

การแบ่งภาพและจำแนกความผิดปกติของเซลล์เม็ดเลือดแดง
ด้วยฐานวิทยา
IMAGE SEGMENTATION AND CLASSIFICATION
OF ABNORMAL RED BLOOD CELL BASED ON MORPHOLOGY



การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2567

KMITL-2024-SC-M-017-013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMAGE SEGMENTATION AND CLASSIFICATION
OF ABNORMAL RED BLOOD CELL BASED ON MORPHOLOGY



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

2024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KMITL-2024-SC-M-017-013



COPYRIGHT 2024

SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อการค้นคว้าอิสระ	การแบ่งภาพและจำแนกความผิดปกติของเซลล์เม็ดเลือดแดงด้วยสัญญาณวิทยา
ชื่อนักศึกษา	นายสุกวิทย์ หนูพันธ์
รหัสนักศึกษา	65056090
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการข้อมูลและการวิเคราะห์) ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
พ.ศ.	2567
อาจารย์ที่ปรึกษาการค้นคว้าอิสระ	ดร.จิรภัทร์ หยกรัตน์ศักดิ์

บทคัดย่อ

เลือดเป็นของเหลวชนิดหนึ่งที่ไหลเวียนอยู่ในร่างกายมีปริมาณคิดเป็น 7-8 เปอร์เซ็นต์ของน้ำหนักตัว มีหน้าที่ในการลำเลียงออกซิเจน สารอาหาร และน้ำไปหล่อเลี้ยงเนื้อเยื่อตามร่างกาย รวมถึงนำของเสียจากส่วนต่าง ๆ ไปกำจัดออกจากร่างกาย ในทางโลหิตวิทยาที่มีการศึกษาเกี่ยวกับรูปร่างของเซลล์เม็ดเลือดแดงพบว่า ความผิดปกติที่เกิดขึ้นกับรูปร่างและสีของเซลล์เม็ดเลือดแดงนั้นมีความสัมพันธ์กับภาวะโรคที่อาจเกิดขึ้นได้ การทดลองนี้ได้พัฒนาแบบจำลองโครงข่ายประสาทเทียมคอนโวลูชันทั้ง 3 แบบ คือ สถาปัตยกรรม ResNet50 InceptionV3 และ VGG16 มาทำเป็นตัวแบบเพื่อใช้ในการแยกประเภทความผิดปกติของเซลล์เม็ดเลือดแดง โดยใช้ข้อมูลภาพของเซลล์เม็ดเลือดแดงจำนวน 13 ประเภทที่ได้จากกระบวนการแบ่งภาพด้วยเทคนิค Watershed Segmentation เป็นข้อมูลที่ใช้สำหรับการเรียนรู้ ในการทดลองนี้แบ่งประเภทของชุดข้อมูลออกเป็น 3 ชนิด คือ ข้อมูลที่ไม่มีการแก้ไขปัญหาความสมดุล ข้อมูลที่แก้ไขด้วยการถ่วงน้ำหนัก และข้อมูลที่ทำ SMOTE ผลการศึกษาพบว่าแบบจำลองโครงข่ายประสาทเทียมคอนโวลูชันที่ใช้สถาปัตยกรรม ResNet50 ที่ใช้ข้อมูลที่ทำ SMOTE ในการฝึกสอนมีประสิทธิภาพในการจำแนกประเภทความผิดปกติของเซลล์เม็ดเลือดแดงจากรูปร่างและสีได้ดีที่สุดที่ความแม่นยำเท่ากับ 91.44 เปอร์เซ็นต์ โดยเมื่อนำตัวแบบที่มีประสิทธิภาพดีที่สุดดังกล่าวมาทำการปรับแต่งโดยใช้ค่า Dropout rate เท่ากับ 0.20 พบว่าสามารถเพิ่มประสิทธิภาพในการจำแนกประเภทความผิดปกติของเซลล์เม็ดเลือดแดงได้สูงสุดที่ค่าความแม่นยำเท่ากับ 95.59 เปอร์เซ็นต์

คำสำคัญ : การแบ่งภาพ โครงข่ายประสาทเทียมคอนโวลูชัน เซลล์เม็ดเลือดแดง

Independent Study Title	Image Segmentation and Classification of Abnormal Red Blood Cell Based on Morphology
Student Name	Mr. Sukawin Nunan
Student ID	65056090
Degree	Master of Science (Data Science and Analytics) KMITL-Digital Analytics and Intelligence Center
Year	2024
Independent Study Advisor	Dr. Jiraphat Yokrattanasak

Abstract

Blood is a fluid that circulates throughout the body, accounting for approximately 7-8% of body weight. It is responsible for transporting oxygen, nutrients, and water to nourish tissues throughout the body, as well as removing waste products from various parts of the body. In hematology, the study of red blood cell morphology has revealed that abnormalities in the shape and color of red blood cells are associated with potential diseases. This study developed three convolutional neural network models, ResNet50, InceptionV3, and VGG16 architectures, as classifiers for red blood cell abnormality classification. The models were trained using 13 categories of red blood cell images obtained from image segmentation using the Watershed Segmentation technique. The dataset was divided into three types: data without data imbalance handling, data handling with weighted method, and data handling with SMOTE method. The results showed that the convolutional neural network model using the ResNet50 architecture trained on SMOTE-processed data achieved the best performance in classifying red blood cell abnormalities from shape and color, with an accuracy of 91.44%. Fine-tuning this model a dropout rate of 0.20 further improved the accuracy to 95.59%.

Keywords : Convolutional Neural Network, Image Segmentation, Red Blood Cell

Acknowledgements

The successful completion of this independent study, "Image Segmentation and Classification of Red Blood Cell Abnormalities Based on Morphology", was made possible by the knowledge imparted by all the professors from the KMITL-Digital Analytics and Intelligence Center, School of Science, King Mongkut's Institute of Technology Ladkrabang. Their insightful knowledge provided the foundation upon which I could build and expand my understanding, for which I would like to express my sincere gratitude.

I am grateful to Dr. Jiraphat Yokrattanasak, my independent study advisor, for her invaluable guidance, advice, and assistance in correcting any errors encountered during the study process. Her encouragement and inspiration empowered me to follow the plan and achieve the desired outcomes.

My deepest thanks to my friends for their support and encouragement, especially when I faced challenges. Their advice provided me with invaluable insights throughout the research process.

Finally, a special thanks to my parents and family for their unwavering support throughout my education. Especially during times of discouragement, they were a pillar of strength and inspiration. I am deeply grateful for their unwavering belief in me.

Sukawin Nunan

Table of contents

	Page
บทคัดย่อ	i
Abstract	ii
Acknowledgements	iii
Table of contents	iv
List of tables	viii
List of figures	ix
Chapter 1 Introduction	1
1.1 Statement and Significance of the Problems	1
1.2 Objectives	2
1.3 Scope of Study	3
1.4 Contribution to Knowledge	3
1.5 Process of Study	3
Chapter 2 Literature Review	4
2.1 Hematology	4
2.1.1 Blood anatomy	4
2.1.2 Red blood cell morphology	6
2.1.3 Blood smear	9
2.1.4 Red blood cell abnormality grading criteria	10
2.2 Image Processing	11
2.2.1 Thresholding	11
2.2.2 Image opening	12
2.2.3 Flood fill	13
2.2.4 Watershed segmentation	13
2.3 Convolution Neural Network: CNN	14
2.3.1 Residual Network: ResNet	16
2.3.2 Inception Architecture Network: InceptionNet	17
2.3.3 Visual Geometry Group Network: VGGNet	18
2.4 Literature Survey	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table of contents (Cont.)

	Page
Chapter 3 Methodology	24
3.1 Research Method	24
3.1.1 Select a topic and study a related document	25
3.1.2 Search for a dataset applicable for this study	25
3.1.3 Explore and preprocess data	25
3.1.4 Fitting selected model architectures to the preprocessed data	25
3.1.5 Result evaluation and comparison	26
3.1.6 Analysis and diagnostics the result	26
3.2 Image Preprocessing and Labeling	26
3.2.1 Color channel selection	26
3.2.2 Image Thresholding	27
3.2.3 Hole filling	28
3.2.4 Removing small object	30
3.2.5 Watershed Transform	30
3.2.6 Cropping Region of Interest and Collecting information	32
3.2.7 Image Padding and Data Labeling	33
3.3 Imbalance Data Handling and Modeling	35
3.3.1 Imbalance Data Handling	37
3.3.1.1 Weighted method	37
3.3.1.2 SMOTE (Synthetic Minority Oversampling Technique)	38
3.3.2 Modeling	39
3.3.2.1 ResNet50	39
3.3.2.2 InceptionV3	43
3.3.2.3 VGG16	46
Chapter 4 Results	49
4.1 Results of ResNet50 architecture with baseline dataset	49
4.2 Results of ResNet50 architecture with weighted dataset	51
4.3 Results of ResNet50 architecture with SMOTE dataset	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table of contents (Cont.)

	Page
4.4 Results of InceptionV3 architecture with baseline dataset	56
4.5 Results of InceptionV3 architecture with weighted dataset	58
4.6 Results of InceptionV3 architecture with SMOTE dataset	60
4.7 Results of VGG16 architecture with baseline dataset	62
4.8 Results of VGG16 architecture with weighted dataset	64
4.9 Results of VGG16 architecture with SMOTE dataset	66
4.10 Model Comparison	68
4.11 Learning Rate and Dropout Rate Fine-Tuning	69
4.11.1 Efficiency of Experiment A1	70
4.11.2 Efficiency of Experiment A2	72
4.11.3 Efficiency of Experiment A3	74
4.11.4 Efficiency of Experiment A4	76
4.11.5 Efficiency of Experiment A5	78
4.11.6 Efficiency of Experiment A6	80
4.11.7 Efficiency of Experiment B1	82
4.11.8 Efficiency of Experiment B2	84
4.11.9 Efficiency of Experiment B3	86
4.11.10 Efficiency of Experiment B4	88
4.11.11 Efficiency of Experiment B5	90
4.11.12 Efficiency of Experiment B6	92
4.11.13 Efficiency of Experiment C1	94
4.11.14 Efficiency of Experiment C2	96
4.11.14 Efficiency of Experiment C3	98
4.11.16 Efficiency of Experiment C4	100
4.11.17 Efficiency of Experiment C5	102
4.11.18 Efficiency of Experiment C6	104
4.12 Fine-Tuning Conclusion	106
Chapter 5 Conclusion and Recommendations	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table of contents (Cont.)

	Page
5.1 Conclusion	108
5.2 Limitations	109
5.3 Recommendations	109
5.4 Future Work	110
References	111
Author Biography	114



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List of tables

Table	Page
2.1 Feature of red cell type and possible pathology	7
2.2 Making thick and thin blood smears	9
2.3 Red blood cell abnormality grading criteria	10
2.4 Example of RBC grading	11
2.5 Method comparison of literature survey	23
3.1 Hyperparameter used in this research	36
3.2 The amount of class distribution of the baseline dataset	36
3.3 Class weight	38
3.4 The amount of class distribution of SMOTE dataset	39
4.1 The classification performance of each model	68
4.2 The classification abilities of each model in each class	69
4.3 List of experiments detail	69
4.4 Performance comparison of fine-tuned models in each experiment set	107

List of figures

Figure	Page
2.1 Blood anatomy	5
2.2 Red blood cell morphology	7
2.3 Opening operation in binary image with 3x3 kernel	12
2.4 Flood fill	13
2.5 (a) Synthetic two dark blobs. (b) Analogy of (a) as the landscape.	14
2.6 CNN architecture	14
2.7 Convolution operation of 4x4 input data with 2x2 kernel	15
2.8 Pooling operations (a) Max pooling (b) Min pooling (c) average pooling	16
2.9 (a) Normal architecture (b) Architecture with residual block	17
2.10 (a) Normal architecture (b) Inception module	18
2.11 VGG-16 architecture	18
2.12 VGG-16 summary	19
3.1 Overall study process	24
3.2 The example of blood smear image	25
3.3 Spitting color channels and plotting intensity histogram code	27
3.4 An example of Intensity histogram got from coding	27
3.5 Otsu's Thresholding code	28
3.6 Example of white mask obtained using Otsu's Thresholding	28
3.7 Hole checking and flood filling code	29
3.8 Brief process of flood filling	29
3.9 Remove small object code	30
3.10 (a) Mask with small object in red circle, (b) Mask after removing small object	30
3.11 Code use for separate white masks	31
3.12 (a) Concatenated white masks (Red arrow), (b) Separated masks obtained from code identified by color	32
3.13 (a) Mask combining code, (b) Combining operation and result of mask combining code	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List of figures (Cont.)

Figure	Page
3.14 (a) Bounding box finding code, (b) Bounding box components	33
3.15 (a) Cropping with information collection code, (b) Example result	33
3.16 Label text file format	34
3.17 Label reading code	34
3.18 The overall concept and process of image padding	35
3.19 The overall concept and process of image labeling	35
3.20 An example cropped dataset	37
3.21 Compute class weight code	38
3.22 SMOTE code	38
3.23 Code used for creating the ResNet50 model	41
3.24 The ResNet50 architecture used in this research	42
3.25 Code used for creating the InceptionV3 model	44
3.26 The InceptionV3 architecture used in this research	45
3.27 Code used for creating the VGG16 model	47
3.28 The VGG16 architecture used in this research	48
4.1 Accuracy and loss graph of ResNet50 architecture with baseline dataset	50
4.2 Confusion matrix result using baseline ResNet50 model	50
4.3 Classification report result using baseline ResNet50 model	51
4.4 Accuracy and loss graph of ResNet50 architecture with weighted dataset	52
4.5 Confusion matrix result using weighted ResNet50 model	52
4.6 Classification report result using weighted ResNet50 model	53
4.7 Accuracy and loss graph of ResNet50 architecture with SMOTE dataset	54
4.8 Confusion matrix result using SMOTE ResNet50 model	55
4.9 Classification report result using SMOTE ResNet50 model	55
4.10 Accuracy and loss graph of InceptionV3 architecture with baseline dataset	56
4.11 Confusion matrix result using baseline InceptionV3 model	57
4.12 Classification report result using baseline InceptionV3 model	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List of figures (Cont.)

Figure	Page
4.13 Accuracy and loss graph of InceptionV3 architecture with weighted dataset	58
4.14 Confusion matrix result using weighted InceptionV3 model	59
4.15 Classification report result using weighted InceptionV3 model	59
4.16 Accuracy and loss graph of InceptionV3 architecture with SMOTE dataset	60
4.17 Confusion matrix result using SMOTE InceptionV3 model	61
4.18 Classification report result using SMOTE InceptionV3 model	61
4.19 Accuracy and loss graph of VGG16 architecture with baseline dataset	62
4.20 Confusion matrix result using baseline VGG16 model	63
4.21 Classification report result using baseline VGG16 model	63
4.22 Accuracy and loss graph of VGG16 architecture with weighted dataset	64
4.23 Confusion matrix result using weighted VGG16 model	65
4.24 Classification report result using weighted VGG16 model	65
4.25 Accuracy and loss graph of VGG16 architecture with SMOTE dataset	66
4.26 Confusion matrix result using SMOTE VGG16 model	67
4.27 Classification report result using SMOTE VGG16 model	67
4.28 Accuracy and loss graph of model in experiment set A1	71
4.29 Confusion matrix result using model in experiment set A1	71
4.30 Classification report result using model in experiment set A1	72
4.31 Accuracy and loss graph of model in experiment set A2	73
4.32 Confusion matrix result using model in experiment set A2	73
4.33 Classification report result using model in experiment set A2	74
4.34 Accuracy and loss graph of model in experiment set A3	75
4.35 Confusion matrix result using model in experiment set A3	75
4.36 Classification report result using model in experiment set A3	76
4.37 Accuracy and loss graph of model in experiment set A4	77
4.38 Confusion matrix result using model in experiment set A4	77
4.39 Classification report result using model in experiment set A4	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List of figures (Cont.)

Figure	Page
4.40 Accuracy and loss graph of model in experiment set A5	79
4.41 Confusion matrix result using model in experiment set A5	79
4.42 Classification report result using model in experiment set A5	80
4.43 Accuracy and loss graph of model in experiment set A6	81
4.44 Confusion matrix result using model in experiment set A6	81
4.45 Classification report result using model in experiment set A6	82
4.46 Accuracy and loss graph of model in experiment set B1	83
4.47 Confusion matrix result using model in experiment set B1	83
4.48 Classification report result using model in experiment set B1	84
4.49 Accuracy and loss graph of model in experiment set B2	85
4.50 Confusion matrix result using model in experiment set B2	85
4.51 Classification report result using model in experiment set B2	86
4.52 Accuracy and loss graph of model in experiment set B3	87
4.53 Confusion matrix result using model in experiment set B3	87
4.54 Classification report result using model in experiment set B3	88
4.55 Accuracy and loss graph of model in experiment set B4	89
4.56 Confusion matrix result using model in experiment set B4	89
4.57 Classification report result using model in experiment set B4	90
4.58 Accuracy and loss graph of model in experiment set B5	91
4.59 Confusion matrix result using model in experiment set B5	91
4.60 Classification report result using model in experiment set B5	92
4.61 Accuracy and loss graph of model in experiment set B6	93
4.62 Confusion matrix result using model in experiment set B6	93
4.63 Classification report result using model in experiment set B6	94
4.64 Accuracy and loss graph of model in experiment set C1	95
4.65 Confusion matrix result using model in experiment set C1	95
4.66 Classification report result using model in experiment set C1	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List of figures (Cont.)

Figure	Page
4.67 Accuracy and loss graph of model in experiment set C2	97
4.68 Confusion matrix result using model in experiment set C2	97
4.69 Classification report result using model in experiment set C2	98
4.70 Accuracy and loss graph of model in experiment set C3	99
4.71 Confusion matrix result using model in experiment set C3	99
4.72 Classification report result using model in experiment set C3	100
4.73 Accuracy and loss graph of model in experiment set C4	101
4.74 Confusion matrix result using model in experiment set C4	101
4.75 Classification report result using model in experiment set C4	102
4.76 Accuracy and loss graph of model in experiment set C5	103
4.77 Confusion matrix result using model in experiment set C5	103
4.78 Classification report result using model in experiment set C5	104
4.79 Accuracy and loss graph of model in experiment set C6	105
4.80 Confusion matrix result using model in experiment set C6	105
4.81 Classification report result using model in experiment set C6	106

CHAPTER 1

INTRODUCTION

1.1 Statement and Significance of the Problems

Blood [1] is the one of connective tissues that flow within a closed system of blood vessels. Blood transports functions are carrying nutrients, oxygen, hormones, enzymes, and nitrogenous wastes to tissues and organs, also responsible for maintaining equilibrium of the body. It has a volume about 7-8 percents of adult human weight or 5-6 liters. Blood consists of two major parts, the first part is blood cell called Cellular Element or Formed Element consists of red blood cell (RBC), white blood cell (WBC) and platelets, the second part is liquid part called Liquid Medium or Plasma.

One of the methods used for examining and analyzing the abnormality of blood called Blood Film or Blood Smear, Blood Film can be prepared by placing a small drop of blood onto a microscope slide and gently sliding another slide over it to create a thin layer of blood before dyeing with Wright-Giemsa stain this dyeing process called Staining Blood Films [2]. Abnormality of blood can be observed from its morphology like shape, color, and size. In red blood cells case, the percentage of each abnormal red blood cell in Blood Film can be used for analyzing and interpreting diseases or health conditions of patient [3], for example, Echinocyte (burr cell) is associated with liver disease, Target cell is associated with thalassemia, Acanthocyte (spur cell) is associated with alcoholic liver disease.

Although Automated Blood – Cell Analyzers is the most widely medical instrument used for examining and analyzing the abnormality of blood nowadays, there are still many small-medium size hospitals that lack access to modern medical instruments. According to accessibility in aspect of professional manpower or even financial problems, Blood Film or Blood Smear is still the popular traditional method that medical personnels use for examining and analyzing blood of patient. This method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

requires 100x microscope with a professionalism of the examiner to examine shape, color, and size of red blood cells one by one cause time consuming and fatigue for medical personnels to classify all red blood cells that may lead to low accuracy result.

At present, image processing technology has developed significantly and many of them are applied with machine learning, for example, neural networks to classify an object in an image. Therefore, the combination of image segmentation and the ability to classification of machine learning would be an option to help medical personnels classify an abnormal red blood cells with the satisfactory result for diseases and health conditions screening.

In this research, segmented red blood cell images from Blood smear will be used for developing neural network models that aim to classify red blood cells into 13 classes according to blood morphology. Segmented images are the inputs obtained from an image segmentation process which consists of using the flood fill image masking technique and the watershed transform technique to separate overlapping objects. The results of the classification through different Neural Network architectures will be compared in aspect of Accuracy, F1-score, Precision and Recall to identify the most suitable Neural Network architectures for use in classifying red blood cells.

1.2 Objectives

The objective of this research aims to

1. To develop a model for classification abnormalities of red blood cell based on morphology.
2. To compare the performance and capabilities of models using ResNet, InceptionNet, and VGGNet architectures trained with the same hyperparameters for abnormal red blood cell classification.
3. To compare the classification performance for abnormal red blood cells in each class across different imbalance dataset handling techniques.

1.3 Scope of Study

The limitation of this research consists of

1. Dataset used in this study is the secondary dataset from “Oxidation in Red Cell Disorders Research Unit (ORDR) Chulalongkorn University” Year 2019 from 1000x Nikon DS-Fi2-L3
2. Abnormality of red blood cells are classified into 13 classes consisting of Normal cell, Macrocyte, Microcyte, Spherocyte, Target cell, Stomatocyte, Ovalocyte, Teardrop, Burr cell, Schistocyte, Hypochromia, Elliptocyte and Uncategorized.
3. Comparison of model’s performance are based on the same dataset and comparing in aspect of Accuracy, F1-score, Precision and Recall.

1.4 Contribution to Knowledge

The expected gain of this research includes.

1. Able to apply image segmentation techniques to another medical image.
2. Developed models can be used in screening and interpreting diseases and health conditions from abnormalities red blood cells based on morphology with satisfactory result.

1.5 Process of Study

1. Gathering and selecting topic.
2. Research the basic Hematology and RBC morphology.
3. Research algorithm and technique of medical image segmentation.
4. Image segmentation testing and parameter tuning.
5. Research algorithm and architecture of neural networks.
6. Develop neural networks model with prepared secondary dataset.
7. Result discussion and conclusion.
8. Prepare a research paper.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 2

LITERATURE REVIEW

This research is a study of image segmentation and classification of red blood cells based on morphology. The researcher has studied and researched the following relevant theories.

2.1 Hematology

Hematology [4] is one of the medical sciences, that study about blood, blood disorder, blood infection, circulatory system, and blood-forming organs are also included. The results of analysis from hematology laboratory are called blood analysis result which hematologist use for diagnosis a disease and disorder of blood, for well-known example, Leukemia, Anemia, and clotting disorder.

2.1.1 Blood anatomy

Blood is a body fluid flow within a closed system of blood vessels, a mixture of about 55 percent plasma and 45 percent blood cells. It has a volume around 7 - 8 percent of body weight. Transporting oxygen and nutrients to lung and other organs, bringing waste products to kidney and liver, fighting an infection, and regulating body temperature are functions of blood. Blood consists of 4 components [5]:

1. Red blood cells: RBC

The shape of a red blood cell is a biconcave disk with a flattened center with a size about 7.2 to 7.9 microns, accounting for about 40 to 45 percent of blood volume. It contains hemoglobin, a special protein which helps carry oxygen to the rest of body and returns carbon dioxide from the body to lung. The average life span of red blood cells is normally 120 days. It is formed by bone marrow and eliminated by liver and spleen.

2. White blood cells: WBC

White blood cells have a shape like kidney bean. The size of white blood cells are approximately 12 to 15 microns, accounting for 1 percent of blood volume. The major type of white blood cell is called lymphocyte, a cell responsible for making antibodies and attacking bacteria, viruses, foreign material, infected cells, and tumors. The volume of white blood cells changes directly in response to body inflammation.

3. Platelets

Platelets are small fragments of blood cells. It helps the blood clotting process by forming a platform at the site of injury which covers the wound and prevents blood from leaking out. A lower than normal number of platelets can be an indicator of abnormality of bone marrow or bone marrow cancer.

4. Plasma

A liquid component of blood that carries red blood cells, white blood cells, and platelets circulate within a closed system of blood vessels. It contributes to 55 percent of blood volume. Plasma serves as a transport medium for delivering nutrients to various organs of the body and returns waste products to the kidneys, liver, lungs, and spleen.

Figure 2.1 illustrates the components flowing within a closed system of blood vessels. These components include red blood cells, white blood cells, platelets, and plasma.

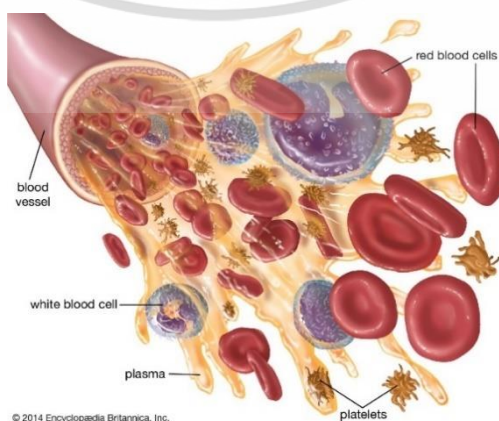


Figure 2.1 Blood anatomy [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อผู้เ้าตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Red blood cell morphology

Red blood cell morphology [7] is a study about the shape and structure of red blood cells. Healthy red blood cells are normally in the shape of biconcave disk with a flattened center and have a red color from its special protein called hemoglobin and have a size around 7 microns. The alteration of shape, size, and color of red blood cells are valuable information for diagnosing and analyzing as it can represent overall health and possible pathology of a patient. The abnormality of red blood cell morphology is categorized in 5 categories:

1. Size variation

Red blood cell size variation is known as anisocytosis, a condition in which the red blood cells are not even in size, results from abnormal cell development. It is frequently found in leukemia and most forms of anemia.

2. Hemoglobin distribution

Hemoglobin distribution or color variation refers to the amount of hemoglobin in the cell or area of red color at central pallor. Red blood cells with normal amount of hemoglobin called normochromia.

3. Shape variation

Variation in shape of red blood cells is known as poikilocytosis. Normal red blood cells are biconcave discs with a flattened center. The other shape of red blood cells is considered poikilocytic cells. Poikilocytosis can be caused by various factors, including genetic disorders, undernutrition, and usage of medicine.

4. Inclusion

Red cell inclusions are abnormal structures found within red blood cells that can be indicative of certain medical conditions. These inclusions can be caused by a variety of factors, such as oxidative stress, iron overload, and abnormal hemoglobin synthesis.

5. Red cell distribution

Red cell distribution is divided into 2 types, Agglutination and Rouleaux. Agglutination is an aggregation of red cells into random clusters or masses

while Rouleaux is a condition in which red cells appear as stacks of coins on the peripheral smear shown in figure 2.2.





























RED BLOOD CELL MORPHOLOGY					
Size variation	Hemoglobin distribution	Shape variation		Inclusions	Red cell distribution
Normal 	Hypochromia 1+ 	Target cell 	Acanthocyte 	Pappenheimer bodies (siderotic granules) 	Agglutination 
Microcyte 	2+ 	Spherocyte 	Helmet cell (fragmented cell) 	Cabot's ring 	Rouleaux 
Macrocyte 	3+ 	Ovalocyte 	Schistocyte (fragmented cell) 	Basophilic stippling (coarse) 	
Oval macrocyte 	4+ 	Stomatocyte 	Tear drop 	Howell-Jolly 	Crystal formation  
Hypochromic macrocyte 	Polychromasia (Reticulocyte) 	Sickle cell 	Burr cell 		

Figure 2.2 Red blood cell morphology [7]

A possible pathology is correlated with abnormal morphology. In this study, researcher aims to classify 11 types of red blood cell. Identifying feature and possible pathology are shown in table 2.1

Table 2.1 Feature of red cell type and possible pathology [7]

Red cell type	Identifying feature	Possible pathology
Macrocyte	Red cell has a diameter approximately 9 microns or larger	Megaloblastic process, High reticulocyte count, Liver diseases, Post-splenectomy, Chemotherapy, Hypothyroidism
Microcyte	Red cell has a diameter less than 7 microns	Iron deficiency, Thalassemic conditions, Sideroblastic anemia, Anemia of chronic disease, Lead poisoning
Spherocyte	Red cell with no central pallor	Immune hemolytic anemia, Hereditary spherocytosis, Post-transfusion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2.1 Feature of red cell type and possible pathology [7] (Cont.)

Red cell type	Identifying feature	Possible pathology
Target cell	Red cell with a bell shape	Hemoglobin C disease, Hemoglobin C trait, Post-splenectomy, Liver disease, Iron deficiency anemia, Any hemoglobin abnormality
Stomatocyte	Red cell with bowl shape and have a mouth shape central pallor	Hemolytic anemia, Alcoholic cirrhosis, Acute alcoholism
Ovalocyte (Egg shape)	Red cell with Egg shape	Myelodysplastic syndrome, Thalassemic syndrome, Megaloblastic process
Elliptocyte (Pencil shape)	Red cell with pencil, rod, cigar shape	Iron deficiency anemia, Hereditary elliptocytosis, Idiopathic myelofibrosis
Teardrop	Red cell with tear shape or pear shape	Thalassemia syndrome, Drug-induced Heinz body formation, Iron deficiency
Burr cell	Red cells with rounded spicules evenly placed over the surface	Renal disease, Liver disease, Burns
Schistocyte	A fragment of red cell without central pallor	Prosthetic heart valve, Microangiopathic H.A., Clostridial infections
Hypochromia	Red cells with a decreased surface-to-volume ratio and a decreased or absent central pallor	Iron deficiency anemia, Thalassemia syndrome

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


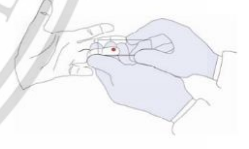
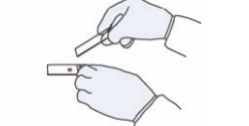
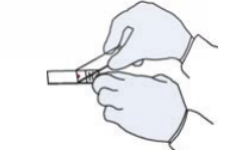
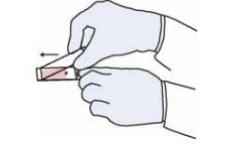
2.1.3 Blood smear

Blood smear [8] also known as peripheral blood smears (PBS) is one of the methods used for assessment of red blood abnormality. Morphologic abnormalities of blood cells are discovered by microscopic examination with the oil immersion lens of well-prepared films of peripheral blood smears with Wright's stain [9]. There are 2 types of blood smear:

1. Thick smears use a large amount of blood, suitable for detecting parasites in blood samples with a low parasite or infection.
2. Thin smears use less blood than thick smears. Red blood cells are arranged in a single layer in this smear type making them suitable for diagnosing parasites from their characteristic morphological.


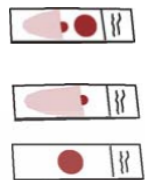
Thick and thin blood smear can be prepared following the steps provided in Table 2.2.

Table 2.2 Making thick and thin blood smears [10]

Process	Illustration
1. Clean the area to be punctured with 70% alcohol, Puncture the ball of the finger	
2. Wipe away the first drop of blood with clean gauze. Touch the next drop of blood with a clean slide. Repeat with several slides	
3. Thin film (a): Bring a clean spreader slide, held at a 45° angle, toward the drop of blood on the specimen slide.	
4. Thin film (b): Wait until the blood spreads along the entire width of the spreader slide.	
5. Thin film (c): While holding the spreader slide at the same angle, push it forward rapidly and smoothly.	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2.2 Making thick and thin blood smears [10] (Cont.)

