



ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

RUPESH KUMAR DEY

A thesis submitted in fulfilment of the
requirements for the award of the degree of
MASTER OF SCIENCE IN ARTIFICIAL INTELLIGENCE

ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION (APU)
SCHOOL OF COMPUTING AND TECHNOLOGY

OCTOBER 2022

DECLARATION OF THESIS CONFIDENTIALITY

Author's full name: **RUPESH KUMAR DEY**

IC No./Passport No.: **950320-07-5255**

Thesis/Project title: **ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING
DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI
TRACT DISEASES**

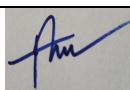
I declare that this thesis is classified as:

- CONFIDENTIAL
- RESTRICTED
- OPEN ACCESS

I acknowledged that Asia Pacific University of Technology & Innovation (APU) reserves the right as follows:

1. The thesis is the property of Asia Pacific University of Technology & Innovation (APU).
 2. The Library of Asia Pacific University of Technology & Innovation (APU) has the right to make copies for the purpose of research only.
 3. The Library has the right to make copies of the thesis for academic exchange.
-

Author's Signature:



Date: 1 October 2022

Supervisor's Name: **DR. VAZEERUDEEN HAMEED**

Date: 3 October 2022

Signature:

Vazeer

DECLARATION OF SUPERVISOR(S)

“We hereby declare that We have read this thesis and in our opinion this thesis is sufficient in terms of scope and quality for the award of the degree of
Master of Science in Artificial Intelligence

Name of Supervisor: **DR VAZEERUDEEN HAMEED**

Signature:


Date: 3 October 2022

Name of Supervisor (II) **HAMAM MOKAYED**

Signature:

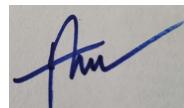

Date: 5 October 2022

DECLARATION OF ORIGINALITY AND EXCLUSIVENESS

I declare that this thesis entitled
**ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP
LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT
DISEASES**

is the result of my own research work except as cited in the references. This thesis has not been accepted for any degree and it is not concurrently submitted in candidature of any other degree.

Signature:



Name: Rupesh Kumar Dey

Date: 1 October 2022

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr Vazeerudeen Hameed, my final project supervisor for the continuous support he has provided me throughout my project. His encouragement, guidance and critical advice has been a major contributor to the development of the project and has helped me immensely in paving its direction and in solving the many issues faced throughout the project. I would also like to thank Dr Imran Medi and Shangar Sivananthan for their advice and constructive comments on my project's implementation and the project's final thesis. Their inputs have helped me in structuring my implementation and in presenting my findings in the final project's thesis. Last but not least, I would like to thank my friends and family for their understanding and support throughout the project.

ABSTRACT

GI Tract related diseases are one of the most prevalent in today's society. Studies have shown that early detection and treatment of these diseases are imperative in improving patients' recovery rate. An important part of this is regular check-ups and monitoring of the GI tract. Wireless Capsule Endoscopy (WCE) is an innovative imaging technology that enables invasive imaging of the GI Tract. It enables medical professionals to visualize the digestive tract and perform diagnosis accordingly. It also significantly reduces the risks and barrier for individuals to perform regular check-ups. Convolutional Neural Networks (CNN) and Image Processing have become very sought-after solutions in the process of developing a Computer Aided Diagnosis (CAD) system for many medical applications including in the domain of GI tract disease diagnosis. CNNs are robust and efficient in automatically extracting features from images at scale, analysing them and performing classification of different diseases. This overcomes the bottleneck faced within the medical industry whereby medical professionals are required to spend a lot of time to analyse large amounts of images captured from the WCE. This project proposes a CNN classification-based solution framework encompassing various network architectures, image processing enhancement techniques and data augmentation methods for a CAD solution for GI tract disease diagnosis by analysing WCE images. Three image processing enhancement techniques were introduced as a pre-processing step to enhance the quality of the features of the raw image prior to performing classification. Leveraging on the image processing enhancement algorithms, data augmentation was performed on the raw data to create a variety of images for the models to learn from. Various network architectures of self-developed architectures, transfer learning feature extraction, transfer learning fine tuning and an ensemble of models were developed. The Accuracy and F1-Score results of the developed CNN models were analysed in detail, putting emphasis on the generalization capability of the developed solutions. Results showed that image processing enhancement improved the CNN models' capability in performing accurate classification. In terms of individual network architectures, the transfer learning fine tuning models performed better as compared to the rest of the architectures. CNN networks trained on the dataset with augmentation are more generalized as compared to CNN networks trained on non-augmented data. The final proposed solution for GI tract CAD CNN network is the ensemble model which managed to achieve an overall accuracy of 97.03% when tested and compared to other proposed architectures across 4 phases of result analysis.

TABLE OF CONTENTS

DECLARATION OF THESIS CONFIDENTIALITY	ii
DECLARATION OF SUPERVISOR(S)	iii
DECLARATION OF ORIGINALITY AND EXCLUSIVENESS	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xiii
LIST OF EQUATIONS	xiv
LIST OF ABBREVIATIONS	xv
CHAPTER 1 INTRODUCTION	1
1.1 General Introduction to GI Tract + Diseases	1
1.2 Current monitoring methods and their limitations	3
1.3 Automating the entire image analysis process with AI	4
1.4 Challenges in developing a Deep Learning CAD solution.	5
1.5 Problem Statement:	6
1.6 Research Questions	7
1.7 Aim	7
1.8 Objectives	7
1.9 Scope of Research:	8
1.10 Significance of Study	8
CHAPTER 2 LITERATURE REVIEW	10
2.1 Introduction to GI Tract Diseases and WCE	10
2.1.1 Ulcerative Colitis	10
2.1.2 Polyps	10
2.1.3 Esophagitis	11
2.1.4 WCE imaging	11
2.2 Relevant Works on CAD Solutions Development	11
2.2.1 Traditional Feature Extraction + Machine Learning Methods:	12
2.2.2 Deep Learning Based Solutions.	13

2.2.3	Self-Developed CNN Architectures:	15
2.2.4	Handling Lack of Available Data	18
2.2.5	Data Augmentation	18
2.2.6	Transfer Learning	20
2.3	Image Processing	25
2.4	Summary	27
CHAPTER 3 METHODOLOGY		30
3.1	Project Definition and Project Proposal	30
3.2	Detailed Research and Project Framework Planning	30
3.3	Detailed Implementation	32
3.3.1	Dataset	33
3.3.2	Dataset Preparation and Analysis	33
3.3.3	Image Enhancement and Augmentation (Dataset Pre-processing)	36
3.3.4	CNN model solution Development	51
3.4	Results, Analysis, Interpretation and Conclusion	66
3.5	Final Documentation and Presentation	66
CHAPTER 4 RESULTS		67
4.1	Image Enhancement	67
4.1.1	Qualitative Analysis of Images Before and After Image Processing	67
4.1.2	Analysis of Evaluation Metric (PSNR SSIM and MSE)	69
4.2	CNN SOLUTION DEVELOPMENT	70
4.2.1	Tagging System	71
4.2.2	Phase 1	73
4.2.3	Phase 2	82
4.2.4	Phase 3	88
4.2.5	Phase 4	90
CHAPTER 5 DISCUSSION		93
5.1	Image Processing Enhancement as a Data Pre-Processing Step	93
5.2	Performing Data Augmentation	94
5.3	Network Architectures	95
5.4	Evaluation Process	96
5.5	Comparing Image Size	97
5.6	Optimization	98
CHAPTER 6 CONCLUSION		101

6.1 Conclusion of Study	101
6.2 Limitations and Future Works	102
REFERENCES	104
APPENDIX A	110
APPENDIX B	116
APPENDIX C	117
APPENDIX D	118
APPENDIX E	120
APPENDIX F	121
APPENDIX G	124
APPENDIX H	131

LIST OF FIGURES

Figure 1: Human Body GI Tract System (Institute, 2022)	2
Figure 2: Convolution Operation	14
Figure 3: Project High Level Framework	30
Figure 4: Detailed Process Flow of Stage 2.	31
Figure 5: Detailed Process Flow of Stage 3	32
Figure 6: Detailed Process flow of Data Analysis from Stage 3.	33
Figure 7: Visualization of each image class. (a) 0_normal, (b) 1_ulcerative_colitis (c) 2_polyps and (d) 3_esophagitis	34
Figure 8: Average Pixel Distribution for each Image Class.	35
Figure 9: Detailed Process flow of Image Enhancement + Augmentation (Dataset Pre-processing) from Stage 3.	37
Figure 10: Detailed Process Flow of CLAHE Image Enhancement (J. Lee et al., 2015).	39
Figure 11: Detailed Process Flow of Rayleigh Image Enhancement	41
Figure 12: Detailed Process Flow of Multiscale Image Enhancement	45
Figure 13: Detailed Process flow of CNN Solution Development from Stage 3.	51
Figure 14: Architecture of 01_Base_model1 (Group 1)	54
Figure 15: Architecture of 02_baseModel2 (Group 2)	54
Figure 17: Architecture of ResNet50-V1 (a) Step-block, (b) Stage1-Block1, (c) FC-Block (S. Wang et al., 2021)	55
Figure 18: Residual Connection (ResNet Building Block)	56
Figure 19: Additional Layers Added to ResNet50-V1 Transfer Learning Models.	56
Figure 20: Comparison of ResNet-V1 and ResNet-V2 Residual Blocks (Van Hieu & Hien, 2020).	57
Figure 21: Architecture of ResNet50-V2 (a) Step-block, (b) Stage1-Block1, (c) FC-Block	58
Figure 22: Additional Layers Added to ResNet50-V2 Transfer Learning Models.	59
Figure 23: Depth wise Separable Convolution Operation (Kumar et al., 2021)	60
Figure 24: (a) Standard Convolution Layer with Batch Normalization and ReLu activation, (b) Depth-wise separable convolution with depth wise and pointwise layers followed by Batch Normalization and ReLu activation (Bardina et al., 2020)	60

Figure 25: Architecture of MobileNet-V1 (W. Wang et al., 2020)	61
Figure 26: Additional Layers Added to MobileNet-V1 Transfer Learning Models.	62
Figure 27: Comparison of MobileNet-V1 against MobileNet-V2 Architectures.	63
Figure 28: Comparison of Image Before and After Undergoing Image Processing Enhancement for each class. (a) 0_normal, (b) 1_ulcerative_colitis, (c) 2_polyps, (d) 3_esophagitis	68
Figure 28: Image Processing Enhancement Results. (a) PSNR, (b) MSE, (c) SSIM.	69
Figure 29: Phase 1 Results	73
Figure 31: Comparison of Test Acc, Val Acc and ValTestAvg Acc for the Best 10 Models in Stage 1.	74
Figure 32: Test F1-Score of the Best 10 Models from Stage 1.	75
Figure 33: Confusion Matrix of the Best Model from Stage 1 (model I_9_1_N) (a) Validation Set, (b) Test Set.	76
Figure 34: Comparison of Average Accuracies Between ResNet Architecture against MobileNet Architecture.	76
Figure 35: Comparison of ValTestAvg Acc and Test Acc of the Models from Stage 1. Categorized by Model Type According to Colour.	77
Figure 36: Averaged ValTestAvg Acc and Test Acc by Model Architecture Type.	77
Figure 37: Averaged ValTestAvg Acc and Test Acc of Models in Stage 1 by (a) Optimizer, (b) Learning Rate Configuration.	78
Figure 38: Count of Best 10 Models from Stage 1 by (a) Image Processing Enhancement Performed, (c) Model Optimizer, (c) Learning Rate Configuration	79
Figure 39: Training Curve of the Best 10 Models from Stage 1.	81
Figure 40: Phase 2 Results	82
Figure 41: Test F1-Score of the Best 10 Models from Stage 2.	83
Figure 42: Confusion Matrix of the Best Individual Model from Stage 2 (Model K_11_6_C_TALR) (a) Validation Set, (b) Test Set.	84
Figure 43: Confusion Matrix of the Ensemble Model from Stage 2 (Model M_12_12345678910_MIXED) (a) Validation Set, (b) Test Set.	84
Figure 44: Training Curve of the 10 Models from Group 11, Stage 2.	86
Figure 45: Averaged ValTestAvg Acc and Test Acc of Models in Stage 2 by (a) Transfer Learning Type, (b) Optimizer, (c) Learning Rate Configuration	87
Figure 46: Comparison of ValTestAvg Acc and Test Acc of the Models from Stage 1 and Stage 2. Categorized by Model Type According to Colour.	88
Figure 47: Phase 3 Results.	88

Figure 48: Comparison of Test F1-Score for Models in Stage 1 against Stage 2 Models and Ensemble Model in Phase 3.	89
Figure 49: Phase 4 Results.	90
Figure 50: Comparison of Test F1-Score for Models in Stage 1 against Stage 2 Models and Ensemble Model in Phase 4	91
Figure 51: Confusion Matrix of the Ensemble Model from Phase 4 Analysis (Model M_12_12345678910_MIXED) (a) Validation Set, (b) Test Set.	92
Figure 52: Comparison of Model 03_TransferLearningFeatureExtraction_ResNet50V1's Training Curve Before and After Optimization (a) Without Optimization, (b) With Dynamic Learning Rate (LrScheduler), (c) With Adam Optimizer, (d) With Adam Optimizer and Dynamic Learning Rate (LrScheduler).	99

LIST OF TABLES

Table 1: Analysis of Input Image Size Distribution	35
Table 2: Summary of Datasets Set 1 - 5.	38
Table 3: Abbreviations for Equation 1	42
Table 4: Abbreviations for Equation 2 and Equation 3.	43
Table 5: Abbreviations for Equation 5	44
Table 6: Abbreviations for Equation 7.	46
Table 7: Abbreviations for Equation 8 and Equation 9	47
Table 8: Abbreviations for Equation 10 to Equation 13	48
Table 9: Abbreviations for Equation 14	49
Table 10: Abbreviations for Equation 15	49
Table 11: Abbreviations for Equation 16	50
Table 12: Abbreviations for Equation 17	50
Table 13: Networks' Training Baseline Parameters.	52
Table 14: Python Libraries used during Development	65
Table 15: Model Tagging Naming Convention	71
Table 16: Best 10 Models from Stage 1.	74

LIST OF EQUATIONS

Equation 1	39
Equation 2	42
Equation 3	42
Equation 4	42
Equation 5	43
Equation 6	44
Equation 7	46
Equation 8	46
Equation 9	47
Equation 10	47
Equation 11	47
Equation 12	47
Equation 13	47
Equation 14	48
Equation 15	48
Equation 16	49
Equation 17	50
Equation 18	50

LIST OF ABBREVIATIONS

Acc	Accuracy
CAD	Computer Aided Diagnosis
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
GI	Gastrointestinal
LR	Learning Rate
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
PSNR	Peak-to-Sound-Noise Ratio
RGB	Red-Green-Blue
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity
SVM	Support Vector Machine
ValTestAvg	Validation-Test-Average
WCE	Wireless Capsule Endoscopy

CHAPTER 1

INTRODUCTION

This chapter provides an overview of the study. The subject domain i.e., GI Tract disease diagnosis and the prevalent issues faced within are first introduced to gain a better understanding. Following which, the application of Artificial Intelligence and Image Processing to automate the diagnosis process are introduced. The challenges faced in automating GI Tract disease diagnosis are then discussed to identify the problems. The problems to work on the project are then streamlined in the problem statement. The research questions, aim, objectives, scope of research and significance of study are then defined. The remaining chapters are organized as followed: Chapter 2 details the literature review. Chapter 3 defines the project's methodology. Chapter 4 presents the results of the project. Chapter 5 analyses and discusses the findings of the project. Last but not least, Chapter 6 concludes the project.

1.1 General Introduction to GI Tract + Diseases

The Gastrointestinal tract, commonly known as the GI tract encompasses the passageway of the human digestive system from the mouth to the anus. It is comprised of the mouth, throat, oesophagus, stomach, the small and large intestines, the rectum, and anus (Institute, 2022). The GI tract is a crucial component of the digestive system as it is the passageway in which food enters and leaves our body and it acts as a medium where digestion and nutrient absorption occurs. Most nutrients required by the human body cannot be utilized directly from the food we eat on a daily basis. It is in the GI tract where most of these complex nutrients are digested by mechanical and chemical means so that they are broken down into their simplest form for bodily absorption and consumption (Hornbuckle et al., 2008). Figure 1 below depicts the GI tract of a human being.

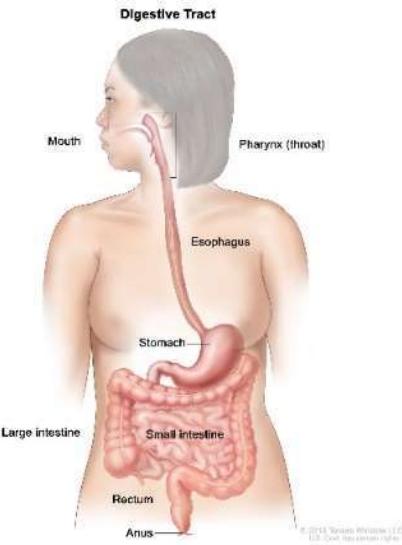


Figure 1: Human Body GI Tract System (Institute, 2022)

GI tract related diseases are one of the most prevalent diseases in our community. The WHO reported that an approximate 765000 deaths were caused by stomach cancer whereas colon and rectum cancer contribute to an approximate of 525000 deaths globally (Rustum et al., 2021). Studies from (Siegel et al., 2017) reported a total of 135430 new GI tract diseases occurrences since 2017, which grew at a staggering rate of 200000 cases annually since 2011. The American Cancer Society also reported an estimated 106180 cases of colon related cancer and 44840 related rectal cancer year to date. It is reported the lifetime risk of developing colorectal cancer is approximately 4.3% for men and 4.0% for women. In the United States, colorectal cancer leads as the 3rd cause of cancer accumulating an approximate 52580 deaths so far in 2022 (Team, 2020). These statistics show how commonly, and severe GI tract related diseases are globally.

Lesions found on GI tract passageway can be categorized into benign, pre-cancerous, early cancer and advanced cancer lesions. Benign GI tract lesions tend to have lower risk of progressing into cancer in the short run. However, if untreated in the long run, these lesions may potentially progress into cancerous cells over time. Pre-cancerous GI lesions onwards have the highest likelihood to develop into cancerous diseases which can be life threatening. Clinical data have shown that the 5-year survival rate of these diseases remain low between 23-27% with some diseases going as low as 4% (Siegel et al., 2017). However, statistics show that the survival rate rises exponentially to around 95% if these lesions are detected during early stages

(Sharif et al., 2021). This clearly shows how imperative early detection of diseases are in saving lives. With early detection, active prevention and intervention steps can be implemented to further improve the recovery rate of individuals from these diseases.

