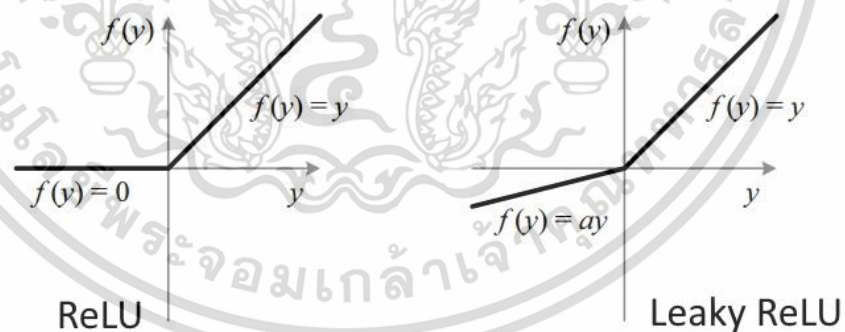
ReLU เป็น ฟังก์ชันการกระตุ้น (Activation Function) ที่ใช้ในการประมวลผล Feature Map และ ReLU เป็นฟังก์ชันการกระตุ้นที่นิยมใช้มากที่สุดเพราะมีประสิทธิภาพสูงที่สุดในโครงข่ายประสาทเทียมแบบคอนโวลูชัน เนื่องจากการเรียนรู้เชิงลึกส่วนใหญ่จะเป็นการคำนวณแบบไม่เชิงเส้นหรือไม่เป็นเส้นตรง (Non Linearity) โดยมีรูปแบบฟังก์ชัน ดังนี้

- ReLu Activation Function

$$f(y) = \begin{cases} 0, & y < 0 \\ y, & y \geq 0 \end{cases} \quad (2.24)$$

- Leaky ReLu Activation Function

$$f(y) = \begin{cases} 0.01y, & y < 0 \\ y, & y \geq 0 \end{cases} \quad (2.25)$$



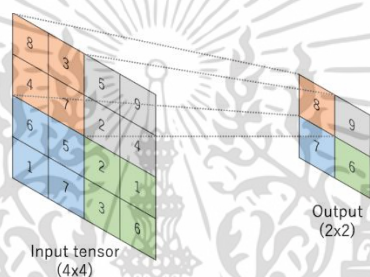
รูปที่ 2.11 กราฟของฟังก์ชันการกระตุ้นแบบ ReLu และ Leaky ReLu

หมายเหตุ. จาก <https://www.quora.com/What-is-leaky-ReLU-activation-and-why-is-it-used>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3 Pooling Layer

การทำ Pooling มีวัตถุประสงค์เพื่อลดขนาด (Down Sampling) ของ Feature Map ให้เล็กลง เพื่อเป็นการลดปริมาณข้อมูลและจำนวนของ Parameter เพื่อลดเวลาในการ คำนวณลง และการแก้ปัญหาในเรื่อง Overfitting โดยมีวิธีการ Pooling ที่นิยมคือ Max pooling และ Mean pooling โดยมีหลักการทำงานคล้ายคลึงเหมือนกับชั้น Convolution คือ มี Pooling Filter เลื่อนผ่านทุกตำแหน่งของ Feature Map และทำการคำนวณ เช่น Max Pooling จะเป็นการนำค่าที่มากที่สุดของ Filter มาเป็น Output หรือ Mean pooling เป็นการนำค่าเฉลี่ยของค่าใน Filter มาเป็น Output

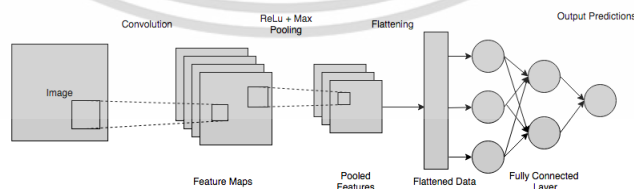


รูปที่ 2.12 การลดขนาด Feature Map ด้วย Max Pooling

หมายเหตุ. จาก <https://analyticsindiamag/maxpooling-in-cnn-and-features/>

2.8.4 Fully connected layer

Fully connected layer เป็นชั้นเชื่อมโยงแบบสมบูรณ์ เป็นชั้นที่ทำหน้าที่ในการจำแนกประเภทของรูปภาพ (Image Classification) ซึ่งทุกโครงข่ายที่อยู่ใน Fully connected layer จะถูกเชื่อมโยงกับ Convolution Layer และ Pooling Layer อย่างสมบูรณ์ จนเป็นโครงสร้างเครือข่ายประสาท (Neural Network)



รูปที่ 2.13 ขั้นตอนการทำงานของโครงข่ายประสาทเทียมแบบคอนโวลูชันนอล

หมายเหตุ. จาก <https://www.oreilly.com/library/view/hands-on-automated-machine>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 งานวิจัยที่เกี่ยวข้อง

2.9.1 Different multivariate techniques for automated classification of MRI data in Alzheimer's disease and mild cognitive impairment

การจำแนกผู้ป่วยอัลไซเมอร์ และผู้ป่วยความจำถดถอย ด้วยการใช้ตัวแปรหลายตัว โดยใช้เทคนิคการเรียนรู้ของเครื่องที่แตกต่างกัน 4 วิธี คือ Orthogonal Projections to Latent Structures (OPLS), Decision Trees (Trees), Artificial Neural Networks (ANN) and Support Vector Machines (SVM) กับชุดข้อมูล 3D MP-RAGE ที่มีความละเอียดสูงและประมวลผลด้วย FreeSurfer จากกลุ่มสำรวจผู้ป่วย 345 คน คือ ผู้ป่วยอัลไซเมอร์ 119 คน ผู้ป่วยความจำถดถอย 116 คน และคนปกติ 110 คน จากการใช้ 10-Fold Cross Validations พบว่าเทคนิค SVM และ OPLS ให้ผลลัพธ์ที่ดีกว่าเทคนิค Trees และ ANN โดยในการจำแนกระหว่างผู้ป่วยอัลไซเมอร์และคนปกติให้ค่าความไวที่ 83% และค่าจำเพาะ 87% สำหรับเทคนิคที่ดีที่สุด และสามารถทำนายผู้ป่วยความจำถดถอยโดยมีฐานข้อมูลจากผู้ป่วยอัลไซเมอร์ได้ความแม่นยำ 86% งานวิจัยชิ้นนี้สามารถพัฒนาต่อได้โดยใช้การเรียนรู้ของเครื่องด้วยเทคนิคที่สูงขึ้น และใช้ข้อมูลที่เกี่ยวข้องกับสมองมาเพิ่มเป็นตัวแปรเพื่อสร้างความซับซ้อนและพัฒนาการเรียนรู้ของเครื่อง

2.9.2 MRI Simulation-Based Evaluation of Image-Processing and Classification Methods

การตรวจสอบความถูกต้องของการวินิจฉัยด้วยภาพ MRI ซึ่งยากต่อการจินตนาการรูปภาพให้เสมือนจริง ส่งผลให้วินิจฉัยได้ลำบาก วัตถุประสงค์ของงานวิจัยนี้คือจำลองภาพเสมือนของ MRI แบบ 3 มิติ อย่างมีประสิทธิภาพโดยใช้เทคนิคการประมวลผลภาพ (Image Processing) Hybrid Bloch, Tissue Template Simulation, Image Contrast, Partial Volume และ Noise โดยภาพประมวลผลภาพเสมือนจริงจะสามารถช่วยให้การวินิจฉัยสามารถทำได้ง่ายขึ้น งานวิจัยนี้สามารถนำเทคนิคการประมวลผลภาพมาใช้ควบคู่กับการเรียนรู้เชิงลึกเพื่อเพิ่มความซับซ้อนและเพิ่มประสิทธิภาพในการวินิจฉัยได้

2.9.3 Automated Classification of Alzheimer's Disease and Mild Cognitive Impairment using A Single MRI and Deep Neural Networks

วัตถุประสงค์ของงานวิจัยนี้คือ การสร้างการเรียนรู้เชิงลึกเพื่อวินิจฉัยผู้ป่วยอัลไซเมอร์ และผู้ป่วยความจำถดถอยโดยใช้ภาพถ่ายสมองแบบตัดขวาง ด้วยการใช้เทคนิค CNN โดยป้อนข้อมูลเป็นรูปภาพ 3D T1-weight และสามารถวินิจฉัยแบบอัตโนมัติด้วยความแม่นยำ 75% ซึ่งความสามารถในการวินิจฉัยอัตโนมัติสามารถทำงานได้ดีโดยไม่ต้องทำ Feature Engineer และช่วยลดความแปรปรวนของภาพจาก MRI ส่งผลให้ผู้ใช้งานสามารถใช้งานได้แม้ประสบการณ์น้อย งานวิจัยนี้สามารถเพิ่มประสิทธิภาพได้ด้วยการเพิ่มข้อมูลภาพ T2 และ Flair เพื่อให้เห็นมุมมองที่แตกต่างและมากขึ้น มาผสมผสานกับการเรียนรู้เชิงลึกเพื่อการวินิจฉัยที่แม่นยำและมีประสิทธิภาพ

2.9.4 Combining MRI and CSF Measures for Classification of Alzheimer's Disease and Prediction of Mild Cognitive Impairment Conversion

วัตถุประสงค์ของงานวิจัยนี้คือตรวจสอบความสามารถในการจำแนกผู้ป่วยความจำถดถอยโดยการผสมผสานระหว่างภาพ MRI และ CSF เปรียบเทียบการใช้ข้อมูลภาพอย่างใดอย่างหนึ่ง ด้วยการใช้เทคนิค Orthogonal Partial Least Squares to Latent Structures (OPLS) จากการทดลอง การผสมผสานระหว่าง MRI และ CSF ให้ผลลัพธ์ที่ดีที่สุด ในงานวิจัยนี้สามารถเพิ่มประสิทธิภาพในการจำแนกได้ด้วยการใช้เทคนิคการเรียนรู้เชิงลึกเพื่อสร้างความซับซ้อน และเปรียบเทียบเทคนิค OPLS กับ เทคนิคอื่นๆ เพื่อพัฒนาให้มีประสิทธิภาพมากขึ้น

2.9.5 Detection of Alzheimer's Disease on Brain MRI using Inception V3 Network

การใช้ภาพ MRI มีอิทธิพลอย่างมากในอุตสาหกรรมทางการแพทย์ โดยช่วยให้เห็นโครงสร้างของอวัยวะภายในร่างกายได้ละเอียดเพิ่มขึ้น รวมไปถึงสมองของมนุษย์ ในบทความนี้จะทำการทดลองจรวจจับโรคสมองเสื่อมโดยใช้การเรียนรู้เชิงลึกด้วยสถาปัตยกรรมโมเดล InceptionV3 จากการทดลองพบว่า สถาปัตยกรรมโมเดล InceptionV3 สามารถให้ความแม่นยำในการตรวจจับโรคสมองเสื่อมได้มากถึง 82.89% และสามารถจำแนกด้วยความเร็ว 0.11 วินาทีต่อภาพ