6. Thick film: Using the corner of a clean slide, spread the drop of blood in a circle (diameter 1-2 cm).	
7. Wait until the thin and thick films are completely dry before staining. Fix the thin film with methanol and let it dry completely before staining. The thick film should not be fixed.	

2.1.4 Red blood cell abnormality grading criteria

Red blood cell abnormality grading criteria [11] is a guideline used for diagnosing and assessing the severity of red blood cells. The Medical Technology Council of Thailand has established criteria for grading abnormality shown in table 2.3.

Table 2.3 Red blood cell abnormality grading criteria

Grade	% abnormal RBC
4+	76-100
3+	51-75
2+	26-50
1+	11-25
few	5-10

The important characteristics of red blood cells are size, shape, staining, distribution, inclusion, and infection. An example of grading of size, shape, and staining characteristic and its description show in table 2.4.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2.4 Example of RBC grading

Grading	Description
anisocytosis 2+ with microcyte 1+, macrocyte few	RBC has a size abnormality 26-50% of all RBC, divided into microcyte cells 11-25% and macrocyte cells 5-10% of its abnormality
Poikilocytosis 2+ with target cell 2+	RBC has a shape abnormality 26-50% of all RBC, that contain target cells 26-50% of its abnormality
hypochromia few	RBC has a color abnormality 5-10% of all RBC.

2.2 Image Processing

Image processing is the field of study that studies the manipulation and analysis of an image using an algorithm. The goal of image processing is to prepare the image for further analysis including noise reduction, scaling, feature extraction and segmentation.

2.2.1 Thresholding

Thresholding is a technique that uses a difference gray level in image to divide object into K classes, widely using K=2 to divide the foreground and background part. One of the most famous and excellent algorithms is Otsu' thresholding [12] by using the estimates of dispersion and location of intensities in each class. The threshold intensity which maximizes the between class variance [13], are given by

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \quad (2.1)$$

And

$$\omega_0 = \Pr(C_0) \quad (2.2)$$

$$\omega_1 = \Pr(C_1) \quad (2.3)$$

$$\mu_0 = \sum_{i=1}^k i \Pr(i|C_0) \quad (2.4)$$

$$\mu_1 = \sum_{i=k+1}^k i \Pr(i|C_1) \quad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Where σ_B^2 is maximum threshold intensity
 ω_0 is probability of background class occurrence
 ω_1 is probability of foreground class occurrence
 μ_0 is background class mean levels
 μ_1 is foreground class mean levels
 i is pixel level
 C_0 is pixels of background class
 C_1 is pixels of foreground class

2.2.2 Image opening

Image opening [14] is the operation used for separating objects in the image by combining erosion and dilation together. First, erosion, this operation erodes away the boundaries of the object by slicing the kernel through the image, a pixel in image will be considered 1 only if all pixel under the kernel is 1. Second, dilation, this operation expands the boundaries of the object by slicing the kernel through the image, a pixel in image will be considered 1 if at least 1 pixel under the kernel is 1. With the erosion followed by dilation the small white object in the image will be removed. This operation is usually used for removing noise and separate boundaries in the image. Opening operation in binary image show in Figure 2.3.

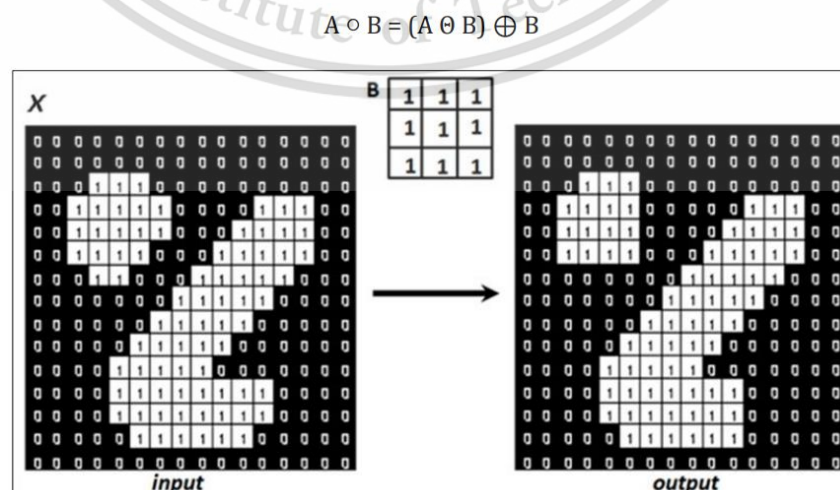


Figure 2.3 Opening operation in binary image with 3x3 kernel [14]

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น มิใช่เพื่อเผยแพร่ในเชิงพาณิชย์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 Flood fill

Flood fill or seed fill is an algorithm used to fill a region of connected pixels with a new color starting at the given pixels called seed point. Flood fill is probably the most known as the "Bucket Fill" application in most art programs.

The flood fill algorithm consists of the following step:

1. Consider the seed point if the color of the seed point is already the same as the new color, do nothing.
2. Choose whether to travel in four directions (N, S, W, E) or eight directions (N, S, W, E, NW, NE, SW, SE).
3. Travel in the selected direction, replace with the new color.
4. Repeat step 2 and 3.

The example of flood fill result show in figure 2.4

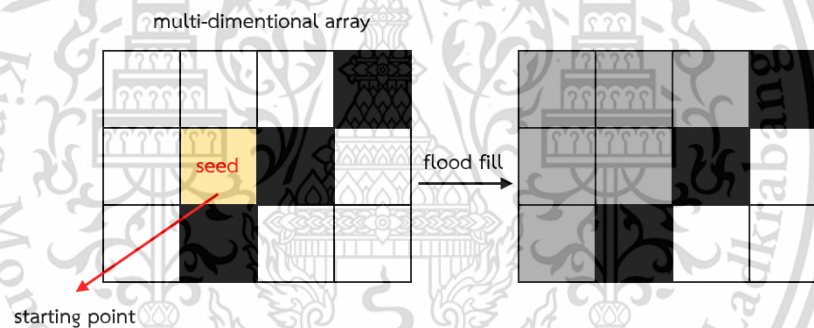


Figure 2.4 Flood fill

2.2.4 Watershed segmentation

Watershed segmentation [15], one of the most popular segmentation techniques, is based on an analogy of flooding a landscape with rain. By assuming each pixel in the image as terrain with varying heights, brighter areas represent peaks and hills, and darker areas symbolize valleys. The main goal of watershed algorithms is to identify the watershed line, which acts as a drainage that separates an adjacent catchment basin, darker areas. These boundaries are calculated through distance transformation. While watershed segmentation offers a straightforward approach and can segment complex objects with irregular

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

shapes, it remains sensitive to noise. The concept of watershed is shown in figure 2.5.

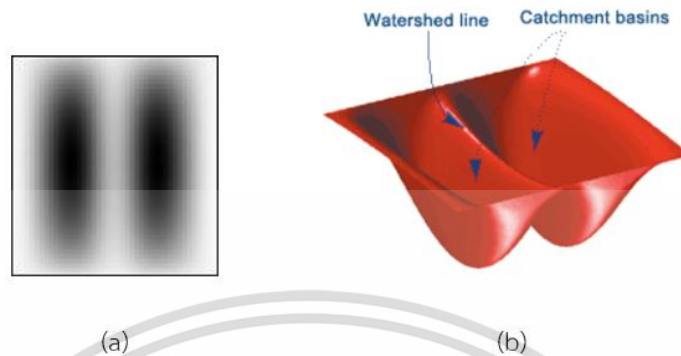


Figure 2.5 (a) Synthetic two dark blobs. (b) Analogy of (a) as the landscape.

2.3 Convolution Neural Network: CNN

Convolution neural network (CNN, ConvNet) [16] [17] is a class of deep neural networks, it was first developed and used around the 1980s. It is mostly applied to analyze image data due to its superior performance with grid-patterned information, the illustrate of CNN architecture show in figure 2.6.

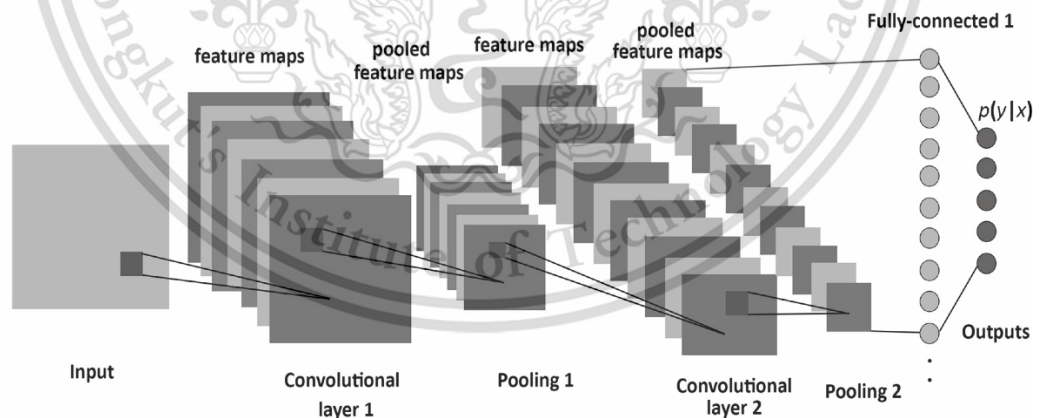


Figure 2.6 CNN architecture [18]

The architecture of CNN is analogous to the connectivity pattern of the human brain, CNN neurons are arranged like the brain's frontal lobe, the area responsible for processing visual stimuli. his arrangement effectively addresses the disadvantage of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

traditional neural networks, requiring feeding of small, reduced-resolution image. CNNs consist of important layer as follow:

1. Convolution layer

Convolution layer is a core component of CNN architecture that performs convolution operation with the input tensor, some of example are feature extraction, padding, and image smoothing. The final output of convolution layer through multiple iteration of convolution operation is known as feature map, activation map or convolved feature. Convolution operation of convolution layer show in figure 2.7.

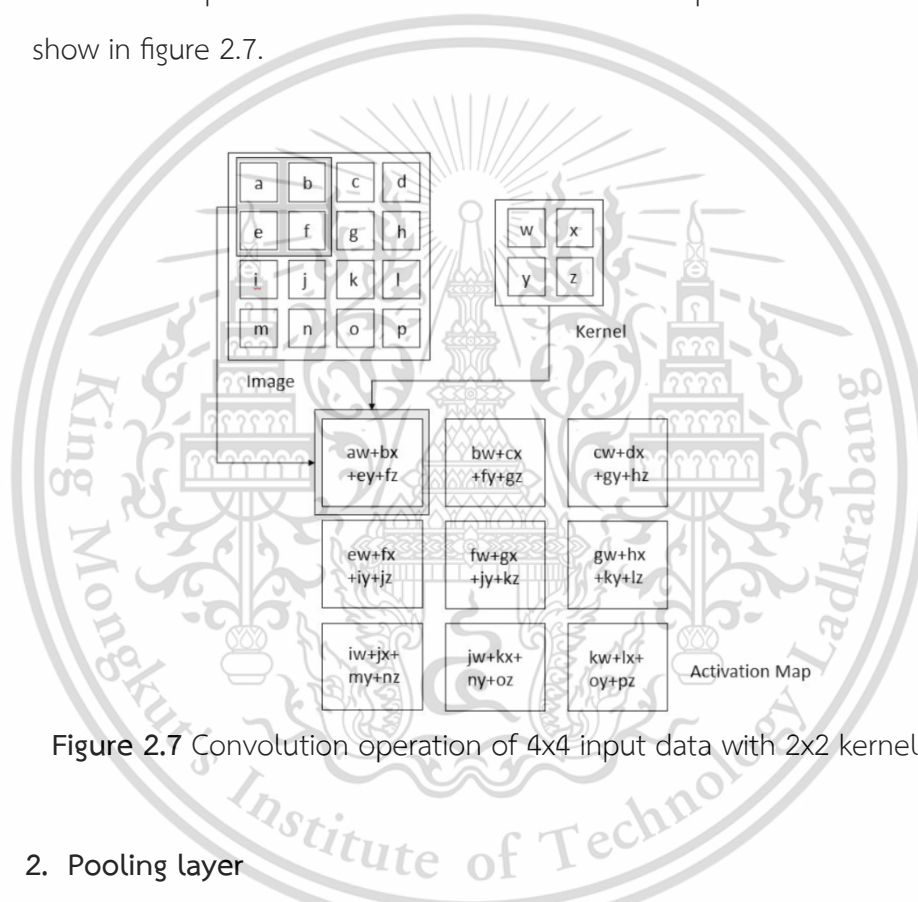


Figure 2.7 Convolution operation of 4x4 input data with 2x2 kernel [19]

2. Pooling layer

Pooling layer functions is to return the notable representative of rectangular neighborhood in grid data. This operation helps in reducing the dimension and complexity of data but also results in some information loss. The several pooling functions including max pooling, return the maximum output of neighborhood, min pooling, return the minimum output of neighborhood, and average pooling, return the average of neighborhood as the output. An illustration of pooling operations is shown in figure 2.8.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

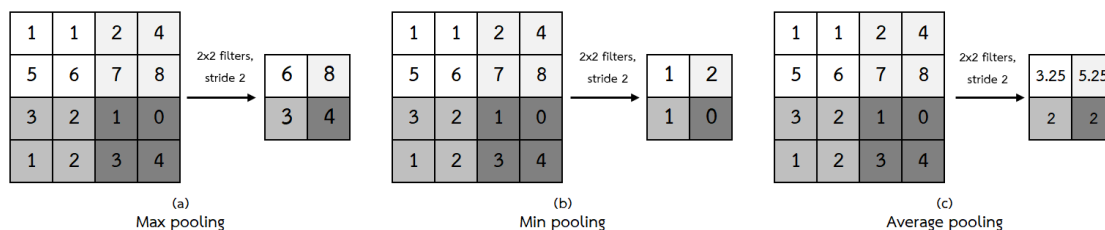


Figure 2.8 Pooling operations (a) Max pooling (b) Min pooling (c) average pooling

Several specific convolutional neural network (CNN) architectures have been developed to address limitations in traditional CNN architectures. Among these, ResNet, InceptionNet, and VGGNet stand out as three of the most successful CNN architectures for medical image classification.

2.3.1 Residual Network: ResNet

ResNet (Residual Network) is a powerful convolutional neural network architecture introduced by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang in 2015. ResNet has overcome the traditional architecture problem where the weights and loss function of earlier layers become smaller, seemingly vanishing. By using a technique known as “skipping connection”, it allows information to flow directly between layers of the network.

In figure 2.9 shows a normal neural network (a) compared to a neural network with residual block (b), The skip connection in the residual block facilitates the flow of input information through various layer functions and then combine with the information x to produce $f(x) + x$ before transferring it to the next activation function. In contrast, a standard neural network requires the information x to traverse through all layers, producing the result $f(x)$ before transferring it to the next activation function.

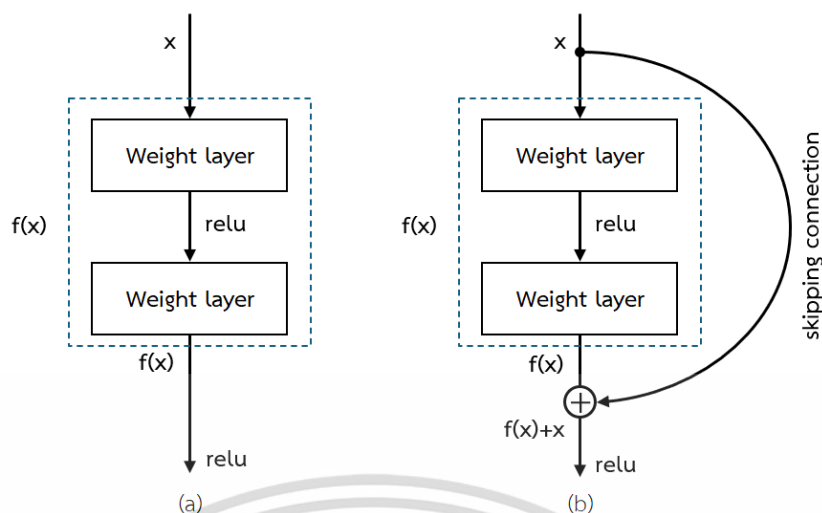


Figure 2.9 (a) Normal architecture (b) Architecture with residual block

2.3.2 Inception Architecture Network: InceptionNet

InceptionNet [20] is a convolution neural network architecture developed by Google in 2014. The structure of InceptionNet is different from other convolution neural networks, to overcome the vanishing gradient problem, instead of deepening the network, InceptionNet chooses to broaden its architecture. It uses the technique known as “Inception module” illustrated in Figure 2.10 (b), featuring a sparsely connected architecture with a block of layers utilizing parallel convolution operations of varying sizes. With the multiple operation in the inception module, it allows the network to learn a combination of local and global features from the input information. This solves the problem of traditional architecture that has a limit in capturing the input feature.

In comparison to other CNN architectures, InceptionNet offers advantages such as flexibility due to its convolutional operations, enabling the model to process information from multiple viewpoints. Additionally, the Inception module technique optimizes resource utilization in processing the model.

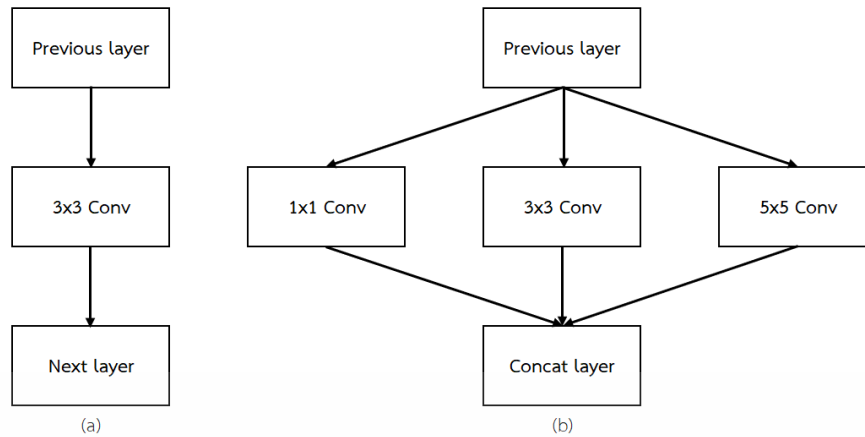


Figure 2.10 (a) Normal architecture (b) Inception module

2.3.3 Visual Geometry Group Network: VGGNet

VGGnet [21] is the one of CNN architecture which belongs to Oxford university. The well-known of VGGnet are VGG-16 and VGG-19, by having 16 and 19 convolution layers respectively. The architecture of VGGNet consists of two main parts: convolution layers, and fully connected layers show in figure 2.11.

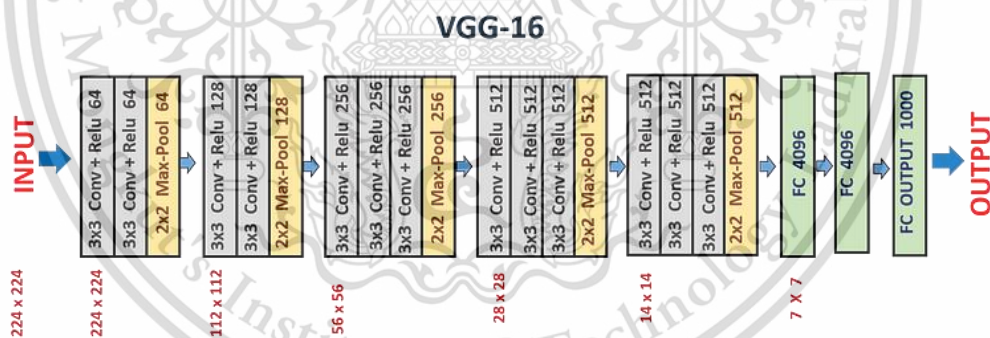


Figure 2.11 VGG-16 architecture

VGGnet accepts 224x224 pixels RGB image as an input, using a 3x3 kernel and ReLU as an activation function. The summary of VGG-16 is shown in figure 2.12.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Layer	Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

Figure 2.12 VGG-16 summary

2.4 Literature Survey

2.4.1 Ismail M. I. Alkafrawi and Zaroug A. Dakhell (2022) conducted a research on Blood Cells Classification Using Deep Learning Technique [22]. The dataset consists of 15,941 images of individual cells obtained from the Core Laboratory at the Hospital Clinic of Barcelona. Researchers aimed to classify red blood cells into eight classes using a pre-trained AlexNet model. The hyperparameters utilized in the model included an initial 0.00075 learning rate, 100 max epochs, and testing data at 20% of the dataset. In the image preprocessing phase, the researchers resized the input images to 227x227 pixels to feed into the first layer of the neural network.

The experimental result revealed that AlexNet was able to classify red blood cells images with an accuracy of 95.08%. However, the researchers did not provide information regarding the confusion matrix and the method used for balancing the data in their study.

2.4.2 Prashanth Kannadaguli (2020) conducted a research on Microscopic Blood Smear RBC Classification using PCA and SVM based Machine Learning [23]. This study aimed to classify four classes of red blood cells using dimension reducing technique along with support vector machine model. In this study, researcher converted an image into grayscale then use the watershed algorithm as a segmentation process, combines

with hit-or-miss transform (HmT) and Hough transform (HT). The dimensionality of the data was reduced using K-Means clustering, followed by PCA to extract the 500 strongest features for each image.

The SVM model was trained on an exact balance data set, consisting of 10,125 training and 3,375 testing images for each class. This experiment achieves the best accuracy at 88% and the model can classify Sickel-cell, Elliptocytes, Echinocytes, and Normal RBC from best to worst, respectively.

2.4.3 Dyah Aruming Tyas et al. (2020) conducted a research on Morphological, Texture, and Color Feature Analysis for Erythrocyte Classification in Thalassemia Cases [24]. In this study, researcher aim to classify nine classes of red blood cells focus on thalassemia case using multilayer perceptron (MLP) with backpropagation learning methods. The Experiment in this study was divided into two sets. The first experiment used the D1 dataset, 7,108 single red blood cell images to determine the effect of the selected features on classification performance. The second experiment used the D2 dataset, 2,247 subset images of D1 to compare the classification performance of their proposed previous method. The hyperparameters utilized in the MLP included 0.01, 0.05, and 0.10 learning rate, the values of 0.2, 0.4, 0.6, and 0.8 for momentum, and the values of 1000, 5000, and 10000 for the number of epochs. The number of hidden layer nodes was obtained using based on Huang equation, $L_1 = \sqrt{(m+n)N} + 2\sqrt{N/(m+2)}$ where L_1 is the number of hidden layer nodes, m is the number of classes, and N is the number of inputs. In the image preprocessing phase, researchers use hole filling operations, watershed distance transform, and circle Hough transform to separate overlapping objects.

The maximum accuracy obtained in this study is 98.11%, maximum specificity is 99.75%, sensitivity is 86.67%, and ROC area 0.99. However, the researchers did not provide information on the method used for balancing the data in their study but focused on the ROC area due to it is not affected by the majority class.

2.4.4 Maria Delgado-Ortet et al. (2020) conducted a research on A Deep Learning Approach for Segmentation of Red Blood Cell Images and Malaria Detection [25]. The study focuses on analyzing red blood cell morphology in cases of Malaria,

with the goal of designing a Clinical Decision Support System (CDSS) for malaria detection using neural networks. The 517 digital images of peripheral blood smears dataset were divided into two sets, dataset A, 186 healthy smear images, and dataset B, 331 infected smear images with 19.7% of red blood cells parasitized. The study employed a Segmentation Neural Network (SNN) to segment the blood smear images. The architecture of SNN used in this study consists of three layers, the first layer or the decoding block consisted of convolution and ReLU to capture semantic information, the second layer or encoding block consisted of up-sampling and deconvolution layers to filter and resize the image, and the last layer, pixel classification layers, enabled the individual labeling of the pixels. To address the issue of imbalanced classes, researcher trained CNN with 27,558 images of masked RBCs, with 50% parasitized cell, which downloaded from the online repository Malaria dataset, and decided to use dataset A and B as the test set instead.

In this experiment, CNN with 13 layers and 30,758 weights is used to train a 181x181 pixels dataset. The output provided in the form of probabilities indicating whether the cell belonged to an infected or non-infected. Conclusion of this research, RBC segmentation of SNN achieved accuracy of 93.72% and CNN achieved accuracy of 95% in classification.

2.4.5 Wongsathon Naksuwan et al. (2022) conducted a research on Deep Learning for Detecting Malaria Parasites of Infected Red Blood Cells in Thin Blood Smear Images [26]. The research focuses on diagnosing malaria infections in thin blood smears obtained from the MP-IDB and the Phramongkutklao College of Medicine dataset, The primary objective of this research is to classify infected cells from normal cells, utilizing DenseNet121 as the classification model. The segmentation is performed through the object detection mechanisms of the YOLOv3 model, resulting in the accurate detection of 453 out of 462 infected cells, as well as the correct identification of 24,288 normal red blood cells. However, there were 2,869 normal red blood cells that were incorrectly detected.

The classification model is trained on 296,636 normal cells and 296,636 infected cells for training, while the validation set consists of 70,152 normal cells and 70,152

infected cells. Image augmentation, involving rotation and random shifting, is employed as the balancing method in this research. With hyperparameters set to a batch size of 128, a learning rate of 0.001, and a weight decay of 0.001, the classification model achieves the best accuracy, sensitivity, and specificity at 98.48%, 90%, and 99.24%, respectively.

2.4.6 Korranat Naruenatthanaset et al. (2020) conducted a research on Red Blood Cell Segmentation with Overlapping Cell Separation and Classification on Imbalanced Dataset [27]. The primary objective of this research is to classify red blood cells in blood smear into twelve classes using eight different sizes of EfficientNet as a classification model along with ellipse fitting method as overlapping cells separation technique. The ellipse fitting method includes identifying concave points on the contour, estimating ellipses through direct fitting, and verifying their alignment with the object contour, resulting in high precision in segmented data. With multiple classes of data, the imbalance problem isn't avoidable, researcher handling the imbalance data with various techniques including data augmentation, weight normalization, upsampling, and focal loss.

In this experiment, each EfficientNet use five fold-cross validation with 80% and 20% for training and testing and with the various of imbalance data handling method it can improve the accuracy of the baseline model from 88.23% to 90.21% with the data augmentation method.

Table 2.5 provides a comparison of the various research approaches presented. It details the dataset, segmentation method, balancing technique, classification method, and best accuracy achieved for each study.

Table 2.5 Method comparison of literature survey

Author	Dataset	Segmentation method	Balancing method	Classification method	Achieved accuracy
Ismail M. I. Alkafrawi & Zaroug A. Dakhell (2022)	Single blood cell dataset 15,941 images, Divided into 8 classes	-	-	AlexNet	95.08%
Prashanth Kannadaguli (2020)	Blood smear dataset Segment into: 40,500 training set, 13,500 testing set, Divided into 4 classes	Watershed algorithm hit-or-miss transform Hough transform	-	SVM with PCA technique	88%
Dyah Aruming Tyas et al. (2020)	Blood smear dataset Segment into: 7,108 images, Divided into 9 classes	Watershed algorithm Circle Hough transform	Handling imbalance dataset by using ROC as the result interpreter	MLP with backpropagation learning	98.11%
Maria Delgado-Ortet et al. (2020)	Malaria blood smear dataset Segment into: 27,558 images, Divided into 2 classes	Segmentation Neural Network	-	CNN with 13 layers and 30,758 weights	95%
Wongsathon Naksuwan et al. (2022)	Malaria blood smear dataset Segment into: 593,272 training set, 140,304 testing set, Divided into 2 classes	YOLOv3 - object detection mechanisms	Augmentation (rotation and random shifting)	DenseNet121	98.48%
Korranat Naruenatthanaset et al. (2020)	Blood smear dataset Segment into: 20,875 images, Divided into 13 classes	Ellipse fitting method	Augmentation, Weight normalization, Upsampling, Focal loss	EfficientNet	90.21%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 3

METHODOLOGY

Research method of image segmentation and classification of red blood cells based on morphology consists of Data understanding process, Image preprocessing process, and Model fitting process.

3.1 Research Method

In this study, researcher studied related documentation about hematology to understand the blood cell types in their dataset and the theorems of image segmentation, a necessary method for obtaining single blood cell images from the blood smear dataset. The overall process of this study is shown in Figure 3.1.

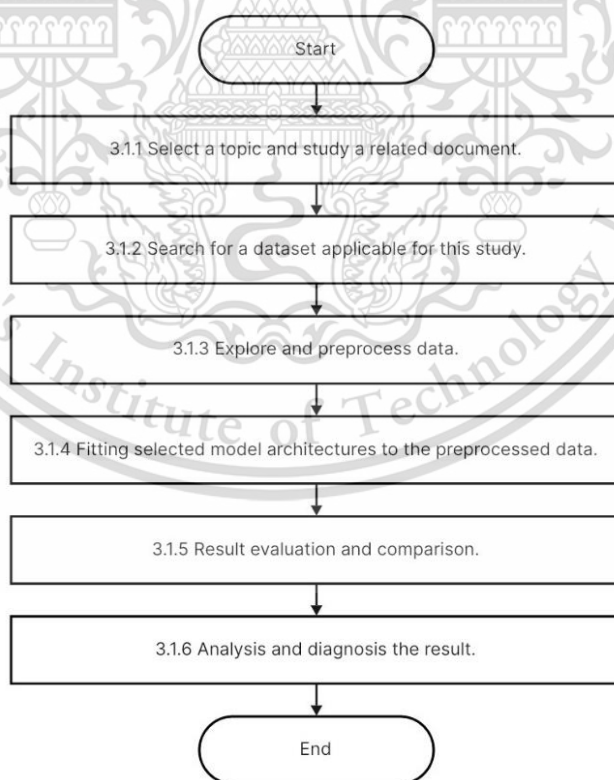


Figure 3.1 Overall study process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 Select a topic and study a related document

Researcher selects “Image segmentation and classification of red blood cells based on morphology”, which is a study of a method for classifying the shape of red blood cells. This method can be linked to pathology. The researcher also studied various theories related to the topic, including hematology, image segmentation, and convolutional neural network architectures.

3.1.2 Search for a dataset applicable for this study

The dataset used in this study is the secondary blood smear dataset from the “Oxidation in Red Cell Disorders Research Unit (ORDR)” at Chulalongkorn University, made public by Mr. Korranat Naruenatthanaset. The dataset consists of 623 blood smear images containing 13 classes of blood cells.

3.1.3 Explore and preprocess data

The dataset used in this study is blood smears, which contain many blood cells in a single image. Image segmentation is necessary to isolate individual red blood cells. These isolated cells are fed into the model architecture to learn the characteristics and patterns of each red blood cell type for classification. The example of blood smear image from this dataset shown in Figure 3.2.

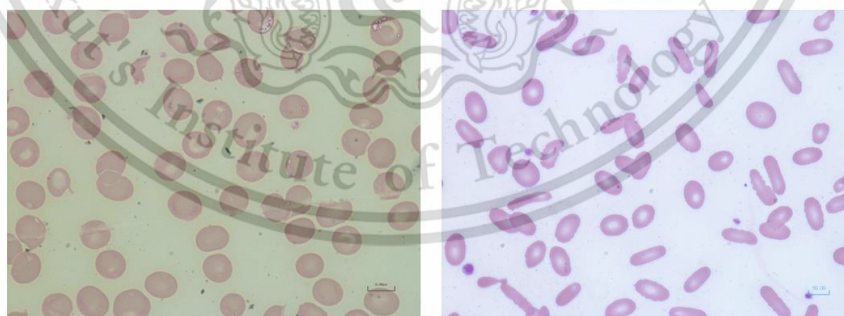


Figure 3.2 The example of blood smear image

3.1.4 Fitting selected model architectures to the preprocessed data

This research will utilize three pre-trained model architectures for fitting the preprocessed data: ResNet50, InceptionNetV3, and VGG16. Their selection is motivated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

by their relatively simple architecture and widespread use in image classification tasks. All previously mentioned architectures will be imported from Keras using TensorFlow and trained with the same hyperparameters, including epoch, learning rate, batch size, and optimizer. Additionally, they will be augmented with the same number of custom layers.

