



Project Report

เรื่อง Diabetic Retinopathy Detection

เสนอ

ผศ.ดร. นริศ หนูหอม

จัดทำโดย

นาย ชนาวีร์	เสถียรธีราภาพ	รหัสนักศึกษา	6313124
นางสาว ปุณยา	หาสินอนันต์	รหัสนักศึกษา	6313134
นางสาว วรนิษฐ์	ดวงแข	รหัสนักศึกษา	6313140

รายงานนี้เป็นส่วนหนึ่งของวิชา EGCO 486 Image Processing

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล

ภาคเรียนที่ 2 ปีการศึกษา 2566

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา EGCO486 Image Processing จัดทำขึ้นเพื่อใช้ประกอบการเรียนการสอนในรายวิชา ซึ่งคณะผู้จัดทำได้รับมอบหมายให้ออกแบบ Model สำหรับจำแนกประเภทเบาหวานขึ้นจอประสาทตา โดยทางคณะผู้จัดทำได้ทำการศึกษาค้นคว้าเพิ่มเติม ฝึกฝนทักษะ และนำสิ่งที่ได้ศึกษาค้นคว้ามาออกแบบ Model สำหรับจำแนกประเภทเบาหวานขึ้นจอประสาทตาเก็บไว้เป็นประโยชน์ต่อการเรียนการสอนของตนเองและรายวิชาต่อไป

คณะผู้จัดทำได้ไปศึกษาค้นคว้า รวบรวมและเรียบเรียงออกมาเป็นรายงานฉบับนี้ ทั้งนี้เนื้อหาประกอบไปด้วยขั้นตอนการดำเนินงาน และค่าความแม่นยำจากโมเดล ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานฉบับนี้จะเป็นประโยชน์แก่ผู้อ่านไม่มากนักน้อย

คณะผู้จัดทำ

นาย ชนาวีร์ เสถียรธรรภาพ

นางสาว ปุณยา หาสินอนันต์

นางสาว วรนิษฐ์ ดวงแข

สารบัญ

เนื้อหา

บทนำ.....	1
- ที่มาและความสำคัญ.....	1
- วัตถุประสงค์.....	1
รายละเอียดชุดข้อมูล.....	2
- ชุดข้อมูล.....	2
- ชุดข้อมูลสำหรับฝึกโมเดล.....	2
- ชุดข้อมูลสำหรับทดสอบโมเดล.....	3
- ตัวอย่างชุดข้อมูล.....	3
- วิเคราะห์ชุดข้อมูล.....	5
ขั้นตอนการดำเนินงาน.....	7
- Machine Learning Model.....	7
- การเตรียมข้อมูล (Data Preprocessing)	7
- Image Resize.....	7
- ทำการลบ Noise ออกจากรูปภาพ.....	7
- ทำการปรับสมดุลความเข้มของภาพ.....	7
- การสร้างชุดข้อมูล (Dataset Creation).....	7
- Microaneurysms และ Exudates Features	7
- Vessels Features.....	8
- Texture Features.....	8
- Color Features.....	8
- Histogram Features.....	9

- การทดลองจับคู่และรวมฟีเจอร์ในการสร้าง Dataset.....	9
- Lesions and Vessels.....	10
- Texture Features.....	10
- Color Features and Histogram Features.....	10
- Lesions, Vessels, and Texture Features.....	11
- All Features.....	11
- การสร้างโมเดล.....	12
- Deep Learning Model.....	14
- การเตรียมข้อมูล (Data Preprocessing)	14
- Image Resize and Save Process.....	14
- Preprocess Image Process.....	14
- Crop Image from Gray.....	14
- Circle Crop.....	14
- Display Gaussian Example.....	14
- Display Crop Gaussian Example.....	15
- Data Generator.....	15
- โมเดลที่เลือกใช้ในการสร้าง Deep Learning.....	16
- ResNet50.....	16
- InceptionV3.....	16
- DenseNet169.....	16
- ขั้นตอนการสร้างโมเดล.....	16
- ประเมินผลโมเดล Deep Learning.....	18
สรุปผลการดำเนินงาน.....	23
บรรณานุกรม.....	24

บทนำ

ที่มาและความสำคัญ

ผู้ป่วยที่เป็นโรคเบาหวานจะมีระดับน้ำตาลในเลือดสูงผิดปกติ ส่งผลให้ผนังหลอดเลือดฝอยเสื่อมทั่วร่างกายรวมทั้งหลอดเลือดฝอยที่จอตาด้วย เลือดและสารต่างๆจะรั่วซึมออกจากหลอดเลือดที่ผิดปกติเหล่านี้และทำให้เกิดภาวะเบาหวานขึ้นจอตา (diabetic retinopathy, DR) ซึ่ง สามารถแบ่งได้เป็น 5 ระยะตามความรุนแรงของโรค no-DR, mild-DR, moderate-DR, severe-DR, และ proliferative-DR ตามลำดับ ผลที่ตามมาของโรคเบาหวานขึ้นจอตาทำให้หลอดเลือดขยายใหญ่ขึ้นและมีของเหลวหรือเลือดรั่วไหลออกมา การตาบอดอาจเกิดขึ้นได้ในกรณีที่อยู่ในระยะ severe-DR และ 2.6% ของการตาบอดทั่วโลกเกิดจากอาการเบาหวานขึ้นตา ปัจจุบันผู้ป่วยเป็นโรคนี้มาอย่างยาวนานและมีแนวโน้มที่จะเพิ่มมากขึ้น การคัดกรองผู้ป่วยที่เป็นโรคเบาหวานขึ้นตาเป็นประจำเป็นสิ่งจำเป็นเพื่อระบุ และรักษาอย่างรวดเร็ว เพื่อลดอาการ เพื่อลดความเสี่ยงของการตาบอด

ในยุคสมัยที่เทคโนโลยีคอมพิวเตอร์ได้กลายเป็นเทคโนโลยีสำคัญที่เข้ามาเกี่ยวข้องกับชีวิตในหลายๆ ด้าน เทคโนโลยีปัญญาประดิษฐ์ (artificial intelligence, AI) เป็นเทคโนโลยีที่ได้เข้าไปมีบทบาทในวิทยาศาสตร์หลากหลายสาขา ในสาขาสุขภาพและการแพทย์ก็เป็นอีกสาขาหนึ่งที่นำเอาเทคโนโลยี AI มาใช้ประโยชน์

วัตถุประสงค์

การศึกษานี้มีวัตถุประสงค์เพื่อเสนอระบบตรวจจับภาวะเบาหวานขึ้นจอตา (diabetic retinopathy, DR) อัตโนมัติโดยใช้การประมวลผลล่วงหน้า (preprocessing) การดึงคุณลักษณะ (feature extraction) และขั้นตอนการจำแนกประเภท (classification) มีการใช้แนวทาง Deep Convolutional Neural Networks (DCNN) และ Machine Learning (ML)

รายละเอียดชุดข้อมูล

ชุดข้อมูล (Dataset)

ชุดข้อมูลที่ได้มานั้นประกอบไปด้วย 2 กลุ่ม คือ ชุดข้อมูลสำหรับการ Train และชุดข้อมูลสำหรับการ Test โดยแต่ละกลุ่มจะประกอบไปด้วย ไฟล์ภาพของจอประสาทตา และไฟล์ CSV ที่ประกอบไปด้วยชื่อไฟล์ และประเภทเบาหวานขึ้นจอประสาทตา

โดยประเภทของเบาหวานขึ้นตาแบ่งออกเป็น 5 ประเภทตามระดับความรุนแรง คือ

ระดับ 0 คือ จอประสาทตาที่ไม่เกิดอาการเบาหวานขึ้นจอประสาทตา

ระดับ 1 – 4 คือ ระดับความรุนแรงของเบาหวานขึ้นตาที่เห็นได้จากภาพตามลำดับ

ชุดข้อมูลสำหรับการฝึกโมเดล (Train Dataset)