1.2 Current monitoring methods and their limitations

The present available methods for monitoring and diagnosing GI tract related diseases is via endoscopy (Take et al., 2015) which is comprised of Gastroscopy, Colonoscopy, and Wireless Capsule Endoscopy (WCE). Gastroscopy and Colonoscopy are the most commonly performed procedures. Gastroscopy is used for examination on the upper sections of the GI tract whereas Colonoscopy is used to monitor the lower sections of the GI tract. These 2 methods however have their limitations. Firstly, they are invasive, and patients need to be present and sedated for the procedure which is troublesome. These procedures also require a lot of resources making them costly for patients and is a major barrier for many patients to perform checks regularly. The procedures also have limitations in terms of the depth and extent of their image capturing capability which leads to issues of continuity and traceability. They are invasive in nature, the patients are also exposed to inherent risks which may lead to complications such as perforations, infections or even reaction to the sedation during the procedure which are common.

WCE on the other hand is a newer method of micro-digestive endoscopy. It is a non-invasive imaging technique which involves the patient ingesting a small camera device which then passes through the entire GI tract whilst capturing and recording images. The WCE capsule records and transmits images to an external receiver in real time. The capsule is then discharged via defecation by the patient and its recordings are then analysed by medical professionals. On average, a WCE can produce between 50000 to 120000 images in a single examination (Du et al., 2019; Rustam et al., 2021). Currently, these large amounts of images need to be manually examined by medical professionals to detect and diagnose the relevant disease. However, the sheer size of the data recorded can be challenging for any individual to go through manually. The proportion of data containing recordings of lesions (if any) are mostly very small in proportion as compared to the entire data recorded for any patient. Medical professionals are required to go through the entire recording which may take hours and herein lies the bottleneck of the diagnosis process. Manual diagnosis by medical professionals may also be subjective depending on the medical professional's experience and opinion of the image.

1.3 Automating the entire image analysis process with AI

The breakthrough developments of technology in the domain of Artificial Intelligence and Image Processing, has unlocked new possibilities of solutions for many problems in the medical diagnosis domain. This includes the detection and classification of various GI tract related diseases by analysing GI tract imaging. Traditional mundane task of going through and analysing thousands of data / images manually can now be replaced with a more automated and efficient pipelines to identify and classify images. Such a solution can be beneficial in not only making accurate diagnosis, but also enables medical practitioners to focus on other core part of their tasks in making medical breakthroughs and focusing on treatments. Hence, a lot of interest and research has been poured into this particular field.

The core problem of diagnosis is a classification problem which involves accurately detecting the presence of lesions in WCE images and classifying the detected lesions into an accurate class of a particular disease. Presently, the general approaches by available literatures can be categorized into 2 main pathways of solution development which are

- a) Image Processing Feature Extraction + Machine Learning algorithms
- b) Deep Learning techniques.

Image Processing Feature Extraction (also known as traditional feature extraction) methods involve manually extracting multiple features using image processing techniques from the raw image to create a dataset with their corresponding classes. These manually handcrafted features are then fed as input into Machine Learning algorithms such as Support Vector Machines (SVM), Random Forest (RF) and K-Nearest Neighbours (KNN) classifiers. The Machine Learning algorithms will learn patterns from different features of each class to be able to distinguish and accurately classify the image's class. These features could pertain to various aspects such as the colour and texture characteristics of the image.

An alternative solution that is also commonly sought after are Deep Learning models. Deep Learning models specifically Convolutional Neural Networks (CNN) have unlocked new potential to carrying Computer Aided Diagnosis (CAD) tasks at scale. These models are developed to automatically learn, analyse, and improve their classification performance from large amounts of data. This makes them ever so dynamic as feature extraction and learning of patterns is done automatically at scale. Multiple features are extracted automatically, and these

features are used by the network to learn patterns from the raw image data. Deep Learning model frameworks also provide associated utilities to handle problems such as lack of training data using Data Augmentation and Transfer Learning which will be discussed in later sections. The great flexibility that CNN architectures offer combined with their performance capability at scale in a dynamic environment have made them a preferred solution for many CAD related problems.

1.4 Challenges in developing a Deep Learning CAD solution.

Though there are relevant works related to the diagnosis of GI tract diseases, many of them gravitate towards binary classification problems such as the works performed by (Pannu et al., 2020) and (Rustum et al., 2021). Binary classification-based solutions may assist in detecting diseases but lack the refinement for practical and real-world applications where streamlining the diseases are of more interest. Moreover, the available works with regards to multiclass classification of GI tract diseases are developed using smaller sized datasets. This can be attributed to the lack of available annotated data for large scale training and testing.

To overcome this issue, many authors like (Hmoud Al-Adhaileh et al., 2021) and (B. Liu et al., 2018) have leveraged on data augmentation techniques (creating synthetic data) and using transfer learning to develop their solution. However, these solutions were developed and tested based on a small sized single source dataset. Training and testing on a small sized dataset, even though augmented may lead to the model to overfit to the data. A lot of the augmentation techniques used in literatures are observed to be geometrical augmentations such as rotation, translation and flipping which may not be as practical in creating newly seen data for the CNN models as the geometric orientation of the images in the GI tract is quite symmetrical.

On the other hand, transfer learning models are trained with a wide variety of image classes with some having output classes of around 1000 classes. Though transfer learning assists in improving the CNN models' performance with a lack of data, it can only do so to a certain extent as these models are developed for a wide variety of classification applications. Hence, taking and applying them directly for medical related CAD applications may not necessarily results in the optimal results, especially when applied in scale. Retraining and fine-tuning these models with sufficient medical images are still required in order to tune these models and generalize them for medical CAD applications. Moreover, transfer learning models are

typically trained with natural images with colourful and clear boundaries whereas medical GI tract images with lesions found in medical field are often unstandardized, have a lack in consistent shape and don't always have clear edges which these models may not be familiar with. Hence, the performance observed may not reflect the model's capability when put into real-life large-scale applications.

A lot of works also primarily focus on the development of a single CNN architecture as the proposed solution. Though this may be less complicated and simpler to develop, but it may not be as robust a solution when applying it at scale as most of the predictions is only dependent on a single model.

Image processing enhancement techniques are known to improve the quality of images that are blurry, lack in consistent shape, have unstandardized edges, lack of clear edges and poorly illuminated. These enhancements technique help to make hidden features more distinguishable in the image. Many of the available works pertaining to image enhancement however only employ simple techniques such as simple Gaussian and Laplacian filtering with a lack of work exploring image processing in depth as a pre-processing step.

1.5 Problem Statement:

Other than methods of self-developed architectures and transfer learning feature extraction with on a single model, it is observed that there is a lack of work pertaining on other techniques such as transfer learning fine tuning and ensemble model development techniques.

There is a lack of work with regards to studying how image processing enhancement can be leveraged as a pre-processing step to improve the quality of the input image prior to feeding it into the CNN classification model. A lot of the enhancement techniques observed being used are simple techniques such as Gaussian blurring and Laplacian sharpening. Though these techniques improve the quality of the image to a certain extent in terms of texture, they do not contribute much to enhance features pertaining to colour, contrast and illumination which are important factors when analysing an image.

Many augmentation techniques used are only geometric augmentation techniques. There is a lack of techniques that manipulate the image in terms of contrast, colour and illumination. In

reality, images of WCE cameras may vary in terms of resolution, colour, and contrast. By only developing a model that is adapt to a single type of image, this may result in the developed model being not adaptable to newly acquired images from different sources of WCE cameras.

1.6 Research Questions

The study addresses the following research questions:

- What architectures, techniques and frameworks can be used to design, develop, and optimize Deep Learning CNN classification models used in Computer Aided Diagnosis (CAD) systems to diagnose various GI Tract related diseases?
- What image processing enhancement techniques focusing on colour and contrast can be used to enhance the improve the quality of features in the raw GI tract WCE images?
- Apart from geometric augmentations, what other augmentation techniques can be used to create a variety of data to generalize the proposed solution?

1.7 Aim

The aim of this study is to design and develop a generalized multiclass CNN classification algorithm to be used in CAD system for diagnosis of various GI tract diseases by analysing WCE GI tract images with varying tract lining lesions.

1.8 Objectives

The following project objectives are defined:

- To research, design and develop suitable CNN classification models to diagnose various GI tract diseases based on analysis of WCE images. This encompasses basic model design, framework selection, architectures, and parameter tuning.
- To research, design and develop suitable image processing enhancement techniques as a pre-processing step to enhance the quality features in the raw images to further improve the performance of the developed solution.
- To analyse and perform exploratory data analysis on the data to understand its characteristics.
- To study and implement alternative non-geometric based data augmentation methods to create a more generalized dataset for developing the solution.

1.9 Scope of Research:

The scope of the project is defined as such:

- The scope of the study is limited to the design and development of a Deep Learning based CNN classification model.
- The raw dataset images used in this study are only limited to images obtained from WCE imaging.
- The dataset used in this study is limited to readily available dataset. The study will not involve acquisition of new raw data from a new source.
- The study is a multiclass classification problem. The number of classes considered in this study is limited to 4 different classes of diseases.
- With respect to image quality enhancement, the image processing techniques will cover only image enhancement techniques and will not encompass image restoration techniques.
- With respect to data augmentation, the data augmentation techniques used this study will cover only colour, texture, and contrast-based augmentations techniques and will not cover geometric techniques to augment the images.

1.10 Significance of Study

This study reinforces prior knowledge related to the subject matter as it widens the scope of parameters considered during the process of solution development. Firstly, a more generalized dataset is used to develop the solution whereby WCE images used are mixed from different sources. Using a variety of data to train the solution aids in making it more adaptable, especially when deployed in a real-world environment. This work also performs an extensive study with regards to image enhancement as a pre-processing step, an area that is observed to be very unexplored in many literatures. The study also proposes an alternative method in performing data augmentation with the goal of increasing the variety of images for the CNN models to learn from. This helps to generalize the model's capability in distinguishing different features in images. Other than traditional self-developed architectures and transfer learning feature extraction models, this project also studies the performance of architectures using transfer learning fine tuning and ensemble model architectures in detail. The framework and findings of this work serves as a reference for future research and development with regards GI tract disease CAD solution development.

The findings from this study also demonstrates the proof of concept that a generalized and practical solution can be developed to detect and classify various GI tract related diseases. This developed CNN solution can then be integrated into a fully developed CAD system to be deployed at commercial scale. In terms of social and industry impact, the realization of such a system consequently contributes to the long-term efforts to automate the entire diagnosis process for various GI tract diseases. As emphasized in Section 1.1, early detection of diseases plays a major role in battling it and hence, enabling early detection with such a system will definitely help in saving many lives. Moreover, it also helps in reducing the workload of medical practitioners. Medical practitioners such as doctors, nurses and researchers can then be more focused on other significant tasks at hand such as treatment plans and patient care. The emotional and cost barrier for patients to come and regularly get their check-ups are also significantly reduced.

CHAPTER 2

LITERATURE REVIEW

This chapter details and summarizes the review of various literatures covering journal articles and conference papers on firstly, GI tract diseases in terms of conceptual medical aspects and secondly, relevant works of CAD based solutions development for GI tract diseases and other related diseases in the medical domain. The conceptual review of GI tract diseases is performed to better understand the subject matter from a conceptual standpoint and to also familiarize with the different classes of diseases being classified. On the other hand, the literature review on relevant CAD works aims to explore, identify, summarize, and provide a comparison of relevant works developed over the recent years. This will be further refined into the various methodologies and network architectures being used as well as results.

2.1 Introduction to GI Tract Diseases and WCE

The output classes for this project covers 4 main classes of disease which are Normal (no disease), Ulcerative Colitis, Polyps, and Esophagitis. This section will provide a summary of each class and their significance. Moreover, this section will also provide an overview of how WCE works and its significance.

2.1.1 Ulcerative Colitis

Ulcerative Colitis is known as an inflammatory bowel disease that causes inflammation and sometimes ulcers, primarily in the colon and rectum. Some of these ulcers eventually develop into pus when left untreated due to long exposure and bleeding which can lead to life threatening complications (National Health Service (NHS), 2022). There are various factors that can cause Ulcerative Colitis. Typically, it presents itself or is detected when a patient comes in with bloody diarrhoea. However, if detected and treated early, its risks can be significantly reduced.

2.1.2 Polyps

Polyps is also another common occurring disease that is associated with cancer. Polyps can be further categorized into 2 types which are hyperplastic (benign and has low probability of progressing into cancerous cells) and adenomatous polyps (has higher tendency to turn into

cancer) (Ganz et al., 2012). Early detection of polyps significantly help doctors in diagnosing and coming up with a treatment plans to prevent its progression into early-stage cancer.

2.1.3 Esophagitis

Esophagitis is a reflux related disease of the GI Tract. It usually affects the upper portion of the digestive tract. It refers to the inflammation or injury of the oesophagus, the portion of the GI tract the connects the mouth to our stomach. Esophagitis is caused by a variety of reasons but is commonly associated with acid reflux, infections allergies or as a reaction to medications. If left untreated over long periods of time, it may lead to damages of the oesophagus lining and may disrupt the normal physiological function of the organs to transport food from the mouth downstream of the GI tract.

2.1.4 WCE imaging

Wireless Capsule Endoscopy (WCE) is a new method of imaging which involves the patient ingesting a small capsule with camera embedded inside it. The capsule then passes through the GI tract captures and transmits the captured images to an external receiver. It is a revolutionary new tool to perform painless imaging of the GI tract and accordingly perform investigation and medical diagnosis. A very attractive point of WCE is that it offers a safer, less hassle and more cost-effective alternative to Endoscopy and Colonoscopy. However, there are still barriers to its application in the industry as its efficacy, tolerance, safety, and performance is not widely accepted within the industry.(Rustum et al., 2021). Moreover, medical professionals are still required to go through thousands of images recorded by the WCE, of which only has small proportion of disease labelled images (if any). A lot of research interest has been poured into the development of a Computer Aided Diagnosis (CAD) systems that will significantly help medical practitioners in analysing and filtering disease related WCE images to pay attention to. CAD systems contribute greatly in the manner of enhancing the raw images and automating the filtering process (Halalli & Makandar, 2018).

2.2 Relevant Works on CAD Solutions Development

This section details relevant works on CAD based solution development for GI tract disease diagnosis as well as other diseases in the medical field. It will cover various methods and architectures of developing the solution, broken down into Feature Extraction + Machine Learning techniques and Deep Learning based techniques. Furthermore, the section will also

cover the review of data pre-processing steps such as data analysis, image augmentation, and image enhancement techniques to improve the performance of the developed solutions.

2.2.1 Traditional Feature Extraction + Machine Learning Methods:

A lot of early solutions developed for GI tract disease CAD solutions use a combination of traditional features extraction methods with Machine Learning Models. It typically involves extracting specific handcrafted features using image processing and using a collection of these features as datapoints to feed into a Machine Learning model to classify.

In their work, (Emam et al., 2015) proposed various methods of feature extraction such as Colour Moment RGB, Colour Moment HSV, Colour Histogram, Local Binary Pattern (LBP) and statistical features from the raw image. (Karkanis et al., 2003) proposed a colour-based feature extraction method to represent different regions of colonoscopy images. These features were then used to detect the presence of polyps within the colon. Discrete Wavelet Decomposition was implemented on each colour channel of the raw image to extract wavelet features, also known as co-occurrence matrices. These co-occurrence matrices correspond to the second order statistical representation of the raw image. Statistical values such as entropy, angular second moment and correlation were then computed as input features using the extracted matrices resulting in a total of 144 different statistical features representing different aspects of the raw image. The authors termed these features as Colour Wavelet Covariance (CWC). The extracted features were then used as input data for a Linear Discriminant Analysis (LDA) algorithm to detect if polys were present or not.

Authors of (Ellahyani et al., 2021) used a combination of both texture features i.e., complete local binary patterns and colour features i.e., Global Local Oriented Edge Magnitude Pattern (Global LOEMP) for feature extraction. Both of these sets of features were then combined together and used as input features into a Support Vector Machine (SVM) and a Multi-Layer Perceptron (MLP) model for final classification to detect ulcers in the GI Tract. (Tuba et al., 2017) proposed block-based segmentation where the average saturation from the HSI, skewness and Kurtosis of the uniform LBP histogram were extracted as features prior to feeding them into an SVM model to distinguish between bleeding images and non-bleeding images. The entire raw image was segregated into several regions. Following which, feature extraction and classification was performed for each of these regions. The SVM model was tuned using a Grid

Search method. The HSI colour model was used to extract features instead of the RGB colour space as the RGB colour model was deemed not suitable for image processing. Colours of bleeding images tend to be more vivid and intense as compared to normal images and hence, there is a distinguishing factor between normal and bleeding images when looked in the perspective of the HSI colour model. On the texture end of features, the Local Binary Pattern (LBP) of the image was constructed, and statistical calculations were performed on the LBP to extract values such as skewness, kurtosis, energy, and entropy. These extracted features are then used as inputs in the SVM model which is trained to label the different regions of the image. Majority voting of the regions in the image was used as the deciding factor to determine the final class of the image.

Based on review, it is observed that the inherent problem with the Feature Extraction + Machine Learning approach is that they are very limited to the specific conditions / environment in which they are built for, specifically with regards to manual feature extraction. For applications where the environment and quality of the image is constant such as in the domain of manufacturing, these sorts of techniques may be effective. However, for situations where the environment and image quality are dynamic such as in the medical domain, this may not be so suitable. Most of the reviewed literatures focused on detecting a single disease where the features are tailored only to a specific disease class. Putting this into the context of multiple classes / diseases however, the variety and combination of features that is required may be too large and complex to cover. This can not only be tedious, resource and time intensive but also does not perform well when put into commercial scale. Moreover, internal body organs are very blurry visually as these images are subjected to a lot of noise, luminance variation, rotation and blurring making them very unstandardized. Thus, manually selecting features to cover so many classes and many different handcrafted features can be unrealistic.

2.2.2 Deep Learning Based Solutions.

Deep Learning and availability of vast amounts of data over the recent years, has unlocked the possibility of improving the automation of many day-to-day tasks. Deep Learning algorithms have better adaptability to dynamic data. Deep Learning models are mathematical algorithms that iteratively and progressively learn representations of huge amounts of data in consecutive layers to produce an output which is then compared to the actual data label. Depending on the difference between the predicted and actual data, an error feedback mechanism iteratively and

systematically improves the mathematical model's parameters which consequently improves its learning capability. When trained with large amounts of data, especially varying data, the model is capable of generalizing to real life situations. This makes them more robust in an ever-changing environment and is a highly sort after solution for many problems in the medical field. Convolutional Neural Networks (CNN) are synonymous with image related problems in Deep Learning. CNN is a specialized neural network that is designed to process 2-Dimensional image data. The basis of the CNN network is the convolutional process.