2.9.6 Deep-learning approach with convolutional neural network for classification of maximum intensity projections of dynamic contrast-enhanced breast magnetic resonance imaging

จุดประสงค์ของงานวิจัยนี้ เพื่อเปรียบเทียบประสิทธิภาพของการเรียนรู้เชิงลึกด้วยเทคนิค CNN เพื่อแยกระหว่างโรคที่ร้ายแรงและโรคที่ไม่ร้ายแรง จากภาพ MRI ที่มีระดับความเข้มข้นสูงสุดและมีการคอนทราสต์แบบไดนามิก ด้วยสถาปัตยกรรมโมเดลดังนี้ DenseNet121, DenseNet169, InceptionResNetV2, InceptionV3, NasNetMobile, และ Xception จากการทดลองพบว่า สถาปัตยกรรมโมเดลที่ให้ความแม่นยำสูงที่สุดคือ InceptionResNetV2 ด้วยค่า specificities 96% และ AUC 0.895



บทที่ 3

การดำเนินงานวิจัย

3.1 ความเป็นมาและความสำคัญ

ภาวะความถดถอยทางสมอง (Mild Cognitive Impairment : MCI) คือ ระยะที่อยู่ระหว่างการเปลี่ยนแปลงของคนปกติและผู้ป่วยโรคสมองเสื่อม (Alzheimer's Disease : AD) ซึ่งผู้ป่วย MCI จะเป็นอาการเริ่มแรกของ ผู้ป่วย AD โดยมีสาเหตุมาจากการเสื่อมสภาพของเซลล์ประสาทในสมอง โดยผู้ป่วยที่เป็น MCI จะมีปัญหาด้านความจำ และสุขภาพจิต ซึ่งจำเป็นต้องได้รับการรักษาและการวินิจฉัยที่ถูกต้อง ทั้งนี้ผู้ป่วย MCI สามารถแบ่งได้ 2 ประเภทคือ ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มแรก (Early - Mild Cognitive Impairment) และผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย (Late - Mild Cognitive Impairment) ดังนั้นการทดลองนี้จะทำการจำแนกประเภทของผู้ป่วยทั้งหมด 4 กลุ่ม ดังนี้

1. ผู้ป่วยปกติ (CN)
2. ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มแรก (Early + MCI)
3. ผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย (Late - MCI)
4. ผู้ป่วยโรคสมองเสื่อม (AD)

3.2 แนวทางการดำเนินงานวิจัย

แนวทางการดำเนินงานวิจัยนี้ จะประกอบไปด้วยขั้นตอนดังนี้

1. การทำความเข้าใจกับข้อมูล (Data Understanding)
2. การเตรียมข้อมูล (Data Preparation)
3. การสำรวจข้อมูล (Exploratory Data Analysis)
4. การสร้างรูปแบบการทำนาย (Modeling)
5. 5.การประเมินผล (Evaluation)

3.3 การทำความเข้าใจกับข้อมูล (Data Understanding)

ในการทำข้อมูลรูปภาพ จะแตกต่างกับข้อมูลทั่วไปเนื่องจากเป็น ข้อมูลที่ไม่มีรูปแบบโครงสร้าง (Unstructured Data) และมีรูปแบบของไฟล์ที่แตกต่างกัน ส่งผลให้ต้องใช้เครื่องมือ (Tools) หรือ คำสั่ง เฉพาะทางในการศึกษา

ภาพที่ศึกษาในงานวิจัยนี้เป็นภาพ MRI ซึ่งเป็นภาพที่ใช้ในเชิงการแพทย์ จะมีนามสกุลของไฟล์ คือ .nii ซึ่งจำเป็นต้องใช้ Tools เฉพาะทางเพื่อการเปิดรูปภาพ โดย Tools ที่ใช้คือ NiBabel เป็น Tools ที่ใช้ในการอ่านไฟล์ทางการแพทย์หรือ Neuro – Imaging และ Tools นี้สามารถอ่านได้หลายนามสกุล เช่น .nii และ .nii.gz เป็นต้น

ข้อมูลชุดนี้ (Datasets) จะมาในรูปแบบของ .nii จึงจำเป็นต้องทำการติดตั้ง NiBabel เพื่อทำการอ่านไฟล์ และมีไฟล์ .csv หรือไฟล์สำหรับบอกรายละเอียดของรูปภาพนั้น (Description File) เพื่อบอกว่า ภาพนั้นคือสมมติว่าเป็น ผู้ป่วยสมองเสื่อม (AD) ,ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มแรก (Early – MCI), ผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย (Late – MCI) และ ผู้ป่วยปกติ (CN) โดยใน Description File จะมี Image ID ซึ่ง ID ของรูปภาพนั้นๆ ทำให้สามารถแบ่งได้ว่า ภาพ ID นั้นเป็น ผู้ป่วยประเภทใด โดยจะทำการสร้าง Folders ทั้งหมด 4 Folders เพื่อเก็บข้อมูลรูปภาพแต่ละประเภทของผู้ป่วย (CN, AD, Early-MCI และ Late-MCI) โดยใช้คำสั่งดังนี้

```
Des = pd.read_csv("/content/drive/MyDrive/ADNI/Description.csv")
map_df = Des[['Image Data ID', 'Group']]
map_df.head()

Image Data ID  Group
0      1105437    CN
1      1204843    CN
2      1137298    CN
3      163874     CN
4      175150     CN

error_filenames = []
for file in file_paths:
    filename = file.split('/')[ -1].split('.')[ -1]
    try:
        if map_df[map_df['Image Data ID'] == filename.split('.')[0]].Group.values[0] == 'CN':
            shutil.copy(file, '/content/drive/MyDrive/ADNI/CN/' + filename)
        elif map_df[map_df['Image Data ID'] == filename.split('.')[0]].Group.values[0] == 'AD':
            shutil.copy(file, '/content/drive/MyDrive/ADNI/AD/' + filename)
        elif map_df[map_df['Image Data ID'] == filename.split('.')[0]].Group.values[0] == 'Early-MCI':
            shutil.copy(file, '/content/drive/MyDrive/ADNI/Early-MCI/' + filename)
        elif map_df[map_df['Image Data ID'] == filename.split('.')[0]].Group.values[0] == 'Late-MCI':
            shutil.copy(file, '/content/drive/MyDrive/ADNI/Late-MCI/' + filename)
    except:
        error_filenames.append(filename)
```

รูปที่ 3.1 คำสั่งในการแยกไฟล์แต่ละประเภทของผู้ป่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเตรียมข้อมูล (Data Preparation)

ในการทำข้อมูลรูปภาพสมอง นี้ข้อมูลมาในรูปแบบของ NiftI format หรือ .nii จึงยังไม่สามารถนำไปวิเคราะห์ได้ เพราะจำเป็นต้องทำให้จากภาพ 3 มิติ เป็นภาพ 2 มิติ ก่อน ส่งผลให้ต้องทำการประมวลผลภาพ (Image Processing) โดยมีขั้นตอนดังนี้

1. การลอกกะโหลก (Skull Stripping)
2. การแปลงภาพจาก 3 มิติ เป็น 2 มิติ
3. การเพิ่มปริมาณข้อมูล (Data Augmentation)

3.4.1 การลอกกะโหลก (Skull Stripping)

ในการลอกกะโหลกจะเป็นจะต้องติดตั้ง Tools เฉพาะทางชื่อ FSL Brain Extraction Tools ซึ่งสามารถติดตั้งได้ทั้งในคอมพิวเตอร์ระบบ Microsoft และ IOS แต่การติดตั้งบน Local Computer จะมีความยุ่งยากในการติดตั้งไปอย่างมากถ้าผู้วิจัยไม่มีความชำนาญมากพอ ดังนั้นในงานวิจัยนี้ จะใช้ Google Colab เป็น Environment หลักในการวิจัยตั้งแต่ต้นจนจบ ซึ่งการติดตั้ง FSL Brain Extraction Tools สามารถติดตั้งบน Google Colab ด้วยคำสั่งดังรูปที่ 3.2

```
import os
!sudo apt -qq install file
!wget https://fsl.fmrib.ox.ac.uk/fsldownloads/fslinstaller.py
!python2 fslinstaller.py

fslpath = "/usr/local/fsl"
os.environ["FSLDIR"] = fslpath
os.environ["PATH"] += os.pathsep + os.path.join(fslpath, 'bin')
!. ${FSLDIR}/etc/fslconf/fsl.sh

!flirt -version
```

รูปที่ 3.2 คำสั่งการติดตั้ง FSL Brain Extraction Tools

หลังจากติดตั้ง FSL Brain Extraction บน Environment ของ Google Colab แล้วจะ
ทำการใช้คำสั่ง ดังรูปที่ 3.3 เพื่อทำการปลอกกะโหลก

```
def skull_strip_nii(original_img, destination_img, frac=0.35):
    ''' Practice skull stripping on the given image, and save
    the result to a new .nii image.
    Uses FSL-BET
    (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide#Main\_bet2\_options:)

    Parameters:
        original_img -- Original nii image
        destination_img -- The new skull-stripped image
        frac -- Fractional intensity threshold for BET
    ...

    btr = fsl.BET()
    btr.inputs.in_file = original_img
    btr.inputs.frac = frac
    btr.inputs.out_file = destination_img
    btr.cmdline
    res = btr.run()

#CN
#REG_DB = '/content/drive/MyDrive/ADNI/CN/'
SKULL_STRIPPED_DB = '/content/drive/MyDrive/ADNI/CN_Strip'

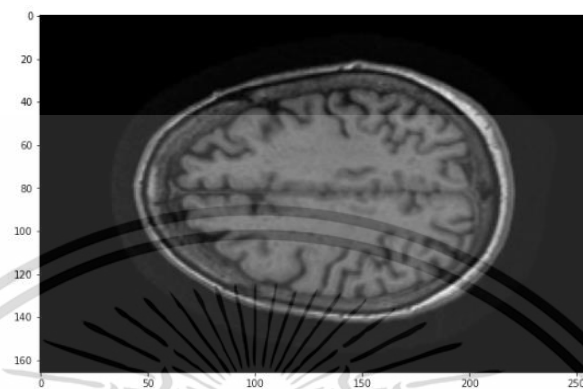
exceptions = []
for path, _, files in os.walk(REG_DB):
    for file in files:
        try:
            img = os.path.join(path, file)
            dest = os.path.join(SKULL_STRIPPED_DB, file)
            skull_strip_nii(img, dest, frac=0.35)
        except RuntimeError:
            exceptions.append(img)

# save the exceptions in case you want to do something about them
# in our case, FSL BET failed with a couple of images, although it
# was a very small amount so they were simply discarded
with open(os.path.join(SKULL_STRIPPED_DB, 'exceptions.txt'), 'w') as f:
    for item in exceptions:
        f.write("%s\n" % item)
```