ชุดข้อมูลสำหรับการฝึกโมเดลจะประกอบไปด้วยไฟล์ภาพของจอประสาทตาจำนวน 4209 ภาพ โดยแต่ละภาพจะมีขนาดไม่เท่ากัน และมีจำนวนภาพแต่ละระดับของเบาหวานขึ้นจอประสาทตาที่ไม่เท่ากัน ดังนี้

- ระดับ 0 : 1948 ภาพ
- ระดับ 1 : 420 ภาพ
- ระดับ 2 : 1051 ภาพ
- ระดับ 3 : 473 ภาพ
- ระดับ 4 : 317 ภาพ

ชุดข้อมูลสำหรับการทดสอบโมเดล (Test Dataset)

ชุดข้อมูลสำหรับการทดสอบโมเดลจะประกอบไปด้วยไฟล์ภาพของจอประสาทตาจำนวน 1968 ภาพ โดยแต่ละภาพจะมีขนาด และจำนวนภาพแต่ละประเภทของภาพเบาหวานขึ้นจอประสาทตาที่ไม่เท่ากัน

- ระดับ 0 : 938 ภาพ
- ระดับ 1 : 197 ภาพ
- ระดับ 2 : 514 ภาพ
- ระดับ 3 : 167 ภาพ
- ระดับ 4 : 152 ภาพ

ตัวอย่างชุดข้อมูล



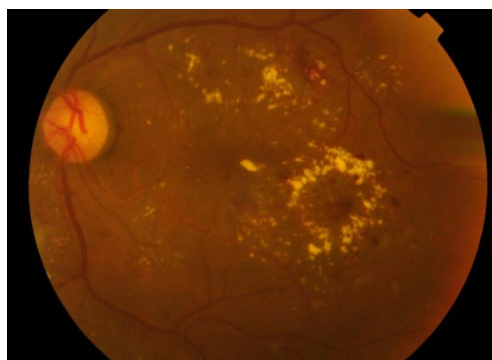
ภาพที่ 1 ภาพจอประสาทตาที่ไม่เกิดอาการเบาหวานขึ้นจอประสาทตา



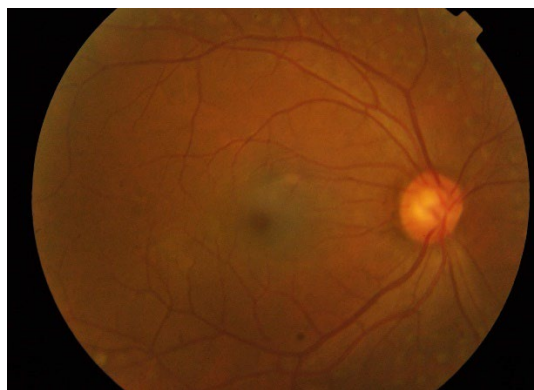
ภาพที่ 2 ภาพจอเบาหวานขึ้นจอประสาทตาที่มีระดับความรุนแรง 1



ภาพที่ 3 ภาพจอเบาหวานขึ้นจอประสาทตาที่มีระดับความรุนแรง 2



ภาพที่ 4 ภาพจอเบาหวานขึ้นจอประสาทตาที่มีระดับความรุนแรง 3



ภาพที่ 5 ภาพจอเบาหวานขึ้นจอประสาทตาที่มีระดับความรุนแรง 4

id_code	diagnosis	id_code	diagnosis
Train_1	2	Test_1	1
Train_2	2	Test_2	3
Train_3	4	Test_3	1
Train_4	2	Test_4	0
Train_5	3	Test_5	3
Train_6	0	Test_6	1
Train_7	0	Test_7	2
Train_8	2	Test_8	0
Train_9	0	Test_9	0
Train_10	2	Test_10	0

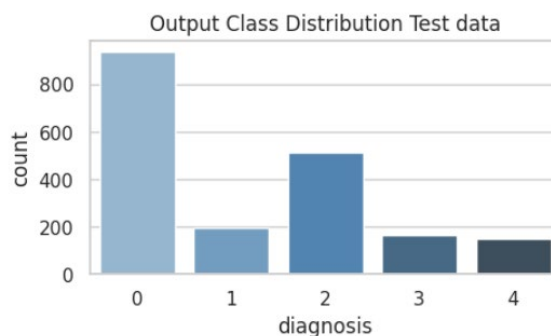
ภาพที่ 6 ตัวอย่างข้อมูลในไฟล์ CSV

วิเคราะห์ชุดข้อมูล (Dataset analysis)

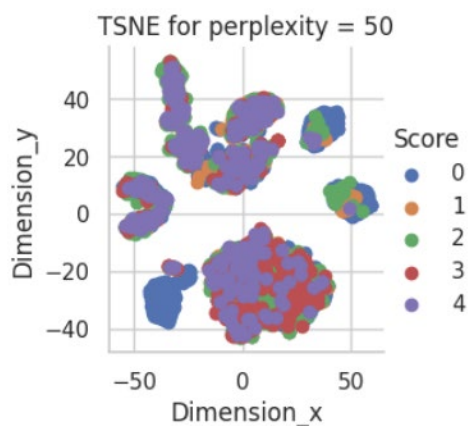
เพื่อให้เข้าใจว่ารูปภาพสามารถแยกออกจากกันในกลุ่มต่างๆ ที่เกี่ยวข้อง สามารถใช้ TSNE เพื่อแสดงภาพใน 2 มิติก่อนได้ ก่อนอื่นเราสามารถแปลงเป็นภาพสีเทา จากนั้นจึงทำให้ภาพเรียบ (flatten) เพื่อสร้างเวกเตอร์ ซึ่งแสดงถึงคุณลักษณะของภาพนั้นได้ โดยกำหนด perplexity ต่างๆ ซึ่งเป็นพารามิเตอร์ของ TSNE ที่ใช้ในการปรับจูนผลลัพธ์



ภาพที่ 7 การกระจายของข้อมูล Train



ภาพที่ 8 การกระจายของข้อมูล Test



ภาพที่ 9 TSNE perplexity = 50

จากผลลัพธ์พบว่าที่ perplexity เท่ากับ 50 จะเห็นการกระจายตัวในการแยกจากกันของคลาสต่าง ๆ ได้ ซึ่งพบว่าคลาส 0 ที่แสดงถึงการไม่พบอาการเบาหวานขึ้นจอประสาทตา (no-DR) เป็นคลาสที่การกระจายตัวโดยไม่ซ้อนทับกับคลาสอื่นมากที่สุด ทำให้สรุปได้ว่าสามารถแยกความแตกต่างของคลาสนี้กับคลาสอื่นได้มากที่สุด ทั้งนี้ผลลัพธ์ที่เกิดขึ้นอาจเกิดจากชุดข้อมูลที่มีภาพของคลาส 0 ที่มากที่สุด

ขั้นตอนการดำเนินการ

ทางคณะผู้จัดทำได้ทำการสร้างและพัฒนาโมเดล Machine Learning และ Deep Learning อย่างละ 3 โมเดล โดยมีรายละเอียด และขั้นตอนการดำเนินการดังนี้

1. Machine Learning Model

1.1 การเตรียมข้อมูล (Data Preprocessing)

1.1.1 Image Resize

ทำการอ่านรูปภาพ และปรับขนาดของรูปภาพให้มีขนาด 512×512 เพื่อให้ภาพมีขนาดมาตรฐานเท่ากันทั้งหมด และเหมาะสมกับการนำเข้าโมเดล

1.1.2 ทำการลบ Noise ออกจากรูปภาพ

ทำการลบ Noise ออกจากรูปภาพ โดยใช้ Median Filtering ขนาด 5×5 เพื่อให้ภาพมีคุณภาพดียิ่งขึ้น