Convolution is a linear mathematical operation that involves the multiplication of a set of weights with the input matrix. These weights are commonly known as filters or kernels. Each kernel size is smaller than the input image data and the multiplication between these 2 arrays are a dot or scalar product. Using a smaller kernel size allows multiple systematic multiplications at different points of the input i.e., different points in the image. Each kernel / filter is designed to extract a specific set of features in the image. The application of the filter across the entire image allows the kernel to discover / extract features at any location within the image i.e., translational invariant. The result of each convolutional operation is a single value. When applied across the entire image at multiple points, the resultant final output is a 2D array that represents the filtering of the input image array which is known as a feature map. This represents the basic operating principle of a CNN. A depiction of this is shown in Figure 2 below that depicts a convolution operation on 7×7 matrix with a 3×3 kernel.

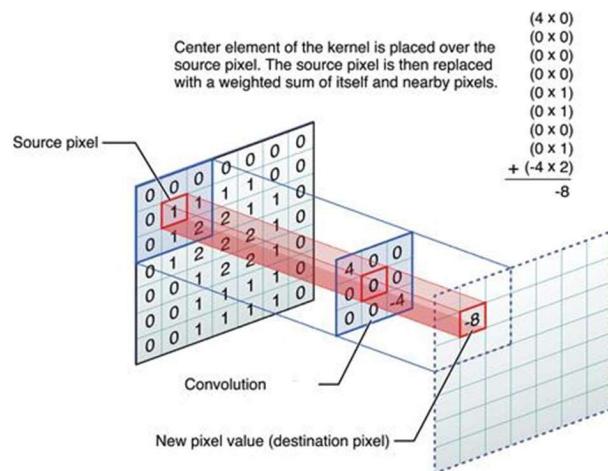


Figure 2: Convolution Operation

This process is repeated through multiple layers of the CNN architecture. A full-scale CNN model applies multiple kernels and generates multiple feature maps to learn from. In most cases, the kernels are not applied on a single channel but multiple channels. Given the scale, this produces high amounts of image data representations in which the model can learn from to accurately perform classifications. A detailed explanation of the working principle of convolutional neural network can be referred to (Namatēvs, 2018; Zhou & Sun, 2020).

2.2.3 Self-Developed CNN Architectures:

(Pannu et al., 2020) developed a CNN architecture to classify WCE images into binary classes of bleeding vs non-bleeding. In their study, the authors proposed using an ensemble of CNN networks. The ensemble of networks are CNNs that have the same architecture which were fed with different batches of input data to learn from. A total of 5 ensemble networks were considered and an aggregation of the classification results from each of these networks were taken as the final output results. The study used images of size 100 x 100 x 3 pixels which was passed through a series of convolutional, batch normalization ReLu and max pool layers prior to flattening the neurons for the final output. The proposed solution was capable of achieving an overall 95% accuracy when tested on static images and 93% when tested on a live video dataset.

(Jia & Meng, 2016) developed an 8-layer CNN model comprised of 3 convolutional layers, 3 max pooling layers and 2 fully connected layers to detect bleeding images in the GI tract. The dataset was trained on a 10000-size dataset of WCE images. The network was trained using the SGD optimizer, over 5000 epochs with a batch size of 100 for each epoch. The learning rate was set at 0.001, with a momentum of 0.9 and a weight decay of 0.004. Inspired with the effectiveness of the SVM model in classification, the authors further modified the proposed network by replacing the final FCN with an SVM based classifier for final prediction. Results showed that the model was capable of performing well with an overall recall of 99.2%.

(Seguí et al., 2016) proposed using a single CNN architecture with different variations of processing the input data i.e., input data dimensions to classify 6 different motility states of the small intestine i.e., Turbid, Bubbles, Clear Blob, Wrinkles, Wall and undefined. What distinguishes them from other methods are the number of channels used for processing. Typical images are processed using single or 3 RGB channels. However, this study included 2 additional

channels which are firstly, the Laplacian of the image brightness channel L and secondly, the Hessian of the image brightness represented by H as inspired by (Santi Segui et al., 2014; Santí Segui et al., 2012). The authors experimented with a late-fusion and an early-fusion workflow. The input size of the CNN network is 100 x 100 x N pixels where N represents the number of input image channels, a manipulated variable in this study. Three convolutional, batch normalization, and pooling layers were then combined to build the network. The network was trained with SGD as the optimizer, with a batch size of 128, learning rate of 0.1 that reduces by a factor of 10 every 100000 iterations. A total of 400000 iterations was considered during training. The proposed late-fusion architecture model considers 3 different networks that takes in input images of RGB, H and L separately. Features maps of these 3 networks are then concatenated at the final convolutional layer prior to flattening them. Each CNN would be updated and trained independently of one another. The early-fusion architecture on the other hand concatenated the RGB, H and L channels at the input head and fed them into the 5-channel input CNN architecture. This allowed the entire network to be trained simultaneously. Results showed that late-fusion workflow produced better results. The authors concluded that merging the channels aided the CNN architecture in improving its discriminatory capability as more feature / representations are learned about the input image data. RGB channels help the network learn more on the colour representations of the data whereas L and H feature maps help the model in learning more about the structure and textural information about the image.

(Shin & Balasingham, 2017) compared the performance of 2 approaches for detection of polyps in colonoscopy images, firstly using Feature Extraction + Machine Learning and secondly, CNN architectures. For the Machine Learning model, the SVM model was adopted using features of Histogram of Oriented Gradient (HOG) combined with hue Histogram features to represent shape and colour features of the image prior to feeding the features into the SVM model. The custom CNN model developed comprised of 3 convolutional and pooling layers for classification. Results showed that the CNN based solution performed better as compared to the Feature Extraction + Machine Learning method. The authors also experimented by comparing the performance of CNN architectures using a single channel input image versus RGB images. Results showed that the solution that performed the best was the CNN model with RGB images.

Authors of (J. Y. He et al., 2018) proposed a CNN solution for hookworm detection by analysing WCE images. The authors proposed 2 separate CNN networks known as edge extraction network and hookworm classification network which were integrated together. 2 separate networks were created to avoid edge feature caching and to speed up the classification performance. In order to combine the features produced from the 2 separate networks, 2 edge pooling layers were introduced to create a combined feature map emphasizing on the tubular regions of the image. The edge extraction CNN model was built based on the Holistically-Nested-Edge (HED) architecture proposed by (Xie & Tu, 2017) comprised of 16 convolutional layers in total. The hookworm classification network on the other hand comprised of a combination of several convolutional + pooling layers and 9 inception feature extraction network models. To combine tubular region information extracted from the edge network, the feature maps extracted from the edge detection network were merged to the hookworm detection network. 11-fold cross validation with leave one out method was used to train and validate the network. The results from 11 folds were averaged to get the final performance score. The learning rate of the model was set to 0.001, with gamma at 0.8, step size of 4000 and with a momentum of 0.9. The network was trained for 20 epochs with a batch size of 20. Data augmentation was performed to handle imbalanced data distribution. The authors compared the performance of the proposed model to a Feature Extraction + SVM model inspired by (Li & Meng, 2012). Sensitivity was used as the evaluation metric. Initial results showed that the SVM showed better performance in terms of accuracy but showed very poor results in terms of sensitivity when tested on the imbalanced dataset. However, the performance of the proposed model was superior to the SVM model when tested on the balanced dataset.

(J. Y. Lee et al., 2019) developed a 2 stage CNN network to detect perforations in the Tympanic Membrane of the ear. The study utilized medical endoscopic images of the ear. Stage 1 of the CNN framework aimed at detecting the presence of a Tympanic Membrane in the image or not. The 2nd stage on the other hand performed binary classification to determine if the Tympanic Membrane had perforations or not. Each CNN architecture stage is comprised of 2 sets of Convolutional + Max Pooling layers. The final layer was then flattened layer prior to predicting the final output. The input image size used was 224 x 224 x 3 pixels. A data split of 60:20:20 train: validation: test was used. The batch size used was 32, trained over 400 epochs. The optimizer used was SGD, with a learning rate of 0.0001 and momentum of 0.9. Results showed that stage 1 CNN was capable of achieving an overall of 98.7% accuracy and stage 2 CNN managed to achieve an 87.2% accuracy.

2.2.4 Handling Lack of Available Data

A very prominent barrier in the development of CNN models is the lack of training data or an imbalance of training data. Images with disease labels in the medical domain are very scarce as compared to images that are normal. This leads to a lack of information for the CNN to learn from during training. When applied to real world applications, the algorithm may not be capable to adapt and generalize well to new and different types of WCE images. Hence, two very commonly used methods are used to overcome this problem which are image augmentation and transfer learning.

2.2.5 Data Augmentation

Data augmentation is a dataset pre-processing step which takes the original dataset images, manipulates them either geometrically, texture wise or even colour wise to create new synthetic images. These new images are then added to the original dataset to increase its size or balance the dataset. Typically, the augmentation is done with controlled parameters to ensure that changes made to these raw images are within desired ranges. There are various alternatives to this by either rotating the images in a certain angle, translating the image by a certain distance, flipping the image or even by adjusting the contrast and colour spectrum of the image. This then creates a different image from the original image which can then be used as a new input datapoint to the model to learn from. With more images and features to learn from, the CNN is better capable to handle a variety of input image data. This section details relevant work presented by authors in performing data augmentation to counter the problem of a lack or imbalance of data points.

To counter the problem of imbalanced data and to increase the dataset size, (J. Y. He et al., 2018) implemented augmentation techniques of crop, flip, rotation, and smoothing using gaussian filters. (Pannu et al., 2020) and (Conference, 2017) implemented two types of augmentation which are geometric augmentations and colour-based augmentation. Geometric based augmentation included rotation of the images at multiple angles. Colour and contrast-based augmentation on the other hand included the Luminance channel stretching, Blurring and Poisson Noise. To alter the luminance of the image, the images were converted from RGB to YCbCr colour spectrum and multiplied with a factor on the Y (Luminance channel). The images were then reconverted to RGB channels. Blurring involved using a Gaussian kernel to blur the

images to reduce the features in the image. Poisson noise was added to the raw images to increase noise.

(Pei et al., 2017) applied geometric augmentations to raw MRI images of the small bowel using flipping and shifting. MRI images are hard to come by, especially those of with small bowel disease. The authors worked on developing an automated CNN + LSTM based network for the diagnosis of small bowel disease detection. The semi-automatic approach aimed to replace manual feature extraction using a CNN instead as the feature extractor. The CNN would be applied to a sequence of frames of images. The feature maps of the frames of images extracted by the CNN were then used as inputs into an LSTM to analyse the sequence and fluctuations of diameter of the small bowel. Based on the fluctuation in diameter length and contraction frequency, a classification would then be made to determine if small bowel disease was present or not.

To overcome the problem of an imbalanced dataset, (Rustam et al., 2021) applied data augmentation technique of flipping. Results showed a significant improvement in the model's performance using augmentation to train, where the model showed an increased accuracy from 80% to 99.3%. The authors however noted that augmenting and creating too many synthetic images may lead to data redundancy (Abbasi & Minhas, 2016; Shorten & Khoshgoftaar, 2019). (Hmoud Al-Adhaileh et al., 2021) used techniques of flipping, zooming, shifting and rotation on the Kvasir dataset (Pogorelov et al., 2017) of GI tract images to classify different types of GI tract diseases. (Takiyama et al., 2018) on the other hand augmented images using rotation. (B. Liu et al., 2018) utilized affine transformation using a combination of rotation, scaling, and horizontal and vertical mirroring to augment image data of gastric pathology images. (J. Y. Lee et al., 2019) applied geometrical augmentations of sheer range, rotation and horizontal flipping to create synthetic Tympanic Membrane images. (Hmoud Al-Adhaileh et al., 2021) performed augmentation using geometrical augmentation techniques of flipping, zooming shifting and rotation. (X. Liu et al., 2018) applied geometric augmentation techniques of rotation, width shifting, height shifting, zooming, horizontal flip, vertical flip, and scale normalization.

2.2.6 Transfer Learning

Transfer learning works on the principle of leveraging on large and complex state of the art CNN networks that were previously trained on very large datasets such as ImageNet by (Fei-Fei et al., 2010) or the MS COCO dataset by (Lin et al., 2014). The principal of transfer learning works such that information learned from one complexly trained model is generalized enough such that it can be repurposed for other related applications, even though not exactly similar. Transfer Learning models can be used in several ways of either via Feature Extraction or via Fine Tuning method. The state-of-the-art networks offer flexibility in terms of adding layers onto it and also the number of layers in its architecture that can be retrained. This makes it a very attractive approach to create CNN based architectures even though there is a lack of training and testing data as the performance of the network are considerably good considering the size of available annotated data. Moreover, given the complexity and amount of knowledge that has been learned by the Transfer Learning models, many of them offer very good results when trained properly. The training time required to achieve the performance is also considerably low as compared to training a CNN architecture from scratch. It is important to note however that transfer learning is not a one stop solution for any classification problems. Its applications should be considered on a case-to-case basis. There are instances of poor performance when transfer learning models are applied for certain applications, specifically in cases where the dataset is small. The network may not be capable of generalizing well especially when seeing new data that it has not been previously trained on (Yao & Doretto, 2010). There are numerous and very famous architectures used for transfer learning such as the ResNet architecture, VGG architecture, Alex Net architecture, MobileNet architecture and many more.

(Rustum et al., 2021) proposed a GI tract bleeding image recognition model using transfer learning of the lightweight MobileNet model. MobileNet was used as a feature extraction model which was added with an additional 3 stages of convolution-max pooling-batch normalization-dropout layers before flattening and final prediction. The problem is a binary classification problem. The authors compared the performance of 2 architectures which were the MobileNet + Additional Layers architecture against Additional Layers as a standalone CNN. Results showed that the combination of the MobileNet + Additional Layers produced the best results, achieving an overall accuracy of 99.3%

(Conference, 2017) explored and studied the performance of 3 different state of the art networks of LeNet, AlexNet, GoogLeNet and VGG networks using transfer learning to detect bleeding from GI tract images. Each of the explored state of the art networks had different training configurations. The LeNet network was developed and trained from scratch without any transfer learning due to its small size. In comparison the other networks of AlexNet, GooLeNet and VGG networks were used as Feature Extraction and Fine Tuning models. Their final layers were replaced with two FCN followed by a Softmax classification layer for a binary class of bleeding vs non-bleeding. The networks were trained using the SGD method with cross entropy loss. The solution was developed using the Caffe framework. A weight decay of 0.0005 was used, with a momentum of 0.9, batch size 32 trained over 30000 epochs was used. The learning rate was set at 0.0001 for LeNet and 0.01 for the other networks. All of the proposed networks were capable of achieving an accuracy of more than 95%. Moreover, the authors also compared the training time for each of these models and results showed that the VGG network architecture took the longest followed by GooLeNet, AlexNet and LeNet respectively.

(Qu et al., 2018) on the other hand performed a step wise Fine Tuning training scheme to train a Deep Learning network inspired by the method in which professionally trained pathologists analyse pathology images to classify between benign and malignant GI tract images. The authors performed 2 stages of training denoted as stage 1 and stage 2. Detailed and labelled malignant and benign images are not commonly available due to the scarcity of these images. Hence, the authors took inspiration on how pathologists first performed a high-level study to determine the type of cell before going into detail to determine if it is benign or malignant to develop their solution. In stage 1, the CNN network would classify pathology cell images between stroma and epithelium cells. Obtaining labelled pathological stroma and epithelium cell images in large quantities is much easier. State of the art pretrained CNNs trained on ImageNet dataset were utilized as the Fine-Tuning networks to classify between stroma and epithelium images. The state-of-the-art networks used were VGG-16, AlexNet and Inception-V3 networks. Performing stage 1 training tuned the network's weights to be more familiar with medical pathology images and their features. In the stage 2, these CNN were then trained on a smaller dataset of malignant and benign pathology images to perform the final classification. Results showed that by performing step wise fine tuning training method, the algorithm was capable of performing significantly better as compared to a single stage training framework where the images are only trained directly on the malignant and benign images.

(Hmoud Al-Adhaileh et al., 2021) performed transfer learning Fine Tuning using GoogleNet, ResNet-50 and AlexNet networks, pretrained on the very large ImageNet dataset to classify between different types of polyps. The final 3 layers of each of the networks were removed and replaced with a fully connected layer, before the final prediction node with a Softmax activation. The models were trained using Adam optimizer. The models were capable of achieving an accuracy of over 95% with an AUC of greater than 98%. The input image size used was 224 x 224 x 3 pixels for the GoogleNet and ResNet-50 architectures. An input image of 227 x 227 x 3 pixels was used for AlexNet. To enhance the image quality, pre-processing steps were performed using averaging filters. The dataset used was the Kvasir dataset comprised of 5000 images equally distributed over 5 classes of dyed-lifted Polyps, Normal Cecum, Normal Pylorus, Polyps, and Ulcerative Colitis (Pogorelov et al., 2017).

(Chen et al., 2018) developed a CNN based CAD system that analysed Narrow-Band Imaging (NBI) of the colon to detect Polyps by analysing the details of histologic features. The goal was to classify between 3 different classes of Neoplastic Polyps, Hyperplastic Polys, and normal case. Transfer Learning with a base model of Inception-V3 model was used by only removing the top layer of the network and replacing it with the input size of the NBI images. The results obtained by the CNN was compared to classification results performed by trained professionals. Results showed only a 10% variation in accuracy of the CNN model as compared to the predictions made by trained professionals. The CNN model was capable of achieving an accuracy of 90.1%, sensitivity of 96.3% and a specificity of 78.1%.

(Ayyaz et al., 2022) proposed a hybrid approach that leveraged on the concepts of Deep Learning and Machine Learning models. The authors used transfer learning with the VGG-19 and AlexNet architectures for feature extraction of WCE GI tract images to classify between 5 different classes of diseases. The total number of datapoints for each class was approximately 500 images. An input image size of 256 x 256 x 3 pixels was used. Both of these networks would be fed the input data simultaneously and their combined output features would total up to an N length array of features. A specific set of selected features would then be fused from each network architecture. This would produce a feature vector which would then be used as inputs into various Machine Learning algorithms of SVM, Bagged Tree, Cosine KNN, KNN, Naïve Bayes and Subspace Discriminant Models which would perform the classification tasks.

The authors used Genetic Algorithm (GA), a commonly used optimization algorithm to determine the best combination of features from the fused N length array. GA works on the principle of exhaustive searching to determine the best set of features. The results showed that the SVM model performed the best with an overall accuracy of 99.8%.

(Takiyama et al., 2018) leveraged on the transfer learning of GooLeNet architecture that is comprised of a total of 22 layers using the Caffe Deep Learning framework. The network aimed to classify and detect 6 different anatomical locations of the GI Tract by analysing Esophagogastroduodenoscopy imaging. The authors paid emphasis on the detection of 3 classes which were lower, middle, and upper regions of the stomach to further diagnose the patient with gastrointestinal diseases. The model was trained on approximately 20000 images and managed to achieve an overall AUC of 99% detection for the upper, middle, and lower stomach. The model was trained using an Adam optimizer with a learning rate of 0.0002. Input images were sized at 244 x 244 x 3 pixels.