3.1.5 Result evaluation and comparison

The evaluation and comparison of results will be based on the same dataset and its class imbalance handling methods. These methods include the baseline dataset, the class weights dataset, and the SMOTE dataset. All datasets will be split into 80% training, 10% validation, and 10% testing sets. The comparison will be based on metrics such as accuracy, F1-score, precision, and recall.

3.1.6 Analysis and diagnostics the result

This step involves analyzing the results obtained from the experiments of training the three different model architectures with different imbalance data handling methods. The goal is to draw conclusions and identify the factors influencing the experimental results and explore possible areas for further development.

3.2 Image Preprocessing and Labeling

Image processing is a crucial step in segmenting individual red blood cells from blood smear images. This process will utilize OpenCV and scikit-image as core libraries, which offer a wide range of functions for image manipulation. The segmentation process consists of the following steps:

3.2.1 Color channel selection

An image typically consists of three color channels: red, green, and blue. To effectively separate the foreground object from the background, color channel selection is a useful method. This method involves finding the channel with the largest difference between the global maximum and a local maximum threshold in its intensity histogram. This difference serves as an indicator of how well the channel

separates the object. Figure 3.3 shows the code used to split the image into its three channels. Figure 3.4 shows the intensity histograms, which represent the global maximum and local maximum value. By analyzing the histograms, we can see that the green channel has the largest difference between its global maximum and local maximum. Therefore, the green channel will be selected for further processing.

```
#Extract Color Channels and Plot Histogram
def extract_plot_channels(Img):
    #Change Image to Grayscale
    Gray = cv2.cvtColor(Img, cv2.COLOR_BGR2GRAY)
    #Extract R G B Channel
    (B, G, R) = cv2.split(Img)
    #Calculate Histogram
    histGray = cv2.calcHist([Gray], [0], None, [256], [0, 256])
    histB = cv2.calcHist([B], [0], None, [256], [0, 256])
    histG = cv2.calcHist([G], [0], None, [256], [0, 256])
    histR = cv2.calcHist([R], [0], None, [256], [0, 256])
    #Plot Intensity Histogram
    plt.plot(histGray, c='black', label = 'Grayscale')
    plt.plot(histB, c='blue', label = 'Blue Channel')
    plt.plot(histG, c='green', label = 'Green Channel')
    plt.plot(histR, c='red', label = 'Red Channel')
    plt.legend(loc = "upper right")
    plt.title('Intensity Histogram')
    return Gray,B,G,R
```

Figure 3.3 Spitting color channels and plotting intensity histogram code

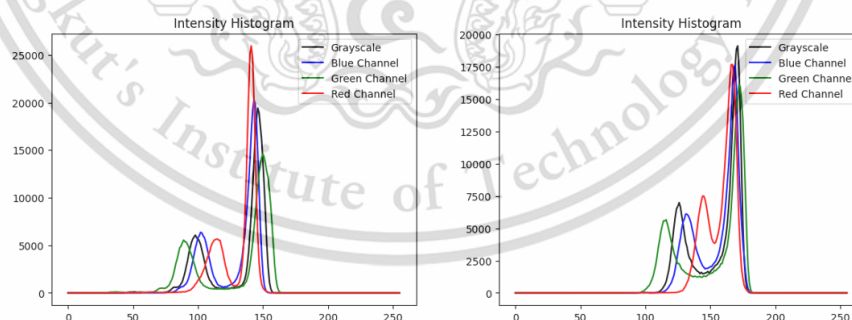


Figure 3.4 An example of Intensity histogram got from coding

3.2.2 Image Thresholding

Thresholding is a technique used to separate the foreground from the background in an image. This experiment will utilize Otsu's Thresholding algorithm for

this task. To improve the effectiveness of thresholding, a Gaussian blur will be applied

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

to the image beforehand. This step helps to reduce the sharpness of the object and minimize noise. The code for this process is shown in Figure 3.5. Figure 3.6 illustrates an example result: a white mask obtained using Otsu's Thresholding.

```
#Thresholding
def thresholding(img):
    #Perform Gaussian Blur
    blurred = cv2.GaussianBlur(img, (3, 3), 0)
    #Otsu's Thresholding
    T, threshInv = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
    white_mask = threshInv
    return white_mask
```

Figure 3.5 Otsu's Thresholding code

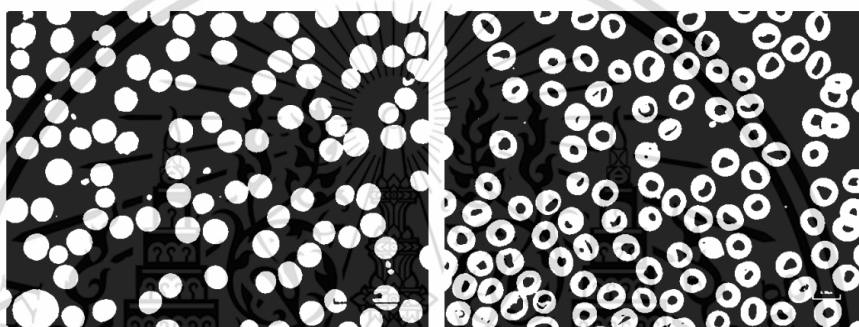


Figure 3.6 Example of white mask obtained using Otsu's Thresholding

3.2.3 Hole filling

As observed in Figure 3.6(b), the object's white mask contains many black holes. These holes occur because some object pixels have intensity values close to the background, even though they belong to the foreground object. To address this issue, this step will perform hole detection and filling. OpenCV's `findContours` and `floodFill` functions will be used for this task. The code for this process is shown in Figure 3.7. Figure 3.8 illustrates a brief process of flood filling.

```

#Hole checking
def hole_detection(mask):
    #Find Contour
    contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_TC89_L1)
    hole = 0
    #Count Hole in Hierarchy
    for h in hierarchy[0]:
        if h[2] != -1:
            hole = hole+1
    return hole

#Flood Filling
def flood_fill(mask):
    #Mask used to Flood Filling.
    im_floodfill = mask.copy()
    #Extract Hight and Width (height x width x channel)
    h, w = mask.shape[:2]
    floodmask = np.zeros((h+2, w+2), np.uint8)
    cv2_imshow(floodmask)
    cv2_imshow(im_floodfill)
    # Floodfill from 0,0 with White Color
    cv2.floodFill(im_floodfill, floodmask, (0,0), 255)
    # Invert floodfilled image
    im_floodfill_inv = cv2.bitwise_not(im_floodfill)
    # Combine the two images to get the foreground.
    filled_mask = mask+im_floodfill_inv
    return filled_mask

```

Figure 3.7 Hole checking and flood filling code

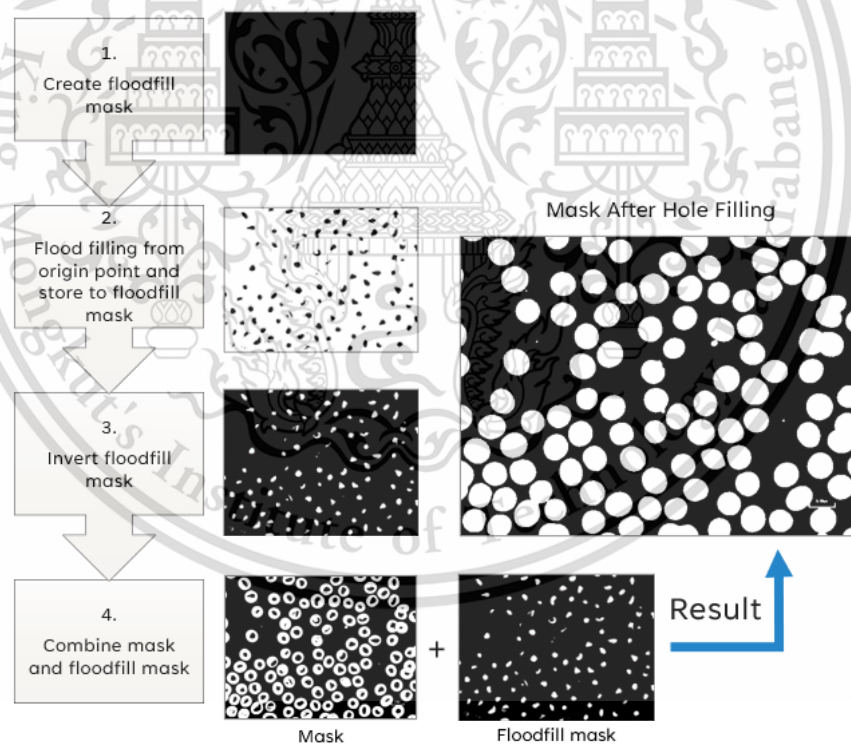


Figure 3.8 Brief process of flood filling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Removing small object

To improve quality of the white mask, this stage removes small objects using the `connectedComponentsWithStats` function from the OpenCV library. This function provides valuable information about components in the image, including size, height, width, centroid, and bounding box coordinates. Figure 3.9 illustrates the code used in this process, and Figure 3.10 shows the resulting output image.

```
#Remove Small Object
def remove_small_object(mask):
    nb_components, output, stats, centroids = cv2.connectedComponentsWithStats(mask)
    #Get Component Size from connectedComponentsWithStats
    sizes = stats[:, -1]
    #Exclude Background Size from sizes
    sizes = sizes[1:]
    nb_components = nb_components - 1
    #Define Minimum Size
    min_size = 200
    #Create Black Image According to Output Size
    im_result = np.zeros_like(output)
    #Check Component Size If Greater Than Threshold Set to White
    for n in range(nb_components):
        if sizes[n] >= min_size:
            im_result[output == n + 1] = 255
    return im_result
```

Figure 3.9 Remove small object code

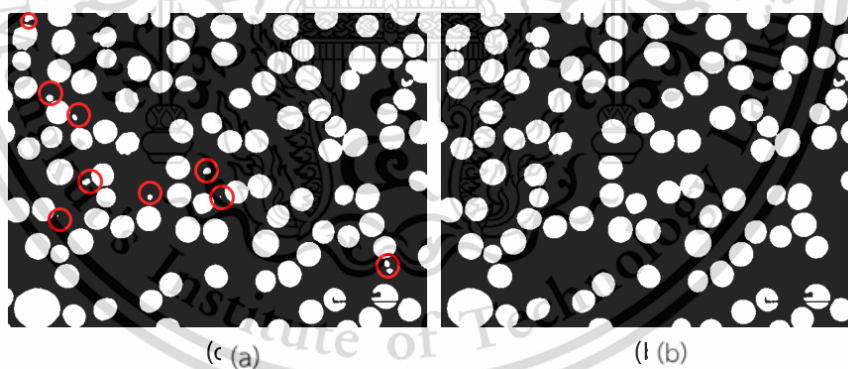


Figure 3.10 (a) Mask with small object in red circle,
(b) Mask after removing small object

3.2.5 Watershed Transform

Some blood smear images in the dataset contain overlapping red blood cells. To isolate these cells, the watershed transform algorithm is used to segment the white blood cell masks from the previous process, which are concatenated together.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The watershed algorithm can be imported from `skimage.segmentation`. The explanation of watershed segmentation follows each step:

1. Distance transformation: The code calculates the distance of each pixel within an object to the nearest background pixel. This results in a grayscale image where brighter areas represent a greater distance from the background.
2. Find local maxima: Peaks are identified within the distance transformed image. These peaks correspond to the centers of the object.
3. Create a new mask: A new mask is created with the same shape as the distance image, with all values set to false except for those corresponding to the peaks, which are set to true.
4. Label the mask: The code assigns unique labels to each connected object within the mask.
5. Watershed segmentation: The watershed function performs segmentation on the distance transformed image, treating it as a topographic map.
6. Return the mask: The code returns a final image containing distinct masks for each individual object.

The code used in this study is shown in Figure 3.11, and Figure 3.12 shows the resulting output of the separated masks.

```

from skimage.feature import peak_local_max
from skimage.segmentation import watershed
from scipy import ndimage as ndi
from skimage import morphology

#Separating Object Mask
def separating_object(mask):
    distance = ndi.distance_transform_edt(mask)
    coords = peak_local_max(distance, min_distance = 25, footprint=np.ones((5,5)), labels=mask)
    np_mask = np.zeros(distance.shape, dtype=bool)
    np_mask[tuple(coords.T)] = True
    markers, _ = ndi.label(np_mask)
    water_mask = watershed(-distance, markers, mask=mask)
    return water_mask

```

Figure 3.11 Code use for separate white masks

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

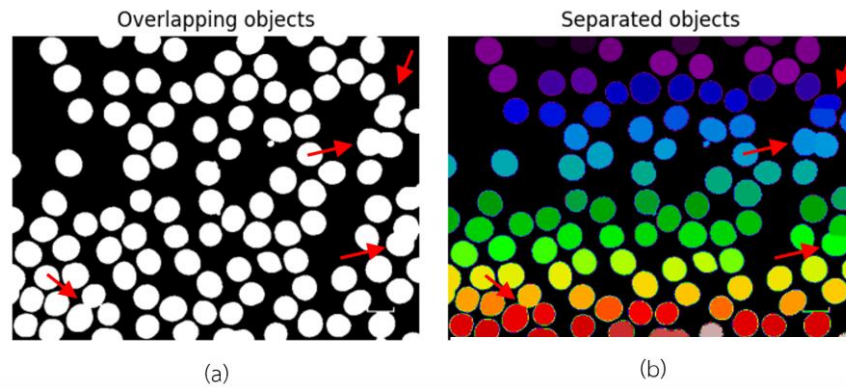


Figure 3.12 (a) Concatenated white masks (Red arrow),
(b) Separated masks obtained from code identified by color

3.2.6 Cropping Region of Interest and Collecting information

This step involves cropping the image to extract the Region of Interest (ROI) and collecting the following data: Running Number of the ROI, Position of the bounding box, and Area of all contours in the image. The collected data are stored in a List. A loop programming is used to combine each white mask with the original dataset image to display the Region of Interest for use in the next process. The code and the result of mask combining, bounding box finding, and cropping with information collection are shown in Figures 3.13, 3.14, and 3.15, respectively.

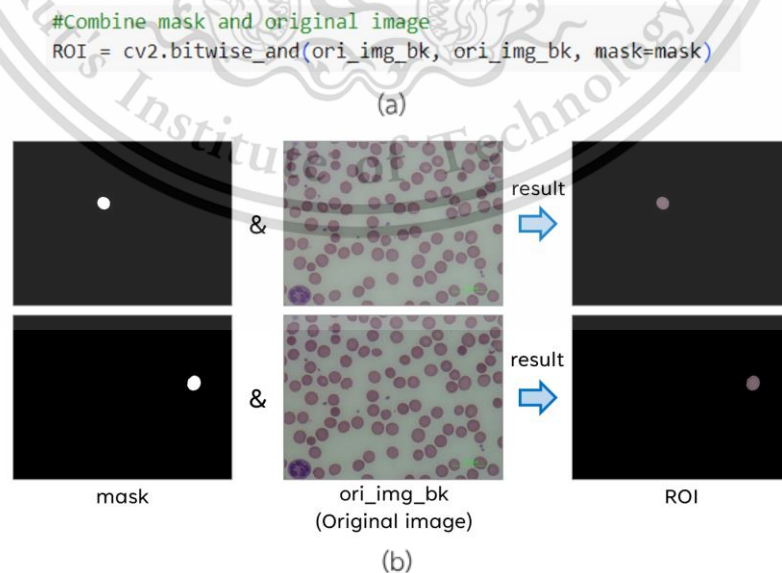


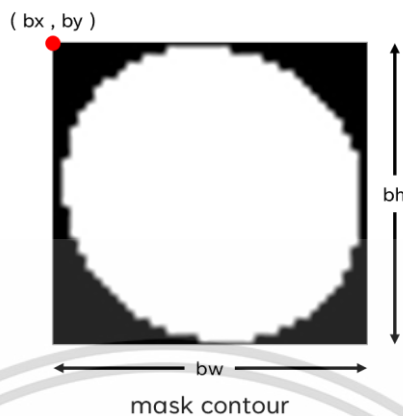
Figure 3.13 (a) Mask combining code,

(b) Combining operation and result of mask combining code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวิชาการเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#Compute Bounding Box
bx, by, bw, bh = cv2.boundingRect(c)
```

(a)



(b)

Figure 3.14 (a) Bounding box finding code, (b) Bounding box components

```
#Crop ROI
cropped_image = ROI[by:by+bh,bx:bx+bw]
print("Component:",cnt_count)
cv2.imshow(cropped_image)
print("Center of this contour",cx,"",cY)
print("Area",cv2.contourArea(c))

#Store Cropped ROI detail
detail=[cnt_count,bx,bx+bw,by,by+bh,cv2.contourArea(c)]
print(detail)
cropped_list.append(cropped_image)
details_list.append(detail)
```

(a)

(b)

Figure 3.15 (a) Cropping with information collection code, (b) Example result

3.2.7 Image Padding and Data Labeling

In this process, cropped red blood cells from the previous step will be padded to a uniform size of 100x100 pixels and labeled with their respective classes from the label text file. The code for this process is separated into three parts: label reading, image padding, and label assignment. The format of the label text file is in the form of X coordinate, Y coordinate, and image label. Figure 3.16 illustrates an example of a text label file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line	X Coordinate	Y Coordinate	Image Label
1	448	278	9
2	464	434	9
3	301	114	9
4	115	66	6
5	322	364	3
6	188	400	3
7	366	246	3
8	538	211	3
9	106	205	3

Figure 3.16 Label text file format

To obtain the label information, the process involves extracting and subtracting the information. The coding is shown in Figure 3.17.

```
#Get Image Label
def get_label(label_path):
    label_list=[]
    #Read label file split by line and strip \n the store in label list
    with open(label_path, "r") as f:
        for line in f:
            data = line.split(" ")
            data[2] = data[2].rstrip("\n")
            for i in range(len(data)):
                data[i] = int(data[i])
            label_list.append(data)
    f.close()
    #Get file name and image name that will be used for naming cropped image
    file_name = label_path.rsplit("/", 1)
    img_name = file_name[1].split(".")[0]
    return label_list, img_name
```

Figure 3.17 Label reading code

After obtaining the information from the text file, padding and labeling will be performed together. The concept for labeling the red blood cells is as follows: If the X and Y coordinates exist within the bounding box of an isolated red blood cell, it will be assigned the corresponding class from the label file. The overall concept of padding and labeling are shown in Figure 3.18 and Figure 3.19 respectively.

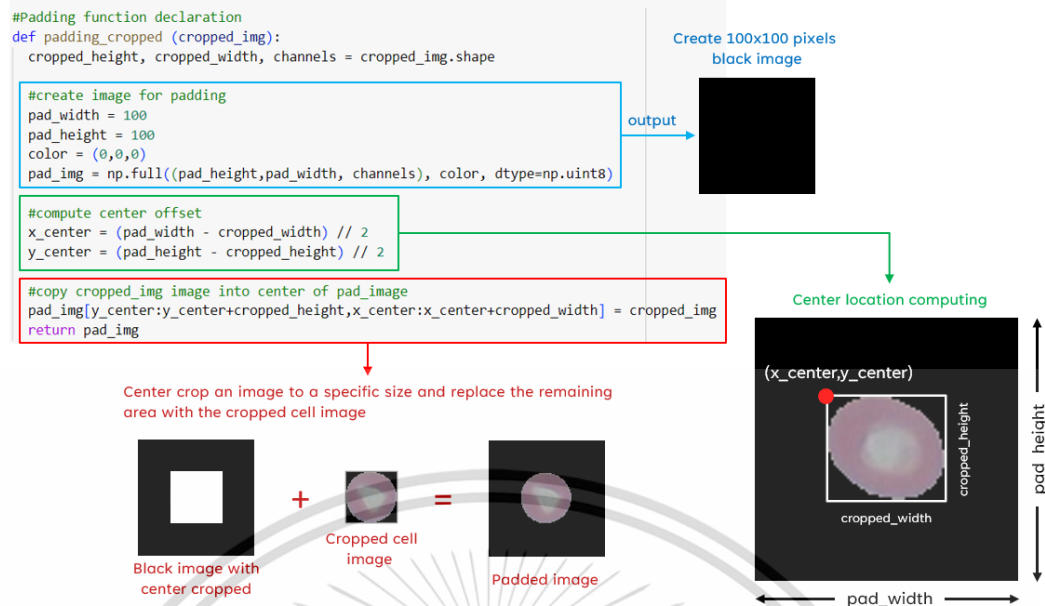


Figure 3.18 The overall concept and process of image padding

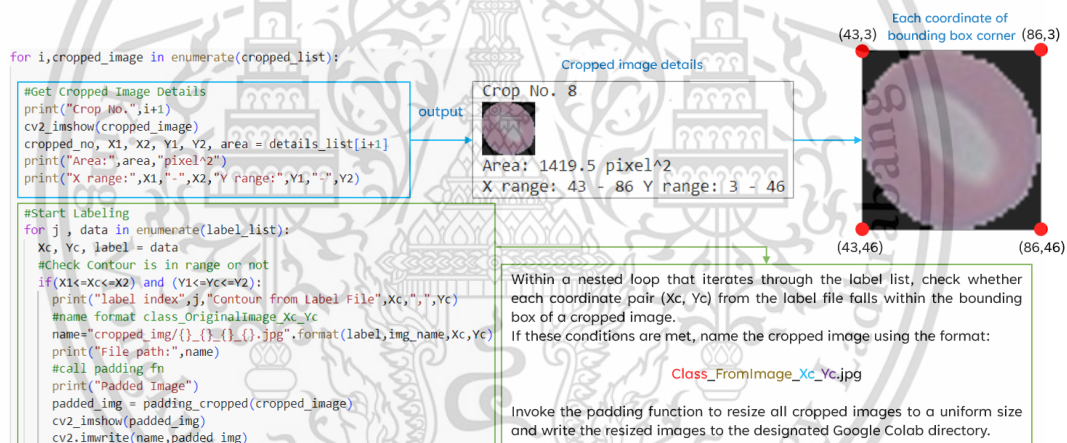


Figure 3.19 The overall concept and process of image labeling

3.3 Imbalance Data Handling and Modeling

This research will compare the performance of three deep learning architectures: ResNet50, InceptionV3, and VGG16. All three models will be trained on the same dataset split into 80% training, 10% validation, and 10% testing sets with stratified sampling, using same hyperparameter settings. Details regarding the hyperparameters are shown in Table 3.1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3.1 Hyperparameter used in this research

Hyperparameter	Value
Epoch range	1-100
Batch size	128
Learning rate	0.001
Loss function	Categorical cross entropy
Activation function	ReLU
Optimizer	Adam

ReLU (Rectified Linear Unit) is chosen as the activation function to determine the output. It behaves by delivering the input directly for positive values, while negative values are set to zero. This characteristic helps reduce the likelihood of vanishing gradients and improves convergence speed during training.

For the optimizer, Adam (Adaptive Moment Estimation) is chosen due to its ability to improve training speeds in deep neural networks and achieve convergence quickly.

The class distribution of the baseline dataset, without performing any imbalance technique, and an example of a cropped data point obtained from the image preprocessing process, are shown in Table 3.2 and Figure 3.20, respectively.

Table 3.2 The amount of class distribution of the baseline dataset

Class Name	Class ID	Amount	Percent Amount
Normal Cell	0	5946	28.38%
Macrocyte	1	682	3.26%
Microcyte	2	438	2.09%
Spherocyte	3	3384	16.15%
Target Cell	4	2349	11.21%
Stomatocyte	5	1987	9.48%
Ovalocyte	6	2075	9.90%
Teardrop	7	295	1.41%
Burr Cell	8	762	3.64%
Schistocyte	9	827	4.16%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3.2 The amount of class distribution of the baseline dataset (Cont.)

Class Name	Class ID	Amount	Percent Amount
Uncategorized	10	145	0.69%
Hypochromia	11	1018	4.86%
Elliptocyte	12	1042	4.97%
Total	20950	100%	

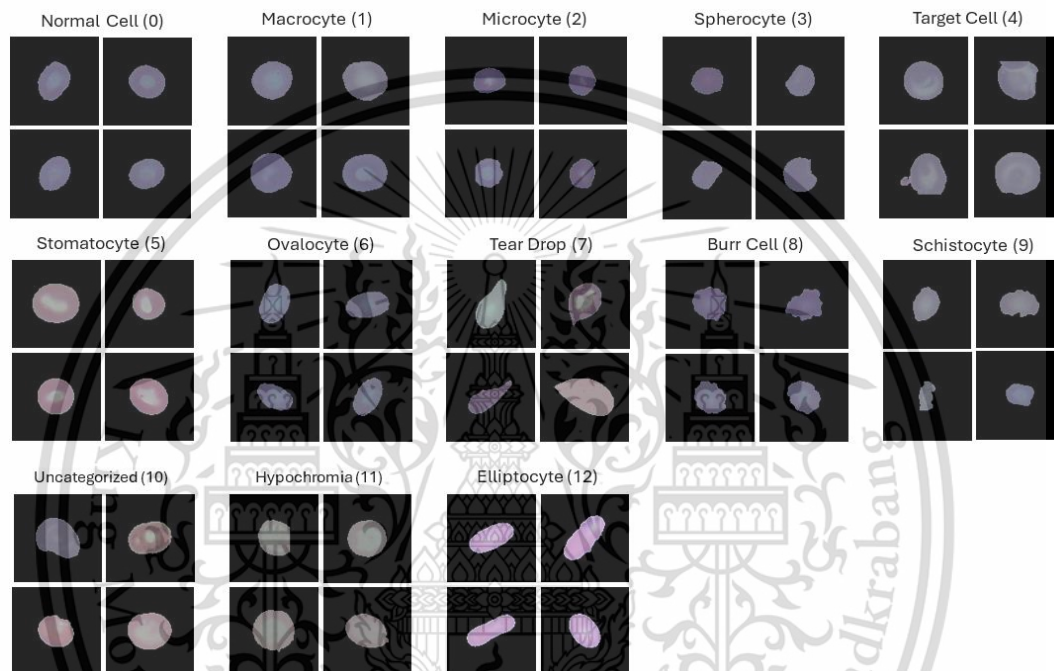


Figure 3.20 An example cropped dataset

3.3.1 Imbalance Data Handling

This study utilizes two techniques for handling class imbalance in the dataset.

3.3.1.1 Weighted method

Weighted method assigns weights to each data point, with a higher weight given to classes with fewer samples. The weight factor is typically proportional to the number of data points in each class. Figure 3.21 illustrates the code used for the weighted method and its result is shown in Table 3.3.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

from sklearn.utils import compute_class_weight
class_weights = compute_class_weight(class_weight = "balanced",
                                   classes = np.unique(train_generator.classes),
                                   y = train_generator.classes)
class_weights = dict(zip(np.unique(train_generator.classes), class_weights))

```

Figure 3.21 Compute class weight code

Table 3.3 Class weight

Class Name	Class	Weight
Normal Cell	0	0.2710
Macrocyte	1	2.3649
Microcyte	2	3.6824
Spherocyte	3	0.4761
Target Cell	4	0.6859
Stomatocyte	5	0.8111
Ovalocyte	6	0.7764
Teardrop	7	5.4612
Burr Cell	8	2.1163
Schistocyte	9	1.9498
Uncategorized	10	11.1107
Hypochromia	11	1.5833
Elliptocyte	12	1.5472

3.3.1.2 SMOTE (Synthetic Minority Oversampling Technique)

SMOTE method addresses class imbalance by synthesizing new data points for the minority class. SMOTE upsamples the minority class until it reaches the same size as the majority class. Figure 3.22 illustrates the code used for SMOTE and the class distribution of the SMOTE dataset is provided in Table 3.4.

```

from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=101)
x_resampled, y_resampled = smote.fit_resample(x_data.reshape(-1,IMG_SIZE*IMG_SIZE*3),y_data)
x_resampled = x_resampled.reshape(-1,IMG_SIZE,IMG_SIZE,3)

```

Figure 3.22 SMOTE code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3.4 The amount of class distribution of SMOTE dataset

Class Name	Class ID	Amount	Percent Amount
Normal Cell	0	5946	7.69%
Macrocyte	1	5946	7.69%
Microcyte	2	5946	7.69%
Spherocyte	3	5946	7.69%
Target Cell	4	5946	7.69%
Stomatocyte	5	5946	7.69%
Ovalocyte	6	5946	7.69%
Teardrop	7	5946	7.69%
Burr Cell	8	5946	7.69%
Schistocyte	9	5946	7.69%
Uncategorized	10	5946	7.69%
Hypochromia	11	5946	7.69%
Elliptocyte	12	5946	7.69%
Total		71488	100%

3.3.2 Modeling

This study employs three pre-trained model architectures from the Keras library and compares their classification performance in terms of accuracy, F1-score, precision, and recall. Additionally, all three models will be augmented with the same number of custom layers.

3.3.2.1 ResNet50

The code used for creating the ResNet50 model is shown in Figures 3.23 and works as follows:

Line 1-8: Defines the create_custom_resnet50 function that takes image size and number of classes as input. It imports a pre-trained ResNet50 model from TensorFlow Keras applications library and sets its argument with 'include_top=False' to exclude the top layers. This allows using an input shape of (100, 100, 3) instead of the default (224, 224, 3). 'pooling='avg''

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

applies global average pooling to the output of the last convolutional block. 'weights='imagenet' is set to speed up the training process.

Line 10-11: 'layer.trainable = False' is set to freeze the weights of all the layers in the pre-trained model.

Line 13-14: Creates a sequential model and adds the pre-trained ResNet50 model as the first layer of the model.

Line 16-25: Adding custom augmented layer for classification which consists of:

1. Dropout(0.5): A dropout layer with a dropout rate of 0.5 to prevent overfitting.
2. Flatten(): Flattens the 3D output into a 1D vector for the fully-connected layer.
3. BatchNormalization(): Normalizes the activations for improved training stability and speed.
4. The final layer with 'num_classes' neurons and Softmax activation ensures the output is a probability distribution.

Line 27-31: Compiles the model and sets the hyperparameters.

```

1 def create_custom_resnet50(img_size, num_classes):
2
3     imported_resnet50 = tf.keras.applications.ResNet50(
4         include_top=False,
5         input_shape=(img_size, img_size, 3),
6         pooling='avg',
7         weights='imagenet'
8     )
9
10    for layer in imported_resnet50.layers:
11        layer.trainable = False
12
13    custom_resnet50 = tf.keras.Sequential()
14    custom_resnet50.add(imported_resnet50)
15
16    custom_resnet50.add(Dropout(0.5))
17    custom_resnet50.add(Flatten())
18    custom_resnet50.add(BatchNormalization())
19    custom_resnet50.add(Dense(128, activation='relu'))
20    custom_resnet50.add(BatchNormalization())
21    custom_resnet50.add(Dropout(0.5))
22    custom_resnet50.add(Dense(64, activation='relu'))
23    custom_resnet50.add(BatchNormalization())
24    custom_resnet50.add(Dropout(0.5))
25    custom_resnet50.add(Dense(num_classes, activation='softmax'))
26
27    adam = Adam(learning_rate=0.001)
28    custom_resnet50.compile(
29        optimizer='adam',
30        loss='categorical_crossentropy',
31        metrics=['accuracy']
32    )
33    return custom_resnet50

```

Figure 3.23 Code used for creating the ResNet50 model

Figure 3.24 illustrates the ResNet50 architecture used in this research. The model consists of 2 layers of pre-trained ResNet50 layers and 10 layers of custom augmented layers. Each block represents a layer and contains a layer name, input shape, and output shape that is provided in array format: (batch size, rows, columns, channels) for 4D arrays and (batch size, size) for 2D arrays.