1.1.3 ทำการปรับสมดุลความเข้มของภาพ

ทำการปรับสมดุลความเข้มของภาพ โดยใช้ Contrast Limited Adaptive Histogram Equalization (CLAHE) เพื่อปรับความสว่าง และความเข้มของภาพให้มีความสมดุล และทำการแยกช่องสี โดยนำช่องสีของสีเขียวมาใช้ เนื่องจากในภาพจอประสาทตาช่องสีเขียวมักจะให้รายละเอียดที่ดีกว่าช่องสีอื่นๆ

1.2 การสร้างชุดข้อมูล (Dataset Creation)

จากนั้นได้ทำการสกัดคุณลักษณะ (features) ต่างๆ จากภาพที่ผ่านการ preprocess แล้ว ได้แก่

1.2.1 Microaneurysms และ Exudates Features:

Microaneurysms และ Exudates เป็นสัญญาณสำคัญของโรคเบาหวานขึ้นจอประสาทตา โดย Microaneurysms เป็นจุดเลือดออก ขนาดเล็กบนจอประสาทตา เกิดจากการอุดตันของหลอดเลือดฝอย และ Exudates เป็นของเหลวที่รั่วออกมาจากหลอดเลือดที่ผิดปกติสะสมบนจอประสาทตา ซึ่งวิธีการตรวจจับจะใช้เทคนิค Top-hat

Morphological Operation และ Thresholding ได้ออกมาเป็น Lesions Score คำนวณจากสัดส่วนของพื้นที่ Lesions ต่อพื้นที่ทั้งหมดของภาพ

1.2.2 Vessels Features:

โรคเบาหวานขึ้นจอประสาทตาส่งผลต่อหลอดเลือดในจอประสาทตา ทำให้เกิดการเปลี่ยนแปลง เช่น การบวมของเส้นเลือด, การตีบ หรือการรั่วของเส้นเลือด จึงมองว่าการประเมินความผิดปกติของเส้นเลือดในจอประสาทตาสามารถช่วยในการวินิจฉัยโรคเบาหวานขึ้นจอประสาทตา ซึ่งในงานนี้ใช้เทคนิค Color Thresholding, Denoising, Morphological Operations และ Connected Component Analysis ออกมาเป็น Vessels Score ซึ่งคำนวณจากสัดส่วนของพื้นที่เส้นเลือดต่อพื้นที่ทั้งหมดของภาพ

1.2.3 Texture Features:

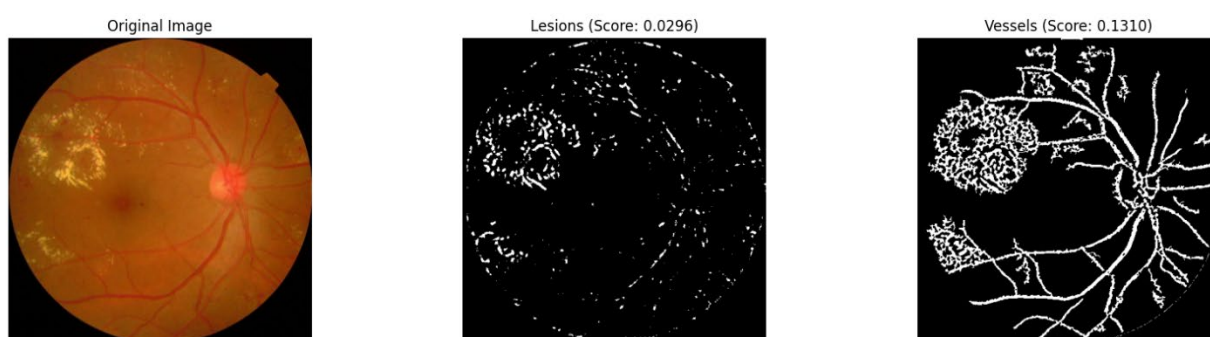
โรคเบาหวานขึ้นจอประสาทตา บางกรณีส่งผลต่อน้ำเยื่อและโครงสร้างของจอประสาทตา ซึ่งสะท้อนให้เห็นในลักษณะเนื้อหาของภาพ โดยใช้เทคนิค Gray-Level Co-occurrence Matrix (GLCM) เพื่อวัดคุณลักษณะเนื้อหาต่างๆ โดยคุณลักษณะที่สกัดได้ ได้แก่ Contrast (ความแตกต่างของพิกเซล), Dissimilarity (ความไม่เหมือนกันของพิกเซล), Homogeneity (ความสม่ำเสมอของพิกเซล), Energy (พลังงานของพิกเซล), และ Correlation (ความสัมพันธ์ของพิกเซล)

1.2.4 Color Features:

โรคเบาหวานขึ้นจอประสาทตาอาจทำให้เกิดการเปลี่ยนแปลงของสีในจอประสาทตา เช่น จุดเลือดออก หรือการสะสมของสารคัดหลั่ง จึงมองว่าการวิเคราะห์ค่าเฉลี่ยและการกระจายของสีในภาพจอประสาทตาสามารถให้ข้อมูลเกี่ยวกับความผิดปกติที่เกี่ยวข้องกับโรคเบาหวานขึ้นจอประสาทตาได้

1.2.5 Histogram Features:

การสกัดคุณลักษณะเป็นการคำนวณฮิสโตแกรมของสีแดง, สีเขียว, และสีน้ำเงินของภาพ โดยจะแสดงการกระจายของความถี่ของสีในภาพ ซึ่งเป็นตัวบ่งชี้ถึงลักษณะเฉพาะของภาพในเชิงสี จากนั้นพีเจอร์ที่สกัดออกมาเป็นเวกเตอร์ของค่าฮิสโตแกรมสีแต่ละสีที่ถูกแปลงเป็นเลขจำนวนเต็มเรียงต่อกัน โดยพีเจอร์ฮิสโตแกรมเป็นเทคนิคเบื้องต้นที่ทำงานได้ดีในการแยกแยะรูปภาพของสิ่งต่างๆ และมีความยืดหยุ่นใช้ในทางการแพทย์ เช่น การวินิจฉัยโรคทางด้านการมองเห็น



ภาพที่ 10 ภาพที่ได้จาก Microaneurysms และ Exudates Features และ Vessels Features

ซึ่งในการวินิจฉัยและติดตามโรคเบาหวานขึ้นจอประสาทตาในทางคลินิก จักษุแพทย์มักใช้วิธีการตรวจหาสัญญาณของโรคเบาหวานขึ้นจอประสาทตา เช่น Microaneurysms, Exudates, เส้นเลือดผิดปกติ, และเลือดออกในจอประสาทตาอยู่แล้ว จึงนำมาเป็นแนวคิดในการทดลองทำ และเมื่อเรานำคุณลักษณะที่สกัดได้จากภาพก่อนหน้ามารวมกับ label ของภาพ (diagnosis) เพื่อสร้างเป็น dataset เบื้องต้น และได้มีการทดลองการจับคู่และรวมพีเจอร์ต่างๆ ในการสร้าง dataset สำหรับการวิเคราะห์ภาพจอประสาทตา

1.3 การทดลองจับคู่และรวมพีเจอร์ในการสร้าง Dataset

เพื่อหาชุดพีเจอร์ที่เหมาะสมสำหรับการจำแนกโรคเบาหวานขึ้นจอประสาทตาจึงได้ทำการทดลองจับคู่และรวมพีเจอร์ต่างๆ ในการสร้าง dataset โดยมีการจับคู่พีเจอร์ 5 รูปแบบดังนี้