(X. Liu et al., 2018) developed 3 different state of the art networks of VGG-16, Inception-V3 and InceptionResNet-V2 models using transfer learning fine tuning to analyse and classify different narrow-bank imaging (MNBI) endoscopy images into 2 different classes of early-gastric cancer (EGC) vs non-early gastric cancer. The authors also analysed the effects of different parameters of training such as dataset size, CNN architecture used, number of CNN model layers to fine tune during training and the effect of input size on the performance of the model. The proposed solution was also compared to traditional Feature Extraction + Machine Learning models. Their results showed that model with the best accuracy, sensitivity and specificity was from the Inception-V3 model with a 98.5%, 98.1% and 98.9% score for accuracy, sensitivity, and specificity respectively. From their experiments, the authors observed that the models performed better when fed a combination of both coarse + fine features as compared to only fine features (where the image is processed and smoothed prior). The authors also observed that unfreezing and training more layers of the transfer learning state of the art models actually led to better model performance for the studied problem. Results also showed that a bigger input image size of around 224 x 224 x 3 pixels and 299 x 299 x 3 pixels produced much better results in terms of accuracy and sensitivity as compared to 80 x 80 x 3 pixels. Lastly, when comparing CNN networks to Feature Extraction + Machine Learning, results showed that the transfer learning CNN models performed significantly better. The Feature

Extraction + Machine Learning approach used LBP, CLBP, Gabor and GLCM features used together with an SVM model.

(B. Liu et al., 2018) developed an automatic detection framework for tumours in gastric pathology images based on the ResNet-50 network architecture. In order to mitigate the issue of a lack of data, the authors used transfer learning. The input image size used was 224 x 224 x 3 pixels. The main evaluation metric used was F1 score. The individual ResNet-50 network was capable of reaching 95.5% F1-score. To increase its performance, the authors generated 5 different sets of training data to train 5 different sets of ResNet-50 networks and used majority voting to finalize the results. This managed to increase the F1 score of the solution to 96%.

Detecting pre-cancerous lesions is very helpful in preventing its progression into GI tract advanced cancer at later stages. In order to automate the accurate detection of pre-cancerous lesions in the GI Tract, (X. Liu et al., 2020) explored and studied various CNN architectures and their training configurations to classify between 3 classes of, Chronic Gastritis, Low Grade Neoplasia and Early Gastric Cancer. A variety of state-of-the-art network architectures of VGG-16, InceptionNet-V3, InceptionResNetV2 and ResNet-50 were used. The authors performed 3 configurations of experiments in which Experiment 1 compared the performance of CNN architectures amongst themselves as feature extractors. Results from Experiment 1 showed that ResNet-50 network performed with the best results. InceptionResnet-V2 had the largest model size as compared to the rest and the Inception-V3 model had the fastest prediction time as compared to the rest of the models. In Experiment 2, the ResNet-50 network was used as feature extractor and the final feature layer was then fed into an SVM. As comparison, a manual Feature Extraction + Machine Learning model solution using linear texture was also developed. The features used were LBP, ULBP, Gabor Filter and GLCM which were then fed into an SVM classifier model. Results from Experiment 2 showed that using CNN as the feature extractor performed significantly better as compared to manual traditional feature extraction method, where the CNN was capable of achieving an overall average accuracy of more than 90% as compared to the manual traditional feature extraction method that only managed to achieve an accuracy of between 50% to 70%. In the final Experiment 3, the authors studied the performance of the ResNet-50 and VGG-16 networks with and without transfer learning. Results showed that the state-of-the-art models trained with transfer learning achieved far superior results of 96% for ResNet-50 and 89% for the VGG-16 network. Results of training

these network from scratch showed that their performance dropped significantly to 25% and 34% for ResNet-50 and VGG-16 respectively. Each of the CNN networks were trained with an input image size of 224 x 224 x 3 pixels for 50 epochs with a batch size 32 images. The optimizer used was SGD, with a learning rate of 0.0001, with a momentum of 0.9.

(Demir et al., 2019) used transfer learning on skin cancer images for early-stage cancer detection. The output classification results was a binary classification of Malignant vs Benign. The authors employed transfer learning using state of the art models of Resnet-101 and Inception-V3 Deep Learning Models. The input images were set at size 224 x 224 x 3 pixels and were trained for 60 epochs at a learning rate of 0.001. The learning rate used was dynamic in the sense that it reduces after each epoch. Results showed that the ResNet-101 model was capable of achieving an overall accuracy of 84.09% and the Inception-V3 model achieved an overall accuracy of 87.42%.

(Saric et al., 2019) compared the performance of VGG-16 architecture and ResNet-50 architecture in detecting lung cancer by analysing histopathological slides of images to aid and speed up the work process for pathologists. The two networks were trained for a binary classification problem using the two aforementioned architectures with the SGD optimizer and with a batch size of 16. Learning rate of the model was set at 0.0001 using binary cross entropy as the loss function. Both models were trained for 17 epochs and evaluated using ROC and accuracy. AUC results showed that the VGG-16 model outperformed the ResNet-50 model. However, in terms of accuracy, the ResNet-50 model outperformed the VGG-16 model having achieved an accuracy of 75.2% vs 70.5% and 93% vs 91.2% for top 1% and Top 5% accuracy respectively.

2.3 Image Processing

Digital Image processing involves subjecting an image through a sequence of algorithms to transform the image from one form to another by manipulating the images using digital computers. By doing so, certain features or information from the image can be enhanced for extraction or highlighted to the final user. Image processing systems usually treats all images as 2D signals when applying certain signal processing methods. A specific domain of image processing algorithms that is very much focused on is for the use of image quality improvement. On a broad scale, this can be categorized into 2 categories which are image restoration

(commonly known as IFM-based solutions) and image enhancement (commonly known as IFM-free techniques). Image restoration works on the principal of recovering suppressed information from an image due to its external environments. These algorithms use mathematical models that represent the mechanics of light propagation through mediums to rebuild the restored image via inverse compensation processing, (Corchs & Schettini, 2010). This technique is more effective and rigorous but can be very detailed in terms of the required physics parameters which are not always readily available. A simpler alternative is image enhancement which is a more qualitative approach where mathematical algorithms do not require much detail. Instead, image enhancement works to manipulate the image to create more distinct features in the image and also to create a visually pleasing image. Image enhancement techniques mainly focus to improve contrast and colour of the images based on pixel intensity re-distribution without considering the dynamic details of the image or light parameters. These techniques manipulate the pixels directly or in a transformed domain. These methods are simpler and intuitive.

(Pannu et al., 2020) performed image pre-processing of histogram equalization and colour palette reduction using minimum variance quantization on WCE GI Tract images. By using image processing, the CNN network was able to improve its accuracy from 91% to 93% when tested on real live video images.

(Ayyaz et al., 2022) implemented a sequence of image processing techniques to enhance the raw images of the GI tract. The authors first attempted to extract dark features from the image using top-hat and bottom-hat filtering, followed by calculating the opening and closing of the image. The resulting image matrix was then subtracted from the original image. In order to reduce noise, the authors applied a 3D median filter which is a non-linear method of denoising an image whilst preserving the details of the original image. (Hmoud Al-Adhaileh et al., 2021) on the other hand applied averaging filter to GI tract images to enhance features within the image.

(Salem et al., 2019) studied and implemented 4 different histogram-based contrast enhancement algorithms to enhance retina, brain endometrium, breast, and knee medical images. These algorithms covered variants of histogram manipulation algorithms encompassing Histogram Equalization (HE), Cumulative Histogram Equalization (CHE), Quadrant Dynamic Histogram

Equalization (QDHE) and Contrast-Limited Adaptive Histogram Equalization (CLAHE). The quality of the enhanced images were compared to the raw image using evaluation metrics of Peak-to-Sound Noise Ratio (PSNR), Mean Squared Error (MSE) and Standard Deviation (SD). The authors noted that HE spreads out the pixel intensity values across the total range in the images to achieve highest contrast. CHE does this much better as compared to HE. QDHE is a brightness-preserving algorithm that reduces several limitations caused by HE. CLAHE on the other hand is best recommended for images with non-uniform intensity distribution across the image. Results showed that the standard deviation of the enhanced images were typically higher compared to the raw image indicating a broader pixel distribution. QDHE and CLAHE produced the least MSE. QDHE gave the highest PSNR for all types of images except retina images. CLAHE was proposed as the best enhancement technique for retina whilst QDHE was proposed as the best for brain, endometrium, breast, and knee images.

(Jintasuttisak & Intajag, 2014) applied Rayleigh CLAHE algorithm on retinal images to enhance the contrast. In their proposed solution, the authors performed the histogram stretching using the HSI colour model instead of RGB. The specific channel that was stretched was the Intensity (I) channel which controls the overall brightness of the image. The raw retinal images were converted into HSI colour model and the Intensity channel (I) was split into 2 different categories of under exposed and high exposed image which would be individually stretched using Rayleigh-CLAHE histogram stretching algorithm prior to recombining the images and finally reconverting the images back into RGB colour model-based image.

2.4 Summary

Out of the 21 literatures reviewed, a total of 19 were closely related to Deep Learning CNN solution-based solutions for medical related CAD. 18 of those literatures were related to classification and 1 was a regression-based network. 7 literatures worked on self-developing their own CNN architectures for the solution whereas the remaining 12 literature were incorporated transfer learning (10 GI tract related and 2 non-GI tract related). Out of the 18 literatures that were classification based, 12 literatures were binary class classification problems whereas 6 were multiclass classification problems. From this it can be observed that the majority of work in the subject domain is heavily focused on binary classification whereby the solutions aimed to detect the presence of a single disease, many of which were bleeding vs non-bleeding. Out of the 6 multiclass based solutions, only 2 were focused on the classification of

a variety of GI tract related diseases, another 2 were only focused on the classification of different types of polyp diseases and the final 2 were not focused on GI tract disease classification and instead revolved around other GI tract related matters such as location and motility of GI tract organs. This shows that there is a gap or less focus on developing a generalized CNN based CAD solution that focuses on different types of GI tract related diseases. Hence, this is identified as a potential area of further research.

With regards to the methodology, a total of 16 literatures pertaining to GI tract disease classification were identified. This is broken down further into 6 self-developed architectures and 10 architectures that incorporated transfer learning. Out of the 10, it is observed that a majority of studies incorporated transfer learning feature extraction models over fine tuning the models (6 feature extraction vs 3 fine tuning vs 1 feature extraction + fine tuning). It can be seen that feature extraction is commonly used as opposed to fine tuning. Moreover, only 1 literature so far has incorporated and compared the performance of both approaches. This provides a room for opportunity to test and compare the performance of transfer learning using feature extraction against fine tuning and comparing their performances against each other.

Only 10 out of the 19 architectures applied data augmentation with a majority of them being geometric based augmentation. 7 out of 10 works incorporated geometric augmentation and only 3 out of 10 applied texture or colour-based augmentation. This is another identified gap whereby there is a lack of research in studying how augmenting the image by manipulating the colour and texture of the image affects the performance of the developed CNN. Moreover, though geometric augmentation may vary the images that is created for the model to learn, it's effects in inducing variability in the data is not as significant as compared to colour and texture-based augmentations.

The three points discussed above highlights the lack of available annotated data covering multiple disease classes. It is observed that a majority of the datasets used are binary class-based datasets. Furthermore, a lot of the datasets that have multiple classes of disease images are also small in size and are taken from a single source i.e., a single WCE camera from the similar patient. There is lack of variety in terms of the types, resolution and generalization of images being used which may results in a not so generalized CNN solution network being developed for real life application. Some of the datasets are also imbalanced. This is also another identified gap whereby, a generalized CNN solution can be developed by sourcing

dataset images more efficiently from different sources of cameras to get varying types of images to train the CNN network.

With regards to input image pre-processing and image quality enhancement, only 3 out of the 19 reviewed literatures performed image pre-processing. Out of the 3 literatures, 2 performed a simple texture enhancement via simple filtering Gaussian or Laplacian filtering and 1 literature performed contrast-based enhancement using global histogram stretching. It is observed that not much work and research has been devoted into studying how more complex algorithms such as colour manipulation, other contrast enhancement techniques and illumination enhancement can further enhance the quality of the input images to help it in improving the performance of the CNN. This is another identified gap that can be contributed to.

With regards to training and solution workflows, a variety of techniques were introduced besides the standalone CNN. There are works that develop an ensemble of CNN models and averaged their results to generalize the solution better. Some proposed solutions even combined CNN with Machine Learning models using the CNN model as a feature extractor only. A few works proposed 2 stage of networks whereby 2 CNN networks are used to run predictions in 2 different stages. There are also proposed solutions that employ 2 stage training whereby the CNN model is trained on 2 different datasets in stages. Several authors also proposed using additional image channels to the CNN architecture instead of the conventional 3 RGB channels. Lastly, some authors also incorporated optimization algorithms to select the best features from a CNN model prior to feeding the features into the Machine Learning model. These proposed workflows of solution development serve as good reference for the development of the project plan and methodology.

Other observations made such as network architectures, state of the art networks used for developing the solution, network training parameters, evaluation parameters and input image sizes were also analysed and considered as reference to develop the CNN architectures used in this study.

A summary of the reviewed works are detailed in Appendix A.

CHAPTER 3

METHODOLOGY

This chapter outlines the detailed methodology of the study. Specifically, the methodology details out the major process flows undertaken during the project with justification of each major workflow. The main evaluation approach taken in this study is quantitatively focused. The entire project took approximately 4.5 months to complete. A high-level framework of the project is detailed in Figure 3 below. The project is broken down into 5 main stages of:



Figure 3: Project High Level Framework

3.1 Project Definition and Project Proposal

The project began with exploratory research on the subject domain to familiarize with the topic and present works. Following exploratory literature review, identified gaps in the body of were determined and further evaluated for better understanding and to streamline the problem statement. Once the problem statement, aim, objectives and project scope were defined, a project proposal was prepared and presented for approval.

3.2 Detailed Research and Project Framework Planning

The research phase of the project is broken down into 2 sections of

- Dataset sourcing
- Solution development research.

An overview of the methodology for research and planning is shown in Figure 4 below.

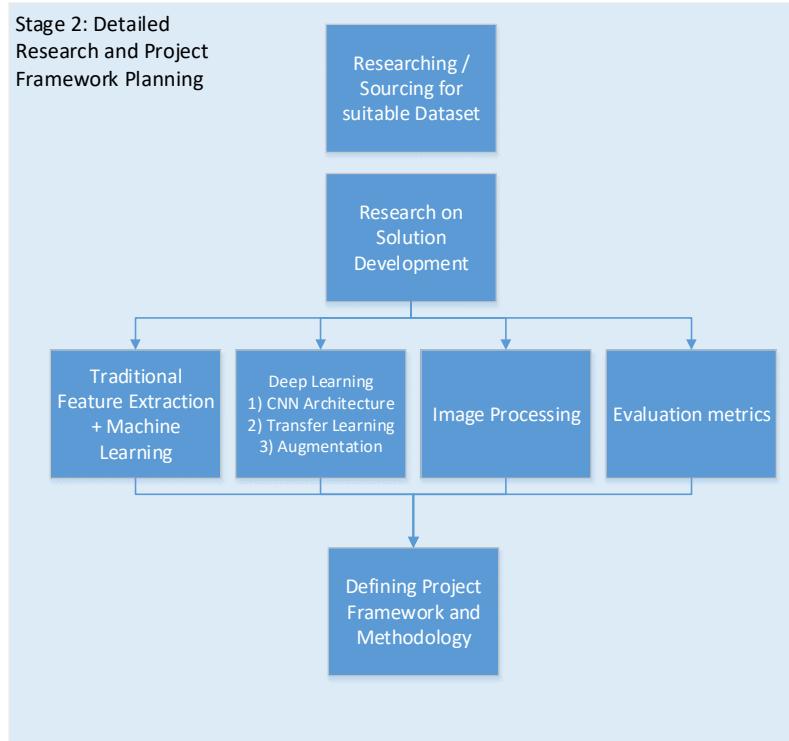


Figure 4: Detailed Process Flow of Stage 2.

Prior to sourcing for the dataset, an overview of the subject domain is first carried out. This covered more on the medical aspects of the GI Tract system, its functionality and diseases associated. This gave a better understanding on the significance of work being carried out and to better comprehend the problem that will be solved.

Subsequently, a suitable dataset was sourced. Several criteria were set as basic requirements to choose the dataset. Firstly, the dataset had to be a multiclass dataset i.e., comprising of more than 2 class. Secondly, the sourced dataset images were a combination of images from different dataset sources i.e., the images were taken from different cameras and different patients. Thirdly, the size of the images had to be sufficiently large i.e., the size should be at least 300 x 300 pixels and above to ensure that the quality of details and information in the images suffice. Moreover, it was also ensured that the images were also of RGB images channel and not only a grayscale. The number of images in each class had to be sufficient and a minimum number of 1000 images was set as a basic required quantity for each class. A preference was set such that there were a balanced or almost balanced number of images for each class for training.

In the research development stage, the process was broken down into 3 main methodologies of solving the problem i.e., Feature Extraction + Machine Learning models, self-developed CNN architectures and transfer learning-based architectures. Concurrently, detailed study on augmentation techniques were also explored to better understand and learn how synthetic data can help improve the overall performance of the solution. As one of the main objectives is to study the effect of image processing on the raw image, research was also carried out on various image processing enhancement techniques to improve the quality of the raw image. Last but not least, research was carried out with respect to evaluation metrics to properly evaluate the proposed solutions and to finally streamline and select the best solution. The detailed literature review of these sections have been detailed in CHAPTER 2.

Finally, upon a clear understanding of the subject, problem and present work in the field, a detailed methodological framework was defined to implement for the current project.

3.3 Detailed Implementation

Figure 5 depicts the overall high-level implementation methodology flowchart defined for this project and is broken down in 7 stages.

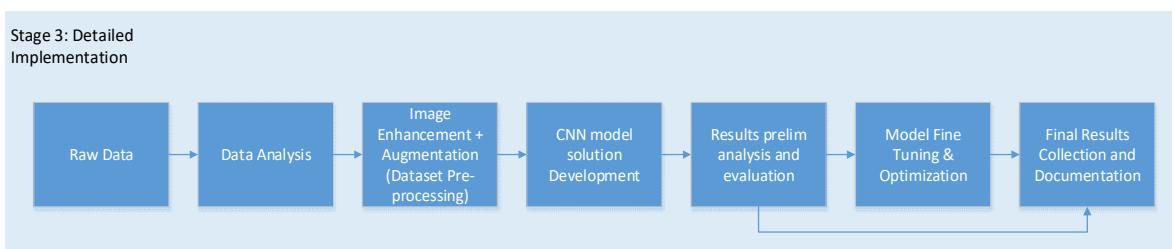


Figure 5: Detailed Process Flow of Stage 3

The implementation began with the sourcing and high-level overview analysis of the selected dataset. Following which, different image pre-processing enhancements were performed on the dataset images to create different sets of data used to augment, train, validate, and test the performance of the proposed CNN solutions. Details of this is detailed in Section 3.3.3 below.