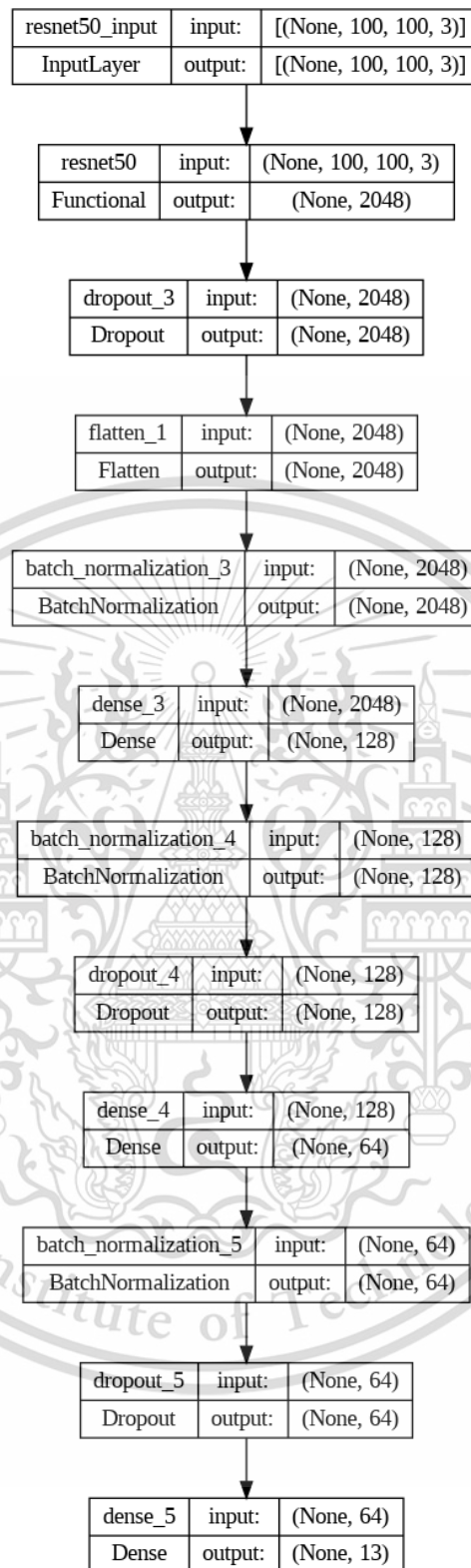


Figure 3.24 The ResNet50 architecture used in this research

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 InceptionV3

The code used for creating the InceptionV3 model shown in Figures 3.25 and works as follows:

Line 1-8: Defines the `create_custom_inceptionv3` function that takes image size and number of classes as input. It imports a pre-trained InceptionV3 model from TensorFlow Keras applications library and sets its argument with `'include_top=False'` to exclude the top layers. This allows using an input shape of (100, 100, 3) instead of the default (299, 299, 3). `'pooling='avg'` applies global average pooling to the output of the last convolutional block. `'weights='imagenet'` is set to speed up the training process.

Line 10-11: `'layer.trainable = False'` is set to freeze the weights of all the layers in the pre-trained model.

Line 13-14: Creates a sequential model and adds the pre-trained InceptionV3 model as the first layer of the model.

Line 16-25: Adding augmented layer for classification which consists of:

1. `Dropout(0.5)`: A dropout layer with a dropout rate of 0.5 to prevent overfitting.
2. `Flatten()`: Flattens the 3D output into a 1D vector for the fully-connected layer.
3. `BatchNormalization()`: Normalizes the activations for improved training stability and speed.
4. The final layer with `'num_classes'` neurons and Softmax activation ensures the output is a probability distribution.

Line 27-31: Compiles the model and sets the hyperparameters.

```

1 def create_custom_inceptionv3(img_size, num_classes):
2
3     imported_inceptionv3 = tf.keras.applications.InceptionV3(
4         include_top=False,
5         input_shape=(img_size, img_size, 3),
6         pooling='avg',
7         weights='imagenet'
8     )
9
10    for layer in imported_inceptionv3.layers:
11        layer.trainable = False
12
13    custom_inceptionv3 = tf.keras.Sequential()
14    custom_inceptionv3.add(imported_inceptionv3)
15
16    custom_inceptionv3.add(Dropout(0.5))
17    custom_inceptionv3.add(Flatten())
18    custom_inceptionv3.add(BatchNormalization())
19    custom_inceptionv3.add(Dense(128, activation='relu'))
20    custom_inceptionv3.add(BatchNormalization())
21    custom_inceptionv3.add(Dropout(0.5))
22    custom_inceptionv3.add(Dense(64, activation='relu'))
23    custom_inceptionv3.add(BatchNormalization())
24    custom_inceptionv3.add(Dropout(0.5))
25    custom_inceptionv3.add(Dense(num_classes, activation='softmax'))
26
27    adam = Adam(learning_rate=0.001)
28    custom_inceptionv3.compile(
29        optimizer='adam',
30        loss='categorical_crossentropy',
31        metrics=['accuracy'])
32    )
33    return custom_inceptionv3

```

Figure 3.25 Code used for creating the InceptionV3 model

Figure 3.26 illustrates the InceptionV3 architecture used in this research. The model consists of 2 layers of pre-trained InceptionV3 layers and 10 layers of custom augmented layers. Each block represents a layer and contains a layer name, input shape, and output shape that is provided in array format: (batch size, rows, columns, channels) for 4D arrays and (batch size, size) for 2D arrays.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

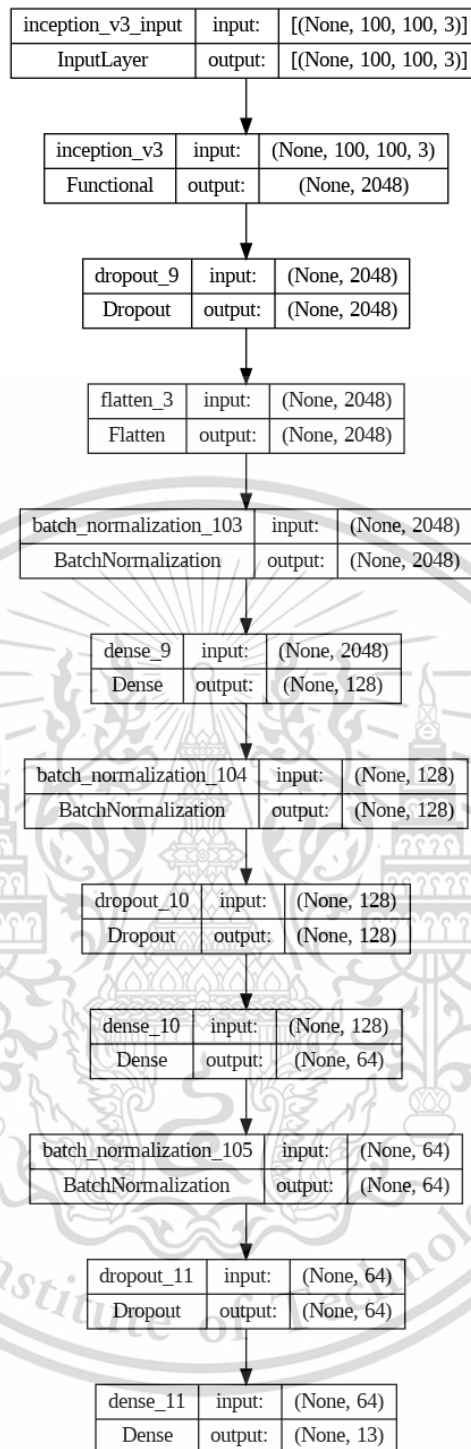


Figure 3.26 The InceptionV3 architecture used in this research

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.3 VGG16

This study chooses VGG16 over VGG19 because the Keras documentation [28] indicates that VGG16 has both higher top-5 accuracy and lower complexity compared to VGG19. The code used for creating the VGG16 model shown in Figures 3.27 and works as follows:

Line 1-8: Defines the `create_custom_vgg16` function that takes image size and number of classes as input. It imports a pre-trained VGG16 model from TensorFlow Keras applications library and sets its argument with `'include_top=False'` to exclude the top layers. This allows using an input shape of (100, 100, 3) instead of the default (224, 224, 3). `'pooling='avg'` applies global average pooling to the output of the last convolutional block. `'weights='imagenet'` is set to speed up the training process.

Line 10-11: `'layer.trainable = False'` is set to freeze the weights of all the layers in the pre-trained model.

Line 13-14: Creates a sequential model and adds the pre-trained VGG16 model as the first layer of the model.

Line 16-25: Adding augmented layer for classification which consists of:

1. `Dropout(0.5)`: A dropout layer with a dropout rate of 0.5 to prevent overfitting.
2. `Flatten()`: Flattens the 3D output into a 1D vector for the fully-connected layer.
3. `BatchNormalization()`: Normalizes the activations for improved training stability and speed.
4. The final layer with `'num_classes'` neurons and Softmax activation ensures the output is a probability distribution.

Line 27-31: Compiles the model and sets the hyperparameters.

```

1 def create_custom_vgg16(img_size, num_classes):
2
3     imported_vgg16 = tf.keras.applications.VGG16(
4         include_top=False,
5         input_shape=(img_size, img_size, 3),
6         pooling='avg',
7         weights='imagenet'
8     )
9
10    for layer in imported_vgg16.layers:
11        layer.trainable = False
12
13    custom_vgg16 = tf.keras.Sequential()
14    custom_vgg16.add(imported_vgg16)
15
16    custom_vgg16.add(Dropout(0.5))
17    custom_vgg16.add(Flatten())
18    custom_vgg16.add(BatchNormalization())
19    custom_vgg16.add(Dense(128, activation='relu'))
20    custom_vgg16.add(BatchNormalization())
21    custom_vgg16.add(Dropout(0.5))
22    custom_vgg16.add(Dense(64, activation='relu'))
23    custom_vgg16.add(BatchNormalization())
24    custom_vgg16.add(Dropout(0.5))
25    custom_vgg16.add(Dense(num_classes, activation='softmax'))
26
27    adam = Adam(learning_rate=0.001)
28    custom_vgg16.compile(
29        optimizer='adam',
30        loss='categorical_crossentropy',
31        metrics=['accuracy']
32    )
33    return custom_vgg16

```

Figure 3.27 Code used for creating the VGG16 model

Figure 3.28 illustrates the VGG16 architecture used in this research. The model consists of 2 layers of pre-trained VGG16 layers and 10 layers of custom augmented layers. Each block represents a layer and contains a layer name, input shape, and output shape that is provided in array format: (batch size, rows, columns, channels) for 4D arrays and (batch size, size) for 2D arrays.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

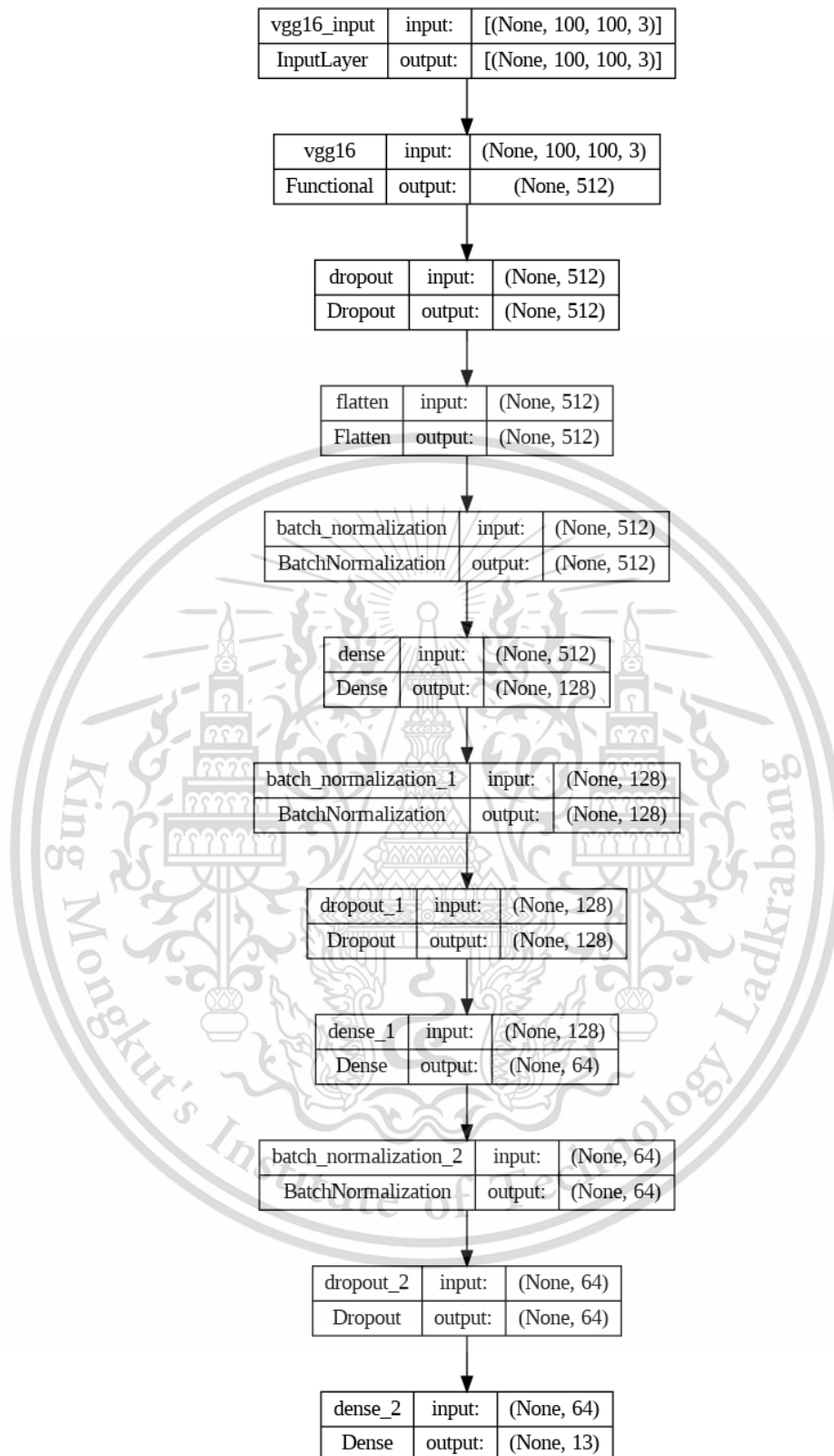


Figure 3.28 The VGG16 architecture used in this research

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 4

RESULTS

Image segmentation and classification of red blood cells based on morphology, this research evaluates the performance of three deep learning architectures including ResNet50, InceptionV3, and VGG16. Each architecture will be tested on three datasets consist of baseline dataset, a dataset with weighted class balancing, and a dataset with SMOTE method handling. The comparison of each model's performance will be based on accuracy, loss, and classification ability for each class. The results for each architecture are presented below.

4.1 Results of ResNet50 architecture with baseline dataset

The results of ResNet50 architecture with baseline dataset consist of accuracy and loss of training and validation set graph, confusion matrix, and classification report.

Figure 4.1 presents the accuracy and loss graphs for the training and validation datasets. The red lines represent the training data, while the blue lines represent the validation data, which acts as a test set for model optimization. The graph shows a clear increase in validation accuracy during the initial epochs, followed by gradual changes. Similarly, the validation loss decreases significantly during the initial epochs. The model was trained for 100 epochs, using the validation loss as the metric for evaluating model performance at its best epoch. The results show that the model learned the data best at epoch 56, with a validation loss of 0.4954, validation accuracy of 0.8467, training loss of 0.5735, and training accuracy of 0.8140. When the best epoch model was tested for classification with the testing dataset, the results are shown in the confusion matrix, Figure 4.2 and classification report, Figure 4.3.

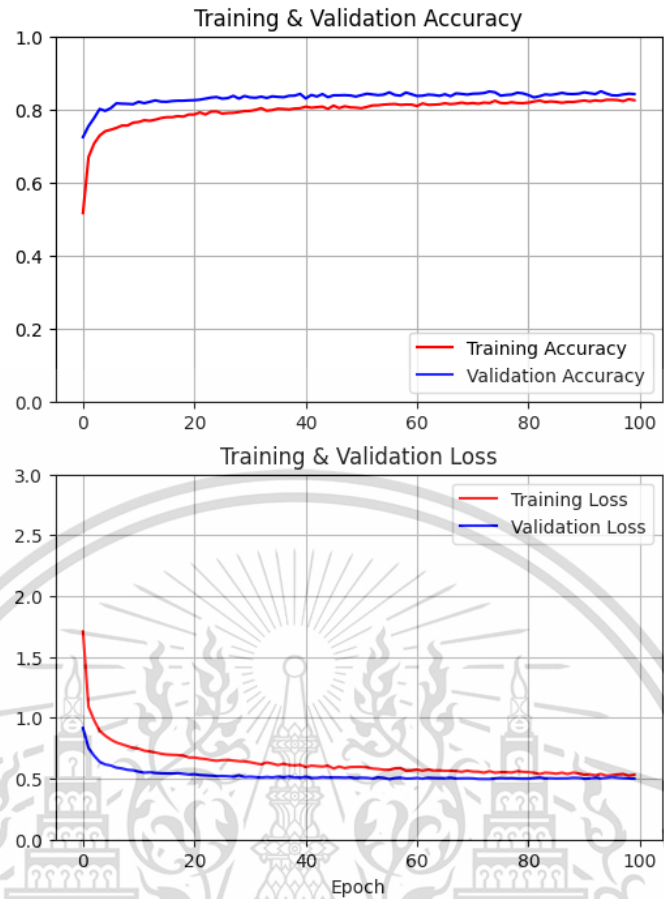


Figure 4.1 Accuracy and loss graph of ResNet50 architecture with baseline dataset

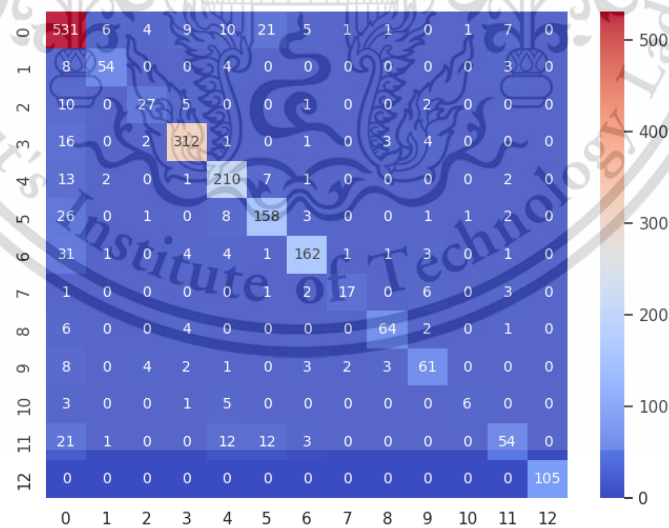


Figure 4.2 Confusion matrix result using baseline ResNet50 model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.7878	0.8909	0.8362	596
Macrocyte	1	0.8438	0.7826	0.8120	69
Microcyte	2	0.7105	0.6000	0.6506	45
Spherocyte	3	0.9231	0.9204	0.9217	339
Target Cell	4	0.8235	0.8898	0.8554	236
Stomatocyte	5	0.7900	0.7900	0.7900	200
Ovalocyte	6	0.8950	0.7751	0.8308	209
Teardrop	7	0.8095	0.5667	0.6667	30
Burr Cell	8	0.8889	0.8312	0.8591	77
Schistocyte	9	0.7722	0.7262	0.7485	84
Uncategorized	10	0.7500	0.4000	0.5217	15
Hypochromia	11	0.7397	0.5243	0.6136	103
Elliptocyte	12	1.0000	1.0000	1.0000	105
accuracy				0.8354	2108
macro avg		0.8257	0.7459	0.7774	2108
weighted avg		0.8359	0.8354	0.8324	2108

Figure 4.3 Classification report result using baseline ResNet50 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1. Elliptocyte, 2.Spherocyte, 3.Burr Cell, 4.Target Cell, 5.Normal Cell, 6.Ovalocyte, 7.Macrocyte, 8.Stomatocyte, 9.Schistocyte, 10.Teardrop, 11.Microcyte, 12.Hypochromia, 13.Uncategorized. The baseline ResNet50 model achieved an average accuracy of 83.54%.

4.2 Results of ResNet50 architecture with weighted dataset

Figure 4.4 presents the accuracy and loss graphs for the training and validation datasets of ResNet50 architecture with weighted dataset. The red lines represent the training data, while the blue lines represent the validation data, which acts as a test set for model optimization. The model was trained for 100 epochs, using the validation loss as the metric for evaluating model performance at its best epoch. The results show that the model learned the data best at epoch 84, with a validation loss of

0.6021, validation accuracy of 0.8050, training loss of 0.5735, and training accuracy of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.8140. When the best epoch model was tested for classification with the testing dataset, the results are shown in the confusion matrix, Figure 4.5 and classification report, Figure 4.6.

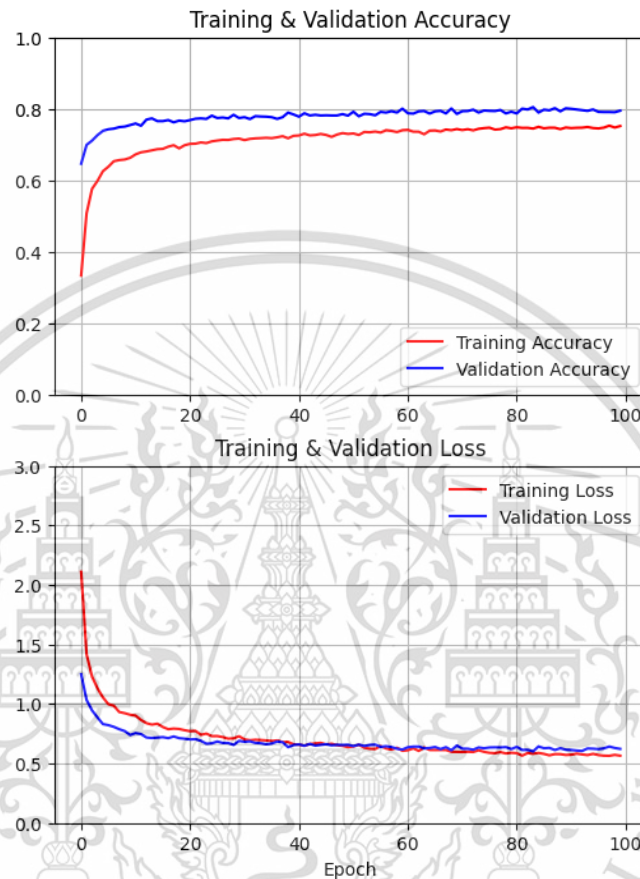


Figure 4.4 Accuracy and loss graph of ResNet50 architecture with weighted dataset

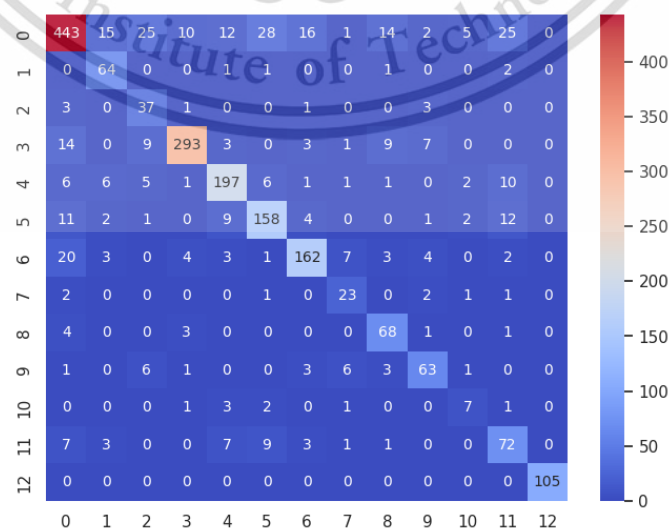


Figure 4.5 Confusion matrix result using weighted ResNet50 model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่เป็นการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8669	0.7433	0.8004	596
Macrocyte	1	0.6882	0.9275	0.7901	69
Microcyte	2	0.4458	0.8222	0.5781	45
Spherocyte	3	0.9331	0.8643	0.8974	339
Target Cell	4	0.8383	0.8347	0.8365	236
Stomatocyte	5	0.7670	0.7900	0.7783	200
Ovalocyte	6	0.8394	0.7751	0.8060	209
Teardrop	7	0.5610	0.7667	0.6479	30
Burr Cell	8	0.6800	0.8831	0.7684	77
Schistocyte	9	0.7590	0.7500	0.7545	84
Uncategorized	10	0.3889	0.4667	0.4242	15
Hypochromia	11	0.5714	0.6990	0.6288	103
Elliptocyte	12	1.0000	1.0000	1.0000	105
accuracy				0.8027	2108
macro avg		0.7184	0.7941	0.7470	2108
weighted avg		0.8206	0.8027	0.8071	2108

Figure 4.6 Classification report result using weighted ResNet50 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Spherocyte, 3.Target Cell, 4.Ovalocyte, 5.Normal Cell, 6.Macrocyte, 7.Stomatocyte, 8.Burr Cell, 9.Schistocyte, 10.Teardrop, 11.Hypochromia, 12.Microcyte, 13.Uncategorized. The weighted ResNet50 model achieved an average accuracy of 80.27%.

4.3 Results of ResNet50 architecture with SMOTE dataset

Figure 4.7 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. The red lines represent the training data, and the blue lines represent the validation data, which is used for model optimization. The model was trained for 100 epochs, using the validation loss to evaluate its performance at the optimal epoch. The results indicate that the model achieved its best performance at epoch 100, with a validation loss of

0.2571, validation accuracy of 0.9173, training loss of 0.4255, and training accuracy of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.8624. The graph suggests that the model's performance might continue to improve with further training. However, for comparison purposes in this research, the model at epoch 100 is used. The classification results for the testing dataset are shown in confusion matrix, Figure 4.8 and classification report, Figure 4.9.

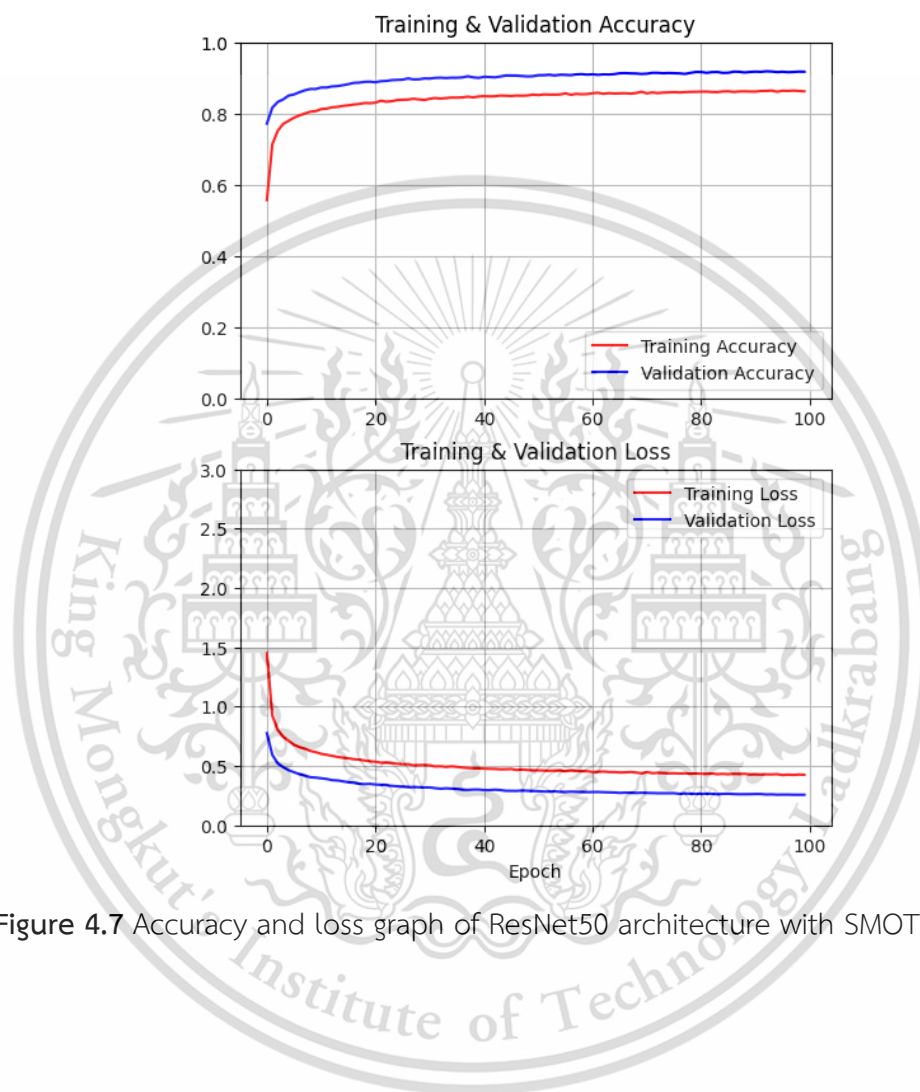


Figure 4.7 Accuracy and loss graph of ResNet50 architecture with SMOTE dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

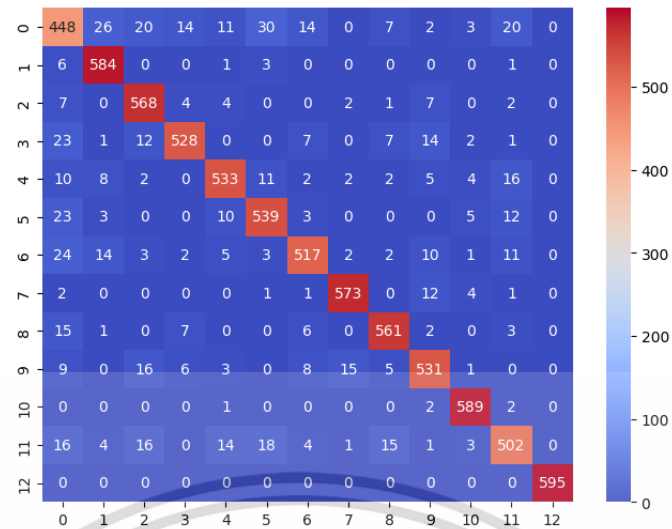


Figure 4.8 Confusion matrix result using SMOTE ResNet50 model

	precision	recall	f1-score	support	
Normal Cell	0	0.7684	0.7529	0.7606	595
Macrocyte	1	0.9111	0.9815	0.9450	595
Microcyte	2	0.8917	0.9546	0.9221	595
Spherocyte	3	0.9412	0.8874	0.9135	595
Target Cell	4	0.9158	0.8958	0.9057	595
Stomatocyte	5	0.8909	0.9059	0.8983	595
Ovalocyte	6	0.9199	0.8704	0.8945	594
Teardrop	7	0.9630	0.9646	0.9638	594
Burr Cell	8	0.9350	0.9429	0.9389	595
Schistocyte	9	0.9061	0.8939	0.9000	594
Uncategorized	10	0.9624	0.9916	0.9768	594
Hypochromia	11	0.8792	0.8451	0.8618	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy			0.9144	7730	
macro avg	0.9142	0.9144	0.9139	7730	
weighted avg	0.9142	0.9144	0.9139	7730	

Figure 4.9 Classification report result using SMOTE ResNet50 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Uncategorized, 3.Teardrop, 4.Macrocyte, 5.Burr Cell, 6.Microcyte, 7.Spherocyte, 8.Target Cell, 9.Schistocyte, 10.Stomatocyte, 11.Ovalocyte,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12.Hypochromia, 13.Normal Cell. The SMOTE ResNet50 model achieved an average accuracy of 91.44%.