1.3.1 Lesions and Vessels:

การรวม Lesions และ Vessels เข้าด้วยกันเพื่อตรวจจับความผิดปกติของโครงสร้างทางกายวิภาคในจอประสาทตา โดย Lesions จะหมายถึงส่วนที่เป็น Microaneurysms และ Exudates ซึ่งเป็นสัญญาณสำคัญของโรคเบาหวานขึ้นจอประสาทตา ในขณะที่ความผิดปกติของเส้นเลือด (Vessels) ก็เป็นอีกลักษณะหนึ่งที่พบได้ในโรคนี้ จึงมองว่าการใช้ฟีเจอร์ทั้งสองร่วมกัน อาจช่วยให้สามารถประเมินทั้งรอยโรคและความผิดปกติของเส้นเลือด ซึ่งเป็นข้อมูลสำคัญในการวินิจฉัยโรค

1.3.2 Texture Features:

การใช้แค่ Texture Features วิเคราะห์ลักษณะพื้นผิวและความแปรปรวนของเนื้อเยื่อในจอประสาทตา โดยในฟีเจอร์จะประกอบไปด้วย Contrast (ความแตกต่างของพิกเซล), Dissimilarity (ความไม่เหมือนกันของพิกเซล), Homogeneity (ความสม่ำเสมอของพิกเซล), Energy (พลังงานของพิกเซล), และ Correlation (ความสัมพันธ์ของพิกเซล) ซึ่งสามารถบรรยายลักษณะเฉพาะของพื้นผิวจอประสาทตาได้ และเมื่อเกิดความผิดปกติของจอประสาทตา อาจส่งผลต่อลักษณะพื้นผิวและความแปรปรวนของเนื้อเยื่อ ดังนั้น Texture Features จึงมีประโยชน์ในการจำแนกบริเวณที่ปกติและผิดปกติของจอประสาทตา

1.3.3 Color Features and Histogram Features:

การรวม Color Features และ Histogram Features เพื่อวิเคราะห์การกระจายของสีและความเข้มสีในภาพจอประสาทตา โดย Color Features ให้ข้อมูลเกี่ยวกับสีโดยรวมในภาพ ในขณะที่ Histogram Features แสดงการกระจายความถี่ของความเข้มสีในแต่ละช่องสี ซึ่งความผิดปกติบางอย่าง เช่น จุดเลือดออกหรือการเปลี่ยนแปลงสี อาจสะท้อนให้เห็นผ่านการเปลี่ยนแปลงของสีและความเข้มสีในภาพจอประสาทตาจึงมองว่าเป็นคู่ที่น่าสนใจ

1.3.4 Lesions, Vessels, and Texture Features:

การรวม Lesions, Vessels และ Texture Features เข้าด้วยกัน เพื่อให้ได้ข้อมูลที่ครอบคลุมทั้งโครงสร้างทางกายวิภาคและลักษณะพื้นผิวของจอประสาทตา ซึ่ง Lesions และ Vessels จะให้ข้อมูลเกี่ยวกับความผิดปกติของโครงสร้าง ในขณะที่ Texture Features ให้ข้อมูลเกี่ยวกับลักษณะพื้นผิวและความแปรปรวนของเนื้อเยื่อ

1.3.5 All Features:

การใช้ฟีเจอร์ทั้งหมดรวมกัน ได้แก่ Lesions, Vessels, Texture Features, Color Features และ Histogram Features เพื่อให้ได้ข้อมูลที่ครอบคลุมและหลากหลายที่สุดในการวิเคราะห์ภาพจอประสาทตา แต่ละชุดฟีเจอร์ให้ข้อมูลที่แตกต่างกันเกี่ยวกับลักษณะของภาพจอประสาทตา เช่น โครงสร้าง พื้นผิว สี และการกระจายความเข้มสีจึงมองว่าการรวมฟีเจอร์ทั้งหมดเข้าด้วยกันช่วยให้โมเดลมีข้อมูลที่หลากหลายในการเรียนรู้และจับรูปแบบที่ซับซ้อนของโรคจอประสาทตาซึ่งอาจช่วยปรับปรุงประสิทธิภาพการจำแนกโรคได้

จากการทดลองจับคู่และรวมฟีเจอร์ในรูปแบบต่างๆ พบว่าการใช้ฟีเจอร์ทั้งหมดรวมกัน (All Features) ให้ผลลัพธ์ที่ดีที่สุดในการจำแนกโรคเบาหวานขึ้นจอประสาทตา โดยโมเดลที่สร้างจาก dataset ที่ใช้ฟีเจอร์ทั้งหมดมีค่าความแม่นยำ (Accuracy) สูงที่สุด รองลงมาคือการใช้ Color Features ร่วมกับ Histogram Features ซึ่งให้ผลลัพธ์ที่ใกล้เคียงกัน ในขณะที่การใช้เพียง Lesions และ Vessels ให้ผลลัพธ์ที่ด้อยที่สุด

เหตุผลที่ใช้ฟีเจอร์ทั้งหมดให้ผลลัพธ์ที่ดีที่สุดอาจเนื่องจากแต่ละชุดฟีเจอร์ให้ข้อมูลที่แตกต่างกันเกี่ยวกับลักษณะของภาพจอประสาทตา เช่น โครงสร้าง พื้นผิว สี และการกระจายความเข้มสี การรวมฟีเจอร์ทั้งหมดเข้าด้วยกันจึงช่วยให้โมเดลมีข้อมูลที่หลากหลายและครอบคลุมในการเรียนรู้และจับรูปแบบที่ซับซ้อนของโรคจอประสาทตา ส่งผลให้สามารถจำแนกโรคได้อย่างแม่นยำมากขึ้น ในทางกลับกัน การใช้เพียง Lesions และ Vessels อาจไม่เพียงพอต่อการจำแนกโรคได้อย่างมีประสิทธิภาพเนื่องจากขาดข้อมูลสำคัญอื่นๆ เช่น ลักษณะพื้นผิวและสีของจอประสาทตา ซึ่งอาจ

บ่งชี้ถึงความผิดปกติที่เกี่ยวข้องกับโรคเบาหวานขึ้นจอประสาทตาได้เช่นกันและสุดท้ายมีการจัดการปัญหา class imbalance โดยใช้เทคนิคของ SMOTE (Synthetic Minority Over-sampling Technique) ซึ่งเป็นวิธีการ over-sampling สำหรับเพิ่มจำนวนตัวอย่างใน class ที่มีน้อย โดยการสร้างตัวอย่างเทียม (synthetic samples) ขึ้นมาจากตัวอย่างเดิมที่มีอยู่ ทำให้ dataset มีจำนวนตัวอย่างที่สมดุลมากขึ้นในแต่ละ class ซึ่งจะช่วยให้โมเดล Machine Learning สามารถเรียนรู้คุณลักษณะของทุก class ได้ดีขึ้น

1.4 การสร้างโมเดล

โมเดลที่ใช้ประกอบไปด้วย 3 โมเดล คือ Gradient Boosting Classifier, Random Forest และ K-Nearest Neighbors

1.4.1 Gradient Boosting Classifier

Gradient Boosting Classifier เป็น Ensemble Learning Algorithm ที่นิยมใช้กันมากในงาน Machine Learning โดยเฉพาะอย่างยิ่งกับปัญหาการจำแนกประเภท (Classification) โดย Gradient Boosting จะทำการสร้าง Decision Tree หลายๆ ต้นแบบต่อเนื่อง (Sequential) โดยในแต่ละรอบจะพยายาม optimize เพื่อแก้ไขข้อผิดพลาดที่เกิดขึ้นในต้นไม้อ่อนหน้า ทำให้ประสิทธิภาพโดยรวมดีขึ้นเรื่อยๆ จนกว่าจะถึงเงื่อนไขที่กำหนด จึงเหมาะกับชุดข้อมูลนี้เพราะ Gradient Boosting สามารถจัดการกับ Feature หลายมิติที่มีความซับซ้อนได้ดี ทำให้ไม่จำเป็นต้องทำ Feature Selection ให้เหลือน้อยลงก่อน โดยโมเดล จะค่อยๆ เรียนรู้ความสำคัญของ Feature แต่ละตัวไปพร้อมๆ กับการสร้างโมเดล