Upon pre-processing, the step that followed was the development, training, validation, and testing of CNN classifiers. A total of 91 CNN architectures were developed and tested throughout the project across 2 stages. For each stage, the results of the developed CNNs were

evaluated, compared, and further optimized via various hyperparameter tuning methods to select the best model in that stage. Details of this is explained in Section 3.3.4. Cross stage model performance comparison was also performed.

The results from each experiment were recorded for evaluation, interpretation, comparison, and final documentation.

3.3.1 Dataset

The dataset was sourced from Kaggle titled WCE Curated Colon Disease Dataset Deep Learning (MONTALBO, 2022). The dataset is comprised of WCE images from a combination of 3 dataset sources which are (Montalbo, 2022; Pogorelov et al., 2017; Silva et al., 2014). The dataset is comprised of 4 different classes which are Normal, Ulcerative Colitis, Polyps and Esophagitis. The dataset is comprised of training, validation, and testing sets. The training dataset has 1050 images for each class totalling the overall training dataset size to 4200 images. The validation set is made up of 150 images for each class totalling up to 600 images. The test set is made up of 300 images for each class totalling up to 1200 images. The quantity of images for all classes in all 3 sets are balanced. The images vary in terms of width and height.

3.3.2 Dataset Preparation and Analysis

Initial data analysis was then performed once the different sets of data. The purpose of the initial data analysis step is to first visualize the data and have a better understanding of it. The process workflow is depicted in Figure 6 below.

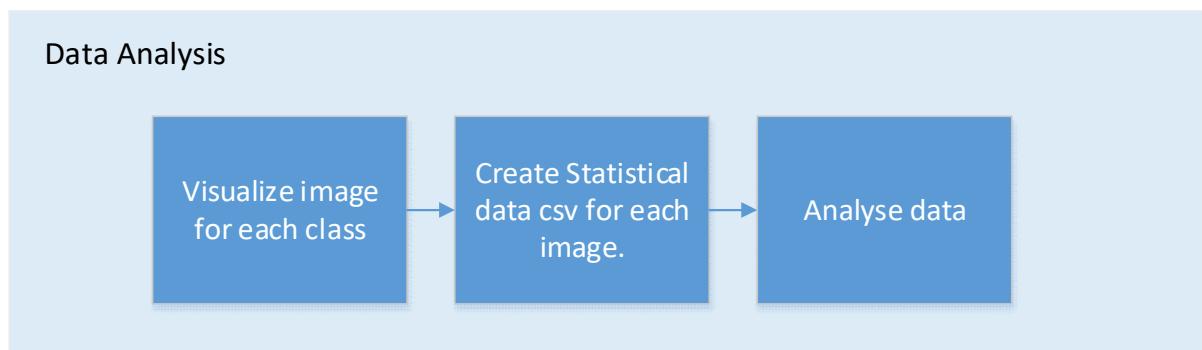


Figure 6: Detailed Process flow of Data Analysis from Stage 3.

A CSV format table compiling statistical data of every image in the dataset was compiled. The contents of the CSV format table include image information such as the image_id, image_file_location, class, width, height, number of channels and the average individual R, G and B values. This was done to summarize and visualize the data in terms its quantity, size and balance between each class. The average R, G and B values for each image was also computed and visualized to better understand the colour channels' distribution for each class. This step was implemented for all 3 train, test and validation sets.

3.3.2.1 Visualization of Raw Images

Figure 7 portrays samples of images from each class.

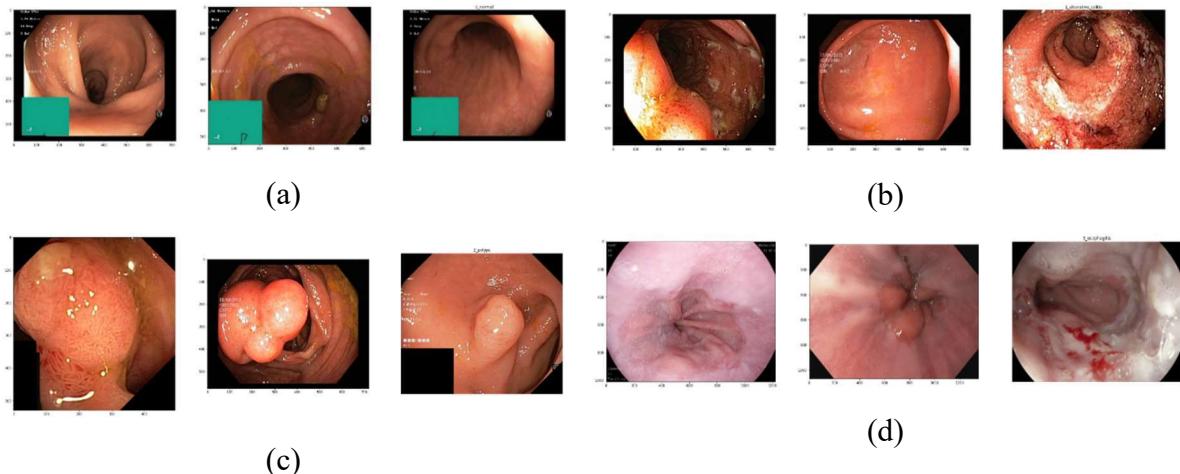


Figure 7: Visualization of each image class. (a) 0_normal, (b) 1_ulcerative_colitis (c) 2_polyps and (d) 3_esophagitis

From a high-level observation, the 0_normal class images look relatively clean and normal as compared to the other classes. For images of 1_ulcerative_colitis a significant number of white lesions are observed in the images. These are the ulcers present within the GI Tract. Image of 2_polyps have abnormal growths along the GI Tract. Lastly, 3_esophagitis images tend to be pale and constricted which can be attributed to the inflammation of the GI Tract.

3.3.2.2 Analysis of size distribution of images

Table 1: Analysis of Input Image Size Distribution

class	Width (pixels)				Height (pixels)			
	mean	median	min	max	mean	median	min	max
0_normal	686.34	720	459	720	559.05	576	530	576
1_ulcerative_colitis	658.81	627	482	1920	558.13	532	447	1080
2_polyps	652.77	622	332	1920	556.04	531	352	1072
3_esophagitis	1256.28	1225	613	1920	998.21	1015	523	1072

Based on analysis, the images range from 332 pixels min to 1920 pixels max with an average pixel width of around 650 pixels and 550 pixels for height. The largest images are from class 3_esophagitis with an average width of 1200 pixels and average height of 990 pixels.

3.3.2.3 Analysis of Distribution of Pixels

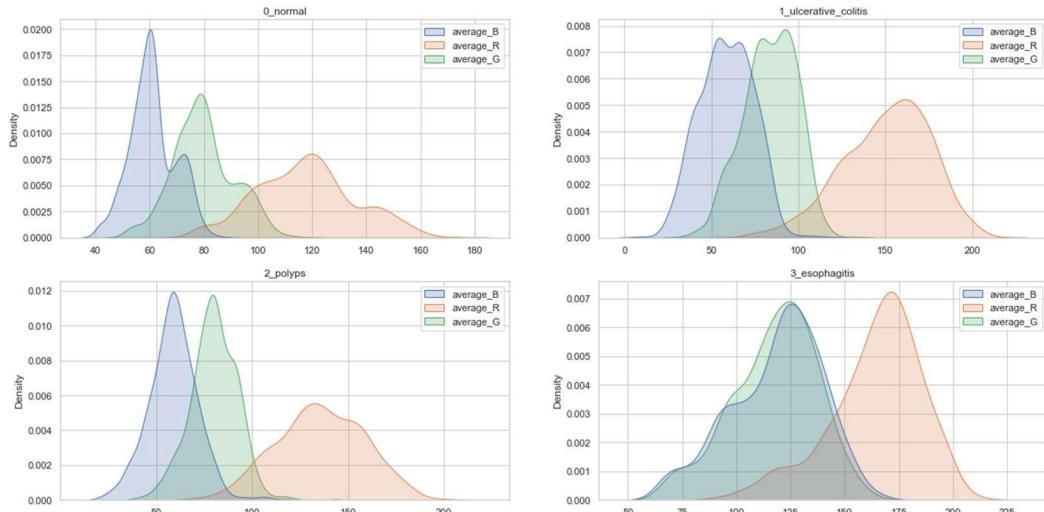


Figure 8: Average Pixel Distribution for each Image Class.

The pixel distribution plot above shown in Figure 8 was obtained by plotting average R, G and B pixel values distribution across all images segregated by each class. This was done to observe the spread and distribution of pixels within each channel.

From the distribution plot shown above, it is observed that the intensity of the Red (R) channel is highly concentrated on the high end of the spectrum range, followed by Green (G) and lastly Blue (B) and this is observed across all classes except class 1_ulcerative_colitis. Across the board, it can be observed that the pixel intensities of all 3 channels are arranged in the order of class 3_esophagitis, 1_ulcerative_colitis, 2_polyps and lastly 0_normal. Each class has its own distinct pixel distribution plot except 1_ulcerative_colitis and 2_polyps that share somewhat similar pixel intensity distribution.

3.3.3 Image Enhancement and Augmentation (Dataset Pre-processing)

Image processing stage is broken down into two major workflows, one is in the form of performing dataset pre-processing and two is in the form of data augmentation. Image Enhancement and Augmentation were performed using Python3 Jupyter notebook as well as Python scripts on PyCharm on a local machine.

In workflow 1, three different image processing algorithms were used to pre-process the raw images. This simulates the situation whereby all the raw images that are being fed to the algorithm as input are initially pre-processed prior. For reproducibility and for ease of use during training, 3 additional sets of datasets were created. Each dataset would contain training, validation, and test sets with the distinction that all images in all 3 of these subsets of data would be subjected to a particular image processing algorithm.

Workflow 2 on the other hand is aimed at performing augmentation to the dataset for the CNN model to learn from. In the augmentation workflow, only the original training data is augmented from the original dataset. The validation and test sets are left untouched. The 3 different image processing algorithms are applied to the training images and added on top of the original data. This augmentation process creates a variety images that have been altered in terms of texture, colour and contrast simulating new images from a new WCE camera source. This aims in aiding the model to learn more generalized features from different images and to be better adapted for real life applications.

This would result in a total of 5 datasets (original + 3 image processed datasets + 1 Augmentation dataset) that will be used for developing the CNN models. These datasets were

saved in different folders and were used accordingly throughout the solution development phase. The overall pathway process of image processing is defined in Figure 9 below.

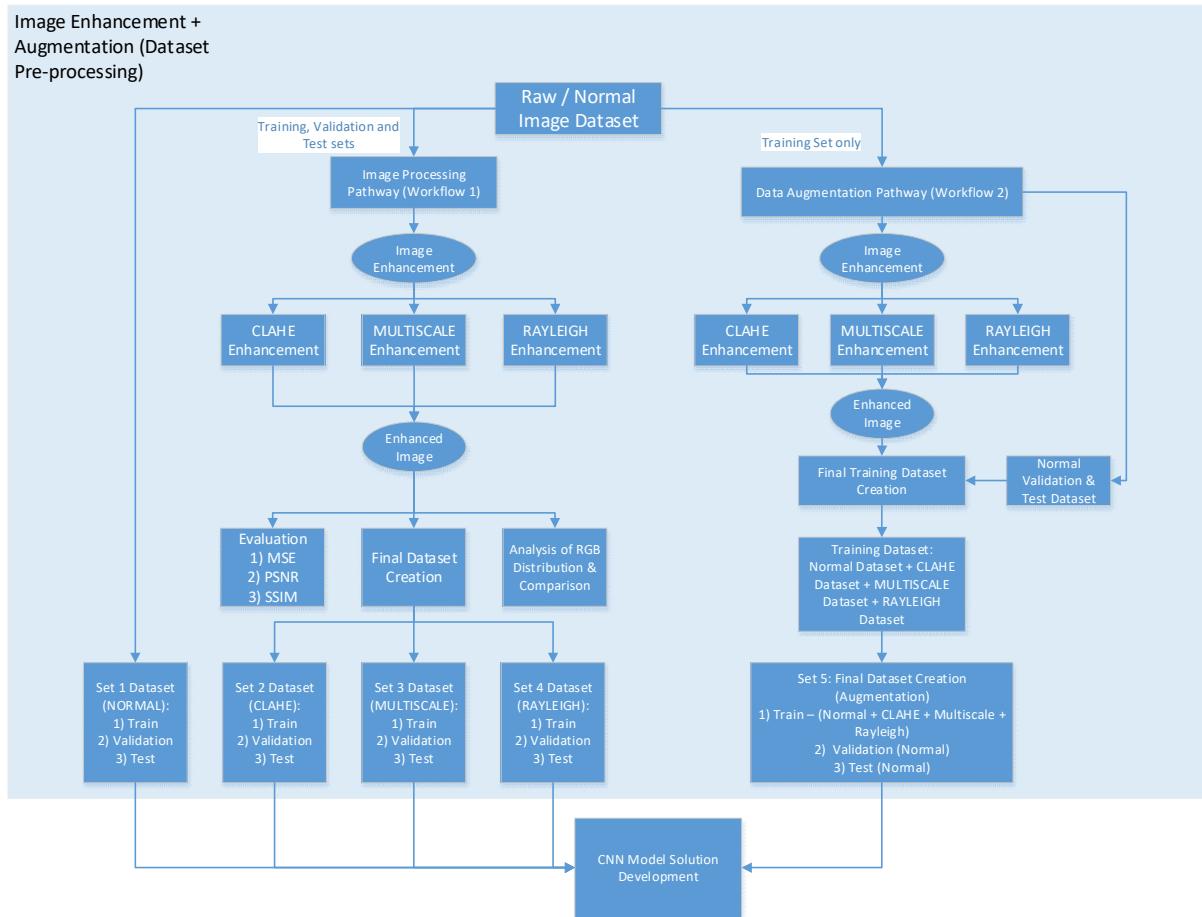


Figure 9: Detailed Process flow of Image Enhancement + Augmentation (Dataset Pre-processing) from Stage 3.

The 3 image enhancement techniques considered in this study are denoted as CLAHE, MULTISCALE and RAYLEIGH techniques. The foundation of these 3 methods are tied back to the manipulation of the histogram of the images to improve contrast and image quality with other processes involved in between.

For a high-level evaluation, the processed image techniques were evaluated using metrics of Mean Squared Error (MSE), Peak to Sound Noise Ratio (PSNR) and lastly Structural Similarity (SSIM). Essentially, these evaluation metrics measure the difference of changes between the raw image and the processed image to estimate the quality of the image enhancement

algorithms. As the size of the dataset is large, an average of these values across the different images in the training set was taken as the final evaluation criteria. Sections 3.3.3.1 to 3.3.3.3 below, details the process of these algorithms.

A summary of the datasets created post image processing is shown in Table 2 below

Table 2:Summary of Datasets Set 1 - 5.

Set	Training Qty	Validation Qty	Testing Qty	Remark
Set 1	4200	600	1200	Training, Validation and Test are UNPROCESSED
Set 2	4200	600	1200	Training, Validation and Test are processed using CLAHE enhancement
Set 3	4200	600	1200	Training, Validation and Test are processed using MULTISCALE enhancement
Set 4	4200	600	1200	Training, Validation and Test are processed using RAYLEIGH enhancement
Set 5	16800	600	1200	Training (UNPROCESSED + CLAHE + MULTISCALE + RAYLEIGH) Validation (UNPROCESSED) Testing (UNPROCESSED)

3.3.3.1 CLAHE Theory

Contrast Limiting Adaptive Histogram Equalization (CLAHE) is an extension to the Global Histogram Equalization. Global Histogram Equalization is one of the earliest contrast enhancement methods developed. It works to essentially spread out the intensity range of the image. However, the algorithm is very general, and a lot of information is typically lost in the images after contrast enhancement. This is attributed to the fact that the histogram stretching is naively done across the entire image and does not consider other additional parameters of the image. Its stretching is also not sensitive to the colour distribution of pixels in different regions of the same image. Different regions may not have different spectrum of pixel distribution that contribute to the features within its region.

An alternative method is to divide the entire image into smaller local blocks of a particular size and perform region specific Global Histogram Equalization. By doing so, the pixel distribution is well confined to that area itself and therefore is less affected by noise. However, if noise is present within the confined region, it will then instead be amplified. To overcome this problem, a contrast limit value is applied whereby any histogram bins above the specified values is clipped and distributed evenly to other bins prior to equalization of the histogram. Bilinear interpolation is then applied to remove artificial boundaries at the border of the tiles. This process is known as CLAHE (Amro, 2022). The process flow of CLAHE on the other hand is depicted in Figure 10.

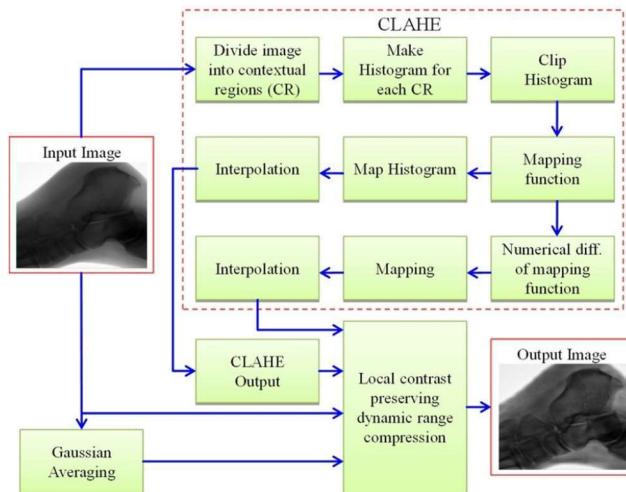


Figure 10: Detailed Process Flow of CLAHE Image Enhancement (J. Lee et al., 2015).

CLAHE was implemented using the `create CLAHE` function from the Python CV2 library. The clip limit for each colour channel was set using Equation 1 below.

$$\text{ClipLimit} = 0.002 * \left(\frac{\text{Width} * \text{Height}}{\text{NumPixelBins}} \right)$$

Equation 1

3.3.3.2 Rayleigh Theory

The Rayleigh histogram stretching enhancement technique is inspired by authors of (Ghani & Isa, 2015). In their proposed work, the authors used the technique for enhancement of features of underwater images. The principal of the enhancement technique combines the principal of image contrast enhancement for better feature prevalence and also for the principal of colour correction to improve the overall quality of the input image. Though the algorithm was initially developed for underwater images contrast enhancement, its foundational working principal makes it flexible and suitable as well to be applied on medical images with several parameter modifications required. Hence with several modifications being made to the algorithm, it was then re-developed for GI tract images enhancement.