4.4 Results of InceptionV3 architecture with baseline dataset

Figure 4.10 shows the accuracy and loss graphs for the training and validation datasets of the InceptionV3 architecture with the baseline dataset. The red lines represent the training data, and the blue lines represent the validation data. The results indicate that the model achieved its best performance at epoch 87, with a validation loss of 0.9182, validation accuracy of 0.6871, training loss of 1.0703, and training accuracy of 0.6317. The classification results for the testing dataset are shown in confusion matrix, Figure 4.9 and classification report, Figure 4.11.

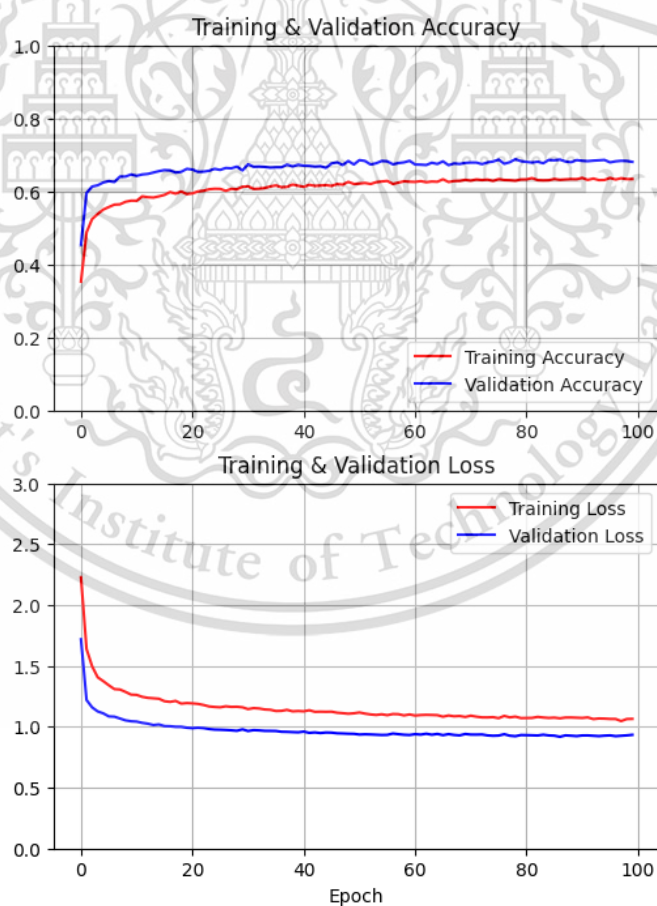


Figure 4.10 Accuracy and loss graph of InceptionV3 architecture with baseline dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

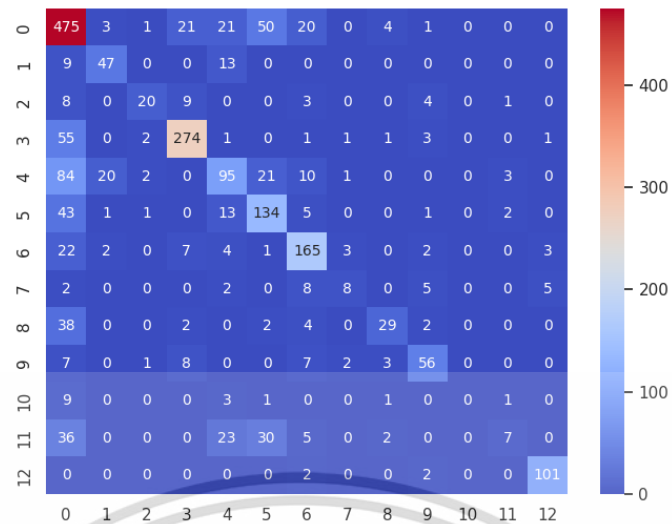


Figure 4.11 Confusion matrix result using baseline InceptionV3 model

	precision	recall	f1-score	support	
Normal Cell	0	0.6028	0.7970	0.6864	596
Macrocyte	1	0.6438	0.6812	0.6620	69
Microcyte	2	0.7407	0.4444	0.5556	45
Spherocyte	3	0.8536	0.8083	0.8303	339
Target Cell	4	0.5429	0.4025	0.4623	236
Stomatocyte	5	0.5607	0.6700	0.6105	200
Ovalocyte	6	0.7174	0.7895	0.7517	209
Teardrop	7	0.5333	0.2667	0.3556	30
Burr Cell	8	0.7250	0.3766	0.4957	77
Schistocyte	9	0.7368	0.6667	0.7000	84
Uncategorized	10	0.0000	0.0000	0.0000	15
Hypochromia	11	0.5000	0.0680	0.1197	103
Elliptocyte	12	0.9182	0.9619	0.9395	105
accuracy				0.6694	2108
macro avg	0.6212	0.5333	0.5515	2108	
weighted avg	0.6633	0.6694	0.6490	2108	

Figure 4.12 Classification report result using baseline InceptionV3 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Spherocyte, 3.Ovalocyte, 4.Schistocyte, 5.Normal Cell, 6.Macrocyte, 7. Stomatocyte, 8.Microcyte, 9.Burr Cell, 10.Target Cell, 11.Teardrop,

12.Hypochromia and the model cannot classify the Uncategorized class. The baseline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นไปใช้ประโยชน์ใด ๆ การทำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InceptionV3 model achieved an average accuracy of 66.94%, This is not a good accuracy value for a classification task.

4.5 Results of InceptionV3 architecture with weighted dataset

Figure 4.13 shows the accuracy and loss graphs for the training and validation datasets of the InceptionV3 architecture with the weighted dataset. The red lines represent the training data, and the blue lines represent the validation data. The results indicate that the model achieved its best performance at epoch 94, with a validation loss of 1.1229, validation accuracy of 0.6148, training loss of 1.1276, and training accuracy of 0.5190. The classification results for the testing dataset are shown in confusion matrix, Figure 4.14 and classification report, Figure 4.15.

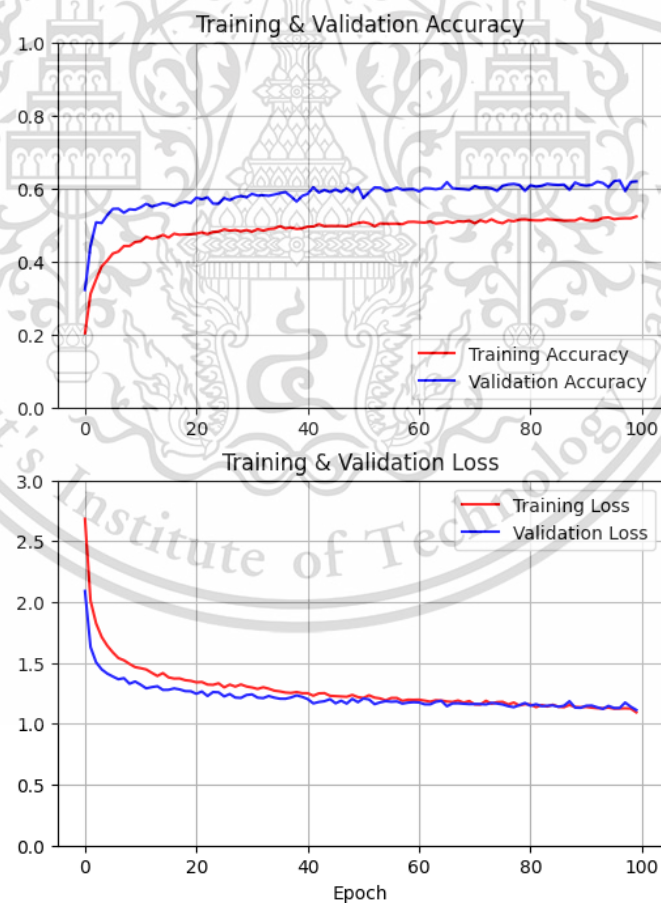


Figure 4.13 Accuracy and loss graph of InceptionV3 architecture with weighted dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

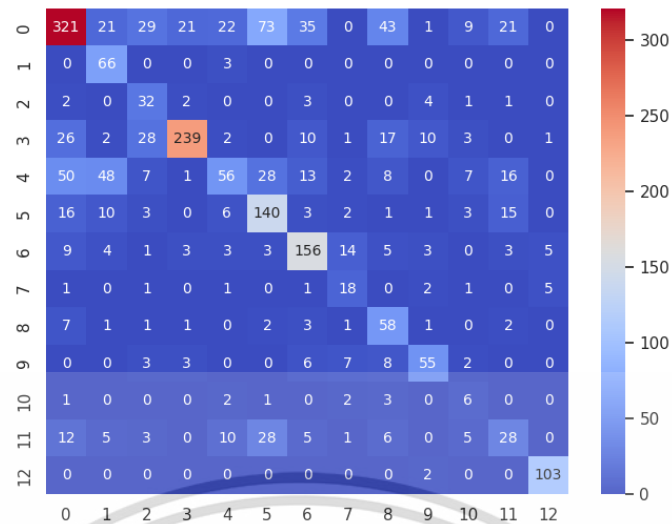


Figure 4.14 Confusion matrix result using weighted InceptionV3 model

	precision	recall	f1-score	support	
Normal Cell	0	0.7213	0.5386	0.6167	596
Macrocyte	1	0.4204	0.9565	0.5841	69
Microcyte	2	0.2963	0.7111	0.4183	45
Spherocyte	3	0.8852	0.7050	0.7849	339
Target Cell	4	0.5333	0.2373	0.3284	236
Stomatocyte	5	0.5091	0.7000	0.5895	200
Ovalocyte	6	0.6638	0.7464	0.7027	209
Teardrop	7	0.3750	0.6000	0.4615	30
Burr Cell	8	0.3893	0.7532	0.5133	77
Schistocyte	9	0.6962	0.6548	0.6748	84
Uncategorized	10	0.1622	0.4000	0.2308	15
Hypochromia	11	0.3256	0.2718	0.2963	103
Elliptocyte	12	0.9035	0.9810	0.9406	105
accuracy			0.6063	2108	
macro avg	0.5293	0.6351	0.5494	2108	
weighted avg	0.6496	0.6063	0.6062	2108	

Figure 4.15 Classification report result using weighted InceptionV3 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Spherocyte, 3.Ovalocyte, 4.Schistocyte, 5.Normal Cell, 6.Stomatocyte, 7.Macrocyte, 8.Burr Cell, 9.Teardrop, 10.Microcyte, 11.Target Cell,

12.Hypochromia, 13.Uncategorized. This model improved the classification of the

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Uncategorized class from 0.0000 to 0.2308 f1-score compared to the baseline InceptionV3 model. It achieved an average accuracy of 60.63%, but this is still not a good accuracy value for a classification task.

4.6 Results of InceptionV3 architecture with SMOTE dataset

Figure 4.16 shows the accuracy and loss graphs for the training and validation datasets of the InceptionV3 architecture with the SMOTE dataset. The red lines represent the training data, and the blue lines represent the validation data. The results indicate that the model achieved its best performance at epoch 99, with a validation loss of 0.7176, validation accuracy of 0.7643, training loss of 0.9726, and training accuracy of 0.6749. The classification results for the testing dataset are shown in confusion matrix, Figure 4.17 and classification report, Figure 4.18.

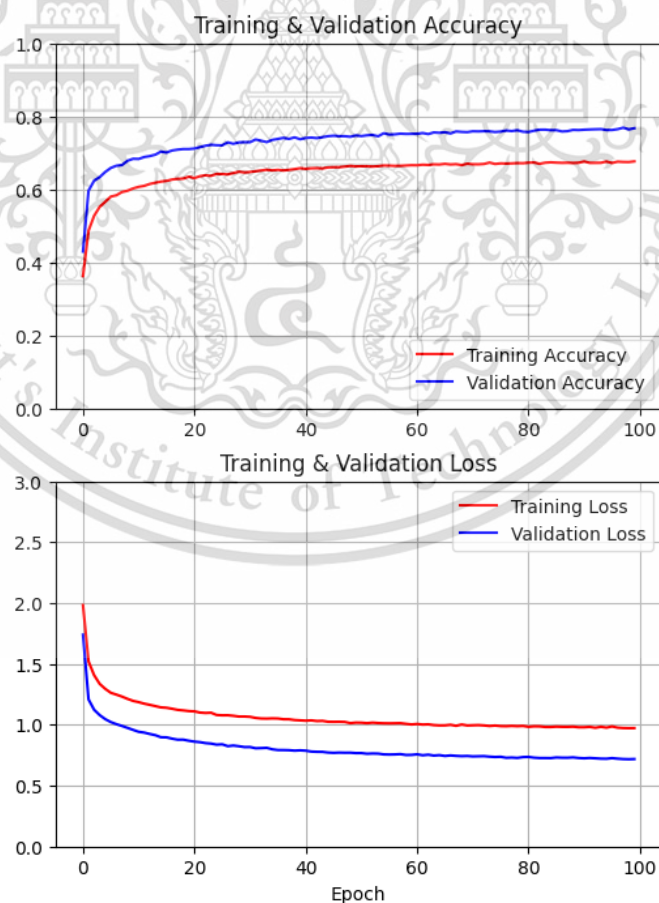


Figure 4.16 Accuracy and loss graph of InceptionV3 architecture with SMOTE dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

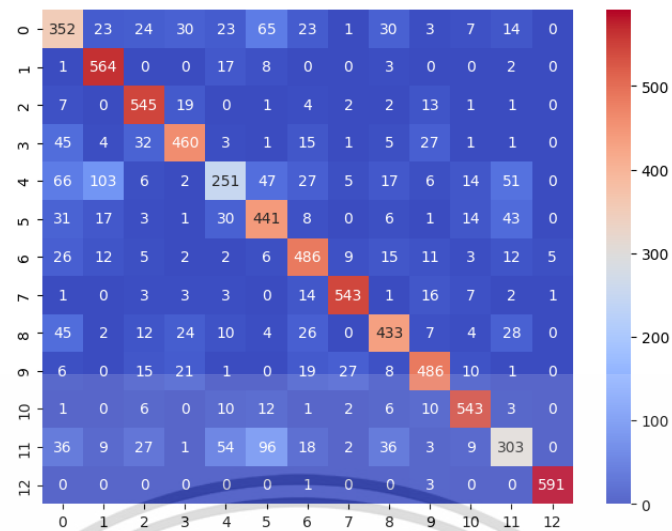


Figure 4.17 Confusion matrix result using SMOTE InceptionV3 model

	precision	recall	f1-score	support	
Normal Cell	0	0.5705	0.5916	0.5809	595
Macrocyte	1	0.7684	0.9479	0.8488	595
Microcyte	2	0.8038	0.9160	0.8562	595
Spherocyte	3	0.8171	0.7731	0.7945	595
Target Cell	4	0.6213	0.4218	0.5025	595
Stomatocyte	5	0.6476	0.7412	0.6912	595
Ovalocyte	6	0.7570	0.8182	0.7864	594
Teardrop	7	0.9172	0.9141	0.9157	594
Burr Cell	8	0.7705	0.7277	0.7485	595
Schistocyte	9	0.8294	0.8182	0.8237	594
Uncategorized	10	0.8858	0.9141	0.8998	594
Hypochromia	11	0.6573	0.5101	0.5744	594
Elliptocyte	12	0.9899	0.9933	0.9916	595
accuracy			0.7759	7730	
macro avg	0.7720	0.7760	0.7703	7730	
weighted avg	0.7720	0.7759	0.7703	7730	

Figure 4.18 Classification report result using SMOTE InceptionV3 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Teardrop, 3.Uncategorized, 4.Microcyte, 5.Macrocyte, 6.Schistocyte, 7.Spherocyte, 8.Ovalocyte, 9.Burr Cell, 10.Stomatocyte, 11.Normal Cell,

12.Hypochromia, 13.Target Cell. This model significantly improved the classification of

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเฉพาะในชั้นเรียน เมื่อผู้ดูแลเนื้อหาเว็บไซต์ได้ดำเนินการจัดทำเอกสารนี้ขึ้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the Uncategorized class from 0.0000 and 0.2308 to 0.8998 f1-score compared to the baseline and weighted InceptionV3 model. It achieved an average accuracy of 77.59%.

4.7 Results of VGG16 architecture with baseline dataset

Figure 4.19 shows the accuracy and loss graphs for the training and validation datasets of the VGG16 architecture with the baseline dataset. The red lines represent the training data, and the blue lines represent the validation data. The results indicate that the model achieved its best performance at epoch 89, with a validation loss of 0.5893, validation accuracy of 0.8131, training loss of 0.8004, and training accuracy of 0.7384. The classification results for the testing dataset are shown in confusion matrix, Figure 4.20 and classification report, Figure 4.21.

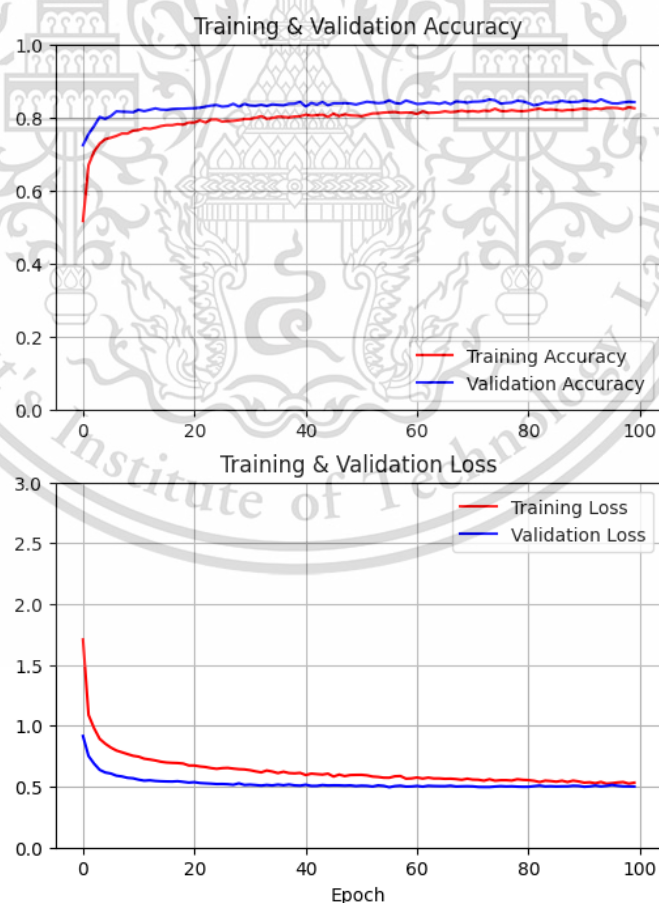


Figure 4.19 Accuracy and loss graph of VGG16 architecture with baseline dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

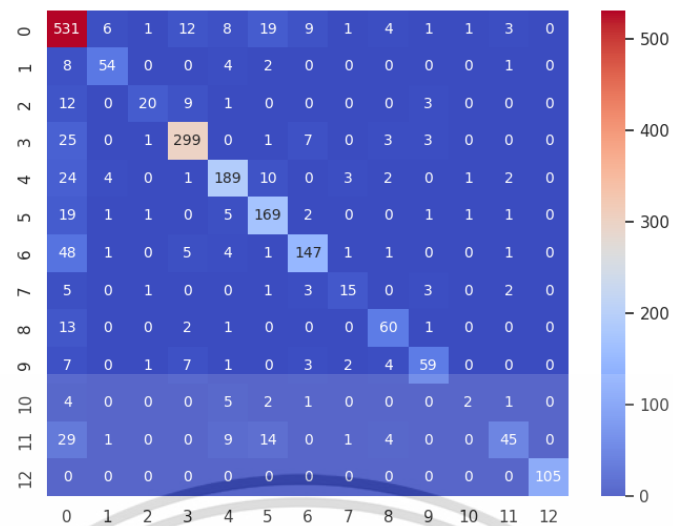


Figure 4.20 Confusion matrix result using baseline VGG16 model

	precision	recall	f1-score	support	
Normal Cell	0	0.7324	0.8909	0.8039	596
Macrocyte	1	0.8060	0.7826	0.7941	69
Microcyte	2	0.8000	0.4444	0.5714	45
Spherocyte	3	0.8925	0.8820	0.8872	339
Target Cell	4	0.8326	0.8008	0.8164	236
Stomatocyte	5	0.7717	0.8450	0.8067	200
Ovalocyte	6	0.8547	0.7033	0.7717	209
Teardrop	7	0.6522	0.5000	0.5660	30
Burr Cell	8	0.7692	0.7792	0.7742	77
Schistocyte	9	0.8310	0.7024	0.7613	84
Uncategorized	10	0.4000	0.1333	0.2000	15
Hypochromia	11	0.8036	0.4369	0.5660	103
Elliptocyte	12	1.0000	1.0000	1.0000	105
accuracy			0.8041	2108	
macro avg	0.7804	0.6847	0.7168	2108	
weighted avg	0.8076	0.8041	0.7980	2108	

Figure 4.21 Classification report result using baseline VGG16 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Spherocyte, 3.Target Cell, 4.Stomatocyte, 5.Normal Cell, 6.Macrocyte, 7.Burr Cell, 8.Ovalocyte, 9.Schistocyte, 10.Microcyte, 11.Hypochromia

and Teardrop, 12.Uncategorized. This model achieved an average accuracy of 80.41%.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 Results of VGG16 architecture with weighted dataset

Figure 4.22 shows the accuracy and loss graphs for the training and validation datasets of the VGG16 architecture with the weighted dataset. The red lines represent the training data, and the blue lines represent the validation data. The results indicate that the model achieved its best performance at epoch 94, with a validation loss of 0.7404, validation accuracy of 0.7542, training loss of 0.8860, and training accuracy of 0.6348. The classification results for the testing dataset are shown in confusion matrix, Figure 4.23 and classification report, Figure 4.24.

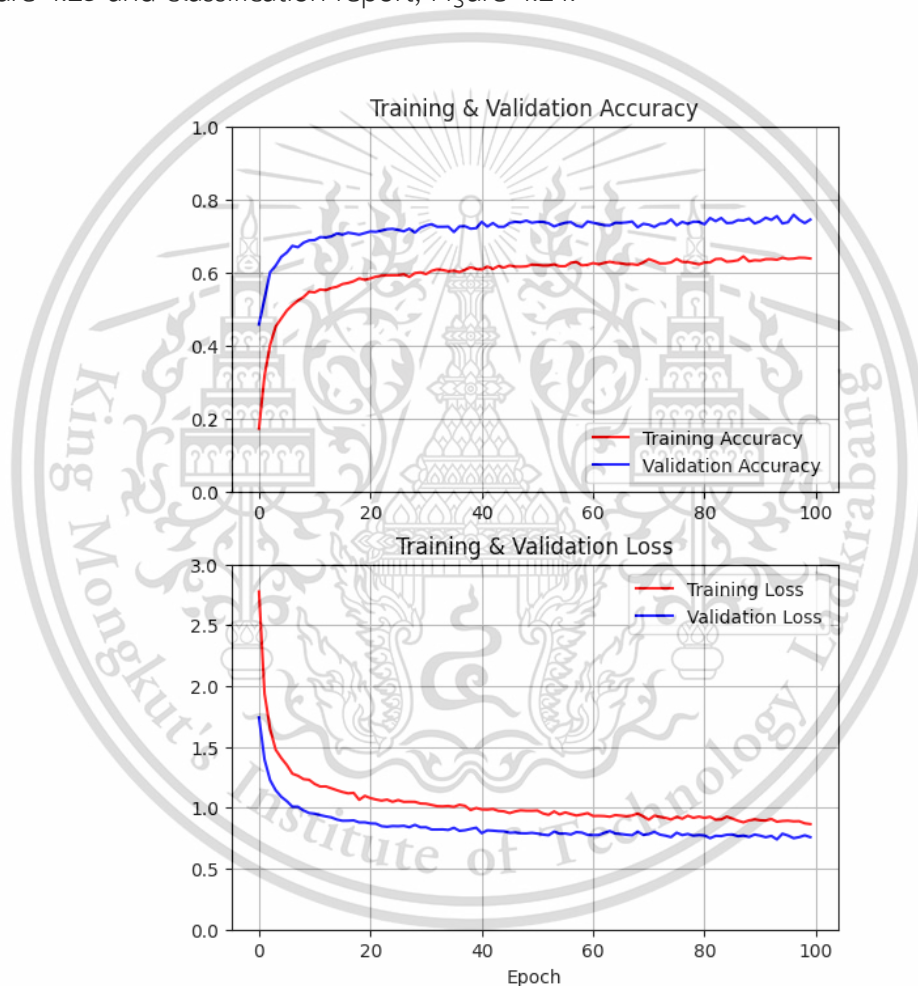


Figure 4.22 Accuracy and loss graph of VGG16 architecture with weighted dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

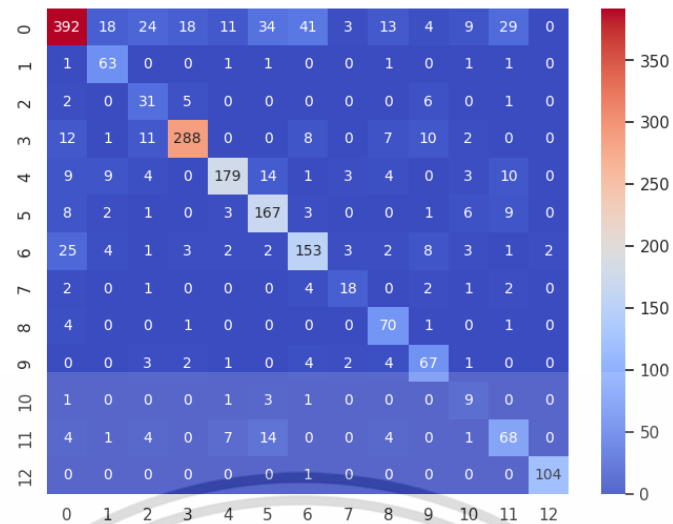


Figure 4.23 Confusion matrix result using weighted VGG16 model

	precision	recall	f1-score	support	
Normal Cell	0	0.8522	0.6577	0.7424	596
Macrocyte	1	0.6429	0.9130	0.7545	69
Microcyte	2	0.3875	0.6889	0.4960	45
Spherocyte	3	0.9085	0.8496	0.8780	339
Target Cell	4	0.8732	0.7585	0.8118	236
Stomatocyte	5	0.7106	0.8350	0.7678	200
Ovalocyte	6	0.7083	0.7321	0.7200	209
Teardrop	7	0.6207	0.6000	0.6102	30
Burr Cell	8	0.6667	0.9091	0.7692	77
Schistocyte	9	0.6768	0.7976	0.7322	84
Uncategorized	10	0.2500	0.6000	0.3529	15
Hypochromia	11	0.5574	0.6602	0.6044	103
Elliptocyte	12	0.9811	0.9905	0.9858	105
accuracy			0.7633	2108	
macro avg	0.6797	0.7686	0.7096	2108	
weighted avg	0.7898	0.7633	0.7686	2108	

Figure 4.24 Classification report result using weighted VGG16 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Spherocyte, 3.Target Cell, 4.Burr Cell, 5.Stomatocyte, 6.Macrocyte, 7.Normal Cell, 8.Schistocyte, 9.Ovalocyte, 10.Teardrop, 11.Hypochromia,

12.Microcyte, 13.Uncategorized. This model achieved an average accuracy of 76.33%.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 Results of VGG16 architecture with SMOTE dataset

Figure 4.25 shows the accuracy and loss graphs for the training and validation datasets of the VGG16 architecture with the weighted dataset. The red lines represent the training data, and the blue lines represent the validation data. The results indicate that the model achieved its best performance at epoch 93, with a validation loss of 0.4366, validation accuracy of 0.8632, training loss of 0.7364, and training accuracy of 0.7666. The classification results for the testing dataset are shown in confusion matrix, Figure 4.26 and classification report, Figure 4.27.

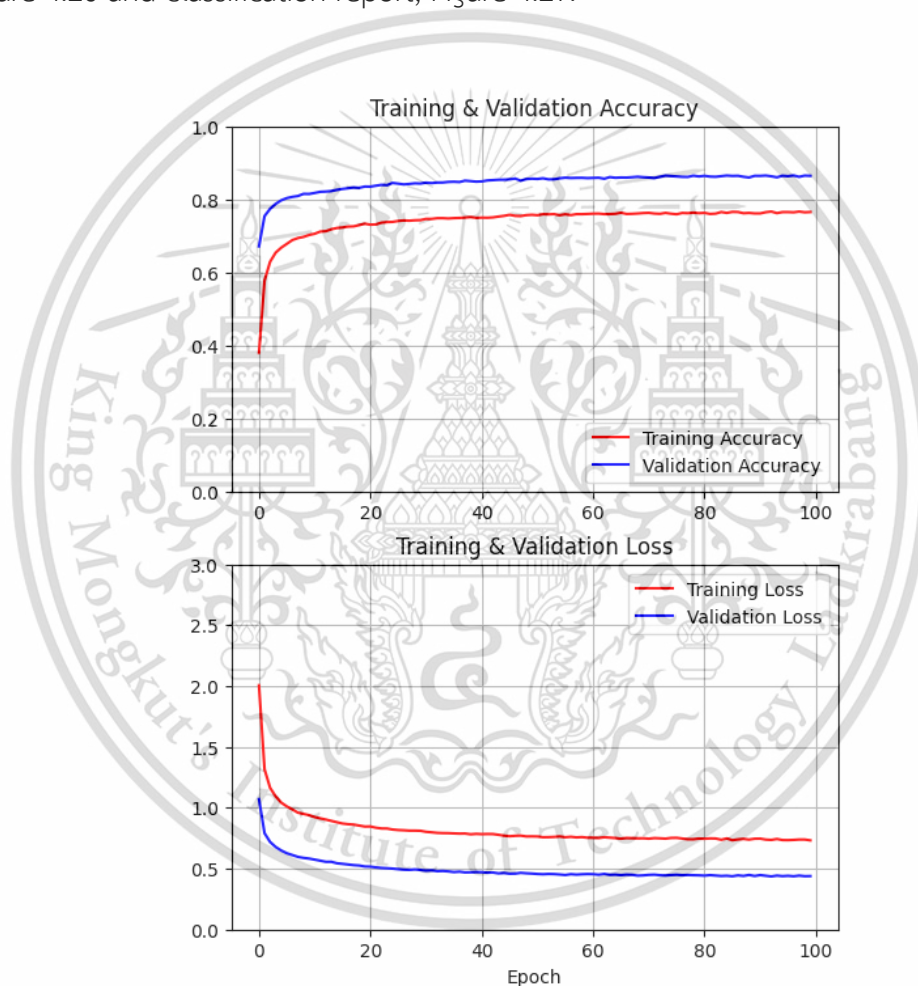


Figure 4.25 Accuracy and loss graph of VGG16 architecture with SMOTE dataset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

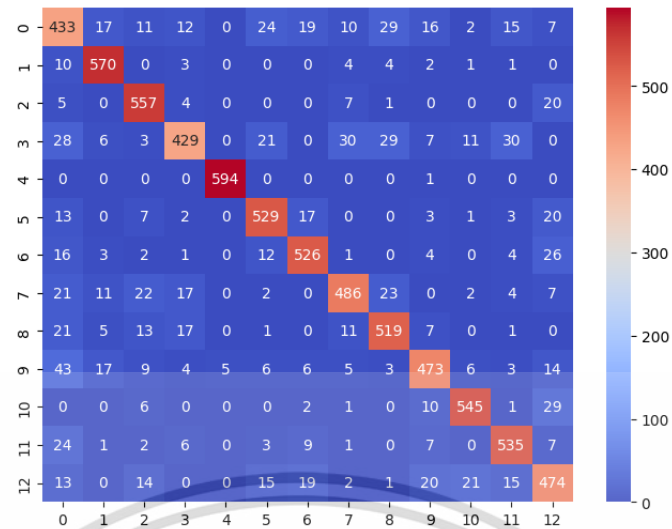


Figure 4.26 Confusion matrix result using SMOTE VGG16 model

	precision	recall	f1-score	support	
Normal Cell	0	0.6950	0.7277	0.7110	595
Macrocyte	1	0.9048	0.9580	0.9306	595
Microcyte	2	0.8729	0.8891	0.8809	595
Spherocyte	3	0.8709	0.8840	0.8774	595
Target Cell	4	0.8663	0.8168	0.8408	595
Stomatocyte	5	0.8536	0.8723	0.8628	595
Ovalocyte	6	0.8727	0.7963	0.8327	594
Teardrop	7	0.9269	0.9175	0.9222	594
Burr Cell	8	0.8814	0.8992	0.8902	595
Schistocyte	9	0.7933	0.8401	0.8160	594
Uncategorized	10	0.8730	0.9377	0.9042	594
Hypochromia	11	0.8667	0.7222	0.7879	594
Elliptocyte	12	0.9917	0.9983	0.9950	595
accuracy			0.8661	7730	
macro avg	0.8669	0.8661	0.8655	7730	
weighted avg	0.8669	0.8661	0.8655	7730	

Figure 4.27 Classification report result using SMOTE VGG16 model

Based on the confusion matrix and classification report, the model can classify red blood cell abnormalities in the following order of performance, measured by f1-score: 1.Elliptocyte, 2.Macrocyte, 3.Teardrop, 4.Uncategorized, 5.Burr Cell, 6.Microcyte, 7.Spherocyte, 8.Stomatocyte, 9.Target Cell, 10.Ovalocyte, 11.Schistocyte, 12.Hypochromia, 13.Normal Cell. This model achieved an average accuracy of 86.61%.