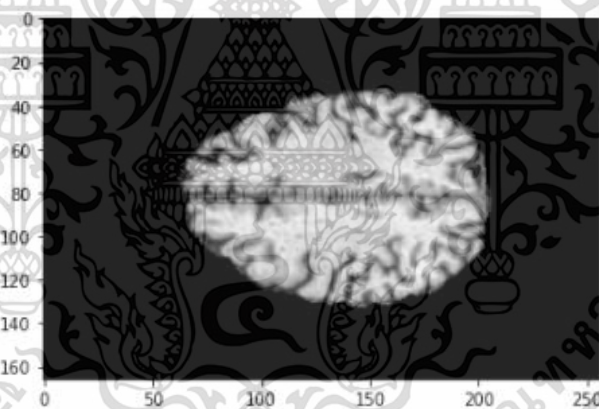
รูปที่ 3.3 คำสั่งการปลอกกะโหลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทำการปกอกกะโหลกด้วย FSL Brain Extraction Tools แล้วจะได้ภาพออกมาดังรูปที่ 3.5 ส่วนรูปที่ 3.4 จะเป็นการรูปก่อนทำการปกอกกะโหลก



รูปที่ 3.4 รูปสมองก่อนทำการปกอกกะโหลก



รูปที่ 3.5 รูปสมองหลังการทำการปกอกกะโหลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การแปลงภาพจาก 3 มิติ เป็น 2 มิติ

เนื่องจากชุดข้อมูลเป็นรูปภาพ 3 มิติ จึงจำเป็นต้องเปลี่ยนชุดข้อมูลเป็นรูปแบบ 2 มิติ เพราะในการทดลองจะใช้รูปภาพแกน Axial ในการทำนาย โดยใช้คำสั่งดังรูปที่ 3.6 และไปเก็บใน Folders เพื่อเตรียมสำหรับไปทำการทำนาย ดังรูปที่ 3.7

```
for x_iterator in range(min_x_range, max_x_range +1):
    axial_image = input_image_data[x_iterator, :, :]
    patient_file_name = patient_id + "_slice_" + str(x_iterator) + "_rotated_axial.png"
    combined_patient_file_name = os.path.join(combined_patient_dir, patient_file_name)
```

รูปที่ 3.6 คำสั่งการแปลงรูปภาพจาก 3 มิติ เป็น 2 มิติ



รูปที่ 3.7 ตัวอย่างรูปภาพที่แปลงจาก 3 มิติ เป็น 2 มิติ

3.4.3 การเพิ่มปริมาณข้อมูล (Data Augmentation)

การใช้เทคนิคการเพิ่มปริมาณข้อมูล (Data Augmentation) เป็นการเพิ่มปริมาณของข้อมูล โดยยังคงคุณภาพของข้อมูล โดรนสร้างจากข้อมูลเดิมรวมไปถึงการหมุนข้อมูลให้อยู่ในองศาที่เหมือนกันหรือต่างกันเล็กน้อย ดังนั้นข้อมูลจะมุมมองที่เพิ่มขึ้น ส่งผลให้สามารถทำการทำนายได้แม่นยำมากขึ้น โดยใช้คำสั่งดังรูปที่ 3.8 และคำสั่งแสดงข้อมูลดังรูปที่ 3.9

```
#Performing Image Augmentation to have more data samples

ZOOM = [.99, 1.01]
BRIGHT_RANGE = [0.8, 1.2]
HORZ_FLIP = True
FILL_MODE = "constant"
DATA_FORMAT = "channels_last"

work_dr = IDG(rescale = 1./255,
              brightness_range=BRIGHT_RANGE,
              zoom_range=ZOOM,
              data_format=DATA_FORMAT,
              fill_mode=FILL_MODE,
              horizontal_flip=HORZ_FLIP)

train_data_gen = work_dr.flow_from_directory(directory=WORK_DIR, target_size=DIM, batch_size=BATCH_SIZE, shuffle=False)
```

รูปที่ 3.8 คำสั่งการเพิ่มปริมาณข้อมูล (Data Augmentation)

```
def show_images(generator,y_pred=None):
    """
    Input: An image generator,predicted labels (optional)
    Output: Displays a grid of 9 images with labels
    """
    # get image labels
    labels =dict(zip([0,1,2,3], CLASSES))

    # get a batch of images
    x,y = generator.next()

    # display a grid of 9 images
    plt.figure(figsize=(10, 10))
    if y_pred is None:
        for i in range(9):
            ax = plt.subplot(3, 3, i + 1)
            idx = randint(0, 6400)
            plt.imshow(x[idx])
            plt.axis("off")
            plt.title("Class:{}".format(labels[np.argmax(y[idx])]))
    else:
        for i in range(9):
            ax = plt.subplot(3, 3, i + 1)
            plt.imshow(x[i])
            plt.axis("off")
            plt.title("Actual:{} \nPredicted:{}".format(labels[np.argmax(y[i])],labels[y_pred[i]]))

    # Display Train Images
    show_images(train_data_gen)
```

รูปที่ 3.9 คำสั่งการแสดงผลข้อมูลจากการทำเพิ่มปริมาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การสำรวจข้อมูล (Exploratory Data Analysis)

ทำการสำรวจข้อมูลเพื่อดูจำนวนรูปภาพหลังจากทำการประมวลผลภาพ (Image Processing) ก่อนจะนำเข้าสู่รูปแบบการคำนวณ โดยดูตามแต่ละประเภทของผู้ป่วย ดังตารางที่ 3.1

ตารางที่ 3.1 จำนวนรูปภาพ

ประเภทของผู้ป่วย	จำนวนรูปภาพ
CN	3200
AD	2240
Early-MCI	896
Late-MCI	64

จากตารางที่ 3.1 จะเห็นว่าจำนวนข้อมูลมีทั้งหมด 6,400 รูป แต่ไม่สมดุลกันหรือห่างกันมาก อาจก่อให้เกิดความผิดพลาดในการทำนายได้ เลยสร้างชุดข้อมูลอีกชุดหนึ่งขึ้นมาโดยการ Over Sampling ด้วยเทคนิค SMOTE(Synthetic Minority Over Sampling Technique) โดยการเพิ่มปริมาณของกลุ่มที่มีจำนวนข้อมูลน้อยให้มีปริมาณข้อมูลที่มากขึ้น

SMOTE เป็นเทคนิคที่ใช้ในการแก้ปัญหาการจำแนกข้อมูลที่ไม่สมดุล เนื่องจากข้อมูลในแต่ละคลาสมีจำนวนแตกต่างกันมาก ทำให้ผลลัพธ์ของการจำแนกอยู่ในข้อมูลกลุ่มมาก ดังนั้นวิธี SMOTE เป็นการเพิ่มจำนวนข้อมูลกลุ่มน้อย ให้มีจำนวนเพิ่มขึ้นโดยการเพิ่มข้อมูลในกลุ่มน้อยนั้นทำให้การกระจายของกลุ่มข้อมูลมีความสมดุลมากขึ้น โดยทำการสุ่มค่าข้อมูลที่อยู่ในกลุ่มข้อมูลน้อยขึ้นมา 1 ค่าหลังจากนั้นพิจารณาค่าข้อมูลใกล้เคียงอีกจำนวน K ค่า(K-nearest neighbor) แล้วคำนวณค่าระยะทาง(Euclidean distance) ระหว่างค่าที่สุ่มกับค่าข้อมูลใกล้เคียงแต่ละค่า เพื่อหาค่าระยะทางที่น้อยที่สุดระหว่างค่าที่สุ่มกับค่าข้อมูลใกล้เคียง จากนั้นจึงสร้างข้อมูลเทียมระหว่างค่าข้อมูลที่สุ่มกับค่าข้อมูลใกล้เคียงตัวที่ให้ค่าระยะทางที่น้อยที่สุด โดยใช้คำสั่งดังรูปที่ 3.10

```

#Performing over-sampling of the data, since the classes are imbalanced

sm = SMOTE(random_state=42)

train_data, train_labels = sm.fit_resample(train_data.reshape(
                                                    -1,
                                                    IMG_SIZE * IMG_SIZE * 3
                                                    ),
                                                    train_labels
                                                    )

train_data = train_data.reshape(-1,
                                IMG_SIZE,
                                IMG_SIZE,
                                3
                                )

print(train_data.shape, train_labels.shape)

```

รูปที่ 3.10 คำสั่งการใช้ SMOTE

ตารางที่ 3.2 จำนวนรูปภาพหลังใช้คำสั่ง SMOTE

ประเภทของผู้ป่วย	จำนวนรูปภาพ
CN	3200
AD	3200
Early-MCI	3200
Late-MCI	3200

หลังจากทำการ SMOTE จะมีรูปภาพทั้งหมด 12,800 รูป ในการวิจัยนี้เราจะทำนายด้วยชุดข้อมูลก่อนทำการ Over Sampling และหลัง Over Sampling เพื่อศึกษาว่าข้อมูลที่ไม่สมดุลกันมากๆ จะส่งผลกับการทำนายหรือไม่ โดยจะแยกชุดข้อมูลเป็น 2 ชุด คือชุดข้อมูลเดิม (Datasets Without SMOTE) และชุดข้อมูลหลังการปรับปรุง (Datasets with SMOTE) หลังจากนั้นจะทำการแบ่งข้อมูลของแต่ละชุด เป็น ข้อมูลสำหรับการเรียนรู้ (Train) 80% ของจำนวนข้อมูลในชุดนั้น และ ข้อมูลสำหรับทดสอบ (Test) 20% ของจำนวนข้อมูลในชุดนั้น เพื่อเตรียมเข้ารูปแบบการทำนาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การสร้างรูปแบบการทำนาย (Modeling)

ในงานวิจัยนี้จะทำนายผลรูปภาพโดยการใช้เทคนิคการจำแนกประเภทรูปภาพ (Image Classification) โดยใช้รูปแบบสถาปัตยกรรมโมเดล 2 รูปแบบ คือ InceptionV3 และ InceptionRestnetV2 ซึ่งมี Hyperparameters ตารางที่ 3.3