In overall, the enhancement process can be broken down into two main phases of firstly, contrast enhancement which is then followed by colour enhancement. The overall process flow of the image processing enhancement technique is depicted in Figure 11 below.

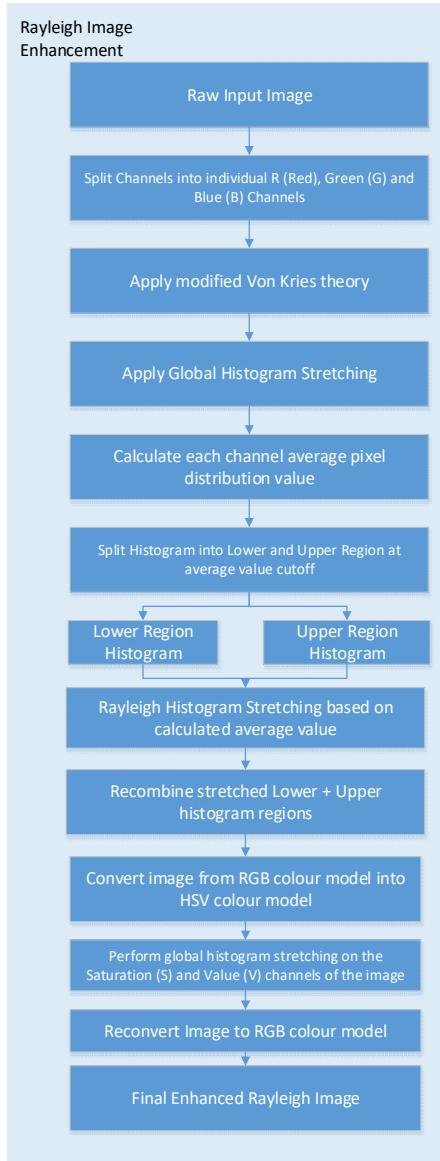


Figure 11: Detailed Process Flow of Rayleigh Image Enhancement

The process flow shown in Figure 11 above was applied to each and every image in the dataset. The first steps involve splitting the image into individual the R, G and B channels as the subsequent histogram manipulation steps involved are performed on individual channels. Subsequently for each channel, the modified Von Kries theory was applied (Iqbal et al., 2010). The modified Von Kries theory takes inspiration of the original Von Kries theory that computes 2 gain factors to be multiplied with the dormant channels of the image. The gain factors are calculated from the dominant colour channel from the image to balance the intensity of illumination for the other colour channels. What differs in the modified Von Kries theory is that

instead of using the dominant colour channel as the basis for gain calculation, the modified Von Kries theory instead considers the median value intensity of the 3 channels as basis when calculating the gain for the other 2 channels. Firstly, the average intensity values for each of the RGB channels were calculated using the Equation 2 below:

$$IC_n_{avg} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H I_n(i,j)$$

Equation 2

Table 3: Abbreviations for Equation 1

Abbreviation	Meaning
IC_n_{avg}	R, G and B channel pixel average intensity respectively for the image
W	Width of image
H	Height of image
$I_n(i,j)$	Pixel intensity at position (i,j)

Following which, the median value channel intensity was determined from the 3 calculated average values to compute the gains. The formula for calculating the gains A and B are shown in Equation 3 and Equation 4 below.

$$A = \frac{median(R_{avg}, G_{avg}, B_{avg})}{min(R_{avg}, G_{avg}, B_{avg})}$$

Equation 3

$$B = \frac{median(R_{avg}, G_{avg}, B_{avg})}{max(R_{avg}, G_{avg}, B_{avg})}$$

Equation 4

Gain A was multiplied with the channel with the lowest average intensity value whereas gain B on the other hand was multiplied with the channel with the highest average intensity value. Subsequently, global histogram stretching was performed on each individual R, G, and B channels where the spread of pixels was stretched over the range of 0 – 255 pixels to provide better pixel distribution of the image. Equation 5 below was used to perform global histogram stretching.

$$P_{out} = (P_{in} - I_{min}) \left(\frac{O_{max} - O_{min}}{I_{max} - I_{min}} \right) + O_{min}$$

Equation 5

Table 4: Abbreviations for Equation 2 and Equation 3.

Abbreviation	Meaning
P_{out}	Final output pixel of image after contrast stretching at any point (i,j) in the image
P_{in}	Initial input pixel value at any point (i,j) in the image
I_{min}	Minimum pixel intensity value of the original input image
I_{max}	Maximum pixel intensity value of the original input image
O_{min}	Minimum desired pixel intensity value of the output image
O_{max}	Maximum desired pixel intensity value of the output image

The histogram for each colour channel were then split into lower and upper regions histogram. The cut-off pixel points in which this was done was the average value for each histogram channel as calculated in Equation 5 above. Based on the calculated average pixel value for each channel, the histogram was then split into lower and upper region. The range of values for the lower region is 0 – average_value whereas for the upper region, the value ranges from average_value – 255. This is done to individually stretch both the lower and upper regions of the histogram. The histogram stretching performed this time is not an overall Global Histogram Stretching and instead was stretched to follow a Rayleigh distribution. This would then produce 2 different set of histograms for the lower and upper region stretched across 0 – 255 pixels

(Hitam et al., 2013). These two histograms would produce images with 2 different contrasts where the lower region histogram produced an under enhanced image and the higher region histogram on the other hand produces an over enhanced image. The formula used to perform this process is depicted in Equation 6 below.

$$\text{RayleighPixel}_{out} = \frac{255(P_{in} - I_{min})}{\alpha^2 (I_{max} - I_{min})} * e^{\frac{-[255(P_{in}-I_{min})]^2}{[2\alpha^2(I_{max}-I_{min})]^2}}$$

Equation 6

Table 5: Abbreviations for Equation 5

Abbreviation	Meaning
RayleighPixel _{out}	Final output pixel of image after Rayleigh contrast stretching at any point in the image
P _{in}	Initial input pixel value at any point in the image
I _{min}	Minimum pixel intensity value of the original input image
I _{max}	Maximum pixel intensity value of the original input image
α	The distribution parameter of Rayleigh distribution

Next, the lower and upper region histograms were recombined to form a final single image histogram for each channel. As both the lower and upper stretched regions values spanned across 0 – 255, these 2 histograms were recombined using averaging whereby the pixel values from the lower and upper histogram were averaged. The 3 channels of processed image were then recombined to form the RGB image.

Finally, in phase 2, colour enhancement on the image was performed. This was done by performing global histogram stretching on the Saturation (S) and Value (V) channels of the image which basically controls the image brightness. The first step in this process was to convert the RGB channel image into HSV format. Global histogram stretching was then applied to both the S and V channels of the image limited to the top and bottom 1% of the image histogram range. This diminished the effects of over and under saturated-ness in the enhanced

image. Post stretching the S and V channels, the image was then reconverted back to its RGB colour channel format to produce the final enhance Rayleigh Image.

3.3.3.3 MULTISCALE Theory

The Relative Global Histogram stretching enhancement technique, hereinafter referred to as the Multiscale technique was inspired by the authors of (Huang et al., 2018). The image enhancement process is similar to that of used in Rayleigh with a slight modification in the way the histogram is stretched, and the proposed colour model used in the study. Similar to that of Rayleigh, the Multiscale method also employs a 2-phase approach of first contrast stretching followed by colour adjustment. The process flow of the entire process is depicted in Figure 12 below.

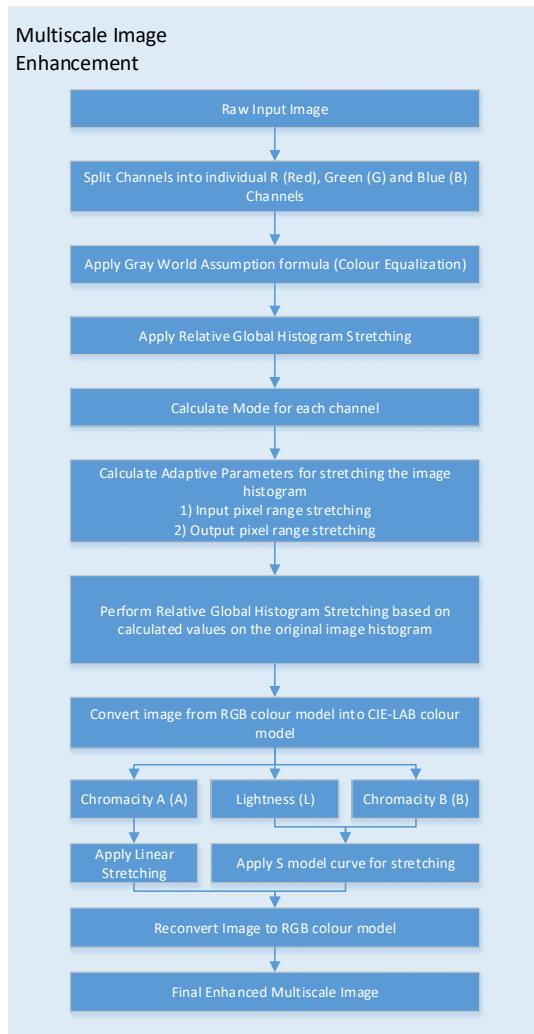


Figure 12: Detailed Process Flow of Multiscale Image Enhancement

The process flow shown in Figure 12 above was applied to each and every image in the dataset. Firstly, the 3 RGB channels were split into individual channels of R, G and B. Instead of applying the modified Von Kries algorithm on the image, the Gray World theory was instead applied to equalize the colours on the channels. The Gray world theory assumes that the average of all 3 channels in the image should be 0.5 as depicted in Equation 7 below.

$$\frac{R_{avg} + G_{avg} + B_{avg}}{3} = 0.5$$

Equation 7

Each pixel in the channels were multiplied with the colour channel equalization coefficient which was computed using the Equation 8 below.

$$\theta_{icn} = \frac{0.5}{IC_n_{avg}}$$

Equation 8

Table 6: Abbreviations for Equation 7.

Abbreviation	Meaning
θ_{icn}	Colour channel equalization coefficient for channels R, G and B
IC_n_{avg}	R, G and B channel pixel average intensity respectively for the image calculated using Equation 2 i.e., R_{avg} , G_{avg} , B_{avg}

In the following step, the Relative Global Histogram stretching was performed as opposed to the traditional Global Histogram stretching. Taking into consideration that each colour channel has different ranges of pixel intensities, histogram stretching should be channel sensitive in order to cater to specific channel pixel distribution. Hence, instead of directly using the minimum and maximum input and output pixel values as per Equation 5 above, these values were instead specifically computed for each colour channel using Equation 9 to Equation 14 below:

$$I_{\lambda min} = P.sort[P.sort.index(a) * \frac{0.1}{100}]$$

Equation 9

$$I_{\lambda max} = P.sort[-(P.len - P.sort.index(a)) * \frac{0.1}{100}]$$

Equation 10

Table 7: Abbreviations for Equation 8 and Equation 9

Abbreviation	Meaning
$I_{\lambda min}$	Minimum pixel intensity value of the original input image. λ represents each channel i.e., RGB
$I_{\lambda max}$	Maximum pixel intensity value of the original input image
P	Set of image pixels present in an image
$P.sort$	Sorted of image pixels set in ascending order
$P.sort.index(a)$	Index number of the mode in the histogram distribution
$P.sort[X]$	Value of image pixels at index X of the ascending sorted data

$$O_{\lambda min} = a_{\lambda} - \sigma_{\lambda}$$

Equation 11

$$\sigma_{\lambda} = 0.655a_{\lambda}$$

Equation 12

$$O_{\lambda max} = \frac{a_{\lambda} + \mu_{\lambda} * \sigma_{\lambda}}{\kappa * t_{\lambda}}$$

Equation 13

$$\frac{\kappa * t_\lambda * I_\lambda}{\sigma_\lambda} \leq \mu_\lambda + 1.526 \leq \frac{\kappa * t_\lambda * 255}{\sigma_\lambda}$$

Equation 14

Table 8: Abbreviations for Equation 10 to Equation 13

Abbreviation	Meaning
$O_{\lambda min}$	Minimum desired pixel intensity value of the output image. λ represents each channel i.e., RGB
$O_{\lambda max}$	Maximum desired pixel intensity value of the output image
I_λ	Pixel intensity value of the original input image. λ represents each channel i.e., RGB
a_λ	Mode pixel value of the channel
σ_λ	Standard deviation of the Rayleigh Distribution
κ	Energy ratio factor
t_λ	Residual energy ratio
μ_λ	Output pixel coefficient

Upon performing Relative Global Histogram stretching, the 3 separate modified R, G and B channels were then recombined and converted into the CIE-LAB format colour channel. The L represents lightness whereas A and B represents different parameters of chromaticity in the image. The histograms of the L channel were stretched using linear stretching whereas the A and B channel histograms were stretched using the S curve model formula shown in Equation 15 below.

$$P_\chi = I_\chi * (\varphi^{1 - \left| \frac{I_\chi}{128} \right|})$$

Equation 15

Table 9: Abbreviations for Equation 14

Abbreviation	Meaning
P_χ	Final output pixel of image after S-curve contrast stretching at any point in the image
I_χ	Pixel intensity value of the original input image. χ represents each channel i.e., a and b
φ	Stretching coefficient

Post stretching, the 3 enhanced channels were the reconverted back in RGB channel-based image to produce the final processed Multiscale image.

3.3.3.4 Evaluation Metrics

As a preliminary evaluation, the 3 different image processing enhancement techniques were evaluated using perceptual metrics of Mean Square Error (MSE), Peak-to-Sound Noise Ratio (PSNR) and Structural Similarity Index (SSIM). These 3-evaluation metrics are generally used to gauge quantitatively gauge the performance of image enhancement techniques.

MSE was computed by taking the cumulative squared error between the enhanced and original image. It was calculated by the Equation 16 below.

$$MSE = \frac{1}{W \times H} \sum_{w,H} [I_{original}(w, h) - I_{enhanced}(w, h)]^2$$

Equation 16

Table 10: Abbreviations for Equation 15

Abbreviation	Meaning
MSE	Calculated Mean Squared Error
$W(w)$	Width of the entire image (coordinate w in width direction)
$H(h)$	Height of the entire image (coordinate h in height direction)
I_x	Pixel intensity at coordinate (w,h)

PSNR on the other hand gives an indication of the ratio of the maximum possible signal to noise. It was calculated using Equation 17 below.

$$PSNR = 20 * \log_{10} \frac{(2^B - 1)}{\sqrt{MSE}}$$

Equation 17

Table 11: Abbreviations for Equation 16

Abbreviation	Meaning
PSNR	Calculated Peak-to-Sound Noise Ratio
B	Bits per sample

Structural Similarity quantifies the degradation to an image that is cause by image processing. SSIM was based on the visible structures of the image. The formula for SSIM is defined as Equation 18 below.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Equation 18

Table 12: Abbreviations for Equation 17

Abbreviation	Meaning
SSIM	Calculated structural similarity index
μ_x	Local mean of original image
μ_y	Local mean of enhanced image
σ_x	Standard deviation of original image
σ_y	Standard deviation of enhanced image

σ_{xy}	Cross-covariance for original and enhanced image
C_1	Regularization constant for luminance
C_2	Regularization constant for contrast

3.3.4 CNN model solution Development

Once image processing enhancement was performed and the relevant datasets were created, the following steps entailed developing the Deep Learning CNN solution. A high-level process flow of the entire CNN solution development phase is shown in Figure 13 below. A detailed process flow for Stage 1 and Stage 2 can be referred to in Appendix B and Appendix C:

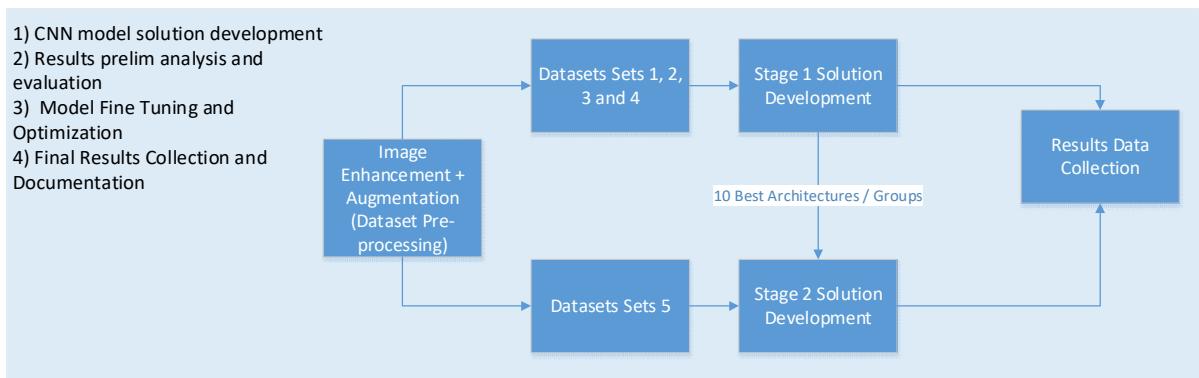


Figure 13: Detailed Process flow of CNN Solution Development from Stage 3.

The entire CNN solution development process was broken down into 2 main stages of Stage 1 and Stage 2. As a baseline, the developed models in Stages 1 and 2 were trained with the following standard hyperparameters as shown in Table 13 unless specifically specified otherwise. Depending on the network architecture, some of these parameters may be modified and specified accordingly.

Table 13: Networks' Training Baseline Parameters.

Parameter	Value
Learning Rate	0.01
Optimizer	Stochastic Gradient Descent (SGD)
Epochs	50
Batch Size	32

In Stage 1, datasets Set 1, 2, 3 and 4 were used to train, validate, test and optimize each architecture. Each architecture was denoted as a group. The main aim for stage 1 is to perform an exhaustive training and comparative evaluation of several proposed CNN solutions that are suitable candidates for the solution. There is a total of 10 architectures i.e., 10 groups defined in Stage 1. The CNNs developed within Stage 1 was further broken down into 2 main types which are self-developed CNN architectures and transfer learning architectures.

Self-developed CNN architectures are architectures that were developed and trained from the ground-up without any prior knowledge being embedded within the network. 2 out of the 10 groups constitute self-developed architectures. The self-developed model architectures were trained, validated, tested and optimized on the 4 different sets of data and the best model from each group was selected as the candidate for Stage 2.