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้เผยแพร่ไปยังเว็บไซต์อื่นโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 Model Comparison

From the experimental results of the classification performed by the 9 models, it can be observed that the accuracy obtained from training with data balanced using SMOTE yields the best results. Next, using the original data without addressing imbalance issues or baseline dataset and finally, using the weighted method to address imbalance issues, resulted in the lowest accuracy overall. However, the weighted method may enhance the capability to classify minor classes with very low quantities, such as in the case of the InceptionV3 and VGG16 architectures. However, the weighted method may reduce the ability to classify certain major classes, which originally had a much larger sample size compared to minor classes. This might be due to the significant reduction in the weighted values assigned to those classes. The weighted method might be more effective when the class imbalance is less severe. This is different from using the SMOTE method, which balances the dataset by sampling, ensuring each class has an equal quantity of samples, allowing the models to learn from each class equally. Therefore, based on experiments with similar hyperparameter criteria and input data, it can be concluded that the ResNet50 architecture, using the SMOTE method to address imbalance issues, performs the best in classifying abnormal red blood cell images with an accuracy of 91.44%. The tables showing the comparison table of classification performance of each model and the classification abilities of each model in each class are presented in Tables 4.1 and 4.2, respectively.

Table 4.1 The classification performance of each model

Model	Best epoch	Accuracy	Macro average	Weighted average
Baseline ResNet50	56	83.54%	77.74%	83.24%
Weighted ResNet50	84	80.27%	74.70%	80.71%
SMOTE ResNet50	100	91.44%	91.39%	91.39%
Baseline InceptionV3	87	66.94%	55.15%	64.90%
Weighted InceptionV3	94	60.63%	54.94%	60.62%
SMOTE InceptionV3	99	77.59%	77.03%	77.03%
Baseline VGG16	89	80.41%	71.68%	79.80%
Weighted VGG16	94	76.33%	70.96%	76.86%
SMOTE VGG16	93	86.61%	86.55%	86.55%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรเผยแพร่หรือนำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4.2 The classification abilities of each model in each class

		Model								
		ResNet50			InceptionV3			VGG16		
		Baselin	Weight	SMOTE	Baselin	Weight	SMOTE	Baselin	Weight	SMOTE
Order of performance measured by f1-score	Normal Cell (0)	5	5	13	5	5	11	5	7	13
	Macrocyte (1)	7	6	4	6	7	5	6	6	2
	Microcyte (2)	11	12	6	8	10	4	10	12	6
	Spherocyte (3)	2	2	7	2	2	7	2	2	7
	Target Cell (4)	4	3	8	10	11	13	3	3	9
	Stomatocyte (5)	8	7	10	7	6	10	4	5	8
	Ovalocyte (6)	6	4	11	3	3	8	8	9	10
	Teardrop (7)	10	10	3	11	9	2	11	10	3
	Burr Cell (8)	3	8	5	9	8	9	7	4	5
	Schistocyte (9)	9	9	9	4	4	6	9	8	11
	Uncategorized (10)	13	13	2	N/A	13	3	12	13	4
	Hypochromia (11)	12	11	12	12	12	12	11	11	12
	Elliptocyte (12)	1	1	1	1	1	1	1	1	1

4.11 Learning Rate and Dropout Rate Fine-Tuning

To optimize the performance of the convolutional neural network model for red blood cell abnormality classification, a fine-tuning process was conducted. This involved evaluating four SMOTE ResNet50 models with varying learning rates and dropout rates. The values used for fine-tuning are shown in Table 4.3.

Table 4.3 List of experiments detail

Experiment set	Learning rate	Dropout rate
A1	0.01	0.10
A2		0.15
A3		0.20
A4		0.25
A5		0.50
A6		0.75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4.3 List of experiments detail (Cont.)

Experiment set	Learning rate	Dropout rate
B1	0.001	0.10
B2		0.15
B3		0.20
B4		0.25
B5		0.50
B6		0.75
C1	0.0001	0.10
C2		0.15
C3		0.20
C4		0.25
C5		0.50
C6		0.75

4.11.1 Efficiency of Experiment A1

Figure 4.28 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.10 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 19, with a validation loss of 0.1639, validation accuracy of 0.9509, training loss of 0.1235, and training accuracy of 0.9572. The classification results for the testing dataset are shown in confusion matrix, Figure 4.29, and classification report, Figure 4.30.

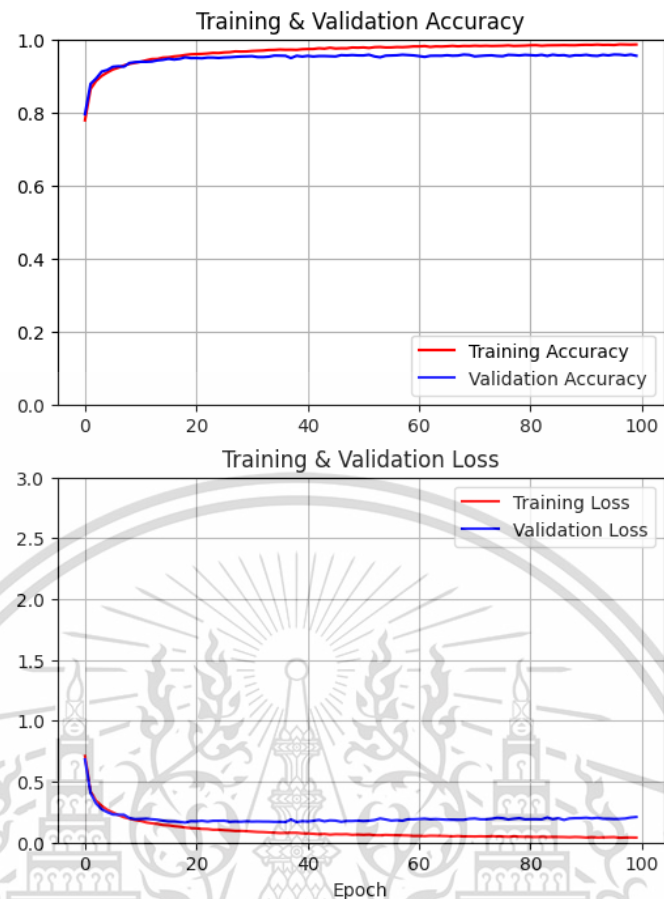


Figure 4.28 Accuracy and loss graph of model in experiment set A1

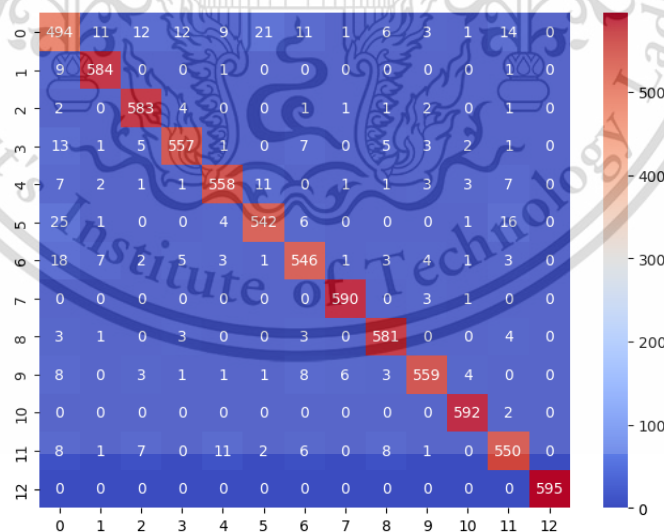


Figure 4.29 Confusion matrix result using model in experiment set A1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8416	0.8303	0.8359	595
Macrocyte	1	0.9605	0.9815	0.9709	595
Microcyte	2	0.9511	0.9798	0.9652	595
Spherocyte	3	0.9554	0.9361	0.9457	595
Target Cell	4	0.9490	0.9378	0.9434	595
Stomatocyte	5	0.9377	0.9109	0.9241	595
Ovalocyte	6	0.9286	0.9192	0.9239	594
Teardrop	7	0.9833	0.9933	0.9883	594
Burr Cell	8	0.9556	0.9765	0.9659	595
Schistocyte	9	0.9671	0.9411	0.9539	594
Uncategorized	10	0.9785	0.9966	0.9875	594
Hypochromia	11	0.9282	0.9259	0.9220	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9484	7730
macro avg		0.9482	0.9484	0.9482	7730
weighted avg		0.9482	0.9484	0.9482	7730

Figure 4.30 Classification report result using model in experiment set A1

4.11.2 Efficiency of Experiment A2

Figure 4.31 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.15 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 35, with a validation loss of 0.1629, validation accuracy of 0.9535, training loss of 0.1216, and training accuracy of 0.9575. The classification results for the testing dataset are shown in confusion matrix, Figure 4.32, and classification report, Figure 4.33.

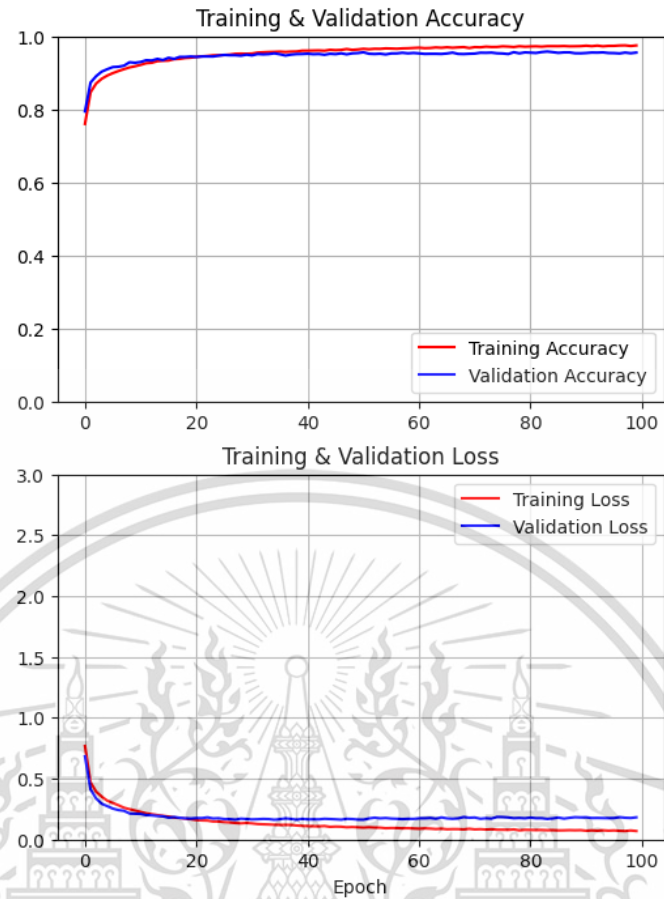


Figure 4.31 Accuracy and loss graph of model in experiment set A2

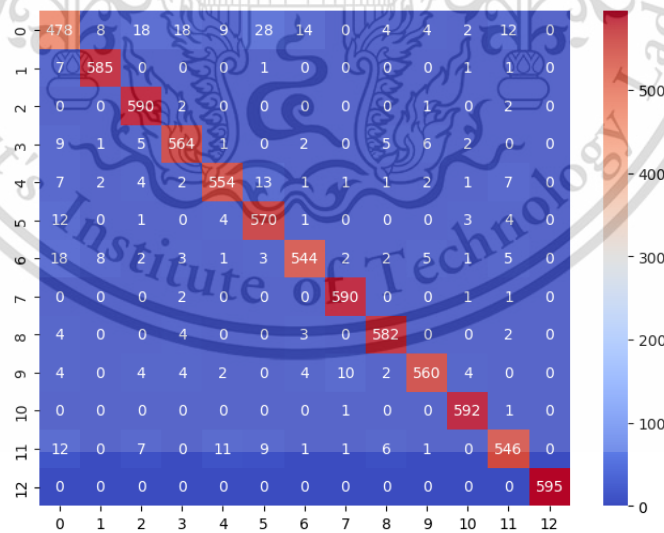


Figure 4.32 Confusion matrix result using model in experiment set A2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8675	0.8034	0.8342	595
Macrocyte	1	0.9685	0.9832	0.9758	595
Microcyte	2	0.9350	0.9916	0.9625	595
Spherocyte	3	0.9416	0.9479	0.9447	595
Target Cell	4	0.9519	0.9311	0.9414	595
Stomatocyte	5	0.9135	0.9580	0.9352	595
Ovalocyte	6	0.9544	0.9158	0.9347	594
Teardrop	7	0.9752	0.9933	0.9842	594
Burr Cell	8	0.9668	0.9782	0.9724	595
Schistocyte	9	0.9672	0.9428	0.9548	594
Uncategorized	10	0.9753	0.9966	0.9858	594
Hypochromia	11	0.9398	0.9192	0.9294	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9508	7730
macro avg		0.9505	0.9508	0.9504	7730
weighted avg		0.9505	0.9508	0.9504	7730

Figure 4.33 Classification report result using model in experiment set A2

4.11.3 Efficiency of Experiment A3

Figure 4.34 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.20 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 77, with a validation loss of 0.1526, validation accuracy of 0.9570, training loss of 0.1178, and training accuracy of 0.9586. The classification results for the testing dataset are shown in confusion matrix, Figure 4.35, and classification report, Figure 4.36.

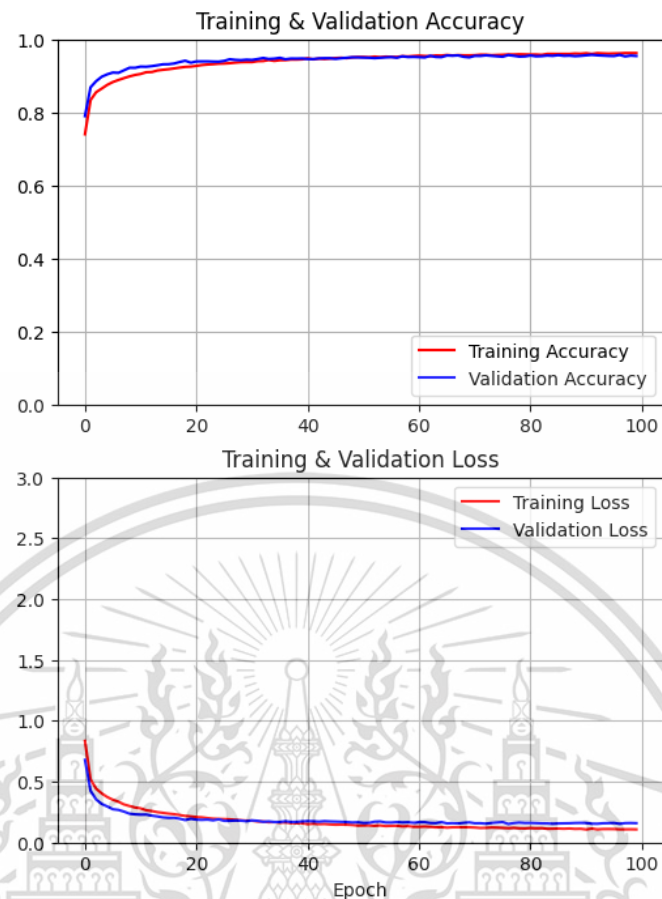


Figure 4.34 Accuracy and loss graph of model in experiment set A3

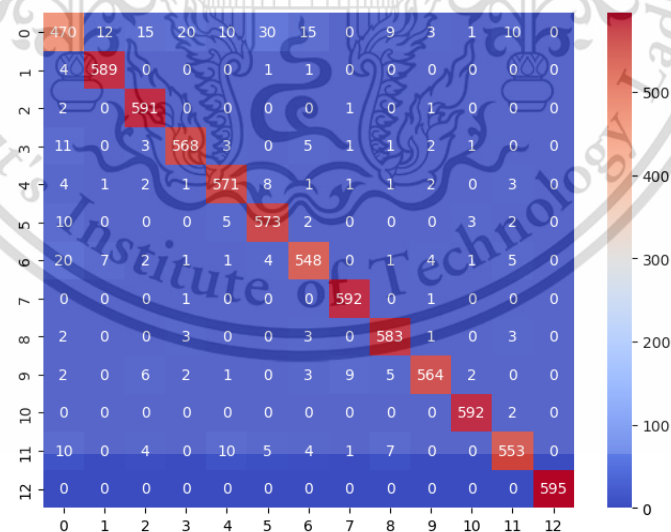


Figure 4.35 Confusion matrix result using model in experiment set A3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8785	0.7899	0.8319	595
Macrocyte	1	0.9672	0.9899	0.9784	595
Microcyte	2	0.9486	0.9933	0.9704	595
Spherocyte	3	0.9530	0.9546	0.9538	595
Target Cell	4	0.9501	0.9597	0.9548	595
Stomatocyte	5	0.9227	0.9630	0.9424	595
Ovalocyte	6	0.9416	0.9226	0.9320	594
Teardrop	7	0.9785	0.9966	0.9875	594
Burr Cell	8	0.9605	0.9798	0.9700	595
Schistocyte	9	0.9758	0.9495	0.9625	594
Uncategorized	10	0.9867	0.9966	0.9916	594
Hypochromia	11	0.9567	0.9310	0.9437	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9559	7730
macro avg		0.9554	0.9559	0.9553	7730
weighted avg		0.9554	0.9559	0.9553	7730

Figure 4.36 Classification report result using model in experiment set A3

4.11.4 Efficiency of Experiment A4

Figure 4.37 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.25 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 80, with a validation loss of 0.1569, validation accuracy of 0.9524, training loss of 0.161, and training accuracy of 0.9448. The classification results for the testing dataset are shown in confusion matrix, Figure 4.38, and classification report, Figure 4.39.

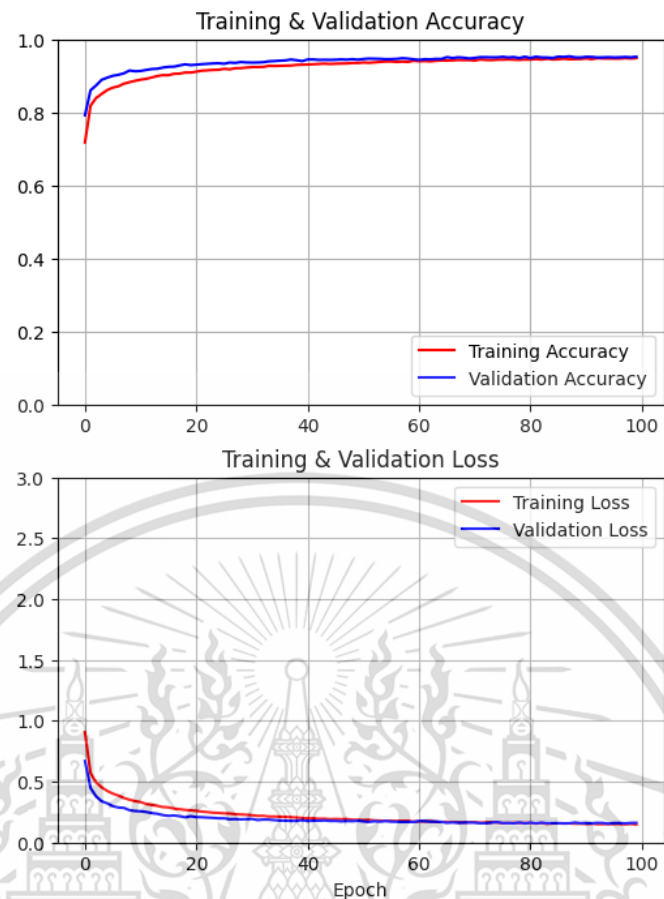


Figure 4.37 Accuracy and loss graph of model in experiment set A4

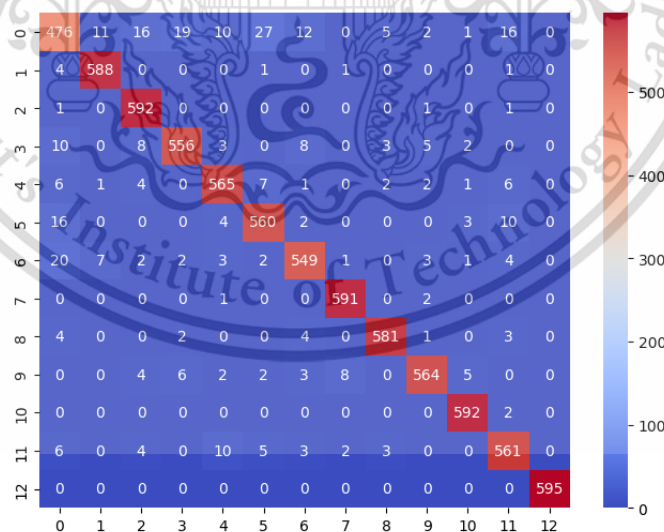


Figure 4.38 Confusion matrix result using model in experiment set A4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8766	0.8000	0.8366	595
Macrocyte	1	0.9687	0.9882	0.9784	595
Microcyte	2	0.9397	0.9950	0.9665	595
Spherocyte	3	0.9504	0.9345	0.9424	595
Target Cell	4	0.9448	0.9496	0.9472	595
Stomatocyte	5	0.9272	0.9412	0.9341	595
Ovalocyte	6	0.9433	0.9242	0.9337	594
Teardrop	7	0.9801	0.9949	0.9875	594
Burr Cell	8	0.9781	0.9765	0.9773	595
Schistocyte	9	0.9724	0.9495	0.9608	594
Uncategorized	10	0.9785	0.9966	0.9875	594
Hypochromia	11	0.9288	0.9444	0.9366	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9534	7730
macro avg		0.9530	0.9534	0.9530	7730
weighted avg		0.9530	0.9534	0.9530	7730

Figure 4.39 Classification report result using model in experiment set A4

4.11.5 Efficiency of Experiment A5

Figure 4.40 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.50 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 100, with a validation loss of 0.2594, validation accuracy of 0.919, training loss of 0.4236, and training accuracy of 0.8627. The classification results for the testing dataset are shown in confusion matrix, Figure 4.41, and classification report, Figure 4.42.

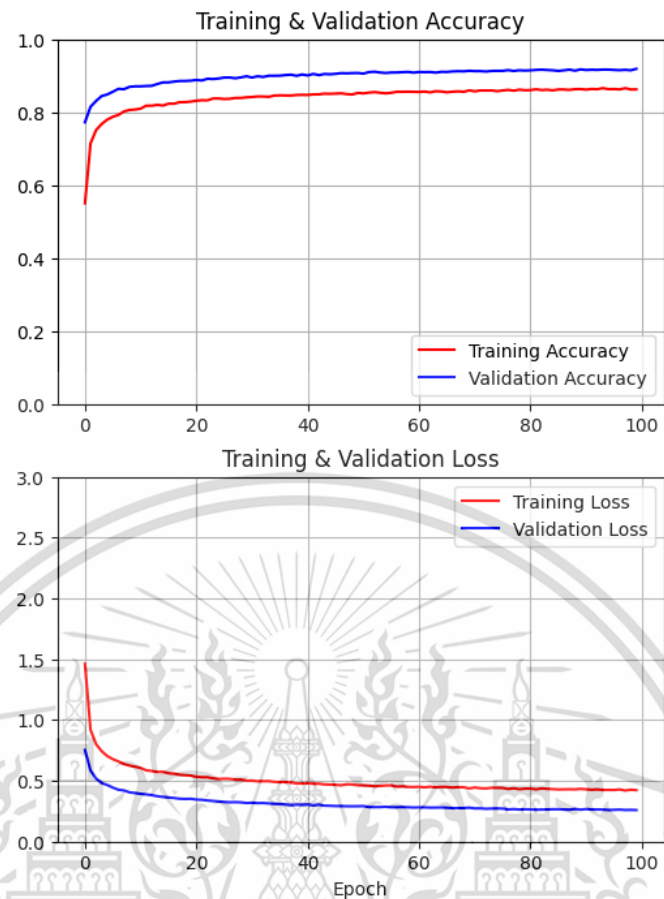


Figure 4.40 Accuracy and loss graph of model in experiment set A5

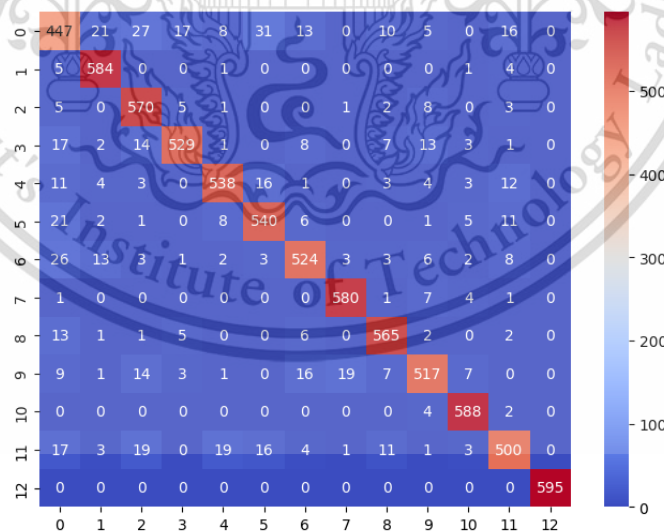


Figure 4.41 Confusion matrix result using model in experiment set A5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.7815	0.7513	0.7661	595
Macrocyte	1	0.9255	0.9815	0.9527	595
Microcyte	2	0.8742	0.9580	0.9142	595
Spherocyte	3	0.9446	0.8891	0.9160	595
Target Cell	4	0.9292	0.9042	0.9165	595
Stomatocyte	5	0.8911	0.9076	0.8993	595
Ovalocyte	6	0.9066	0.8822	0.8942	594
Teardrop	7	0.9603	0.9764	0.9683	594
Burr Cell	8	0.9278	0.9496	0.9385	595
Schistocyte	9	0.9102	0.8704	0.8898	594
Uncategorized	10	0.9545	0.9899	0.9719	594
Hypochromia	11	0.8929	0.8418	0.8666	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9155	7730
macro avg		0.9153	0.9155	0.9149	7730
weighted avg		0.9153	0.9155	0.9149	7730

Figure 4.42 Classification report result using model in experiment set A5

4.11.6 Efficiency of Experiment A6

Figure 4.43 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.75 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 98, with a validation loss of 0.5215, validation accuracy of 0.8414, training loss of 0.9066, and training accuracy of 0.7338. The classification results for the testing dataset are shown in confusion matrix, Figure 4.44, and classification report, Figure 4.45.

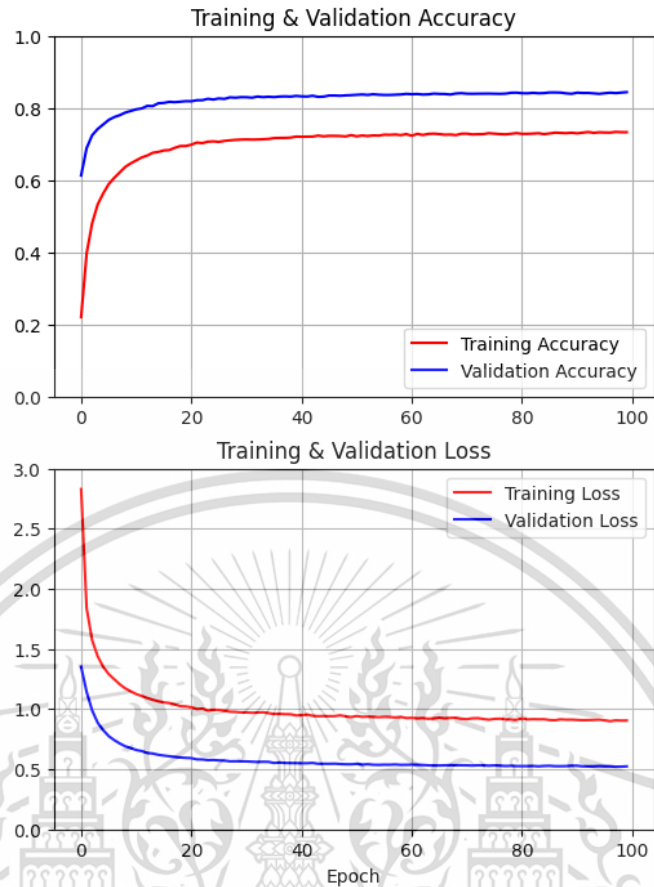


Figure 4.43 Accuracy and loss graph of model in experiment set A6

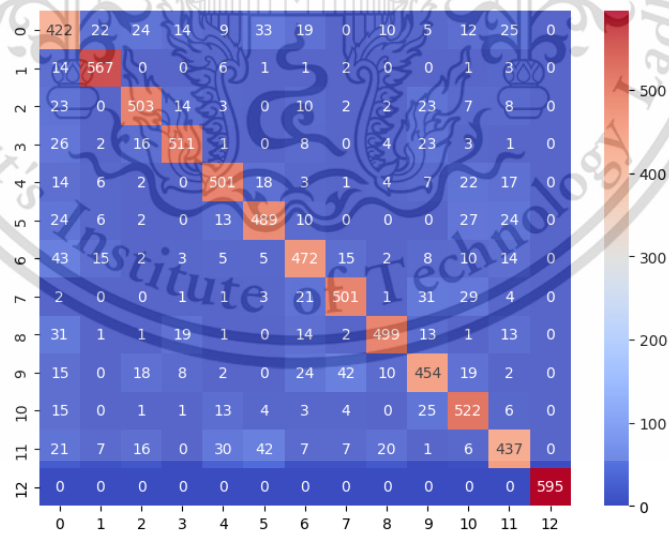


Figure 4.44 Confusion matrix result using model in experiment set A6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.6492	0.7092	0.6779	595
Macrocyte	1	0.9058	0.9529	0.9287	595
Microcyte	2	0.8598	0.8454	0.8525	595
Spherocyte	3	0.8949	0.8588	0.8765	595
Target Cell	4	0.8564	0.8420	0.8492	595
Stomatocyte	5	0.8218	0.8218	0.8218	595
Ovalocyte	6	0.7973	0.7946	0.7960	594
Teardrop	7	0.8698	0.8434	0.8564	594
Burr Cell	8	0.9040	0.8387	0.8701	595
Schistocyte	9	0.7695	0.7643	0.7669	594
Uncategorized	10	0.7921	0.8788	0.8332	594
Hypochromia	11	0.7888	0.7357	0.7613	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.8374	7730
macro avg		0.8392	0.8374	0.8377	7730
weighted avg		0.8392	0.8374	0.8378	7730

Figure 4.45 Classification report result using model in experiment set A6

4.11.7 Efficiency of Experiment B1

Figure 4.46 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.001 learning rate and a 0.10 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 19, with a validation loss of 0.1639, validation accuracy of 0.9509, training loss of 0.1235, and training accuracy of 0.9572. The classification results for the testing dataset are shown in confusion matrix, Figure 4.47, and classification report, Figure 4.48.