ตารางที่ 3.3 ค่า Hyperparameters ของ InceptionV3 และ InceptionRestnetV2

Hyperparameters	Value
Input Value	176
Input Channels	3
Activation Function	ReLU
Optimizer Algorithm	Adam
Learning Rate	0.0001
Epochs	100
Batch SIZE	128

```

inception_model = InceptionV3(input_shape=(176, 176, 3), include_top=False, weights='imagenet')

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_data_format.h5
87916544/87910968 [=====] - 1s 0us/step
87924736/87910968 [=====] - 1s 0us/step

for layer in inception_model.layers:
    layer.trainable=False

custom_inception_model = Sequential([
    inception_model,
    Dropout(0.5),
    GlobalAveragePooling2D(),
    Flatten(),
    BatchNormalization(),
    Dense(512, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(256, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dropout(0.5),
    BatchNormalization(),
    Dense(4, activation='softmax')
], name = "inception_cnn_model")

```

รูปที่ 3.11 คำสั่งการใช้งาน InceptionV3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#Defining a custom callback function to stop training our model when accuracy goes above 99%

class MyCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('acc') > 0.99:
            print("\nReached accuracy threshold! Terminating training.")
            self.model.stop_training = True

my_callback = MyCallback()

#ReduceLRonPlateau to stabilize the training process of the model
rop_callback = ReduceLRonPlateau(monitor="val_loss", patience=3)

METRICS = [tf.keras.metrics.CategoricalAccuracy(name='acc'),
            tf.keras.metrics.AUC(name='auc'),
            tfa.metrics.F1Score(num_classes=4)]

CALLBACKS = [my_callback, rop_callback]

custom_inception_model.compile(optimizer='rmsprop',
                               loss=tf.losses.CategoricalCrossentropy(),
                               metrics=METRICS)

custom_inception_model.summary()

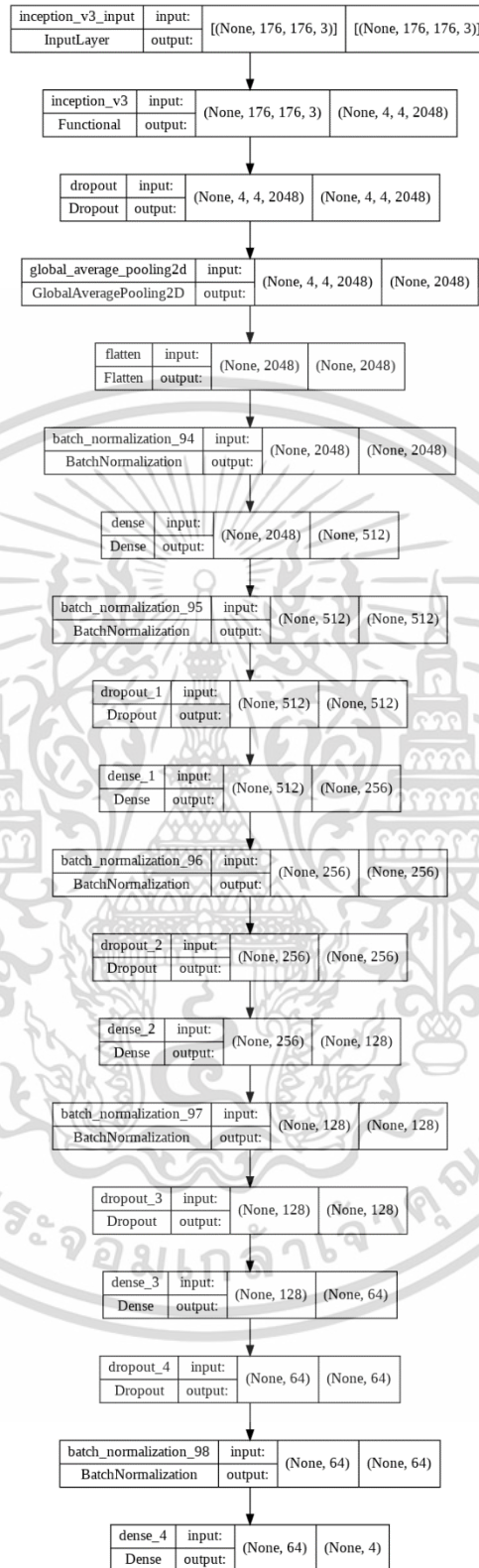
#Fit the training data to the model and validate it using the validation data
EPOCHS = 100

history = custom_inception_model.fit(train_data,
                                     train_labels,
                                     validation_data=(val_data, val_labels),
                                     callbacks=CALLBACKS,
                                     epochs=EPOCHS)

```

รูปที่ 3.11 คำสั่งการใช้งาน InceptionV3 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 สถาปัตยกรรมโมเดลของ InceptionV3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

InceptionResNetV2_model = tf.keras.applications.InceptionResNetV2(input_shape=(176, 176, 3), include_top=False, weights='imagenet')

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_resnet_v2/inception_resnet_v2_weights_t
219062272/219055592 [=====] - 2s 0us/step
219070464/219055592 [=====] - 2s 0us/step

for layer in InceptionResNetV2_model.layers:
    layer.trainable=False

custom_InceptionResNetV2_model = Sequential([
    InceptionResNetV2_model,
    Dropout(0.5),
    GlobalAveragePooling2D(),
    Flatten(),
    BatchNormalization(),
    Dense(512, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(256, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dropout(0.5),
    BatchNormalization(),
    Dense(4, activation='softmax')
], name = "InceptionResNetV2_cnn_model")

#Defining a custom callback function to stop training our model when accuracy goes above 99%
class MyCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('acc') > 0.99:
            print("\nReached accuracy threshold! Terminating training.")
            self.model.stop_training = True

my_callback = MyCallback()

#ReduceLRonPlateau to stabilize the training process of the model
rop_callback = ReduceLRonPlateau(monitor="val_loss", patience=3)

METRICS = [tf.keras.metrics.CategoricalAccuracy(name='acc'),
            tf.keras.metrics.AUC(name='auc'),
            tfa.metrics.F1Score(num_classes=4)]

CALLBACKS = [my_callback, rop_callback]

custom_InceptionResNetV2_model.compile(optimizer='rmsprop',
                                       loss=tf.losses.CategoricalCrossentropy(),
                                       metrics=METRICS)

custom_InceptionResNetV2_model.summary()

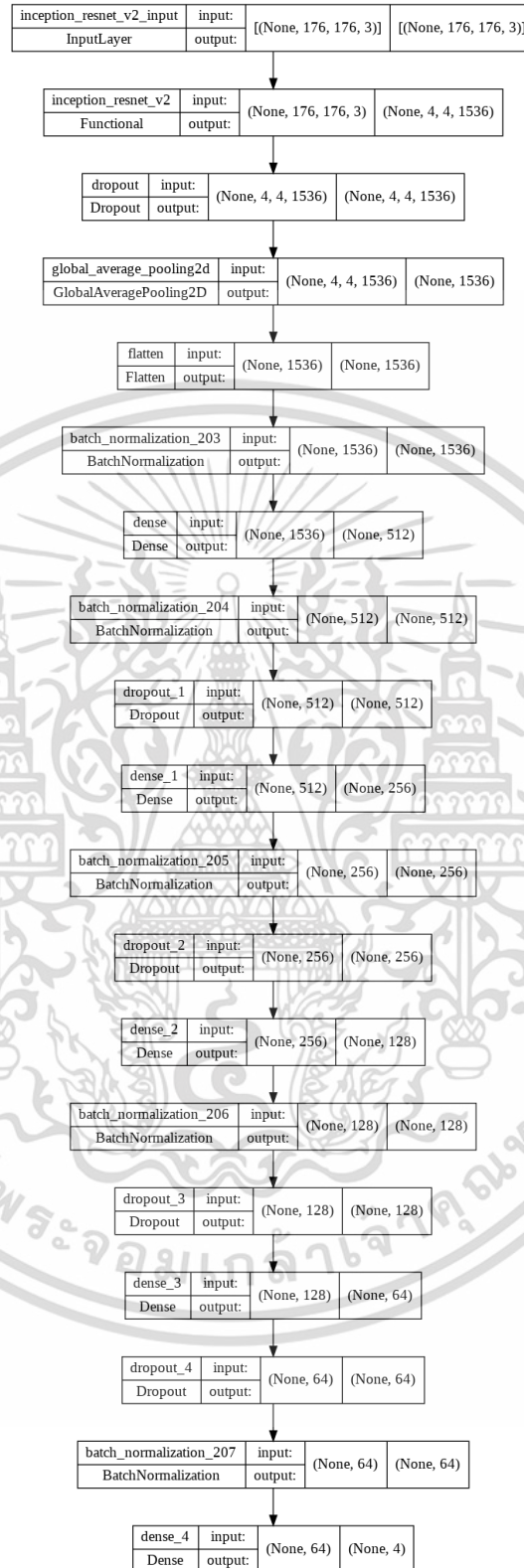
#Fit the training data to the model and validate it using the validation data
EPOCHS = 100

history = custom_InceptionResNetV2_model.fit(train_data,
                                             train_labels,
                                             validation_data=(val_data, val_labels),
                                             callbacks=CALLBACKS,
                                             epochs=EPOCHS
                                             )

```

รูปที่ 3.13 คำสั่งการใช้งาน InceptionRestnetV2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 สถาปัตยกรรมโมเดลของ InceptionRestnetV2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การประเมินผล (Evaluation)

การทดสอบและประเมินประสิทธิภาพโมเดลในงานวิจัยนี้จะใช้มาตรวัดเพื่อวัดประสิทธิภาพของโมเดล มาตรวัดที่ถูกใช้เพื่อทำการประเมินการสร้างโมเดลในแต่ละโมเดลมีทั้งหมด 6 มาตรวัด ได้แก่ Accuracy, Loss, F1-Score, AUC, Balance Accuracy Score และ Matthew's Correlation Coefficient โดยมีตัวอย่างผลการประเมินประสิทธิภาพโมเดล ดังรูปที่ 3.15