Transfer learning models on the other hand used state of the art CNN models that were prior trained on ImageNet dataset and have prior knowledge. These models were modified and repurposed for classification of GI tract diseases. A total of 4 different state of the art model architectures were selected which are ResNet50-V1, ResNet50-V2, MobileNet-V1 and MobileNet-V2. Two modes of training these models were implemented which were feature extraction method and fine-tuning method. Under feature extraction method, the state-of-the-art models (known as the base network) were taken directly, and their weights were preloaded with the weights obtained from the ImageNet training. Several dense layers and a final classification node were then added to the end of the feature extraction network. It is important to note that under feature extraction training mode, none of the weights within the CNN base networks were modified during training and only the weights of the added layers were modified. For fine tuning on the other hand, the architecture of these CNN models still remained the same

as the feature extraction models with the distinction that several layers in the base networks were unlocked, enabling their weights to be updated during training. Each of these transfer learning-based architectures state-of-the-art models were trained, validated and tested on datasets Set 1, 2, 3 and 4. A total of 4 groups constituted transfer learning feature extraction and the remaining 4 constitute transfer learning fine tuning. Similar to the self-developed models, the best performing model from each group was selected as candidates for Stage 2. At the end of stage 1, the best 10 different trained, validated and tested model architectures from each group were selected and further developed in Stage 2.

In stage 2, dataset Set 5 (the augmentation dataset) was used to train, validate, and test the performance of the 10 best different models from Stage 1. Dataset Set 5 was used to simulate data augmentation whereby the images processed in Section 3.3.3 were used as synthetic image data to train the models. The aim of this mode of training is to induce variability in the range of images that was used to train the models. This is to simulate the variety of WCE image sources that is learned by the CNN models and hence will improve the CNN model's capability in generalization and performance in real life applications. However, it is important to note that as a control, the images in validation and testing were kept as that of Set 1.

An additional 3 groups of training, validation, and testing were performed in Stage 2 which encompassed training and testing the 10 models on dataset Set 5 with:

- The same parameters from Stage 1 specifically with 50 epochs, trained, validated, and tested separately as 10 models.
- All other parameters from Stage 1 remained the same except increasing the epochs from 50 to 100, trained, validated, and tested separately as 10 models.
- Ensemble method whereby the predictions from the 10 models were merged and considered collectively to predict the final results. The finalization of results was determined by using max voting.

The results obtained from Stage 2 training was also be collected and analysed.

3.3.4.1 01_Base_model1 (Group 1)

The architecture of the network is depicted on Figure 14 below

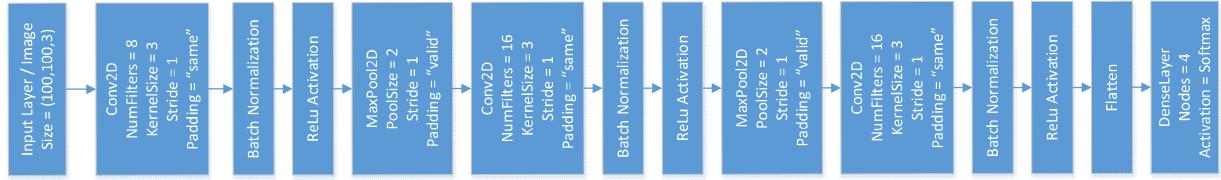


Figure 14: Architecture of 01_Base_model1 (Group 1)

The input image size for this architecture is 100 x 100 x 3 pixels which is smaller than the other architectures used in this study. An architecture with a smaller input image size was developed to study the effect of image size on the performance of the models. Four models were created each trained of datasets Set 1,2,3 and 4. The suffix of the model's name in Appendix D and Appendix E denotes the dataset it was trained on.

The best amongst the 4 initial models was then optimized further as detailed in Section 3.3.4.7 below. The best model amongst the 7 was then selected based on the evaluation method specified in Section 3.3.4.8 below.

3.3.4.2 02_baseModel2 (Group 2)

The architecture of Base Model 2 took consideration of the architecture of Base Model 1 and further expanded the architecture with additional convolutional layers creating a larger network. This is to cater to a larger input image size of 224 x 224 x 3 pixels. The architecture of Base Model 2 is shown in Figure 15 below:

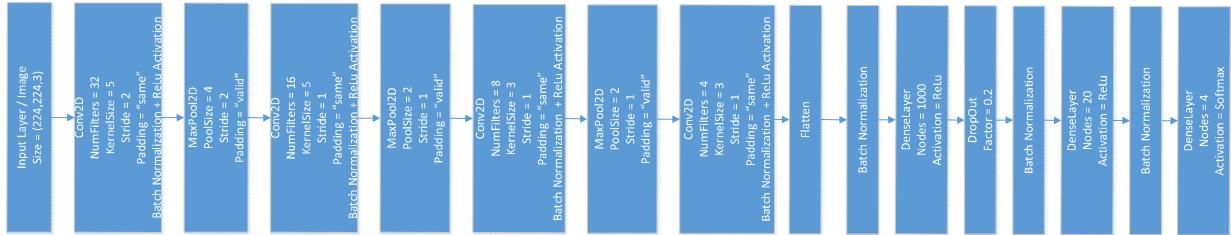


Figure 15: Architecture of 02_baseModel2 (Group 2)

Optimization and selection of the best model is similar to that of Base Model 1 as detailed in Section 3.3.4.1.

3.3.4.3 ResNet50-V1 (Group 3 and 7)

ResNet50-V1 is the state-of-the-art CNN architecture that won the ImageNet Classification challenge in 2015. The network is comprised of a total of 177 layers and its architecture schematic can be shown in Figure 16 below. Based on literature review, a lot of authors have used Resnet50-V1 as the base network in their proposed solution.

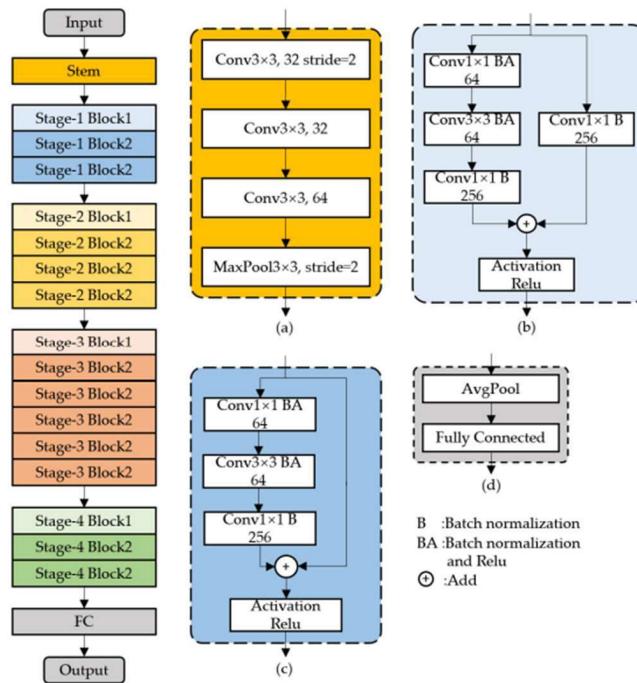


Figure 16: Architecture of ResNet50-V1 (a) Step-block, (b) Stage1-Block1, (c) FC-Block (S. Wang et al., 2021)

The entire network can be broken down into 16 different blocks with 49 convolutional layers using different types of filters for feature extraction. The filter kernels extract features followed by an averaging and max pooling layers which help to pool and reduce the size of the feature vectors extracted. Batch normalization is also introduced to regularize the feature values. The main feature of the ResNet50-V1 architecture are residual blocks as shown in Figure 17 below.

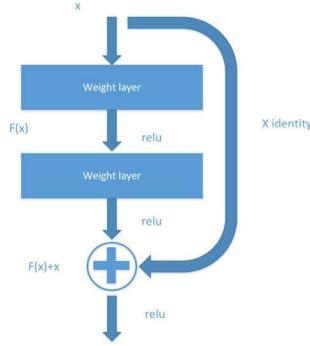


Fig. 3. ResNet building block

Figure 17: Residual Connection (ResNet Building Block)

Residual blocks work on the principle of feeding the output of one layer to the layer succeeding it and directly into layers which are located several layers away from it. This is in contrast to the working principle of traditional neural networks that only feed inputs directly to the next layer in stages. As traditional networks increase significantly in size, a gradient vanishing problem exists where the context of information learned in earlier layers are lost as the depth of the network increases significantly. Residual connections help in overcoming this by transferring prior knowledge from earlier layers directly to upper layers and combining them with the features learned from direct layers. A better understanding of the ResNet50 architecture can be referred to (He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, 2015)

The ResNet50-V1 architecture is used as the base model. The final detection head of the base network was removed and instead replaced with additional dense layers + batch Normalization + SoftMax output layer of 4 classes as shown in Figure 18 below.

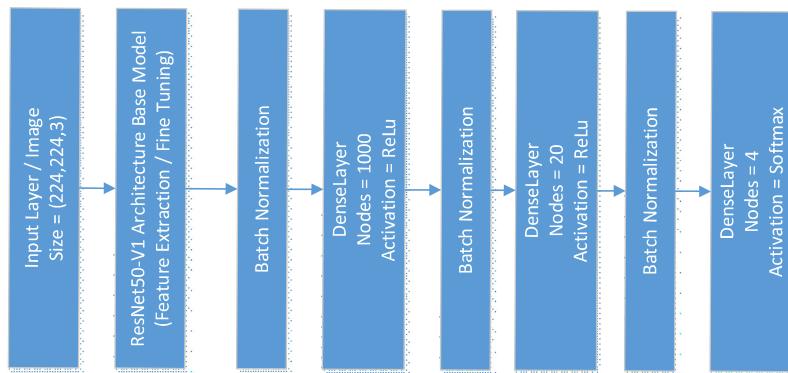


Figure 18: Additional Layers Added to ResNet50-V1 Transfer Learning Models.

As mentioned in Section 2.2.6, the training for transfer learning employed 2 modes of feature extraction and fine-tuning. In the feature extraction mode, none of the layers within the original ResNet50-V1 architecture was unlocked for re-training i.e., none of the weights of the networks were updated during training and only the weights the additional blocks were updated during training. In the fine-tuning mode of training, the last 34 layers in the ResNet50-V1 architecture network were unlocked for re-training. This allowed a portion of the base network model to learn new features from the datasets which will allow it to adapt better. Optimization and selection of the best model was done similar to that of Base Model 1 above.

3.3.4.4 ResNet50-V2 (Group 4 and 8)

ResNet50-V2 is an updated version of the ResNet50-V1 architecture. When tested on the ImageNet dataset, it performed better as compared to the initial ResNet50-V1 architecture. The main changes made in the ResNet50-V2 architecture was mainly on the architecture of the residual connections. Instead of adding the residual connections to n-layers ahead prior to activation, the residual connections were instead added after the activation function. A comparison of the skip connection architecture is shown in Figure 19 below. The ResNet50-V2 architecture is shown in Figure 20 below. A clear understanding of the architecture can be referred to (K. He et al., 2016).

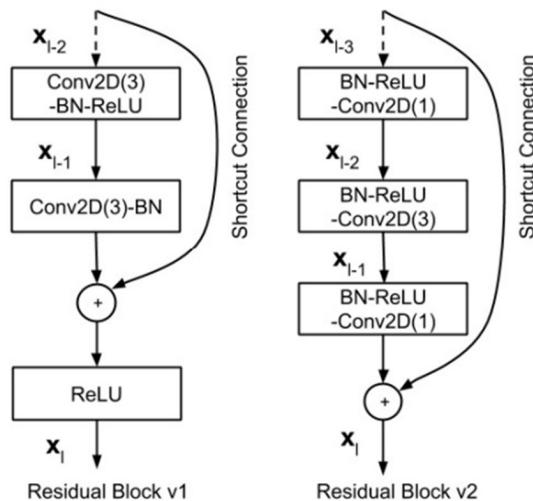


Figure 19: Comparison of ResNet-V1 and ResNet-V2 Residual Blocks (Van Hieu & Hien, 2020).

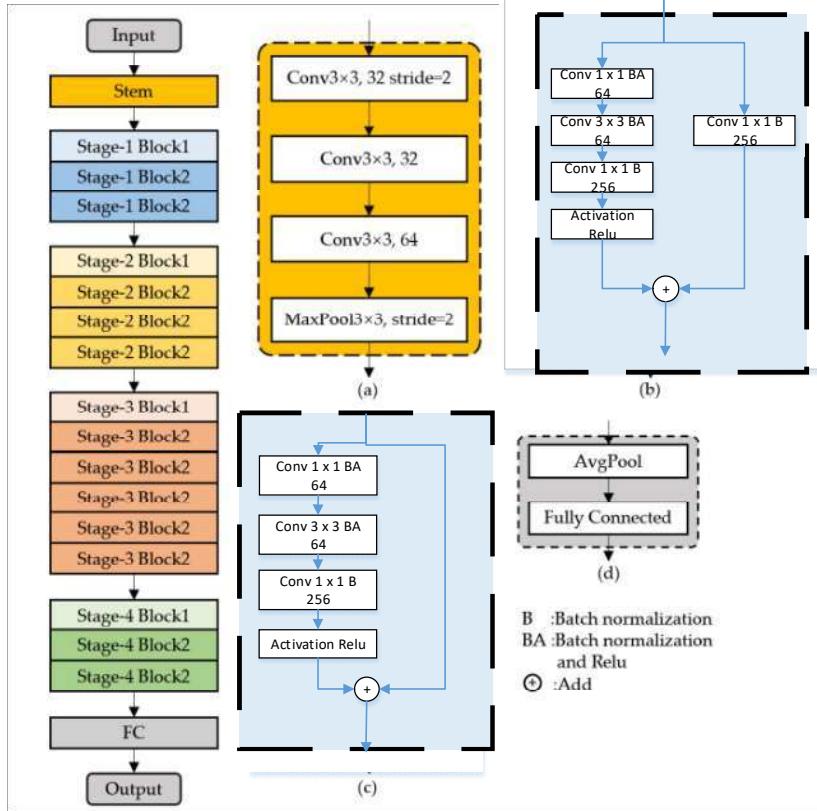


Figure 20: Architecture of ResNet50-V2 (a) Step-block, (b) Stage1-Block1, (c) FC-Block

The same modes of feature extraction and fine-tuning training were also applied to ResNet50-V2 architecture. The ResNet50-V2 network was used as the base network and the same additional dense layers + batch Normalization + SoftMax output layer of 4 classes layers as shown in Figure 21 were added to the end of the layers. In fine tuning, 38 layers of the networks were unfrozen for training. Optimization and selection of the best model was similar to that of Base Model 1.

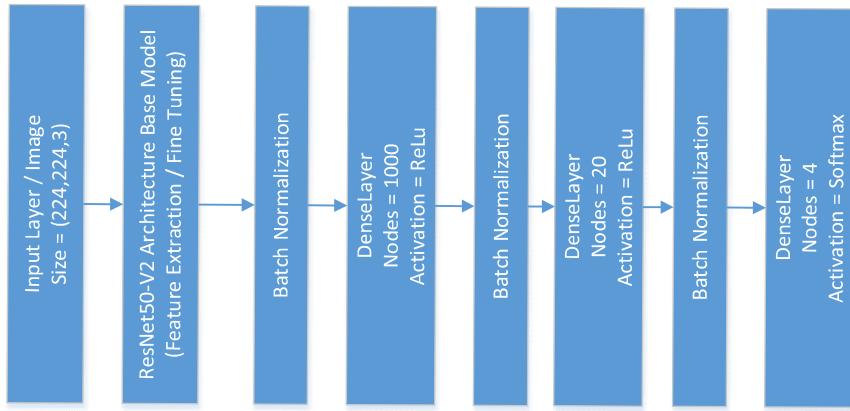


Figure 21: Additional Layers Added to ResNet50-V2 Transfer Learning Models.

3.3.4.5 MobileNet-V1 (Group 5 and 9)

The size of the ResNet50 models are relatively huge in size and hence the speed of detection may be slow. Hence, an alternative lightweight network that has a balanced classification performance against speed is the MobileNet architecture. The MobileNet-V1 architecture is a lightweight CNN network that has significant lesser number of parameters whilst still being able to perform well with high classification accuracy (Howard et al., 2017). The unique architectural feature in MobileNet is the application of depth-wise separable convolution.

As opposed to the traditional convolutional method, depth wise separable convolution is an alternative way of performing convolution in images in such a way that it will require lesser parameters and number of computational complexities in the entire process. The traditional, convolution performs convolution with kernels across the depths of channels of the image simultaneously. Performing this at scale for N layers of feature maps in a CNN would require high and expensive computational multiplications for the filters to convolve across the entire image. Depth wise convolution solves this issue by segregating the process into 2 steps of depth-wise convolution (filtering-stage) and point-wise convolution (combination stage). In depth-wise convolution, the filters kernels perform convolution independently across each channel depth to obtain feature maps, where the number of feature maps equals to the depth of the image (Image depth is typically 3 for RGB input image). These extracted feature maps are stacked on one another. Then, pointwise convolution is performed where $N \times 1 \times 1 \times$ depth sized kernels perform convolution on the stacked feature maps to obtain N final feature maps of the image. A schematic representation of this can be seen in Figure 22 below.