1.4.2 Random Forest Classifier

Random Forest Classifier เป็น Ensemble Learning อีกประเภทหนึ่ง ที่นิยมใช้กันอย่างแพร่หลาย โดยหลักการคือการสร้าง Decision Tree หลายๆ ต้นแบบสุ่ม (Random) และนำผลการทำนายจากทุกต้นมารวมกัน (Aggregation หรือ Bagging) เพื่อให้ได้ผลลัพธ์สุดท้าย โดย Random Forest มักให้

ประสิทธิภาพดีกว่า Decision Tree เดี่ยวๆ และลดปัญหา Overfitting ได้ เนื่องจากการสุ่มเลือก Feature และข้อมูลมาใช้ในแต่ละต้น ทำให้ Tree ไม่ซ้ำกัน จึงเหมาะกับชุดข้อมูลนี้เพราะ Random Forest สามารถจัดการกับ Feature ที่มีจำนวนมากได้ดี เพราะมีการสุ่มเลือก Feature Subset มาสร้าง Tree แต่ละต้น ทำให้ Feature ที่ไม่เกี่ยวข้องหรือสำคัญน้อยไม่ส่งผลกระทบต่อทุกต้น และโมเดลยังคงความแม่นยำสูงได้

1.4.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbor (KNN) เป็น Algorithm แบบ Instance-based Learning ที่อาศัยหลักการเปรียบเทียบความคล้ายคลึงระหว่าง Data Point ที่ต้องการทำนาย กับ Data Point อื่นๆ ใน Training Set โดยพิจารณาจาก Feature ต่างๆ ของ Data นั้น วิธีนี้ไม่มีการสร้าง Model ที่ชัดเจนเหมือน Decision Tree แต่จะจำ Training Data ทั้งหมดเอาไว้ และอาศัยค่า K ที่กำหนด เพื่อหาเพื่อนบ้านใกล้เคียงมาช่วยในการตัดสินใจ โดยเหตุผลที่เลือกลองใช้ KNN กับ Feature ที่เรา Extract มาจากภาพจอประสาทตา เพราะ KNN เป็น Algorithm ที่เหมาะกับปัญหาที่มี Feature ที่ต่อเนื่อง (Continuous) เป็นส่วนใหญ่ ซึ่ง Feature ของเราก็เป็นค่าตัวเลขต่อเนื่องแทบทั้งหมด ทั้งที่ได้จากการวิเคราะห์เนื้อภาพ สี รูปร่าง และองค์ประกอบอื่นๆ โดยในการเทรนโมเดลข้างต้นได้ใช้ค่าพารามิเตอร์เป็น default และใช้เทคนิคอย่าง Cross Validation ในการแบ่ง Training Set ออกเป็น Training และ Validation เพื่อใช้ในการ Tune Model โดยเลือก K-Fold ที่แบ่งข้อมูลออกเป็น K ส่วนเท่าๆกัน แล้วสลับกันเป็น Train/Validation ไปเรื่อยๆ เพื่อประเมินประสิทธิภาพเฉลี่ยจากหลายๆ Split ส่วนในการวัดผล ได้นำโมเดลไปใช้กับ Test Set ที่ไม่เคยนำมาใช้ Train หรือ Validate โมเดลเลย เพื่อช่วยประเมินความสามารถของโมเดลในการทำนายข้อมูลใหม่ได้

2. Deep Learning Model

2.1 การเตรียมข้อมูล (Data Preprocessing)

การเตรียมข้อมูลให้เหมาะสมกับการทำงานของโมเดล และลดความผิดพลาดในการจำแนกประเภทของเบ้าหวานขึ้นจอบประสาทตา โดยมีขั้นตอนดังนี้

2.1.1 Image Resize and Save Process

ทำการอ่านรูปภาพ และปรับขนาดของรูปภาพให้มีขนาด 512×512 เพื่อให้เหมาะสมกับการนำเข้าโมเดล และมีการแบ่งการประมวลผลเป็นหลายกระบวนการ (multithreading) เพื่อเพิ่มประสิทธิภาพในการประมวล หลังจากทีรูปภาพถูกปรับขนาดจะถูกบันทึกลงในไฟล์เดอร์ที่กำหนด

2.1.2 Preprocess Image Process

ทำการตัดส่วนที่ไม่สำคัญออก และปรับขนาดของภาพ เพื่อลดความซับซ้อนของการประมวล และเหมาะสมกับการนำเข้าโมเดล

2.1.3 Crop Image from Gray

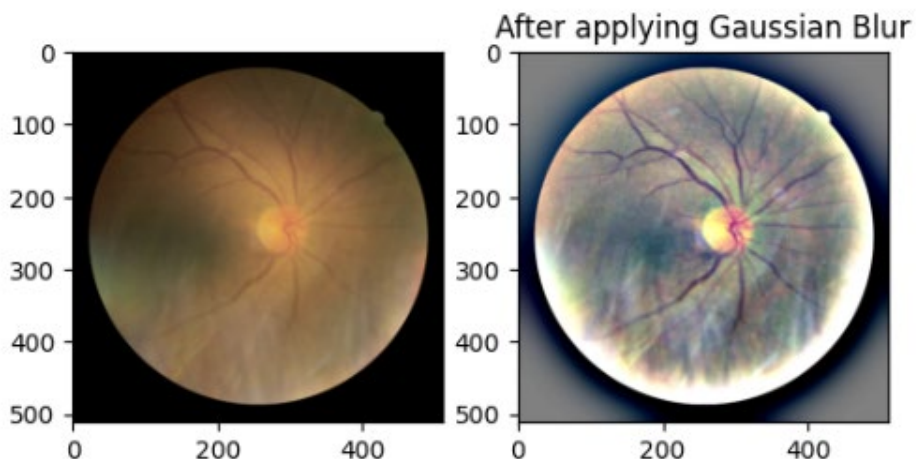
ทำการแปลงภาพเป็นสีเทา (Gray Scale) และทำการตัดขอบของภาพออก เพื่อลดการแสดงผลของพื้นหลังที่ไม่จำเป็น นอกจากนี้การทำการกระบวนการนี้จะช่วยให้ภาพที่ได้มีขนาดเล็กลง และลดความซับซ้อนของการประมวลผลได้

2.1.4 Circle Crop

ทำการตัดส่วนที่ไม่สำคัญของภาพออก โดยใช้วงกลมเพื่อกำหนดส่วนที่ต้องการ และทำการใช้ Gaussian Blur เพื่อทำให้ภาพดูนุ่มนวลมากขึ้น

2.1.5 Display Gaussian Example

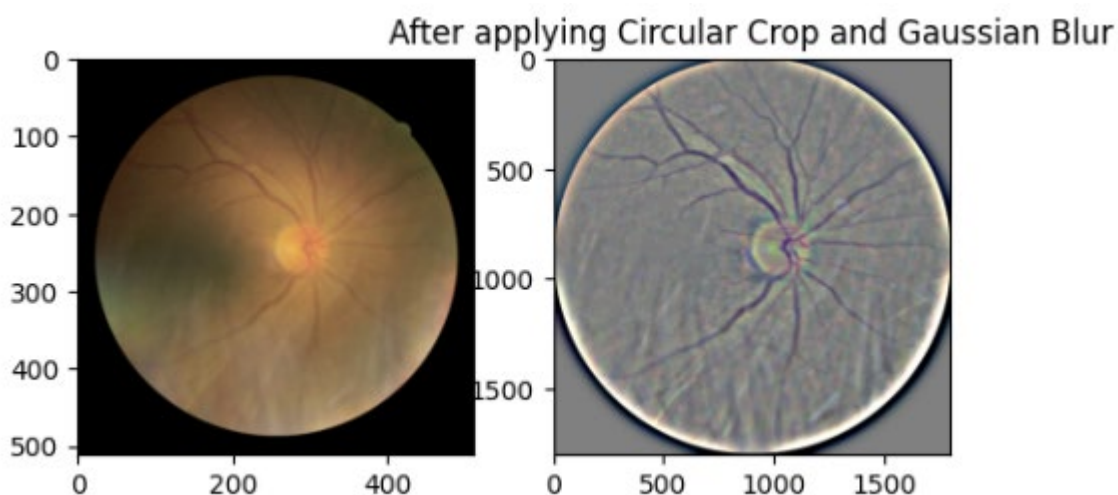
แสดงตัวอย่างของภาพหลังจากการใช้ Gaussian Blur เพื่อให้เห็นการเปลี่ยนแปลงของภาพ