Epoch	Loss	acc	auc	f1_score
Epoch 1/100	1.4790	0.3805	0.6519	0.3789
Epoch 2/100	1.0382	0.5303	0.8059	0.5228
Epoch 3/100	0.8653	0.6063	0.8635	0.5984
Epoch 4/100	0.7705	0.6548	0.8935	0.6508
Epoch 5/100	0.7197	0.6835	0.9075	0.6826
Epoch 6/100	0.6751	0.6978	0.9166	0.6978
Epoch 7/100	0.6434	0.7231	0.9263	0.7234
Epoch 8/100	0.6156	0.7302	0.9317	0.7316
Epoch 9/100	0.5922	0.7461	0.9372	0.7474
Epoch 10/100	0.5655	0.7591	0.9431	0.7604
Epoch 11/100	0.5563	0.7729	0.9450	0.7735
Epoch 12/100	0.5350	0.7827	0.9495	0.7832
Epoch 13/100	0.4924	0.7955	0.9566	0.7965
Epoch 14/100	0.4725	0.8121	0.9600	0.8123
Epoch 15/100	0.4589	0.8214	0.9624	0.8215
Epoch 16/100	0.4454	0.8249	0.9646	0.8251
Epoch 17/100	0.4413	0.8306	0.9651	0.8308
Epoch 18/100	0.4161	0.8397	0.9688	0.8395
Epoch 19/100	0.3922	0.8500	0.9722	0.8505
Epoch 20/100	0.3998	0.8472	0.9711	0.8470
Epoch 21/100	0.3744	0.8564	0.9750	0.8567
Epoch 22/100	0.3682	0.8612	0.9756	0.8611
Epoch 23/100	0.3399	0.8751	0.9786	0.8751

รูปที่ 3.15 ตัวอย่างการวัดประสิทธิภาพโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

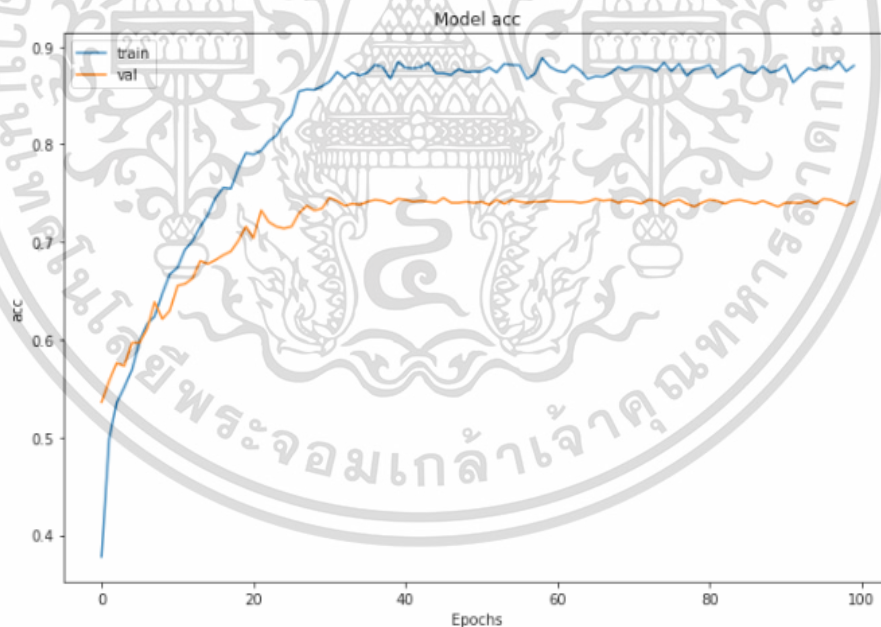
บทที่ 4

ผลการทดลองและการอภิปรายผล

ในงานวิจัยนี้จะมีการทดลองสร้างโมเดลจำแนกผู้ป่วยแต่ละประเภท ด้วยสถาปัตยกรรม InceptionV3 และ InceptionResnetV2 โดยจะมีทั้งหมด 4 การทดลองโดยแบ่งเป็นเป็นสถาปัตยกรรมละ 2 การทดลอง โดยในแต่ละการทดลอง จะใช้ชุดข้อมูล 2 ชุดข้อมูล คือ ชุดข้อมูลก่อน Over Sampling และ ชุดข้อมูลหลัง Over Sampling

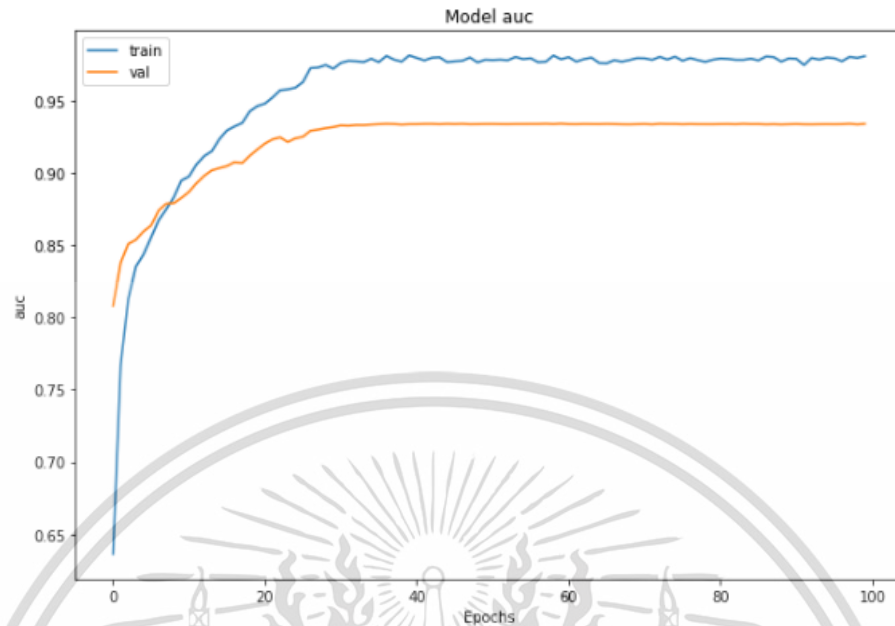
4.1 ผลการทดลองสถาปัตยกรรม InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling

ผลการทดลองนี้จะประกอบไปด้วยกราฟ Accuracy, AUC, Loss ,Confusion Matrix, ค่า Precision, Recall และ F1-Score ของผู้ป่วยแต่ละประเภท, ค่า Balance Accuracy Score และ Matthew's Correlation Coefficient ดังรูปที่ 4.1, 4.2, 4.3, 4.4 และตารางที่ 4.1 ,4.2 ตามลำดับ

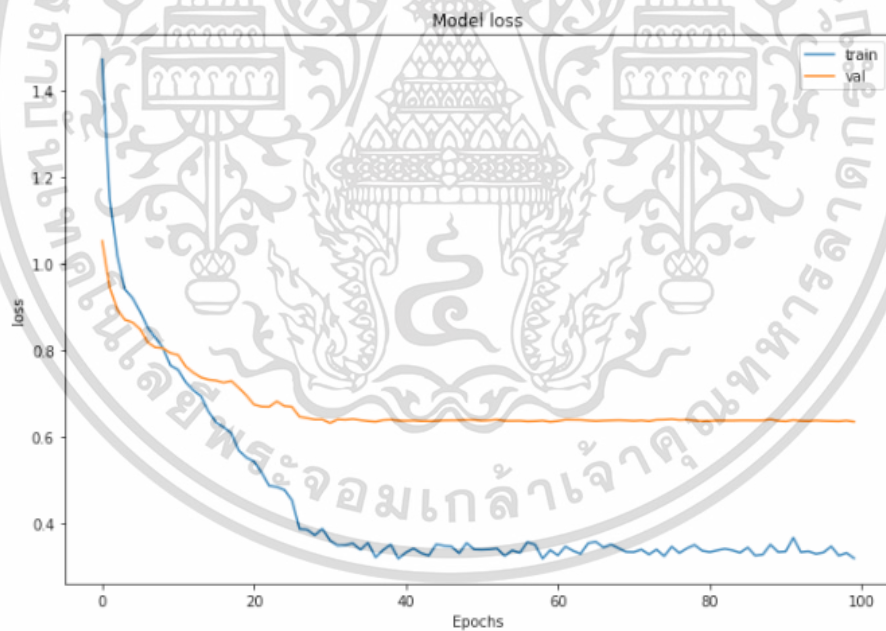


รูปที่ 4.1 กราฟ Accuracy ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

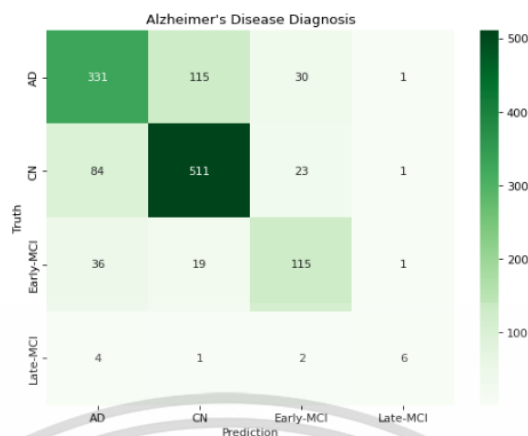


รูปที่ 4.2 กราฟ AUC ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling



รูปที่ 4.3 กราฟ Loss ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 กราฟ Confusion Matrix ของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling

จากรูปที่ 4.4 สามารถนำหาค่า Precision, Recall, F1-Score และ Support เพื่อวัดประสิทธิภาพของสถาปัตยกรรมโมเดลในการจำแนกโรคแต่ละประเภทดังตารางที่ 4.1 และสามารถวัดประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionV3 ได้ดังตารางที่ 4.2

ตารางที่ 4.1 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling

	Precision	Recall	F1-Score	Support
AD	0.73	0.69	0.71	477
CN	0.79	0.83	0.81	619
Early-MCI	0.68	0.67	0.67	171
Late-MCI	0.67	0.46	0.55	13

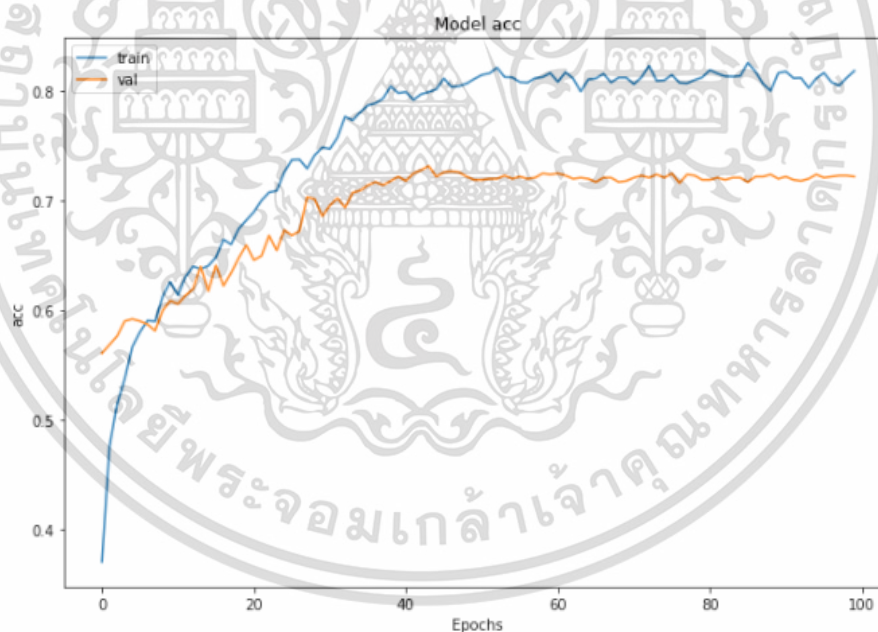
จากตารางที่ 4.1 จะพบว่าจากค่า F1 Score แล้วสถาปัตยกรรมโมเดลนี้สามารถจำแนกโรคของผู้ป่วยแต่ละประเภทจากประสิทธิภาพมากที่สุดถึงน้อยที่สุด โดยเรียงลำดับดังนี้ 1. ผู้ป่วยปกติ 2. ผู้ป่วยโรคสมองเสื่อม 3. ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มต้น 4. ผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย

ตารางที่ 4.2 การวัดประสิทธิภาพของ InceptionV3 ด้วยชุดข้อมูลก่อน Over Sampling

Metrics	Score
Balanced Accuracy Score	66.34%
Matthew's Correlation Coefficient	59.15%

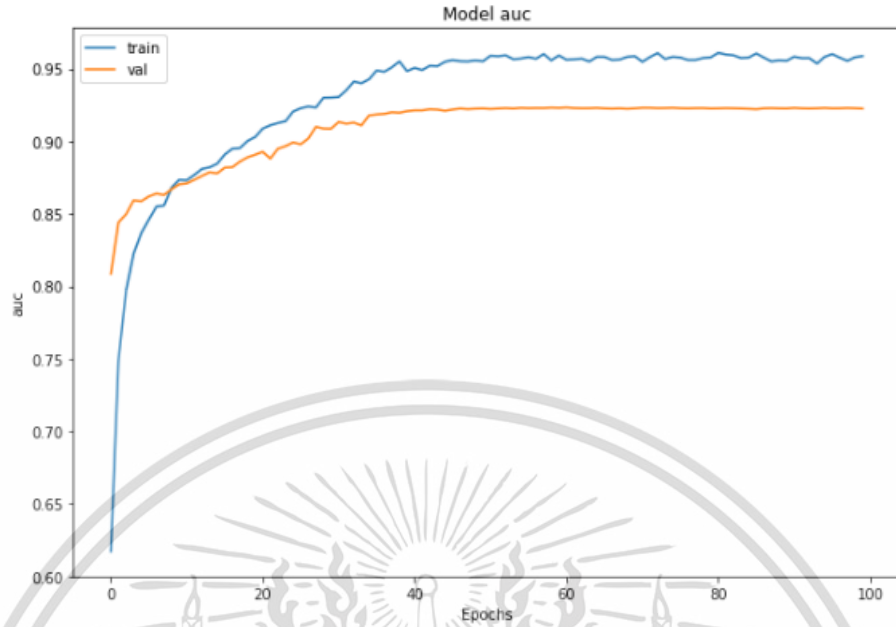
4.2 ผลการทดลองสถาปัตยกรรม InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling

ผลการทดลองนี้จะประกอบไปด้วยกราฟ Accuracy, AUC, Loss ,Confusion Matrix, ค่า Precision, Recall และ F1-Score ของผู้ป่วยแต่ละประเภท, ค่า Balance Accuracy Score และ Matthew's Correlation Coefficient ดังรูปที่ 4.5, 4.6, 4.7, 4.8 และตารางที่ 4.3 ,4.4 ตามลำดับ

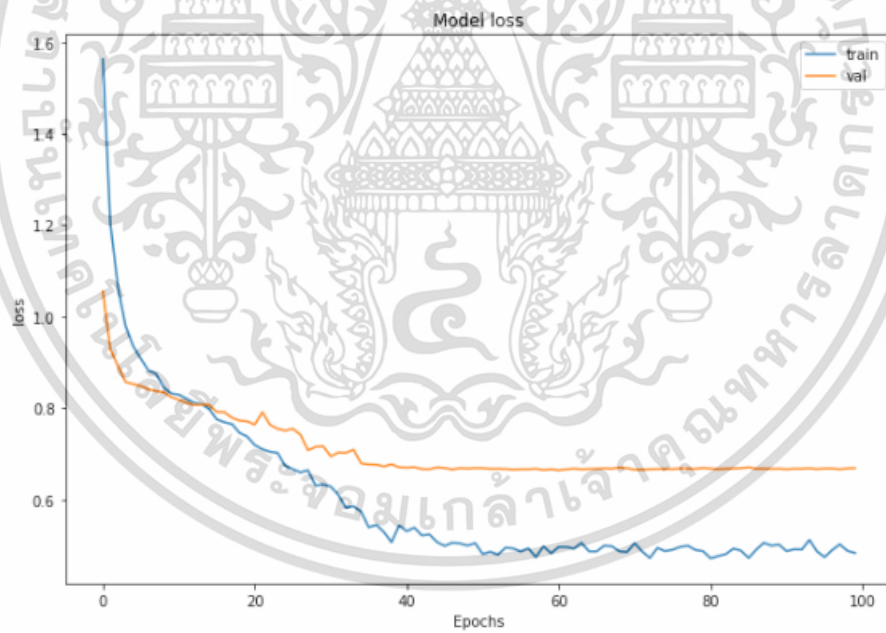


รูปที่ 4.5 กราฟ Accuracy ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

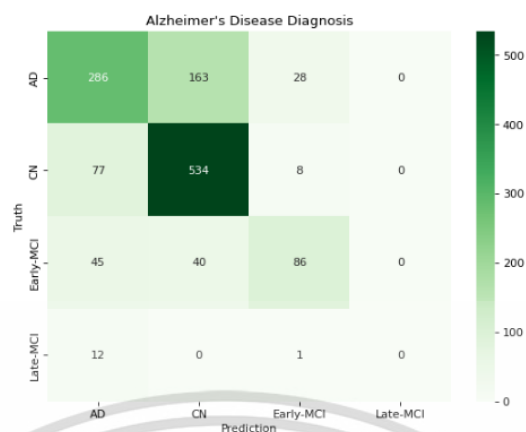


รูปที่ 4.6 กราฟ AUC ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling



รูปที่ 4.7 กราฟ Loss ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 กราฟ Confusion Matrix ของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling

จากรูปที่ 4.8 สามารถนำหาค่า Precision, Recall, F1-Score และ Support เพื่อวัดประสิทธิภาพของสถาปัตยกรรมโมเดลในการจำแนกโรคแต่ละประเภทดังตารางที่ 4.3 และสามารถวัดประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionRestnetV2 ได้ดังตารางที่ 4.4

ตารางที่ 4.3 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling

	Precision	Recall	F1-Score	Support
AD	0.68	0.60	0.64	477
CN	0.72	0.86	0.79	619
Early-MCI	0.70	0.67	0.67	171
Late-MCI	0.37	0.40	0.37	13

จากตารางที่ 4.3 จะพบว่าจากค่า F1 Score แล้วสถาปัตยกรรมโมเดลนี้สามารถจำแนกโรคของผู้ป่วยแต่ละประเภทจากประสิทธิภาพมากที่สุดถึงน้อยที่สุด โดยเรียงลำดับดังนี้ 1. ผู้ป่วยปกติ 2. ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มต้น 3. ผู้ป่วยโรคสมองเสื่อม 4. ผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย

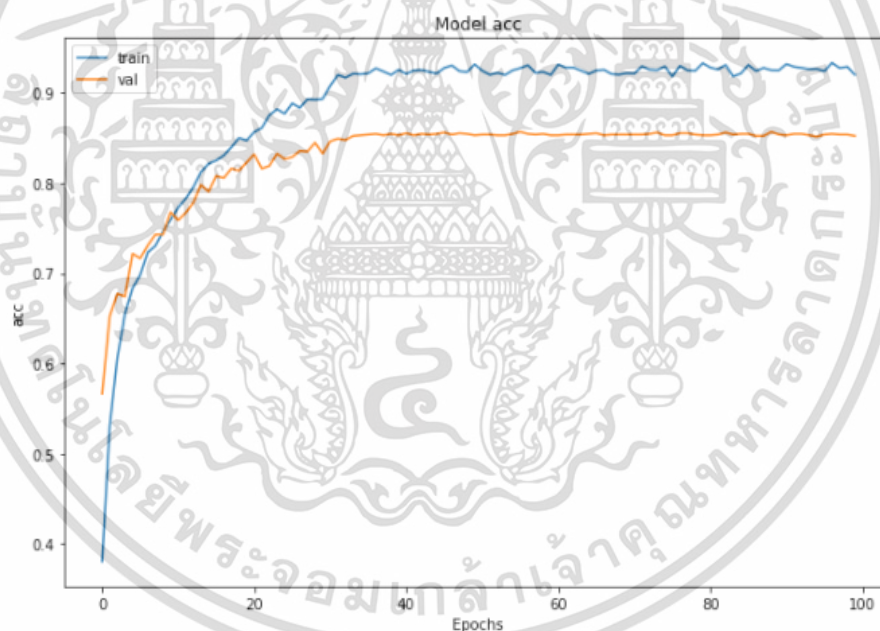
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การวัดประสิทธิภาพของ nceptionRestnetV2 ด้วยชุดข้อมูลก่อน Over Sampling

Metrics	Score
Balanced Accuracy Score	49.13%
Matthew's Correlation Coefficient	50.76%

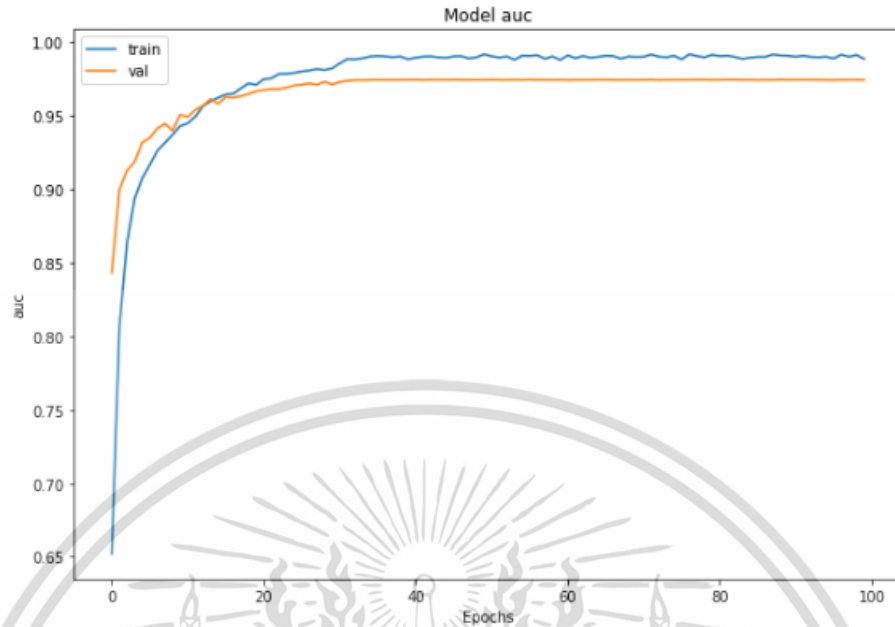
4.3 ผลการทดลองสถาปัตยกรรม InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling

ผลการทดลองนี้จะประกอบไปด้วยกราฟ Accuracy, AUC, Loss ,Confusion Matrix, ค่า Precision, Recall และ F1-Score ของผู้ป่วยแต่ละประเภท, ค่า Balance Accuracy Score และ Matthew's Correlation Coefficient ดังรูปที่ 4.9, 4.10, 4.11, 4.12 และตารางที่ 4.5 ,4.6 ตามลำดับ

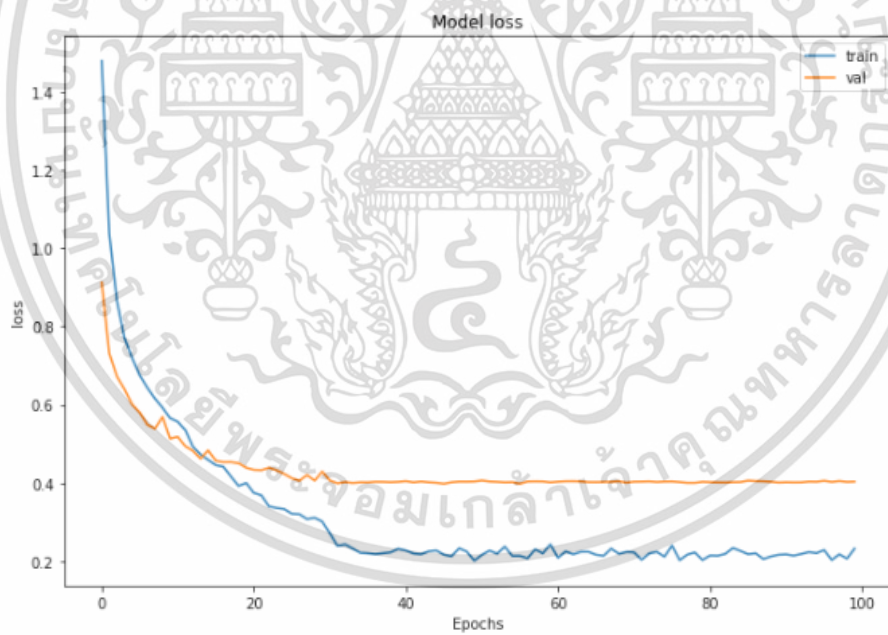


รูปที่ 4.9 กราฟ Accuracy ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

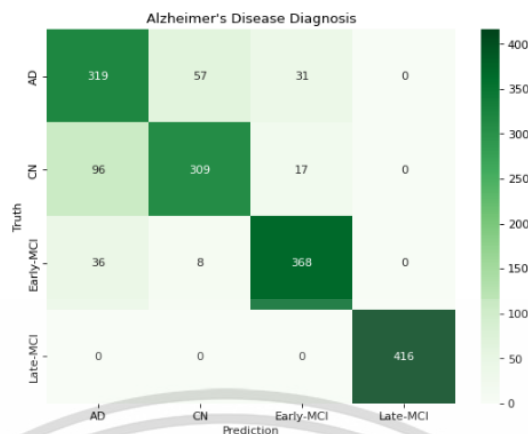


รูปที่ 4.10 กราฟ AUC ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling



รูปที่ 4.11 กราฟ Loss ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 กราฟ Confusion Matrix ของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling

จากรูปที่ 4.12 สามารถนำหาค่า Precision, Recall, F1-Score และ Support เพื่อวัดประสิทธิภาพของสถาปัตยกรรมโมเดลในการจำแนกโรคแต่ละประเภทดังตารางที่ 4.5 และสามารถวัดประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionV3 ได้ดังตารางที่ 4.6

ตารางที่ 4.5 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling

	Precision	Recall	F1-Score	Support
AD	0.71	0.78	0.74	407
CN	0.83	0.73	0.78	422
Early-MCI	0.88	0.89	0.89	412
Late-MCI	1.00	1.00	1.00	416

จากตารางที่ 4.5 จะพบว่าจากค่า F1 Score แล้วสถาปัตยกรรมโมเดลนี้สามารถจำแนกโรคของผู้ป่วยแต่ละประเภทจากประสิทธิภาพมากที่สุดถึงน้อยที่สุด โดยเรียงลำดับดังนี้ 1. ผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย 2. ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มต้น 3. ผู้ป่วยปกติ 4. ผู้ป่วยโรคสมองเสื่อม

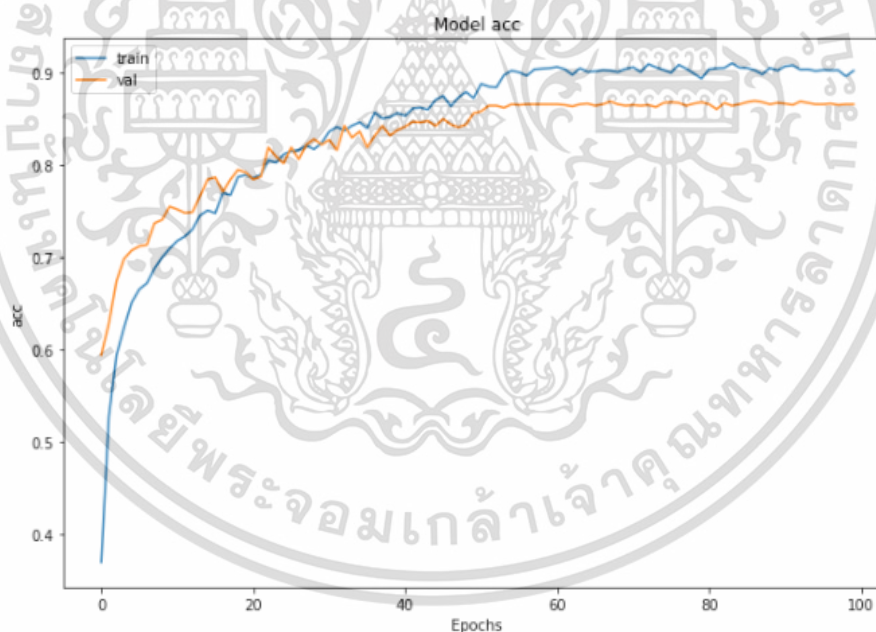
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 การวัดประสิทธิภาพของ InceptionV3 ด้วยชุดข้อมูลหลัง Over Sampling

Metrics	Score
Balanced Accuracy Score	85.23%
Matthew's Correlation Coefficient	80.37%

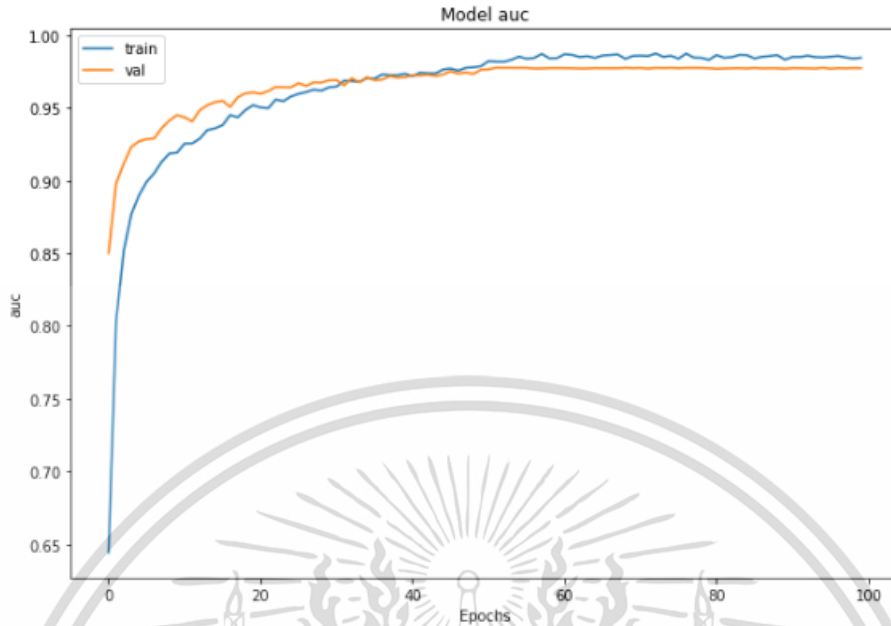
4.4 ผลการทดลองสถาปัตยกรรม InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling

ผลการทดลองนี้จะประกอบไปด้วยกราฟ Accuracy, AUC, Loss ,Confusion Matrix, ค่า Precision, Recall และ F1-Score ของผู้ป่วยแต่ละประเภท, ค่า Balance Accuracy Score และ Matthew's Correlation Coefficient ดังรูปที่ 4.13, 4.14, 4.15, 4.16 และตารางที่ 4.7 ,4.8 ตามลำดับ