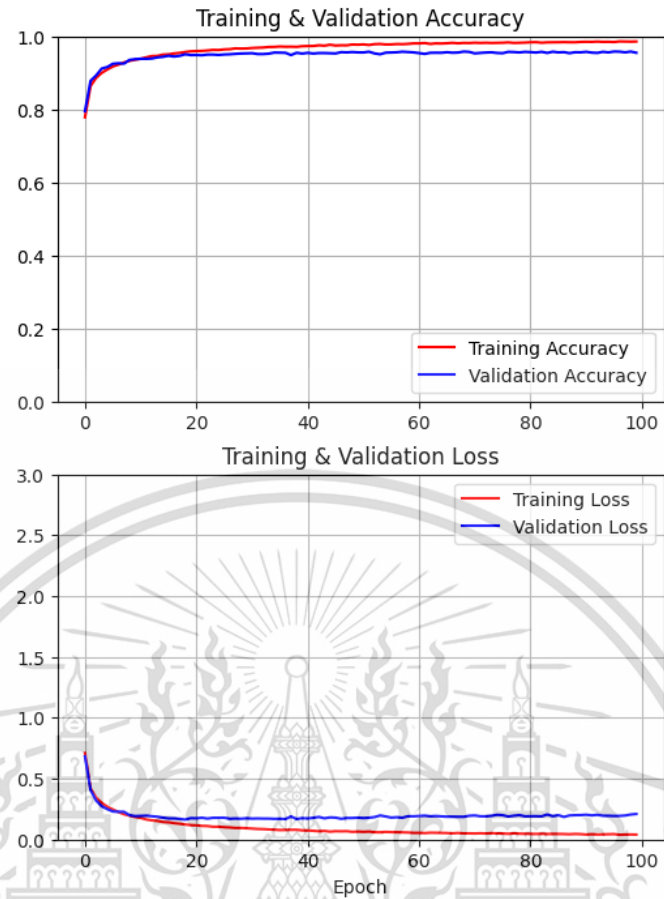


Figure 4.46 Accuracy and loss graph of model in experiment set B1

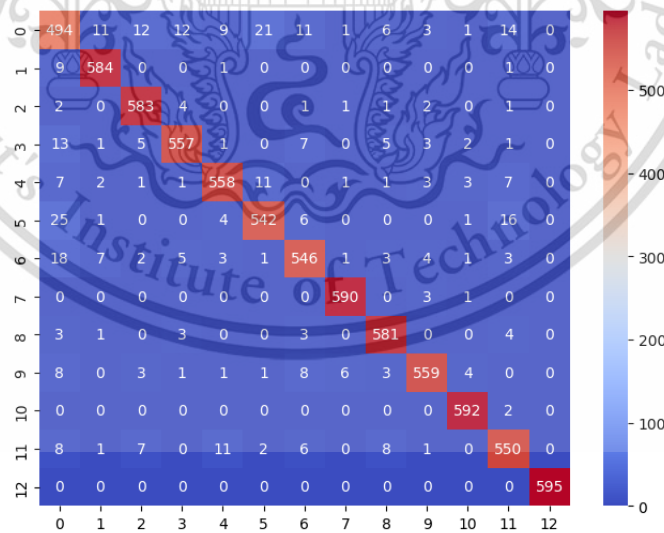


Figure 4.47 Confusion matrix result using model in experiment set B1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8416	0.8303	0.8359	595
Macrocyte	1	0.9605	0.9815	0.9709	595
Microcyte	2	0.9511	0.9798	0.9652	595
Spherocyte	3	0.9554	0.9361	0.9457	595
Target Cell	4	0.9490	0.9378	0.9434	595
Stomatocyte	5	0.9377	0.9109	0.9241	595
Ovalocyte	6	0.9286	0.9192	0.9239	594
Teardrop	7	0.9833	0.9933	0.9883	594
Burr Cell	8	0.9556	0.9765	0.9659	595
Schistocyte	9	0.9671	0.9411	0.9539	594
Uncategorized	10	0.9785	0.9966	0.9875	594
Hypochromia	11	0.9182	0.9259	0.9220	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9484	7730
macro avg		0.9482	0.9484	0.9482	7730
weighted avg		0.9482	0.9484	0.9482	7730

Figure 4.48 Classification report result using model in experiment set B1

4.11.8 Efficiency of Experiment B2

Figure 4.49 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.001 learning rate and a 0.15 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 35, with a validation loss of 0.1629, validation accuracy of 0.9535, training loss of 0.1216, and training accuracy of 0.9575. The classification results for the testing dataset are shown in confusion matrix, Figure 4.50, and classification report, Figure 4.51.

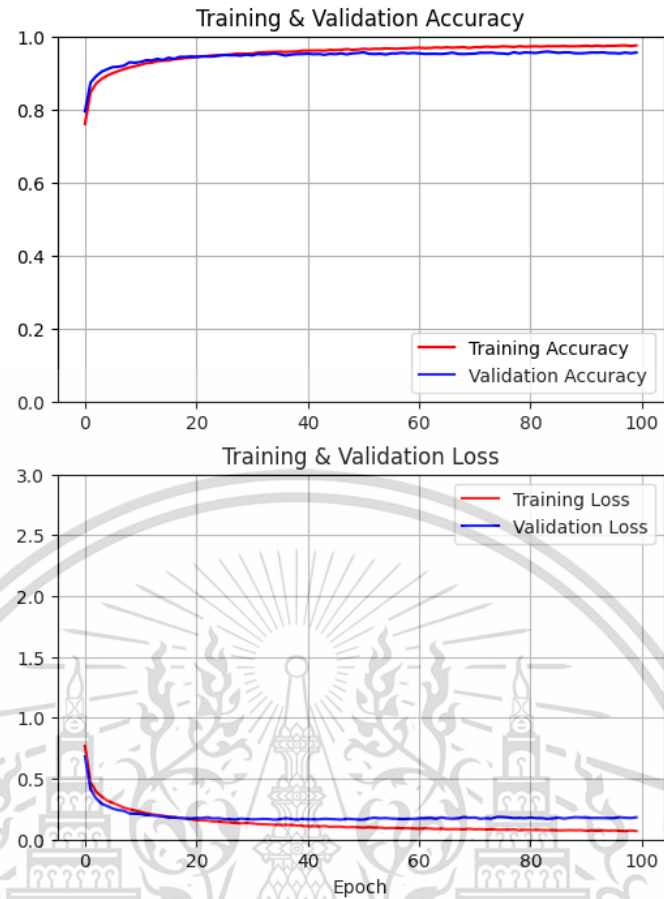


Figure 4.49 Accuracy and loss graph of model in experiment set B2

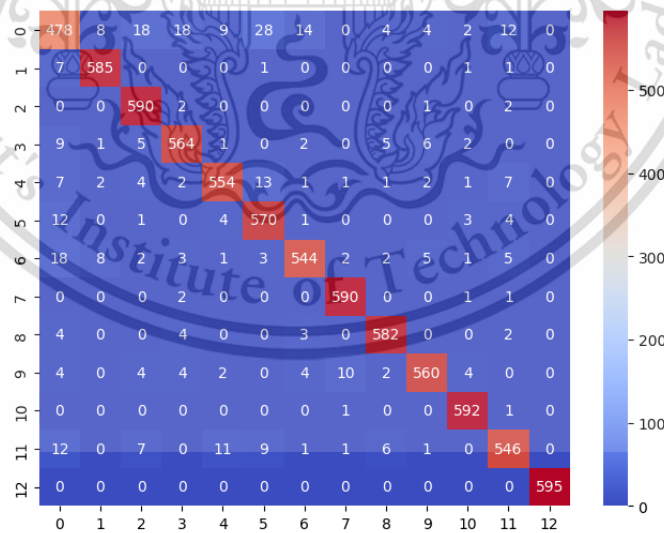


Figure 4.50 Confusion matrix result using model in experiment set B2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8675	0.8034	0.8342	595
Macrocyte	1	0.9685	0.9822	0.9758	595
Microcyte	2	0.9350	0.9916	0.9625	595
Spherocyte	3	0.9416	0.9479	0.9447	595
Target Cell	4	0.9519	0.9311	0.9414	595
Stomatocyte	5	0.9135	0.9580	0.9352	595
Ovalocyte	6	0.9544	0.9158	0.9347	594
Teardrop	7	0.9752	0.9933	0.9842	594
Burr Cell	8	0.9668	0.9782	0.9724	595
Schistocyte	9	0.9672	0.9428	0.9548	594
Uncategorized	10	0.9753	0.9966	0.9858	594
Hypochromia	11	0.9398	0.9192	0.9294	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9508	7730
macro avg		0.9505	0.9508	0.9504	7730
weighted avg		0.9505	0.9508	0.9504	7730

Figure 4.51 Classification report result using model in experiment set B2

4.11.9 Efficiency of Experiment B3

Figure 4.52 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.001 learning rate and a 0.20 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 77, with a validation loss of 0.1526, validation accuracy of 0.9570, training loss of 0.1178, and training accuracy of 0.9586. The classification results for the testing dataset are shown in confusion matrix, Figure 4.53, and classification report, Figure 4.54.

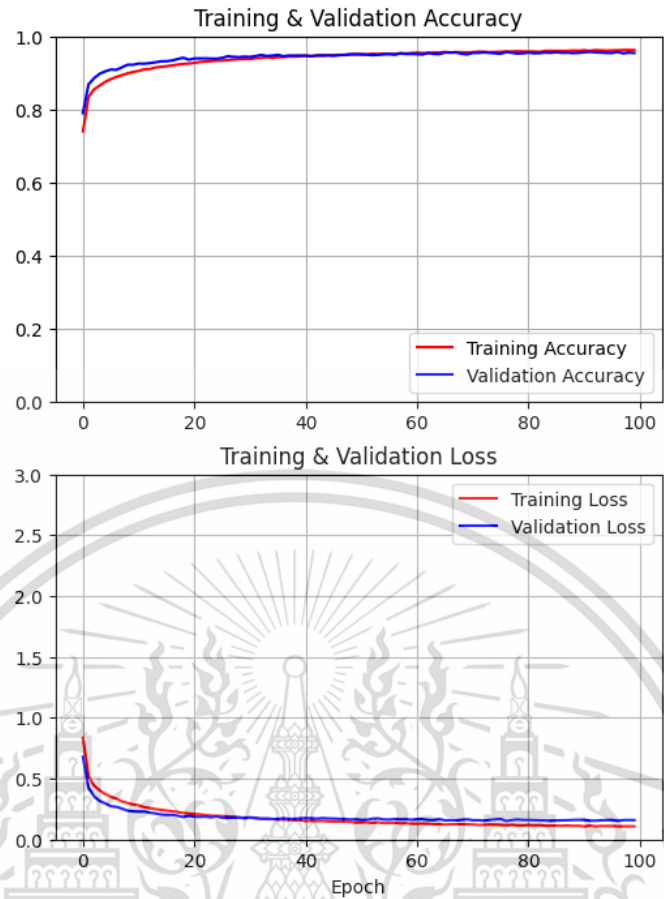


Figure 4.52 Accuracy and loss graph of model in experiment set B3

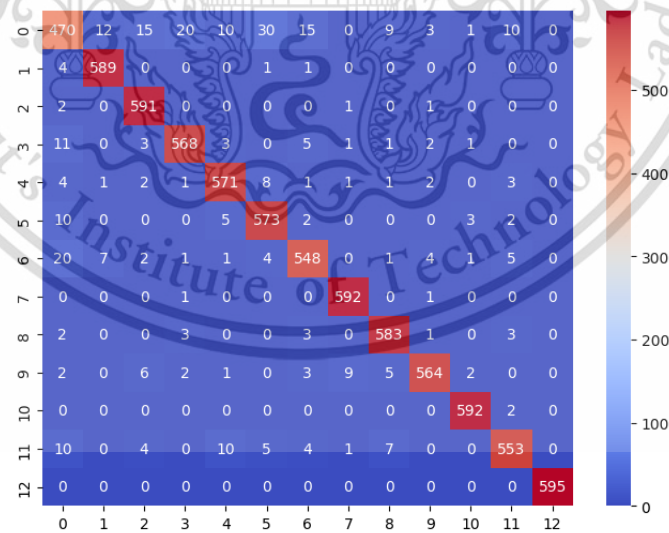


Figure 4.53 Confusion matrix result using model in experiment set B3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8785	0.7899	0.8319	595
Macrocyte	1	0.9672	0.9899	0.9784	595
Microcyte	2	0.9486	0.9933	0.9704	595
Spherocyte	3	0.9530	0.9546	0.9538	595
Target Cell	4	0.9501	0.9597	0.9548	595
Stomatocyte	5	0.9227	0.9630	0.9424	595
Ovalocyte	6	0.9416	0.9226	0.9320	594
Teardrop	7	0.9785	0.9966	0.9875	594
Burr Cell	8	0.9605	0.9798	0.9700	595
Schistocyte	9	0.9758	0.9495	0.9625	594
Uncategorized	10	0.9867	0.9966	0.9916	594
Hypochromia	11	0.9567	0.9310	0.9437	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9559	7730
macro avg		0.9554	0.9559	0.9553	7730
weighted avg		0.9554	0.9559	0.9553	7730

Figure 4.54 Classification report result using model in experiment set B3

4.11.10 Efficiency of Experiment B4

Figure 4.55 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.001 learning rate and a 0.25 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 95, with a validation loss of 0.1549, validation accuracy of 0.9513, training loss of 0.1509, and training accuracy of 0.9475. The classification results for the testing dataset are shown in confusion matrix, Figure 4.56, and classification report, Figure 4.57.

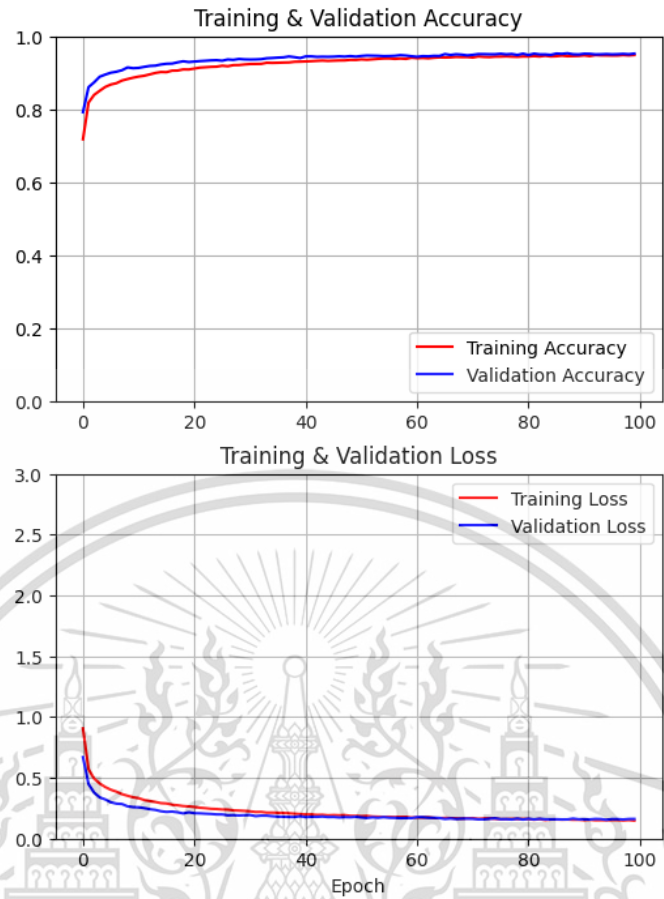


Figure 4.55 Accuracy and loss graph of model in experiment set B4

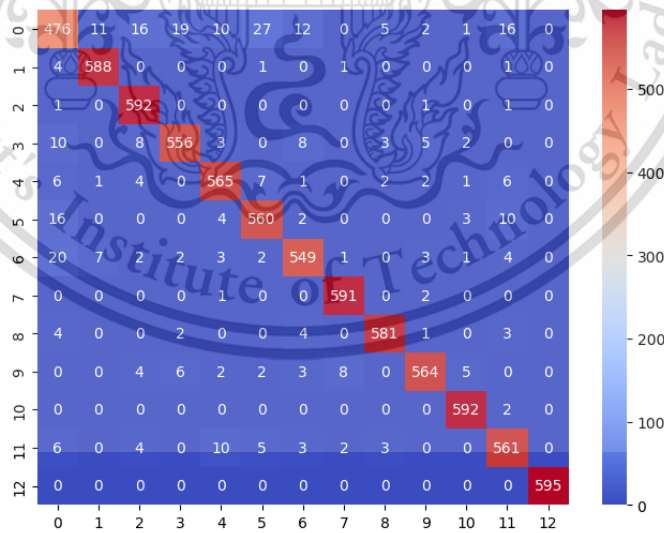


Figure 4.56 Confusion matrix result using model in experiment set B4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8766	0.8000	0.8366	595
Macrocyte	1	0.9687	0.9882	0.9784	595
Microcyte	2	0.9397	0.9950	0.9665	595
Spherocyte	3	0.9504	0.9345	0.9424	595
Target Cell	4	0.9448	0.9496	0.9472	595
Stomatocyte	5	0.9272	0.9412	0.9341	595
Ovalocyte	6	0.9433	0.9242	0.9337	594
Teardrop	7	0.9801	0.9949	0.9875	594
Burr Cell	8	0.9781	0.9765	0.9773	595
Schistocyte	9	0.9724	0.9495	0.9608	594
Uncategorized	10	0.9785	0.9966	0.9875	594
Hypochromia	11	0.9288	0.9444	0.9366	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9534	7730
macro avg		0.9530	0.9534	0.9530	7730
weighted avg		0.9530	0.9534	0.9530	7730

Figure 4.57 Classification report result using model in experiment set B4

4.11.11 Efficiency of Experiment B5

Figure 4.58 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.50 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 100, with a validation loss of 0.2571, validation accuracy of 0.9173, training loss of 0.4255, and training accuracy of 0.8624. The classification results for the testing dataset are shown in confusion matrix, Figure 4.59, and classification report, Figure 4.60. This result is the same as the topic 4.5 Results of ResNet50 architecture with SMOTE dataset.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Figure 4.58 Accuracy and loss graph of model in experiment set B5

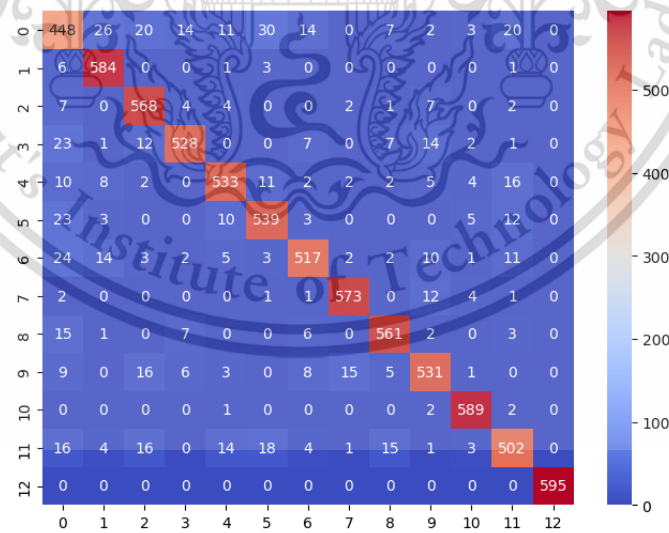


Figure 4.59 Confusion matrix result using model in experiment set B5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.7684	0.7529	0.7606	595
Macrocyte	1	0.9111	0.9815	0.9450	595
Microcyte	2	0.8917	0.9546	0.9221	595
Spherocyte	3	0.9412	0.8874	0.9135	595
Target Cell	4	0.9158	0.8958	0.9057	595
Stomatocyte	5	0.8909	0.9059	0.8983	595
Ovalocyte	6	0.9199	0.8704	0.8945	594
Teardrop	7	0.9630	0.9646	0.9638	594
Burr Cell	8	0.9350	0.9429	0.9389	595
Schistocyte	9	0.9061	0.8939	0.9000	594
Uncategorized	10	0.9624	0.9916	0.9768	594
Hypochromia	11	0.8792	0.8451	0.8618	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9144	7730
macro avg		0.9142	0.9144	0.9139	7730
weighted avg		0.9142	0.9144	0.9139	7730

Figure 4.60 Classification report result using model in experiment set B5

4.11.12 Efficiency of Experiment B6

Figure 4.61 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.01 learning rate and a 0.75 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 98, with a validation loss of 0.5215, validation accuracy of 0.8414, training loss of 0.9066, and training accuracy of 0.7338. The classification results for the testing dataset are shown in confusion matrix, Figure 4.62, and classification report, Figure 4.63.

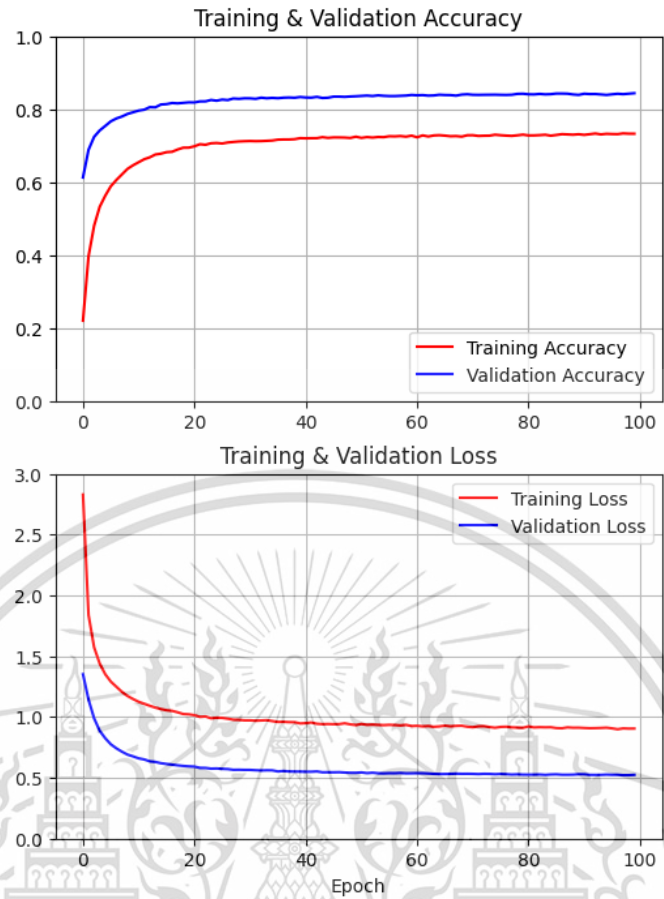


Figure 4.61 Accuracy and loss graph of model in experiment set B6

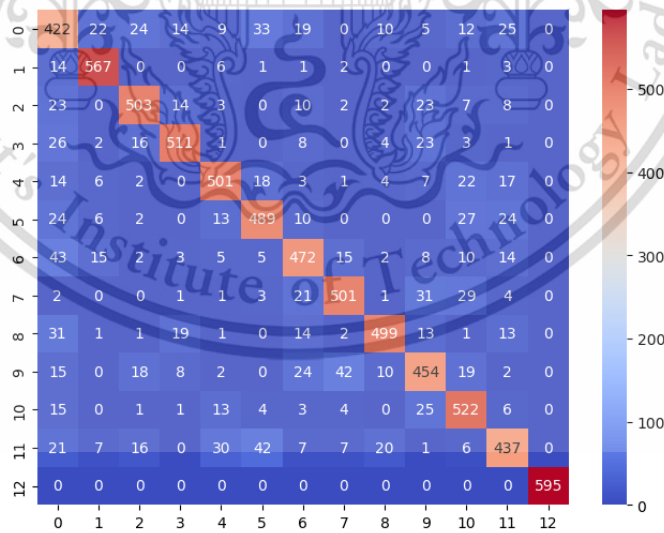


Figure 4.62 Confusion matrix result using model in experiment set B6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.6492	0.7092	0.6779	595
Macrocyte	1	0.9058	0.9529	0.9287	595
Microcyte	2	0.8598	0.8454	0.8525	595
Spherocyte	3	0.8949	0.8588	0.8765	595
Target Cell	4	0.8564	0.8420	0.8492	595
Stomatocyte	5	0.8218	0.8218	0.8218	595
Ovalocyte	6	0.7973	0.7946	0.7960	594
Teardrop	7	0.8698	0.8434	0.8564	594
Burr Cell	8	0.9040	0.8387	0.8701	595
Schistocyte	9	0.7695	0.7643	0.7669	594
Uncategorized	10	0.7921	0.8788	0.8332	594
Hypochromia	11	0.7888	0.7357	0.7613	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.8374	7730
macro avg		0.8392	0.8374	0.8377	7730
weighted avg		0.8392	0.8374	0.8378	7730

Figure 4.63 Classification report result using model in experiment set B6

4.11.13 Efficiency of Experiment C1

Figure 4.64 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.0001 learning rate and a 0.10 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 19, with a validation loss of 0.1639, validation accuracy of 0.9509, training loss of 0.1235, and training accuracy of 0.9572. The classification results for the testing dataset are shown in confusion matrix, Figure 4.65, and classification report, Figure 4.66.

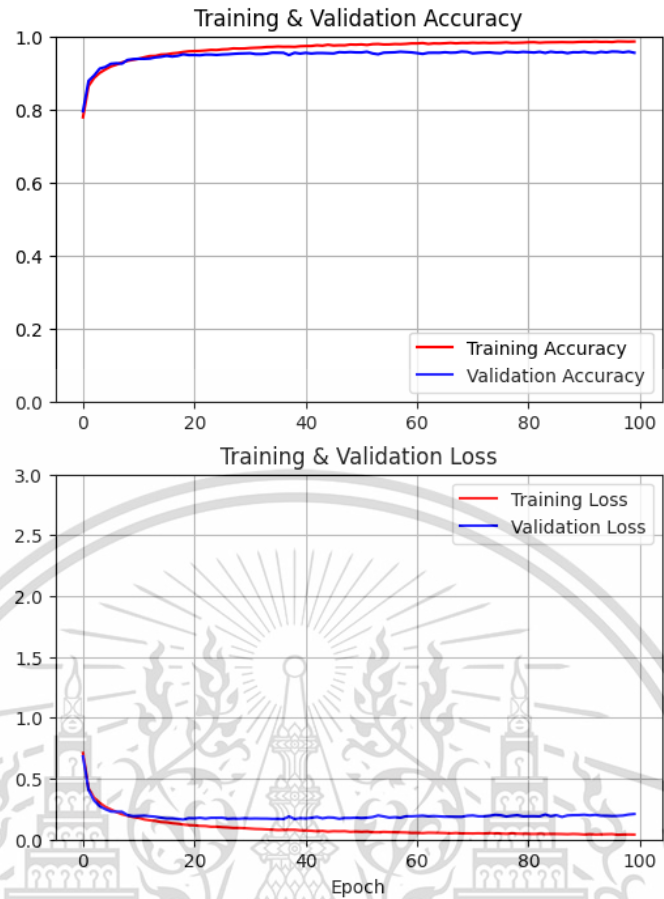


Figure 4.64 Accuracy and loss graph of model in experiment set C1

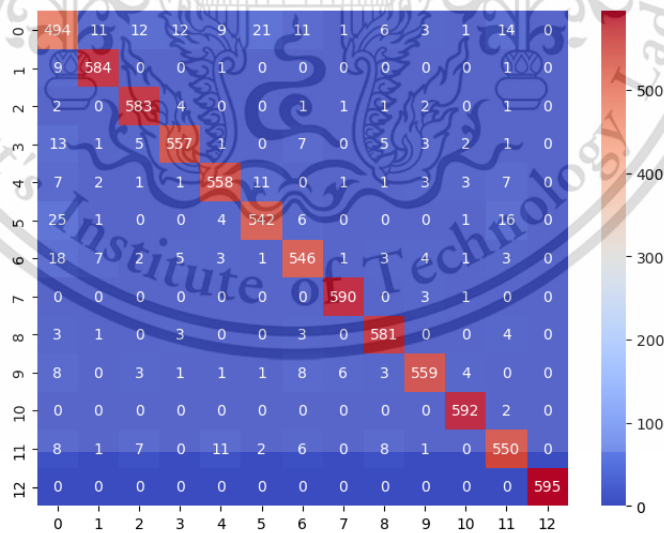


Figure 4.65 Confusion matrix result using model in experiment set C1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8416	0.8303	0.8359	595
Macrocyte	1	0.9605	0.9815	0.9709	595
Microcyte	2	0.9511	0.9798	0.9652	595
Spherocyte	3	0.9554	0.9361	0.9457	595
Target Cell	4	0.9490	0.9378	0.9434	595
Stomatocyte	5	0.9377	0.9109	0.9241	595
Ovalocyte	6	0.9286	0.9192	0.9239	594
Teardrop	7	0.9833	0.9933	0.9883	594
Burr Cell	8	0.9556	0.9765	0.9659	595
Schistocyte	9	0.9671	0.9411	0.9539	594
Uncategorized	10	0.9785	0.9966	0.9875	594
Hypochromia	11	0.9182	0.9259	0.9220	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9484	7730
macro avg		0.9482	0.9484	0.9482	7730
weighted avg		0.9482	0.9484	0.9482	7730

Figure 4.66 Classification report result using model in experiment set C1

4.11.14 Efficiency of Experiment C2

Figure 4.67 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.0001 learning rate and a 0.15 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 35, with a validation loss of 0.1629, validation accuracy of 0.9535, training loss of 0.1216, and training accuracy of 0.9575. The classification results for the testing dataset are shown in confusion matrix, Figure 4.68, and classification report, Figure 4.69.