ภาพที่ 11 ตัวอย่างภาพที่ทำ Gaussian Blur

2.1.6 Display Crop Gaussian Example

แสดงตัวอย่างของภาพหลังจากการใช้กระบวนการ Circle Crop และ Gaussian Blur เพื่อให้เห็นการเปลี่ยนแปลงของภาพหลังจากผ่านการประมวลผลทั้งสองขั้นตอน



ภาพที่ 12 ตัวอย่างภาพที่ทำ Circular Crop และ Gaussian Blur

2.1.7 Data Generator

ทำการสร้างภาพเพื่อฝึกสอนเพิ่มเติมจากภาพภายในชุดข้อมูลโดยใช้การแปลงภาพแบบต่างๆ เช่น การเลียนแบบ และการหมุนภาพ เพื่อเพิ่มประสิทธิภาพในการฝึกโมเดล และเพื่อป้องกันการเกิดโอเวอร์ฟิตติ้ง (Overfitting) ในโมเดล

2.2 โมเดลที่เลือกใช้ในการสร้าง Deep Learning

โดยทางคณะผู้จัดทำได้ทำการสร้างโมเดล Deep learning ด้วยกันทั้งหมด 3 โมเดล โดยใช้โมเดล ResNet50, InceptionV3 และ DenseNet169

2.2.1 ResNet50

ResNet50 เป็นโมเดล Convolutional Neural Network (CNN) ที่มีชื่อเสียงด้านการทำนายภาพ โดยมีจำนวนชั้นเป็น 50 ชั้น ซึ่งประกอบไปด้วยชั้น Convolutional, Batch Normalization, Activation, Max Pooling, และ Fully Connected Layers

ResNet50 มีความชำนาญในการจำแนกประเภทของภาพอย่างมีประสิทธิภาพ เช่น การจำแนกประเภทของวัตถุ หรือภาพ X-ray

2.2.2 InceptionV3

โมเดล InceptionV3 เป็นโมเดล Convolutional Neural Network (CNN) ที่ถูกพัฒนาขึ้นโดยทีมงาน Google โดยมีลักษณะการออกแบบที่ซับซ้อน โดยใช้ชั้นการสร้างโมเดลแบบ Inception ซึ่งมีการใช้ Convolution Layers ที่มีขนาด และการเชื่อมต่อที่แตกต่างกัน

2.2.3 DenseNet169

โมเดล DenseNet169 เป็นโมเดล Convolutional Neural Network (CNN) ที่มีความลึก 169 layers ถูกสร้างขึ้นโดยทีมวิจัยของ Facebook AI Research (FAIR) โดยมีคุณสมบัติ และโครงสร้างที่เฉพาะ

โมเดล DenseNet169 เหมาะกับงานที่ต้องการการจำแนกหมวดหมู่ของภาพ เช่น การจำแนกวัตถุในภาพ, การจำแนกสายพันธุ์ของสัตว์ หรือการจำแนกโรคจากรูปภาพของเซลล์ หรือรังไข่

2.3 ขั้นตอนการสร้างโมเดล

2.3.1 ทำการโหลดโมเดล โดยใช้ฟังก์ชัน ‘application’ จากไลบรารี Keras เพื่อใช้ในการสร้างโมเดล

2.3.2 การกำหนดค่า weights ทั้งสามโมเดลจะกำหนด weights = 'imagenet' และกำหนดให้ไม่รวมชั้น Fully Connected Layer ด้วยพารามิเตอร์ 'include_top = False' เพื่อให้เลือกเฉพาะส่วน Convolution Base

2.3.3 เพิ่มชั้น Global Average Pooling และ Dropout เพื่อลดการเกิด overfitting โดยใช้ฟังก์ชัน 'GlobalAveragePooling2D' และ Dropout ตามลำดับ

2.3.4 เพิ่มชั้น Dense ที่มี 2048 หน่วย และใช้ activation function เป็น Relu (activation = 'relu')

2.3.5 ในชั้นสุดท้ายใส่ชั้น Dense ที่มีจำนวนเท่ากับจำนวนคลาสทั้งหมด (n_out) โดยใช้ activation function เป็น softmax (activation = 'softmax')

2.3.6 ใช้ 'Model' จากไลบรารี Keras เพื่อรวมโมเดลด้วย input tensor และ output tensor เพื่อให้ได้โมเดลที่สมบูรณ์ และสามารถเรียกใช้งานได้ง่าย

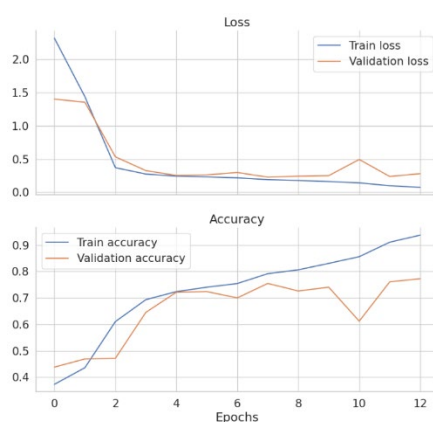
2.3.7 หลังจากทำการสร้างโมเดลเรียบร้อยแล้ว ทำการฝึกโมเดล โดยการฝึกโมเดลจะฝึกเฉพาะ 5 ชั้นสุดท้ายก่อน (Trainable เป็น True เฉพาะ 5 ชั้นสุดท้าย) โดยใช้ optimizer เป็น adam และ loss function เป็น categorical_crossentropy หลังจากนั้นทำการ fine-tune โมเดล โดยในการ fine-tune จะทำการปรับ loss function ให้เป็น binary_crossentropy (loss = 'binary_crossentropy') และ Trainable ทุกชั้นให้เป็น True และใช้ callback ที่จำเป็นในการฝึกโมเดล เช่น EarlyStopping เพื่อหยุดการฝึกเมื่อไม่มีการดีเทรสนในค่า validation loss และ ReduceLROnPlateau เพื่อลดอัตราการเรียนรู้ของ optimizer ในกรณีที่ค่า validation loss ไม่ดี เมื่อฝึกเสร็จจะคืนค่า list ของ callback ที่สร้างขึ้น เมื่อทำการ fine-tune โมเดลเรียบร้อยแล้ว โมเดลที่ถูก fine-tune จะคืนค่าข้อมูลการฝึกโมเดลทั้งหมดในรูปแบบของ history ซึ่งรวมถึงค่า loss และ accuracy ที่ได้ระหว่างการฝึกโมเดล โดยกำหนดค่าพารามิเตอร์ที่ใช้ในการฝึกก่อนการ fine-tune, การ fine-tune ดังนี้