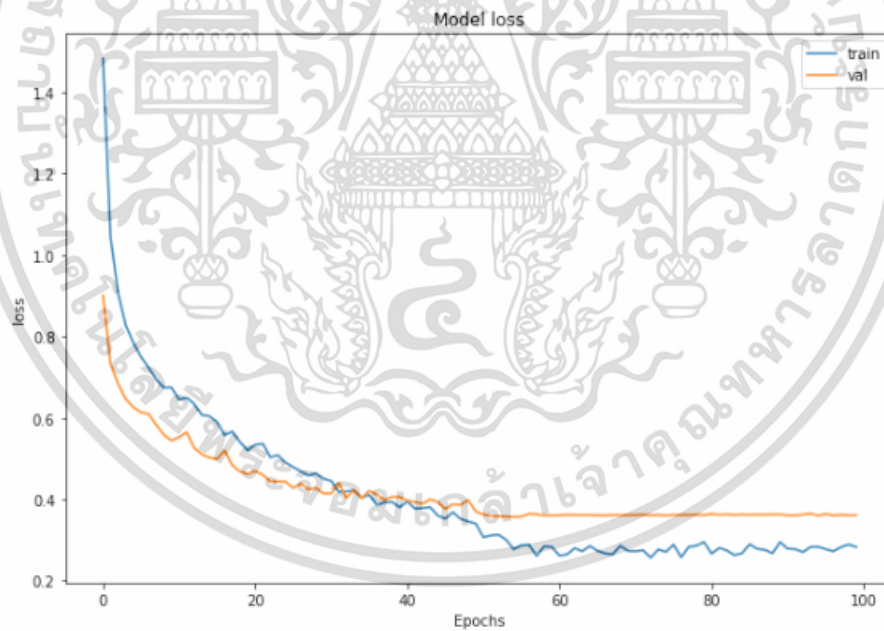


รูปที่ 4.13 กราฟ Accuracy ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

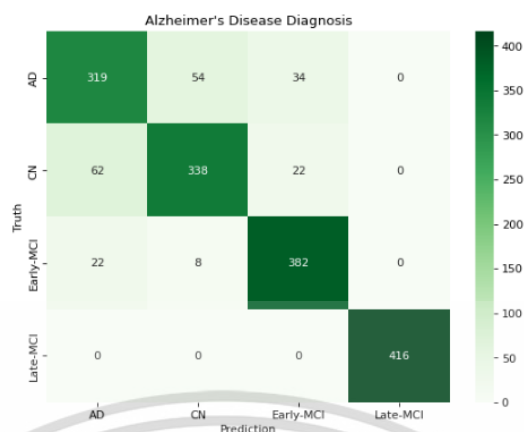


รูปที่ 4.14 กราฟ AUC ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling



รูปที่ 4.15 กราฟ Loss ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 กราฟ Confusion Matrix ของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling

จากรูปที่ 4.16 สามารถนำหาค่า Precision, Recall, F1-Score และ Support เพื่อวัดประสิทธิภาพของสถาปัตยกรรมโมเดลในการจำแนกโรคแต่ละประเภทดังตารางที่ 4.7 และสามารถวัดประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionRestnetV2 ได้ดังตารางที่ 4.8

ตารางที่ 4.7 การวัดประสิทธิภาพการจำแนกแต่ละประเภทผู้ป่วยของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling

	Precision	Recall	F1-Score	Support
AD	0.79	0.78	0.79	407
CN	0.84	0.80	0.82	422
Early-MCI	0.87	0.93	0.90	412
Late-MCI	1.00	1.00	1.00	416

จากตารางที่ 4.7 จะพบว่าจากค่า F1 Score แล้วสถาปัตยกรรมโมเดลนี้สามารถจำแนกโรคของผู้ป่วยแต่ละประเภทจากประสิทธิภาพมากที่สุดถึงน้อยที่สุด โดยเรียงลำดับดังนี้ 1. ผู้ป่วยภาวะความถดถอยทางสมองระยะสุดท้าย 2. ผู้ป่วยภาวะความถดถอยทางสมองระยะเริ่มต้น 3. ผู้ป่วยปกติ 4. ผู้ป่วยโรคสมองเสื่อม

ตารางที่ 4.8 การวัดประสิทธิภาพของ InceptionRestnetV2 ด้วยชุดข้อมูลหลัง Over Sampling

Metrics	Score
Balanced Accuracy Score	87.80%
Matthew's Correlation Coefficient	83.77%

จากตารางที่ 4.2, 4.4, 4.6 และ 4.8 สามารถสร้างตารางเพื่อเปรียบเทียบค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionV3 และ InceptionRestnetV2 ระหว่างการทดลองกับชุดข้อมูลก่อนทำการ Over Sampling ด้วยวิธีการ SMOTE กับชุดข้อมูลหลังทำการ Over Sampling ด้วยวิธีการ SMOTE ดังตารางที่ 4.9

ตารางที่ 4.9 ตารางเปรียบเทียบประสิทธิภาพสถาปัตยกรรมก่อนและการทำงาน Over Sampling

Metrics	Score			
	InceptionV3		InceptionRestnetV2	
	Without Over Sampling	With Over Sampling	Without Over Sampling	With Over Sampling
Balanced Accuracy Score	66.34%	85.23%	49.13%	87.80%
Matthew's Correlation Coefficient	59.15%	80.37%	50.76%	83.77%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงานวิจัย

จากผลการทดลองการเปรียบเทียบระหว่างชุดข้อมูลที่ทำการ Over Sampling กับชุดข้อมูลที่ไม่สมดุลกัน (Imbalance Datasets) ที่นำเข้าสู่กระบวนการจำแนกประเภทของโรคผู้ป่วยด้วยสถาปัตยกรรมโมเดล InceptionV3 และ InceptionRestnetV2 จะพบว่าชุดข้อมูลที่ผ่านการ Over Sampling จะส่งผลให้ทั้ง 2 สถาปัตยกรรมโมเดลนั้นมีประสิทธิภาพหรือความแม่นยำที่มากขึ้น โดยค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionV3 กับชุดข้อมูลก่อน Over Sampling มีค่า Balance Accuracy Score 66.34% และ Matthew's Correlation Coefficient 59.15% และค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionV3 กับชุดข้อมูลหลัง Over Sampling มีค่า Balance Accuracy Score 85.23% และ Matthew's Correlation Coefficient 80.37% และค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionRestV2 กับชุดข้อมูลก่อน Over Sampling มีค่า Balance Accuracy Score 49.13% และ Matthew's Correlation Coefficient 50.76% และค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionRestnetV2 กับชุดข้อมูลหลัง Over Sampling มีค่า Balance Accuracy Score 87.80% และ Matthew's Correlation Coefficient 83.77%

จากการทดลองเปรียบเทียบการเปรียบเทียบสถาปัตยกรรมโมเดลกับทั้ง 2 ชุดข้อมูล จะพบว่าหลังจากทำการ Over Sampling เพื่อเพิ่มประสิทธิภาพและความแม่นยำของสถาปัตยกรรมแล้วสถาปัตยกรรมโมเดล InceptionRestnetV2 จะให้ประสิทธิภาพและความแม่นยำที่มากกว่า InceptionV3 แต่จะใช้เวลาในการคำนวณที่มากกว่าด้วยเช่นกัน โดยค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionRestnetV2 มีค่า Balance Accuracy Score 87.80% และ Matthew's Correlation Coefficient 83.77% และค่าประสิทธิภาพของสถาปัตยกรรมโมเดล InceptionV3 มีค่า Balance Accuracy Score 85.23% และ Matthew's Correlation Coefficient 80.37%

5.2 ข้อเสนอแนะ

เนื่องจากข้อจำกัดด้วยทรัพยากร และเวลา เลยส่งผลให้งานวิจัยนี้ไม่ได้ดีเท่าที่ควร ทั้งนี้ผู้วิจัยมีความเห็นว่าสามารถปรับปรุงและพัฒนาได้ต่อไปในอนาคต ดังนี้

5.2.1 การใช้ชุดข้อมูลที่เป็น 3 มิติในการจำแนก

จากการวิจัยพบว่า การจำแนกแค่ 2 มิติอาจไม่ได้ให้ผลทำนายที่ครอบคลุมทั้งหมด การใช้ชุดข้อมูล 3 มิติ จะสามารถให้ผลการจำแนกที่ลึกกว่าและสามารถบอกถึงปัญหาจากสมองส่วนใด จนสามารถทำให้รังสีแพทย์สามารถตัดสินใจได้ง่ายขึ้น

5.2.2 การใช้สีพื้นที่ความร้อน (Heat Map)

จากการวิจัยพบว่า ภาพที่นำมาจำแนกจะมีลักษณะที่บ่งบอกถึงประเภทของผู้ป่วยได้ แต่ก็มีโอกาสที่จะทำนายผิดพลาดเนื่องจากสีของภาพที่คล้ายกัน ดังนั้นการใช้สีพื้นที่ความร้อน ณ จุดที่บ่งบอกได้ว่าผู้ป่วยคนนี้เป็น ผู้ป่วยประเภทใด จะสามารถทำให้สถาปัตยกรรมโมเดลแม่นยำและมีประสิทธิภาพมากขึ้น

บรรณานุกรม

- [1] Shaswati Roy & Pradipta Maji. (2015). A simple skull stripping algorithm for brain MRI. doi: 10.1109/ICAPR.2015.7050671
- [2] Vitaly Bushaev. (2018). Understanding RMSprop — faster neural network learning. [Online]. Available: <https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a>
- [3] Dario Radečić. (2020). Softmax Activation Function Explained. [Online]. Available: <https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60>
- [4] Yiming Ding et al. (2018). A Deep Learning Model to Predict a Diagnosis of Alzheimer Disease by Using 18F-FDG PET of the Brain. doi: 10.1148/radiol.2018180958
- [5] Google. (2021). Advanced Guide to Inception v3 on Cloud TPU. [Online]. Available: <https://cloud.google.com/tpu/docs/inception-v3-advance>
- [6] Saman Sarraf & Ghassem Tofghi. (2016). DeepAD: Alzheimer’s Disease Classification via Deep Convolutional Neural Networks using MRI and fMRI. doi: <https://doi.org/10.1101/070441>
- [7] Stanford. CS 230 - Deep Learning. [Online]. Available: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>
- [8] Jyoti Islam. (2018). Brain MRI analysis for Alzheimer’s disease diagnosis using an ensemble system of deep convolutional neural networks. doi: 10.1186/s40708-018-0080-3

- [9] Kaiming He et al. (2015). Deep Residual Learning for Image Recognition. doi: 10.1109/cvpr.2016.90
- [10] Deepak Battini. (2018). Implementing Drop Out Regularization in Neural Networks. [Online]. Available: <https://www.tech-quantum.com/implementing-drop-out-regularization-in-neural-network>
- [11] Yechong Huang et al. (2019). Diagnosis of Alzheimer's Disease via Multi-Modality 3D Convolutional Neural Network. doi: 10.3389/fnins.2019.00509
- [12] Kasper Marstal et al. (2016). SimpleElastix: A User-Friendly, Multilingual Library for Medical Image Registration. doi: 10.1109/CVPRW.2016.78
- [13] Hanafy M. Ali. (2017). MRI Medical Image Denoising by Fundamental Filters. doi: 10.5772/intechopen.72427
- [14] S. Ahmed et al. (2019). Ensembles of Patch-Based Classifiers for Diagnosis of Alzheimer Diseases. doi: 10.1109/ACCESS.2019.2920011
- [15] Eduardo Nigri et al. Explainable Deep CNNs for MRI-Based Diagnosis of Alzheimer's Disease [Online]. Available: <https://deepai.org/publication/explainable-deep-cnns-for-mri-based-diagnosis-of-alzheimer-s-disease>