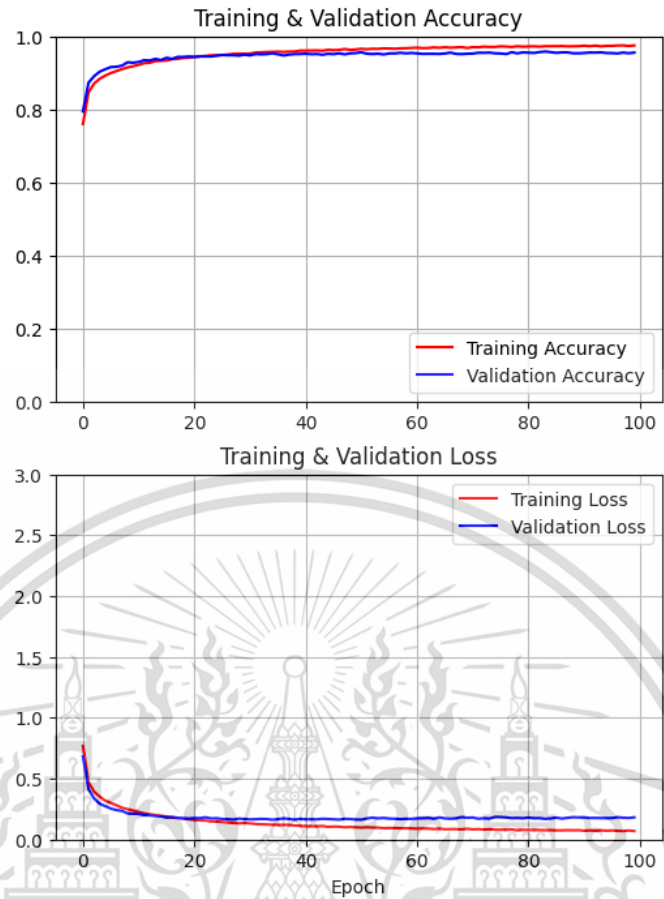


Figure 4.67 Accuracy and loss graph of model in experiment set C2

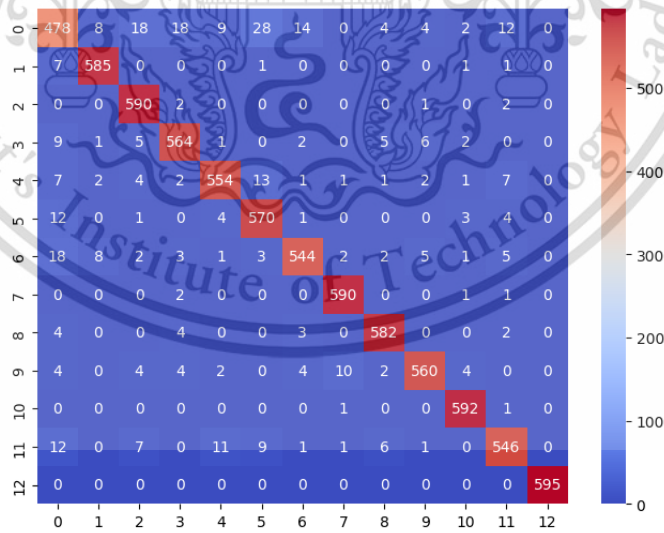


Figure 4.68 Confusion matrix result using model in experiment set C2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8675	0.8034	0.8342	595
Macrocyte	1	0.9685	0.9832	0.9758	595
Microcyte	2	0.9350	0.9916	0.9625	595
Spherocyte	3	0.9416	0.9479	0.9447	595
Target Cell	4	0.9519	0.9311	0.9414	595
Stomatocyte	5	0.9135	0.9580	0.9352	595
Ovalocyte	6	0.9544	0.9158	0.9347	594
Teardrop	7	0.9752	0.9933	0.9842	594
Burr Cell	8	0.9668	0.9782	0.9724	595
Schistocyte	9	0.9672	0.9428	0.9548	594
Uncategorized	10	0.9753	0.9966	0.9858	594
Hypochromia	11	0.9398	0.9192	0.9294	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9508	7730
macro avg		0.9505	0.9508	0.9504	7730
weighted avg		0.9505	0.9508	0.9504	7730

Figure 4.69 Classification report result using model in experiment set C2

4.11.14 Efficiency of Experiment C3

Figure 4.70 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.0001 learning rate and a 0.20 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 35, with a validation loss of 0.1629, validation accuracy of 0.9535, training loss of 0.1216, and training accuracy of 0.9575. The classification results for the testing dataset are shown in confusion matrix, Figure 4.71, and classification report, Figure 4.72.

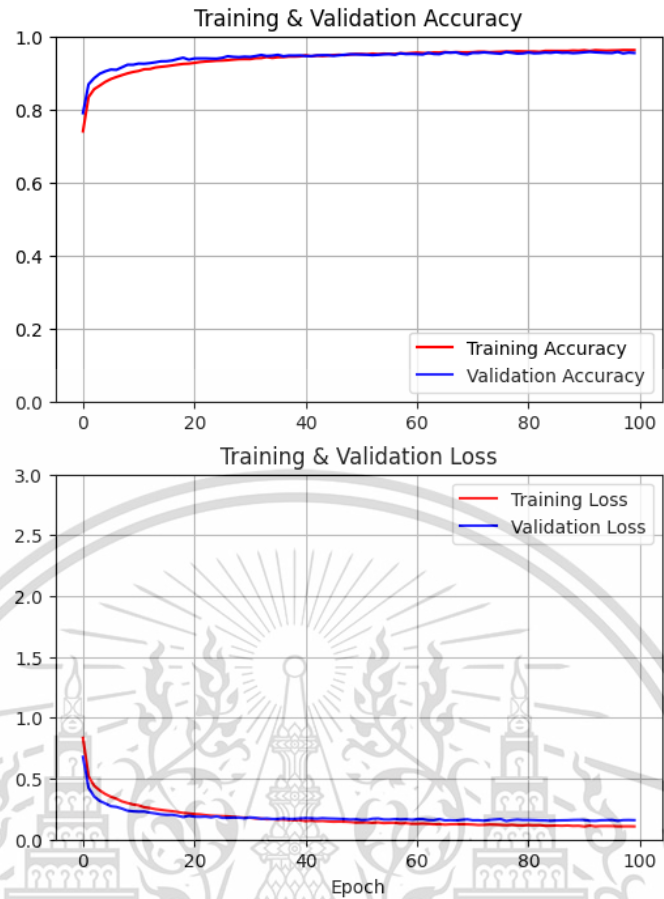


Figure 4.70 Accuracy and loss graph of model in experiment set C3

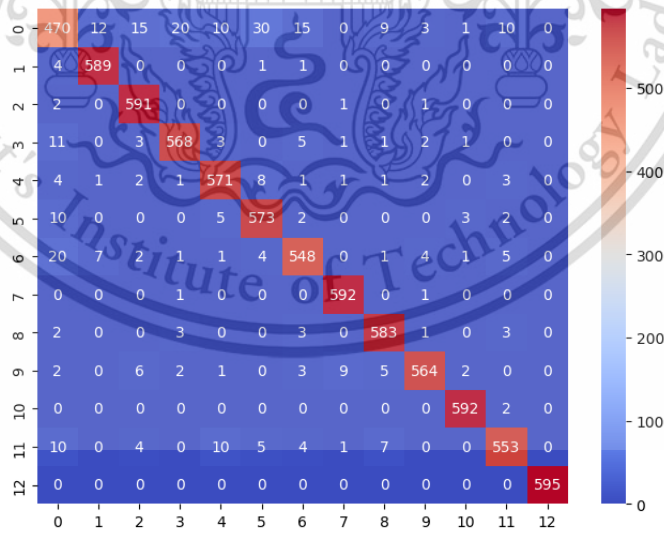


Figure 4.71 Confusion matrix result using model in experiment set C3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8785	0.7899	0.8319	595
Macrocyte	1	0.9672	0.9899	0.9784	595
Microcyte	2	0.9486	0.9933	0.9704	595
Spherocyte	3	0.9530	0.9546	0.9538	595
Target Cell	4	0.9501	0.9597	0.9548	595
Stomatocyte	5	0.9227	0.9630	0.9424	595
Ovalocyte	6	0.9416	0.9226	0.9320	594
Teardrop	7	0.9785	0.9966	0.9875	594
Burr Cell	8	0.9605	0.9798	0.9700	595
Schistocyte	9	0.9758	0.9495	0.9625	594
Uncategorized	10	0.9867	0.9966	0.9916	594
Hypochromia	11	0.9567	0.9310	0.9437	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9559	7730
macro avg		0.9554	0.9559	0.9553	7730
weighted avg		0.9554	0.9559	0.9553	7730

Figure 4.72 Classification report result using model in experiment set C3

4.11.16 Efficiency of Experiment C4

Figure 4.73 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.0001 learning rate and a 0.25 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 80, with a validation loss of 0.1569, validation accuracy of 0.9524, training loss of 0.1610, and training accuracy of 0.9488. The classification results for the testing dataset are shown in confusion matrix, Figure 4.74, and classification report, Figure 4.75.

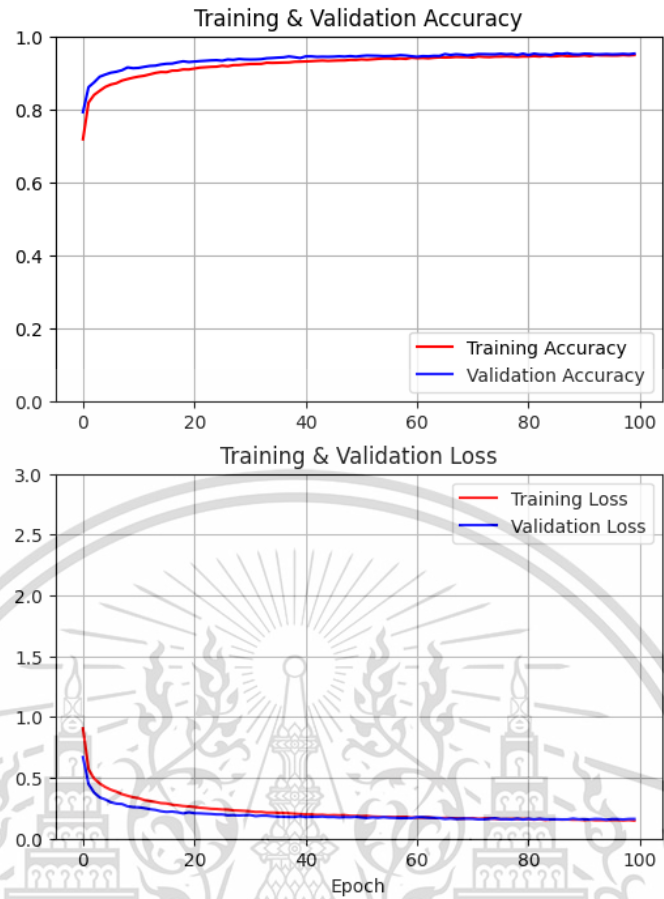


Figure 4.73 Accuracy and loss graph of model in experiment set C4

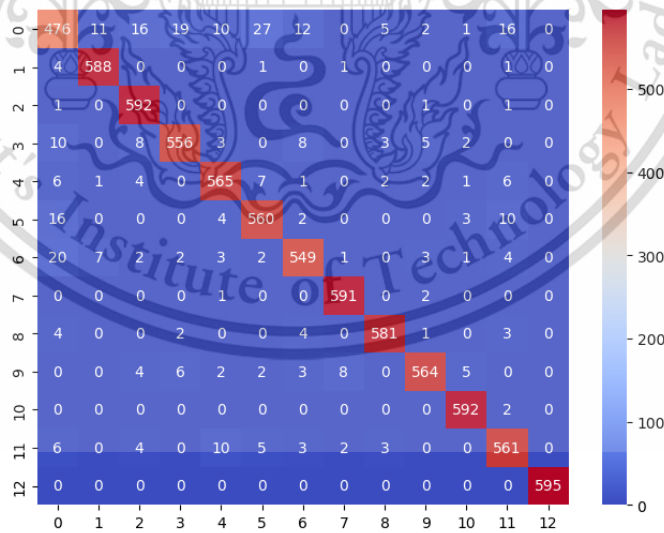


Figure 4.74 Confusion matrix result using model in experiment set C4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.8766	0.8000	0.8366	595
Macrocyte	1	0.9687	0.9882	0.9784	595
Microcyte	2	0.9397	0.9950	0.9665	595
Spherocyte	3	0.9504	0.9345	0.9424	595
Target Cell	4	0.9448	0.9496	0.9472	595
Stomatocyte	5	0.9272	0.9412	0.9341	595
Ovalocyte	6	0.9433	0.9242	0.9337	594
Teardrop	7	0.9801	0.9949	0.9875	594
Burr Cell	8	0.9781	0.9765	0.9773	595
Schistocyte	9	0.9724	0.9495	0.9608	594
Uncategorized	10	0.9785	0.9966	0.9875	594
Hypochromia	11	0.9288	0.9444	0.9366	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9534	7730
macro avg		0.9530	0.9534	0.9530	7730
weighted avg		0.9530	0.9534	0.9530	7730

Figure 4.75 Classification report result using model in experiment set C4

4.11.17 Efficiency of Experiment C5

Figure 4.76 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.0001 learning rate and a 0.50 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 100, with a validation loss of 0.2594, validation accuracy of 0.919, training loss of 0.4236, and training accuracy of 0.8627. The classification results for the testing dataset are shown in confusion matrix, Figure 4.77, and classification report, Figure 4.78.

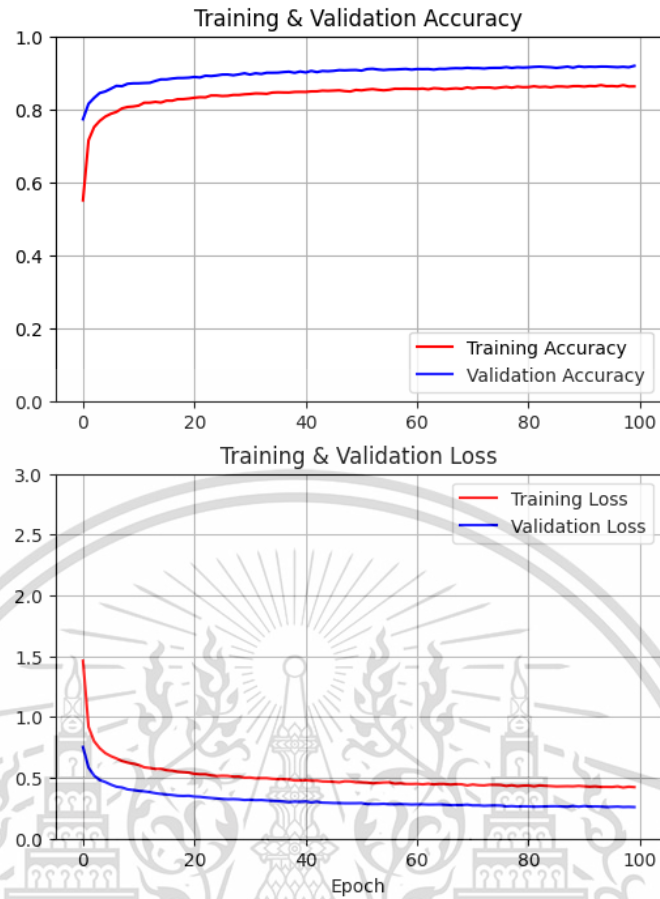


Figure 4.76 Accuracy and loss graph of model in experiment set C5

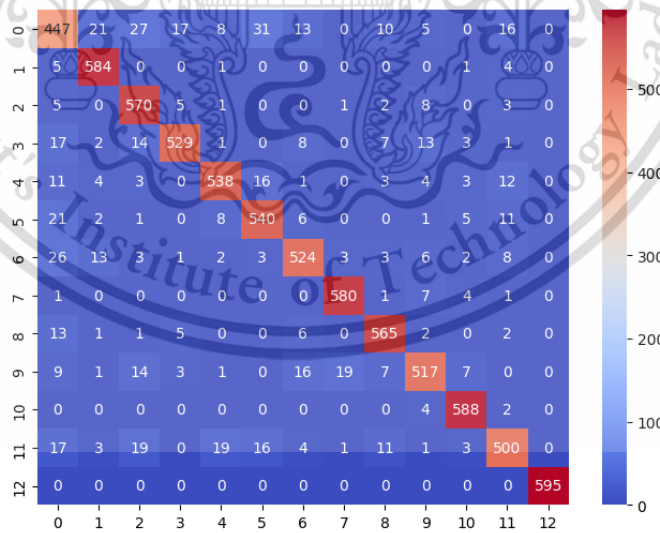


Figure 4.77 Confusion matrix result using model in experiment set C5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.7815	0.7513	0.7661	595
Macrocyte	1	0.9255	0.9815	0.9527	595
Microcyte	2	0.8742	0.9580	0.9142	595
Spherocyte	3	0.9446	0.8891	0.9160	595
Target Cell	4	0.9292	0.9042	0.9165	595
Stomatocyte	5	0.8911	0.9076	0.8993	595
Ovalocyte	6	0.9066	0.8822	0.8942	594
Teardrop	7	0.9603	0.9764	0.9683	594
Burr Cell	8	0.9278	0.9496	0.9385	595
Schistocyte	9	0.9102	0.8704	0.8898	594
Uncategorized	10	0.9545	0.9899	0.9719	594
Hypochromia	11	0.8929	0.8418	0.8666	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.9155	7730
macro avg		0.9153	0.9155	0.9149	7730
weighted avg		0.9153	0.9155	0.9149	7730

Figure 4.78 Classification report result using model in experiment set C5

4.11.18 Efficiency of Experiment C6

Figure 4.79 shows the accuracy and loss graphs for the training and validation datasets of the ResNet50 architecture with the SMOTE dataset. A 0.0001 learning rate and a 0.75 dropout rate were used in this model. The results indicated that the model achieved its best performance at epoch 98, with a validation loss of 0.5215, validation accuracy of 0.8414, training loss of 0.9066, and training accuracy of 0.7338. The classification results for the testing dataset are shown in confusion matrix, Figure 4.70, and classification report, Figure 4.81.

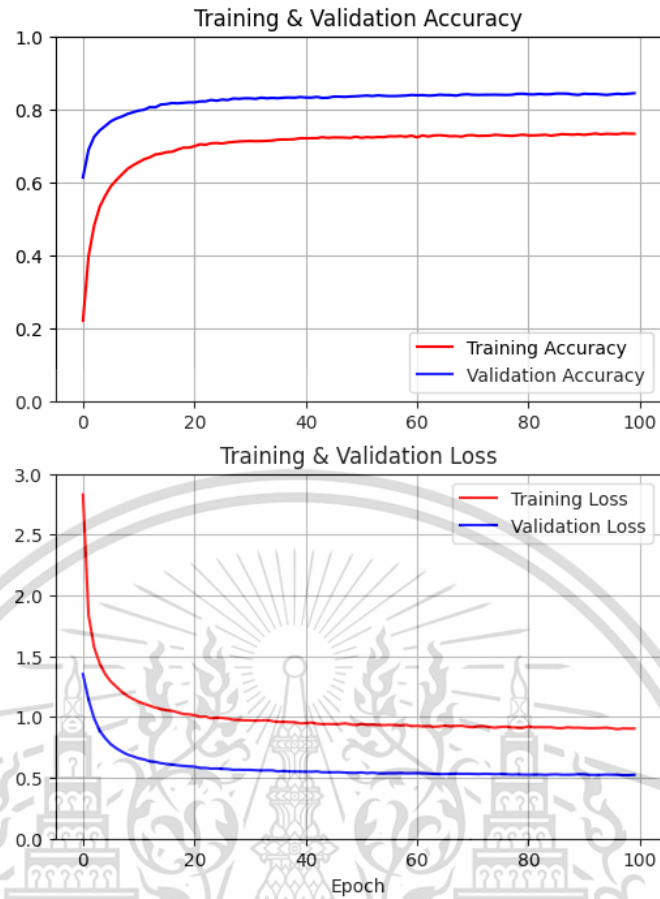


Figure 4.79 Accuracy and loss graph of model in experiment set C6

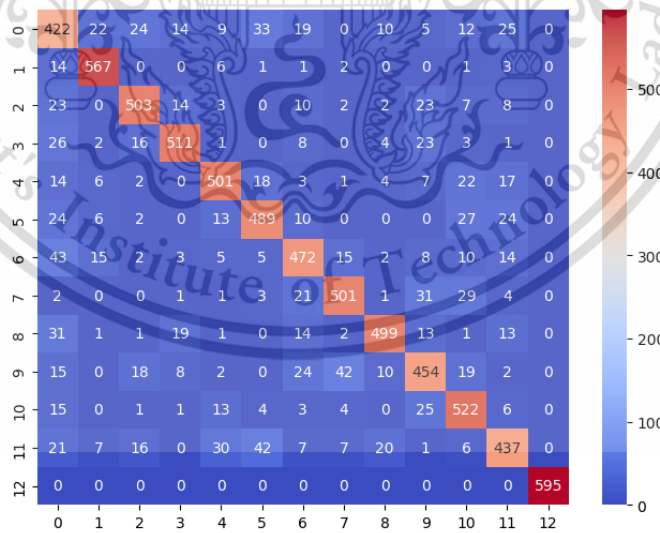


Figure 4.80 Confusion matrix result using model in experiment set C6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		precision	recall	f1-score	support
Normal Cell	0	0.6492	0.7092	0.6779	595
Macrocyte	1	0.9058	0.9529	0.9287	595
Microcyte	2	0.8598	0.8454	0.8525	595
Spherocyte	3	0.8949	0.8588	0.8765	595
Target Cell	4	0.8564	0.8420	0.8492	595
Stomatocyte	5	0.8218	0.8218	0.8218	595
Ovalocyte	6	0.7973	0.7946	0.7960	594
Teardrop	7	0.8698	0.8434	0.8564	594
Burr Cell	8	0.9040	0.8387	0.8701	595
Schistocyte	9	0.7695	0.7643	0.7669	594
Uncategorized	10	0.7921	0.8788	0.8332	594
Hypochromia	11	0.7888	0.7357	0.7613	594
Elliptocyte	12	1.0000	1.0000	1.0000	595
accuracy				0.8374	7730
macro avg		0.8392	0.8374	0.8377	7730
weighted avg		0.8392	0.8374	0.8378	7730

Figure 4.81 Classification report result using model in experiment set C6

4.12 Fine-Tuning Conclusion

From the fine-tuning experiment result, using varying learning rates of 0.01, 0.001, and 0.0001 shows a near-divergence direction into a global minimum of gradient descent at approximately the same epoch and have minimal impact on accuracy in this experiment.

In contrast, the difference in dropout rates significantly affects the achieved accuracy. With a range of dropout rates of 0.10, 0.15, 0.20, 0.25, 0.50, and 0.75, based on the environmental settings in this experiment, a 0.20 dropout rate achieves the best accuracy among all values at 95.59%, which is shown in Table 4.4.

The experiments show that the value of accuracy starts to increase with an increasing dropout rate, then slowly falls off at a 0.25 dropout rate and further. It means this model trained with the blood cell dataset, a small dropout rate may lead to lower accuracy because the model learns too much specific patterns during training process and at higher dropout rate drop too many parameters, weakening the effectiveness of learning during training process.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

However, it is important to note that the result of choosing a dropout rate in this experiment does not represent that this specific rate is appropriate for other datasets or training methods.

Table 4.4 Performance comparison of fine-tuned models in each experiment set

Experiment set	Learning rate	Dropout rate	Best Epoch	Accuracy
A1	0.01	0.10	19	94.84%
A2		0.15	35	95.08%
A3		0.20	77	95.59%
A4		0.25	80	95.34%
A5		0.50	100	91.55%
A6		0.75	98	83.74%
B1	0.001	0.10	19	94.84%
B2		0.15	35	95.08%
B3		0.20	77	95.59%
B4		0.25	95	95.34%
B5		0.50	100	91.44%
B6		0.75	98	83.74%
C1	0.0001	0.10	19	94.84%
C2		0.15	35	95.08%
C3		0.20	77	95.59%
C4		0.25	80	95.34%
C5		0.50	100	91.55%
C6		0.75	98	83.74%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

This research studies image segmentation and classification of red blood cells based on morphology. It presents an experimental study comparing the performance of various Convolutional Neural Networks (CNNs) in classification red blood cells according to their abnormality types. The models were trained using the same hyperparameters and dataset, but with different approaches to handling data imbalance.

5.1 Conclusion

This research develops a model for classifying red blood cell abnormalities. In the data preparation step, image preprocessing is crucial. This step utilizes watershed segmentation to separate red blood cells from blood smear images. This technique is considered a basic method for images with less complexity and readily understandable. It results in 20,950 images of individual red blood cells. For the Classification Model Development step, architectures such as ResNet50, InceptionV3, and VGG16 are employed to develop the models. All three models share the same hyperparameter settings and are trained on the same dataset. However, the data goes through three different approaches to address data imbalance: baseline dataset, weighted method, and SMOTE method.

The evaluation results reveal that models using SMOTE to handle data imbalance achieve the highest accuracy. Models trained on data with no imbalance handling or baseline dataset come in second, while those using the weighted method achieve the lowest accuracy across all three architectures. The model built with the ResNet50 architecture achieves the highest overall accuracy of 91.44%. With the dropout rate fine-tuned to a value of 0.20, the model is able to improve its accuracy to 95.59%.

5.2 Limitations

The following limitations were found in this study:

1. The dataset used in this study is secondary data obtained from only one source. This source contains highly imbalanced classes. The imbalance is significant, with the major class being around 41 times more than the minor class after segmentation.
2. In this research, the model is trained on Google Colab, a cloud computing platform, despite its limitations in GPU and RAM resources. Due to the high amount of data in the image dataset, hyperparameters were chosen to enable training within a single notebook using a specific random seed.

5.3 Recommendations

In this study, the experiment can be further developed and improved in the following ways:

1. Instead of using only the handling imbalance method, gathering the data from other sources for training a model may lead to better performance.
2. Applying a different image segmentation method could reduce the time consumption of the segmentation process and improve the quality of the input image, which will affect the learnability of the model during the training process.
3. The weight method can be calculated using many formulas. Using different formulas may lead to different results.
4. Using other hyperparameters setting or changing the input dimension may increase the classification performance.
5. There are many other CNN architectures that might be better suited for this type of dataset.

5.4 Future Work

An ensemble approach can be implemented to combine the results of different high-performing models. This approach has the potential to improve the overall classification accuracy across all classes. Different model architectures may excel at classifying different classes. By combining their predictions, the ensemble leverages these complementary strengths.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

References

- [1] Vorapan Sirivatanauksorn, Chatchawan Srisawat, Neelobol Neungton, Patarabutr Masaratana and Somsong Lekhakula. “ชีวเคมีของเลือด.” [Online]. Available : https://www.si.mahidol.ac.th/department/biochemistry/home/md/lecture/biochemistry_of_blood.pdf. 2019.
- [2] Supinun Speek-Saichua. “Color Aids HEMATOLOGY.” [Online]. Available : <https://drive.google.com/file/d/1FIqhFCSgWGRjNDZ5bBZEgaQWHDMTnnwS/view>. 2010.
- [3] Kitti Torcharus. “Diagnosis from the Blood Smear” **Royal Thai Army Medical Journal**. 2010, vol.63, no.4. Oct-Dec 2010. pp. 247-250.
- [4] University of Rochester Medical Center Rochester. “Hematology.” [Online]. Available : <https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=85&contentid=P00958>. 2023.
- [5] American Society of Hematology. “Blood Basics.” [Online]. Available : <https://www.hematology.org/education/patients/blood-basics>. 2023.
- [6] Conley, C.L., Schwartz, R.S. “blood biochemistry.” [Online]. Available : <https://www.britannica.com/science/blood-biochemistry>. 2023.
- [7] Harmening, D.M. “Clinical Hematology and Fundamentals of Hemostasis.” [Online]. Available : https://resources.fadavis.com/harmening/documents/1_2326_2416.pdf. 2009.
- [8] Parasitology Faculty of Medicine Chulalongkorn University. “Preparation of Blood Film.” [Online]. Available : <http://www.cai.md.chula.ac.th/lesson/lesson4808/html/01.html>. 2023.
- [9] Lynch, E.C. “Clinical Methods: The History, Physical, and Laboratory Examinations.” [Online]. Available : https://www.ncbi.nlm.nih.gov/books/NBK263/pdf/Bookshelf_NBK263.pdf 1990.
- [10] Laboratory Identification of Parasites of Public Health Concern. “Preparation of blood smears.” [Online]. Available : https://www.cdc.gov/dpdx/resources/pdf/benchaid/malaria/malaria_procedures_benchaid.pdf. 2023.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [11] The Medical Technology Council of Thailand. “เกณฑ์มาตรฐานการจัดระดับความผิดปกติของเม็ดเลือดแดง.” [Online]. Available : <https://mtcouncil.org/download/เกณฑ์มาตรฐานการจัดระดับ/>. 2015
- [12] Yang, X., Shen, X., Long, J. and Chen, H. “An Improved Median-based Otsu Image Thresholding Algorithm” **AASRI Procedia**. 2012, vol.3, pp. 468-473.
- [13] Otsu, N. “A Threshold Selection Method from Gray-Level Histograms” **IEEE Transactions on Systems, Man, and Cybernetics**. 1979, vol.9, no.1. Jan 1979. pp. 62 - 66.
- [14] Bhutada, S., Yashwanth, N., Dheeraj, P. and Shekar, K. “Opening and closing in morphological image processing” **World Journal of Advanced Research and Reviews**. 2012, vol.14, no.3. pp. 687-695.
- [15] Lou, S., Pagani, L., Zeng, W., Jiang, X. and Scott, P.J. “Watershed segmentation of topographical features on freeform surfaces and its application to additively manufactured surfaces” **Precision Engineering**. 2020, vol.63, pp. 177-186.
- [16] Craig, L. “convolutional neural network (CNN).” [Online]. Available : <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>. 2023.
- [17] Yamashita, R., Nishio, M., Richard, K.G.D., and Togashi, K. “Convolutional neural networks: an overview and application in radiology” **Insights into Imaging**. 2018, vol.9, pp. 611-629.
- [18] Albelwi, S. and Mahmood, A. “A Framework for Designing the Architectures of Deep Convolutional Neural Networks” **Entropy**. 2017, vol.19, no.6.
- [19] Mishra, M. “Convolutional Neural Networks, Explained.” [Online]. Available : <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>. 2020.
- [20] Shaikh, J. “Deep Learning in the Trenches: Understanding Inception Network from Scratch.” [Online] Available : <https://www.analyticsvidhya.com/blog/2018/10/understanding-inception-network-from-scratch/>. 2023.
- [21] Bangar, S. “VGG-Net Architecture Explained.” [Online] Available : <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>. 2022.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [22] Alkafrawi, I.M.I. and Dakhell, Z.A. “Blood Cells Classification Using Deep Learning Technique” **2022 International Conference on Engineering & MIS (ICEMIS)**. 2022, pp. 1-6.
- [23] Kannadaguli, P. “Microscopic Blood Smear RBC Classification using PCA and SVM based Machine Learning” **2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT)**. 2020, pp. 82-86.
- [24] Tyas, D.A., Hartati, S., Harjoko, A. and Ratnaningsih, T. “Morphological, Texture, and Color Feature Analysis for Erythrocyte Classification in Thalassemia Cases” **IEEE Access**. 2020, vol.8, pp. 69849-69860.
- [25] Ortet, M.D., Molina, A., Alférez, S., Rodellar, J. and Merino, A. “A Deep Learning Approach for Segmentation of Red Blood Cell Images and Malaria Detection” **Entropy**. 2020, vol.22, no.6.
- [26] Wongsathon Naksuwan, Picha Suwannahitatorn, Chakrit Watcharopas and Pakaket Wattuya. “Deep Learning for Detecting Malaria Parasites of Infected Red Blood Cells in Thin Blood Smear Images” **2022 5th Asia Conference on Machine Learning and Computing (ACMLC)**. 2022, pp. 77-82.
- [27] Korranat Naruenatthanaset, Thanarat H. Chalidabhongse, Duangdao Palasuwan, Nantheera Anantrasirichai and Attakorn Palasuwan. “Red Blood Cell Segmentation with Overlapping Cell Separation and Classification on Imbalanced Dataset” **arXiv**. 2020.
- [28] "Keras Applications" [Online] Available : <https://keras.io/api/applications>. 2024.

Author Biography

Name	Mr. Sukawin Nunan
Date of Birth	5 May 1993
Email Address	n.sukawin@gmail.com
Education	(2015) Bachelor of Engineering in Electrical Engineering GPA 3.07 King Mongkut's University of Technology North Bangkok



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้