- BATCH_SIZE = 8
- EPOCHS = 30
- WARMUP_EPOCHS = 2
- LEARNING_RATE = 1e-4
- WARMUP_LEARNING_RATE = 1e-3

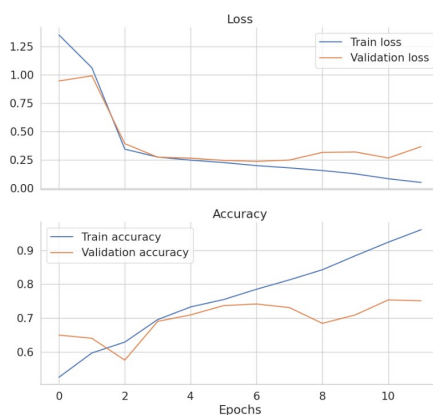
- HEIGHT = IMG_SIZE
- WIDTH = IMG_SIZE
- CHANNEL = 3
- N_CLASSES = df_train_train['diagnosis'].nunique()
- ES_PATIENCE = 5
- RLROP_PATIENCE = 3
- DECAY_DROP = 0.5

2.4 ประเมินผลโมเดล Deep Learning

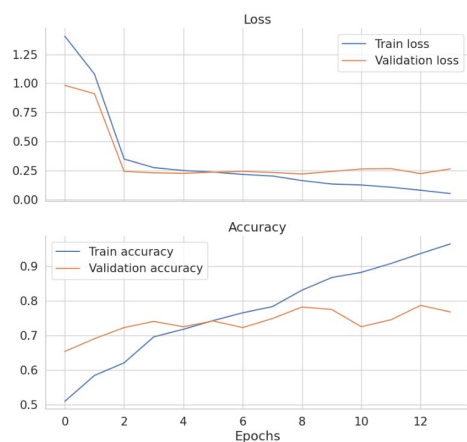
2.4.1 ใช้ขนาดรูปภาพ 512 * 512 และไม่ใช้ EarlyStop



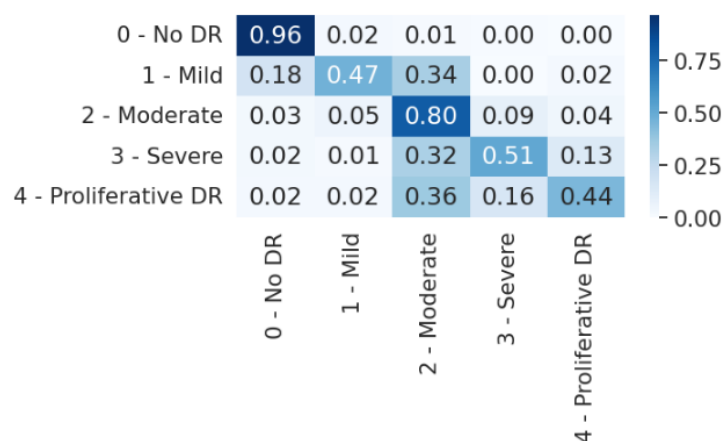
ภาพที่ 13 กราฟแสดง accuracy และ loss ต่อ epochs ของโมเดล ResNet50



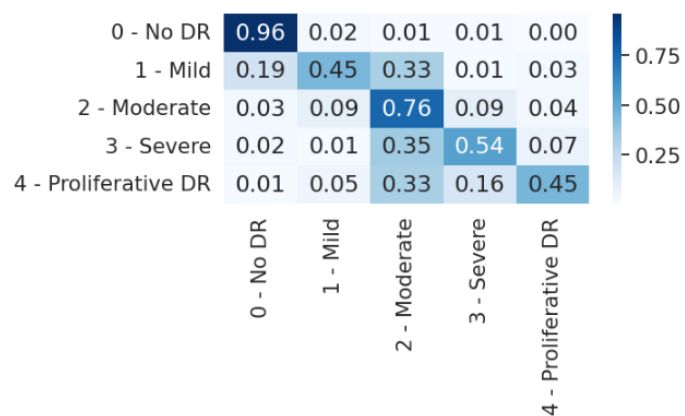
ภาพที่ 14 กราฟแสดง accuracy และ loss ต่อ epochs ของโมเดล InceptionV3



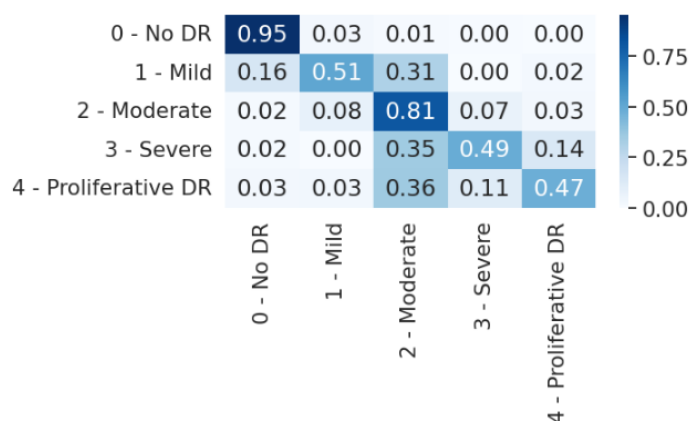
ภาพที่ 15 กราฟแสดง accuracy และ loss ต่อ epochs ของโมเดล DenseNet169



ภาพที่ 16 ภาพ confusion matrix ความถี่การทำนายของชุดข้อมูล test โดยโมเดล ResNet50



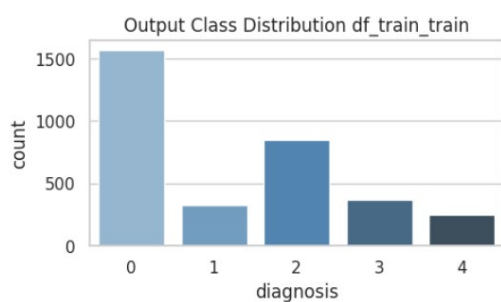
ภาพที่ 17 ภาพ confusion matrix ความถี่การทำนายของชุดข้อมูล test โดยโมเดล InceptionV3



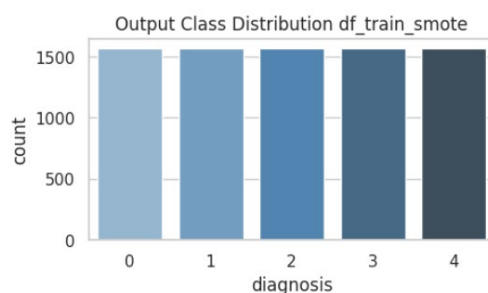
ภาพที่ 18 ภาพ confusion matrix ความถี่การทำนายของชุดข้อมูล test โดยโมเดล DenseNet169

2.4.2 เปรียบเทียบการแก้ Imbalance รูปภาพ ด้วย SMOTE

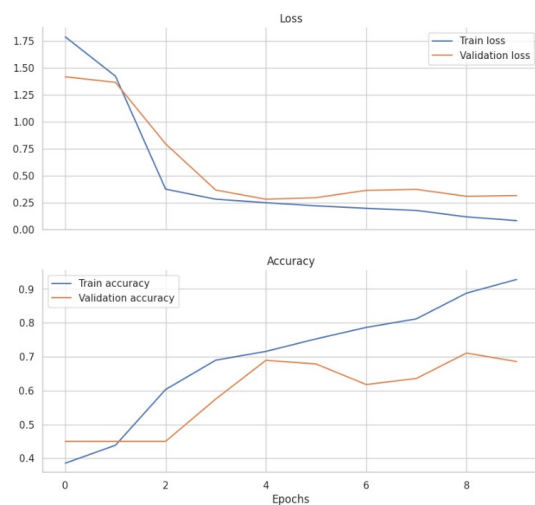
โดยการทดสอบจะใช้ภาพ Train ขนาด 256 * 256 และใช้ Early stop โดยจะการทำ SMOTE หลังจาก Preprocessing image เรียบร้อยแล้ว



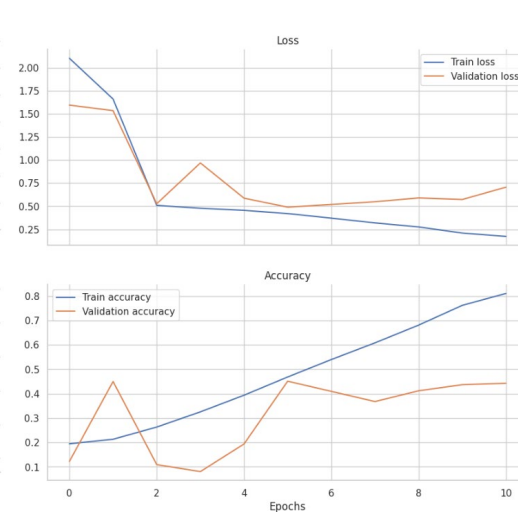
ภาพที่ 19 ภาพการกระจายก่อนการทำ SMOTE



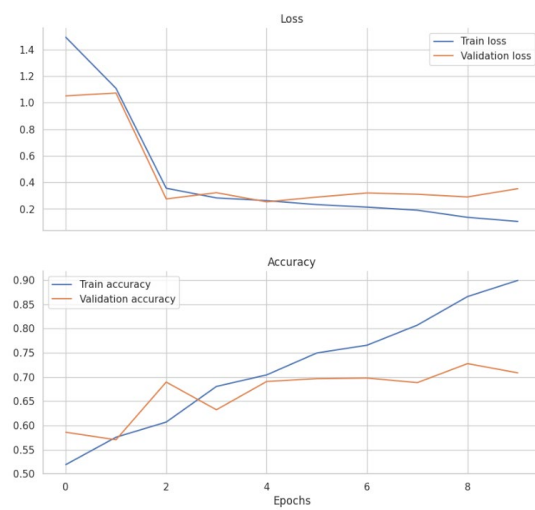
ภาพที่ 20 ภาพการกระจายหลังการทำ SMOTE



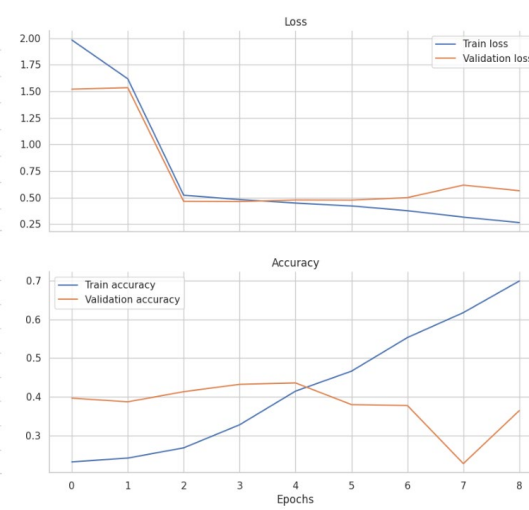
ภาพที่ 21 โมเดล ResNet50 ที่ไม่ทำ SMOTE



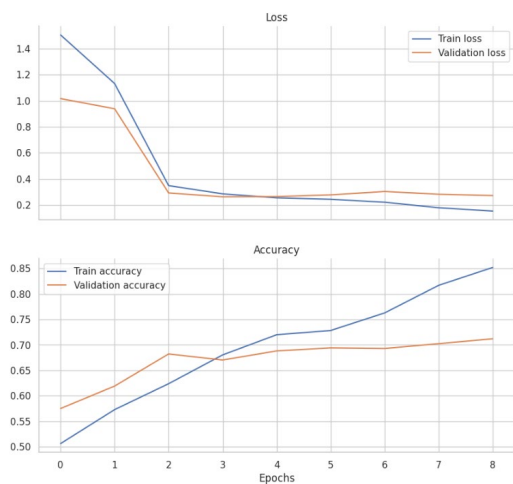
ภาพที่ 22 โมเดล ResNet50 ที่ทำ SMOTE



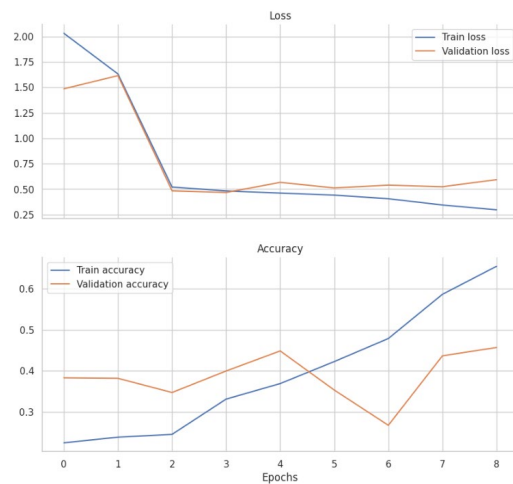
ภาพที่ 23 โมเดล InceptionV3 ที่ไม่ทำ SMOTE



ภาพที่ 24 โมเดล InceptionV3 ที่ทำ SMOTE



ภาพที่ 23 โมเดล InceptionV3 ที่ไม่ทำ SMOTE



ภาพที่ 24 โมเดล InceptionV3 ที่ทำ SMOTE

สรุปผลการดำเนินงาน

Model	Training Accuracy (%)	Test Accuracy (%)
Gradient Boosting	74.06 \pm (1.76)	66.10
Random Forest	90.08 \pm (0.87)	67.58
KNN	83.65 \pm (1.02)	58.58
ResNet50	99.85	79.22
InceptionV3	99.82	78.20
DesnseNet169	99.85	79.52

(ตารางแสดง Accuracy ของแต่ละโมเดล)

จากการทดลองพบว่าในส่วนของ Machine Learning โมเดล Random Forest ให้ความแม่นยำในการทำนาย (predict accuracy) ชุดข้อมูลทดสอบ (test dataset) มากที่สุดที่ 67.58 % โดยลำดับถัดมาจะเป็น Gradient Boosting ที่มีความแม่นยำ 66.10 % และ KNN ที่มีความแม่นยำ 58.58 % ตามลำดับ ในส่วนของ Deep Learning โมเดล DesnseNet169 ให้ความแม่นยำในการทำนายมากที่สุดที่ 79.52 % โดยลำดับถัดมาจะเป็น ResNet50 ที่มีความแม่นยำ 79.22 % และ InceptionV3 ที่มีความแม่นยำ 78.20 % ตามลำดับ โดยการเปรียบเทียบระหว่างการใช้ SMOTE ในการปรับจํานวนรูปภาพที่ไม่สมดุลกันของการทำ Deep Learning Model จากผลลัพธ์พบว่าการทำ SMOTE นั้นจะทำให้ค่า Loss เพิ่มขึ้นและค่า Accuracy ต่ำลง ซึ่งไม่เป็นผลดีต่อโมเดล

บรรณานุกรม

[1] <https://cs231n.stanford.edu/reports/2022/pdfs/20.pdf>

[2]

https://www.researchgate.net/publication/372862444_DIABETIC_RETINOPATHY_DETECTION_USING_MACHINE_LEARNING_AND_DEEP_LEARNING_TECHNIQUES_A_REVIEW

[3]

https://www.researchgate.net/publication/342120641_Diabetic_Retinopathy_Detection_using_Machine_Learning

[4] <https://medium.com/data-espresso/scale-or-standardize-or-normalize-ef905613f275>

[5] https://www.si.mahidol.ac.th/th/division/diabetes/admin/knowledges_files/24_44_1.pdf

[6] <https://keras.io/api/applications/resnet/>

[7] <https://keras.io/api/applications/densenet/>

[8] <https://keras.io/api/applications/inceptionv3/>

[9] <https://dc.etsu.edu/cgi/viewcontent.cgi?article=5868&context=etd>

[10] <https://medium.com/@fatimazahra.belharar/enhancing-classification-accuracy-for-imbalanced-image-data-using-smote-41737783a720>