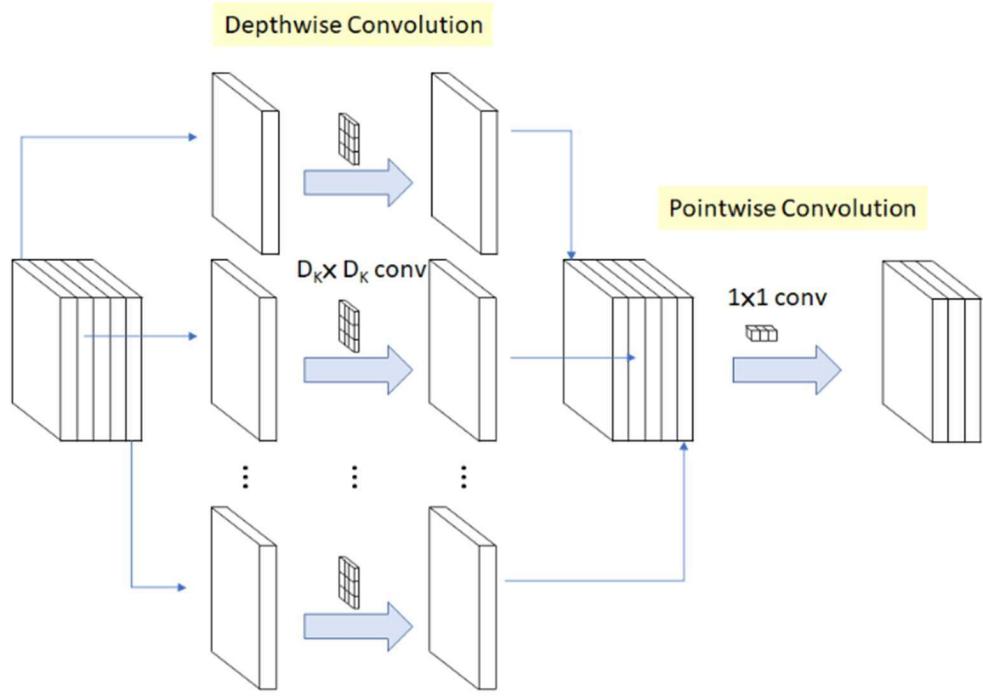


Figure 22: Depth wise Separable Convolution Operation (Kumar et al., 2021)

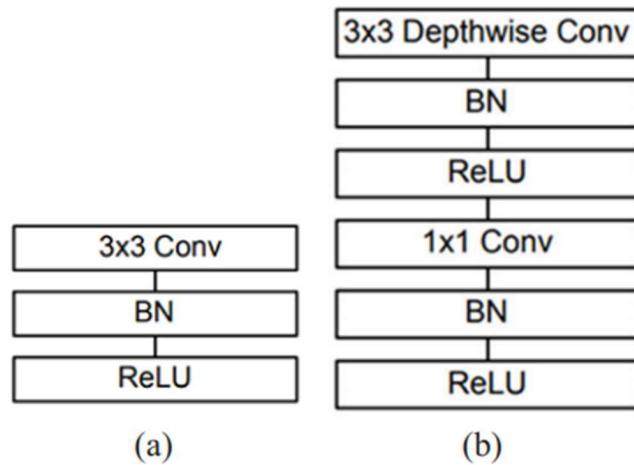


Figure 23: (a) Standard Convolution Layer with Batch Normalization and ReLu activation, (b) Depth-wise separable convolution with depth wise and pointwise layers followed by Batch Normalization and ReLu activation (Bardina et al., 2020)

The MobileNet architecture is depicted in Figure 24 below:

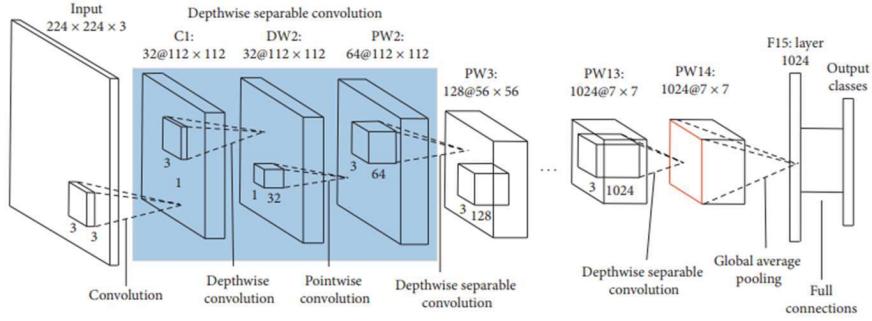


FIGURE 1: Architecture of MobileNet.

Figure 24: Architecture of MobileNet-V1 (W. Wang et al., 2020)

The MobileNet-V1 architecture is comprised of 32 filters, followed by 19 residual bottleneck layers. The MobileNet-V1 architecture has been used for many different medical applications such as done by (Chaturvedi et al., 2021) for detecting skin cancer achieving over 83.1% accuracy, (Apostolopoulos & Mpesiana, 2020) for classification of Covid-19 image dataset for detecting viral and bacterial pneumonia cases and (Da Nóbrega et al., 2018) for classifying lung nodules malignancy.

Due to its lightweight, the application of MobileNet makes it very attractive for applications in embedded systems where there are low computation resources, especially in the medical domain. (Rustum et al., 2021) cited that Mobile Net architecture is best for embedded vision applications which is what is being done here to analyse WCE images in real time. A detailed understanding of the MobileNetV1 architecture can be referred to (Howard et al., 2017).

The MobileNet-V1 architecture was trained in the same manner as the other transfer learning architectures with feature extraction and fine-tuning modes. The MobileNet-V1 network was used as the base network and the same layers as shown in Figure 25 were added to the end of the layers. In fine tuning, the last 36 layers of the base network were unlocked.

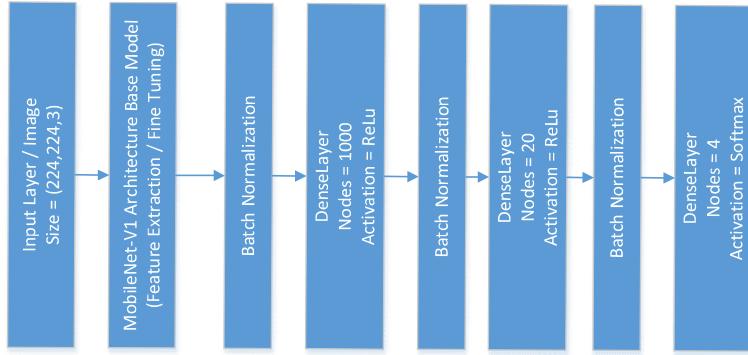


Figure 25: Additional Layers Added to MobileNet-V1 Transfer Learning Models.

Optimization and selection of the best model was similar to that of Base Model 1 above.

3.4.2.6 MobileNet-V2 (Group 6 and 10)

MobileNet-V2 is an updated and improved version of the MobileNet-V1 architecture. The MobileNet-V2 architecture takes the concept of depth-wise separable convolution and develops it further in terms of 2 aspects which are:

- Linear bottlenecks between the layers. (Sandler et al., 2018) cited that these linear layers are important to prevent non-linearities from diminishing too many details extracted from the image.
- Introducing shortcut connections between the bottlenecks in the depth wise separable convolution that encode the intermediate inputs and outputs while the inner layer enhances the model's capability in translating low level concepts like the image pixels to high level descriptions i.e., the image categories. These shortcut connections also reduce training time and also gives better accuracy.

The MobileNet-V2 architecture retains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. Figure 26 below depicts the changes made to the MobileNet-V2 architecture. Depth-wise convolution and point-wise convolution is still retained with the distinction being that point-wise convolution is performed first with non-linearity, followed by depth wise convolution, followed by another point-wise convolution without non-linearity. Additionally, the skip connection architecture which is incorporated by the ResNet architecture is also depicted.

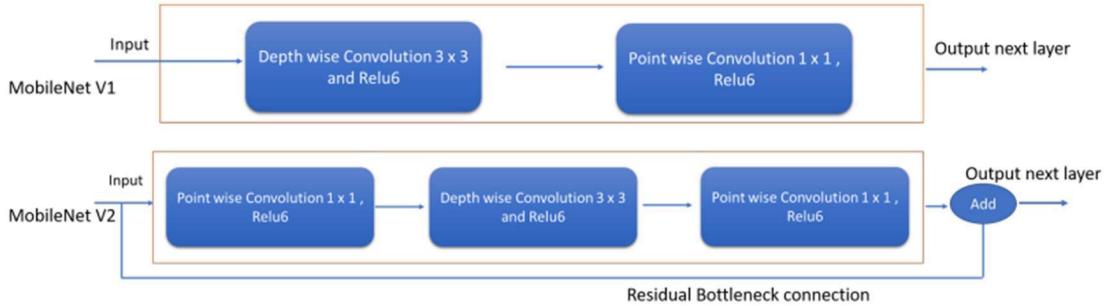


Figure 26: Comparison of MobileNet-V1 against MobileNet-V2 Architectures.

The MobileNetV2 architecture was trained in the same modes as the other transfer learning architectures with feature extraction and fine tuning. The MobileNet-V2 network was used as the base network and the same layers as shown in Figure 25 were added to the end of the layers. In fine tuning, the last 40 layers of the base network was unlocked. Optimization and selection of the best model was similar to that of Base Model 1 above.

3.3.4.6 Stage 2 Models (Group 11, 11A and 12)

Upon selection of the best 10 CNN architecture from Stage 1, the finalized architectures were then trained, validated, and tested on the Set 5 dataset each with their own respective best parameters from Stage 1. These models belong to Group 11. The performance of these 10 individual models were evaluated.

A final optimization step was undertaken by increasing the number of epochs from 50 to 100 to allow the network architecture to learn for longer period, as it was notice in many of the learning training history curve that the learning during training did not plateau within 50 epochs. These group of models were denoted as Group 11A. The results from the original and optimized networks from Stage 2 were recorded and analysed further.

Finally, an ensemble comprised of the optimized 10 models from Stage 2 is built whereby max voting is used to finalize the results. This final model is denoted as Group 12.

3.3.4.7 Optimization

Model optimization was performed in 3 aspects. Firstly, by modifying the optimizer of the CNN from Stochastic Gradient Descent with the Adam optimizer. This also resulted in a smaller learning rate as the default learning rate for SGD is 0.01 and the default learning rate of Adam is 0.001.

Secondly, to study the effects of learning rate variations, the learning rate during training was converted from a static learning rate to a dynamic learning rate where the learning rate of the algorithm remained constant for the first half of the total epochs. After which the learning rate was halved every epoch. This allowed the model to learn on a high level during the first few epochs of training before finally stabilizing its learning during the final few epochs of training. This was done by implementing a Learning Rate Scheduler (LrScheduler) during training.

Finally, a combination the Adam optimizer and dynamic learning rate was implemented to observe how the synergistic effects of both these parameters improve the model's training performance.

3.3.4.8 Evaluation

The main evaluation metrics used throughout the development were Accuracy and F1-Score. Accuracy was used as the primary evaluation metric and F1-score was used as a secondary metric to validate the results or trends observed from Accuracy. Evaluation was done by using the average of the testing and validation sets' accuracies.

Within Stage 1, as a start, a total of 4 architectures would be trained in each group. Upon completion of training, each of the 4 architectures would then be tested using the entire corresponding validation set. The results produced would then be used as a determinant on the best model for each group to further optimize. Optimization, which was explained in Section 3.3.4.7 would then produce 3 additional models. These 3 additional models would then be tested on the validation set as well. The complete 7 architectures would also additionally be tested on the corresponding entire test set. An average of results from the validation and test sets would then be used to determine the best performing model from each group to select the best 10 models from Stage 1.

In Stage 2, the best 10 selected model architectures from Stage 1 were then trained, validated, tested, and optimized on dataset Set 5. Additionally, an ensemble of these 10 models would also be developed and tested. Similar to Stage 1, the evaluation for Stage 2 considered the average of the validation and test set accuracy as the primary metric and the test F1 score results as the secondary metric. The results would then be collected and analysed accordingly.

3.3.4.9 Tools and Frameworks

Python programming was used as the primary language of development. Initial data analysis and was done primarily on a local machine using Jupyter Python Notebook. Experimentation of the image processing enhancement algorithms was initially done on a local machine using Jupyter Python Notebook as well. However, once the algorithms were fully developed, they were then packaged into individual Python functions in a Python script file to be utilized on a large scale for preparation of the datasets Set 2, 3, 4 and 5. This was implemented using the PyCharm IDE. The specifications of the local machine used was Intel i7-6700HQ CPU @ 2.6GHz with 20 GB of RAM.

CNN model development and testing on the other hand was done entirely on Paperspace Gradient which is a web based Jupyter IDE cloud computing platform that enables Machine Learning model development with access to high powered Graphics Processing Units. Training of the models were primarily done using NVIDIA's P5000 GPU. The Tensorflow Machine Learning framework was used to develop the CNN models. Results and data analysis on the other hand was performed using PowerBI.

Table 14 details the Python libraries used throughout the solution development.

Table 14: Python Libraries used during Development

Python Library	Purpose
Os	Used for operating system dependent functionality.
Pandas	Used for data manipulation and analysis.
Numpy	Used for performing operations on large arrays / matrices.
Pathlib	Used to work with / navigating project files and directories
Matplotlib	Used for plotting charts and visualizations and to plot images.

Random	Random number generator
PIL (pillow)	Used for opening, manipulating, and saving images.
CV2`	Used for image processing.
Seaborn	Used for plotting charts and visualizations.
Copy	To copy variables to a separate memory address.
Skimage	Used for Image processing.
Natsort	Used for sorting.
Math	Used to perform mathematical operations.
Sklearn	Used for building classification report and confusion matrix.
Sys	Used for checking system variables.
Tensorflow	Used for Deep Learning CNN model solution development.

3.4 Results, Analysis, Interpretation and Conclusion

The results of the CNN models developed throughout Stage 1 and 2 were recorded, analysed, and interpreted. Following which, discussions on the observations were made along with final Conclusion. This is detailed in CHAPTER 4 to CHAPTER 6 below.

3.5 Final Documentation and Presentation

Concurrent to development, the final documentation and presentation detailing the entire solution process and results was prepared.

CHAPTER 4

RESULTS

This chapter presents the findings and results of this study. This section also analysed the results accordingly and performed a comparative study of the proposed solutions. The analysis was broken down into 4 stages which are Image Enhancement, CNN Solution Development Stage 1, CNN Solution Development Stage 2 and finally a comparison of the CNN models from Stage 1 against Stage 2 results' analysis was performed to better understand the performance of the various CNN model solutions developed at scale under various parameter settings. Analysis was also performed to understand how each of these parameters affected the performance of the solution which ties back to the objectives and aim of this study. Ultimately, the goal is to propose the best model / solution for practical applications.

4.1 Image Enhancement

Image enhancement analysis was performed both qualitatively and quantitatively to visualize and gauge the performances of the image processing enhancement techniques on a high level.

4.1.1 Qualitative Analysis of Images Before and After Image Processing

Qualitative analysis of the images was performed to better visualize and compare the quality of the images before and after undergoing the 3 different types of image processing enhancements. The enhancement of features and the improvement in contrast were observed specifically for features that distinguish one class from the other. Figure 27 below depicts the comparison of images for each class.

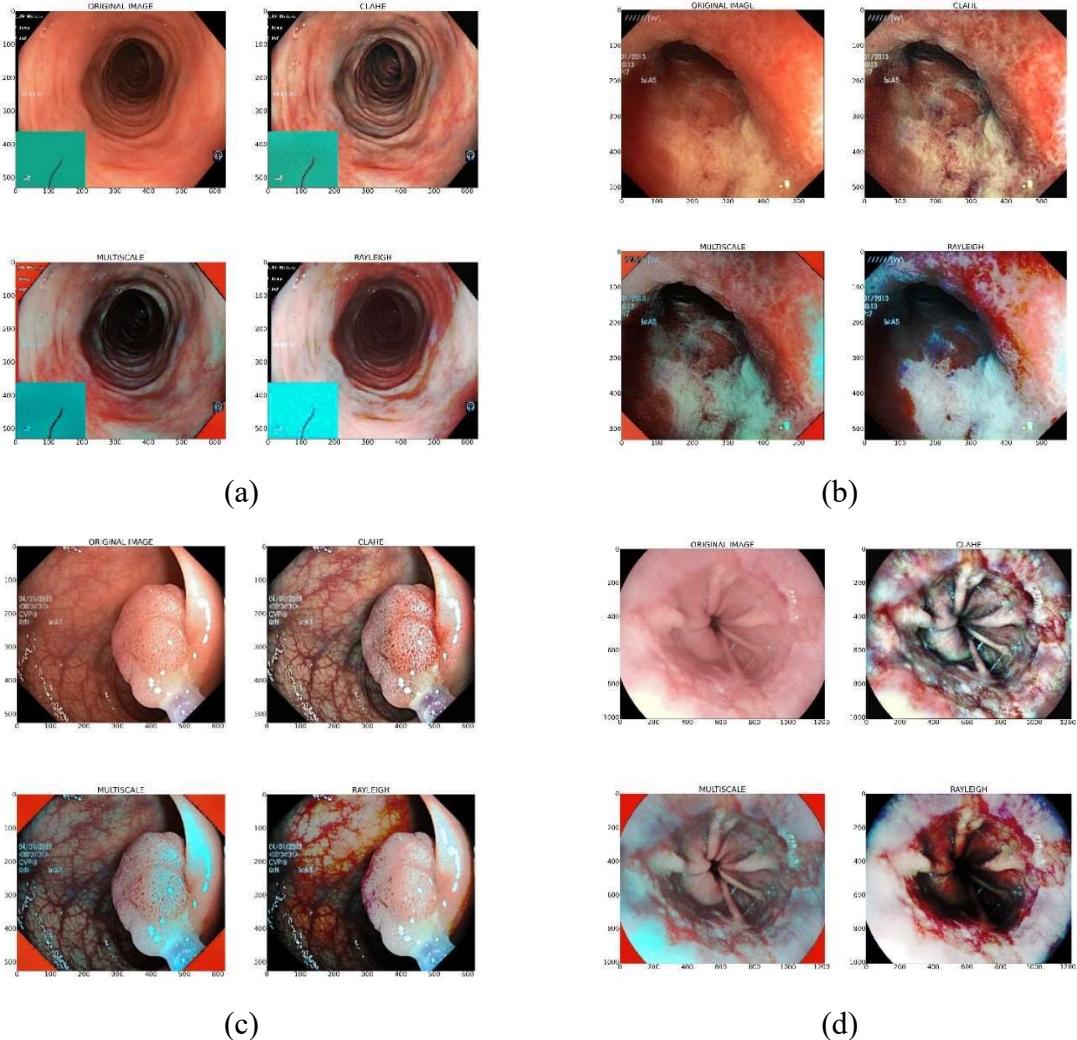


Figure 27: Comparison of Image Before and After Undergoing Image Processing Enhancement for each class. (a) 0_normal, (b) 1_ulcerative_colitis, (c) 2_polyyps, (d) 3_esophagitis

Visually by inspecting Figure 27 above it can be observed that the original images' features were not very distinct. By undergoing the 3 different image enhancement techniques, it can be observed that the features i.e., the edges and contrast of the image improved significantly. This gives better distinction of the different regions within the GI Tract images.

For class 1_ulcerative_colitis, the ulcer can be vaguely seen on the bottom right corner of the original unprocessed image. However, after processing the images, it can be observed that the colour and texture features of the ulcer which are the white or yellowish coloured lesions in the images were enhanced, making them more distinguishable for detection. With regards to class 2_polyyps, it can be observed that the contrast of the image had significantly improved.

Vasculatures in the image which were initially not visible are now clearly visible, inclusive of those on the polyps itself. The texture and the borders of the polyp that differentiates it from the healthy GI tract sections is also clearer providing a visual separation between the two. Lastly, the improvement in contrast of the enhanced 3_esophagitis images as compared to its original can also be clearly seen. The colour and texture features of the constricted or inflamed sections of the images were enhanced and their features are more distinctive.

4.1.2 Analysis of Evaluation Metric (PSNR SSIM and MSE)

In addition, the processed images were also analysed and evaluated quantitatively in terms of PSNR, MSE and SSIM metrics to gauge the performance of each of the image processing enhancement techniques. As an indicator, higher values of PSNR and SSIM relates to a better enhancement whereas a lower MSE correlates to a better enhancement.

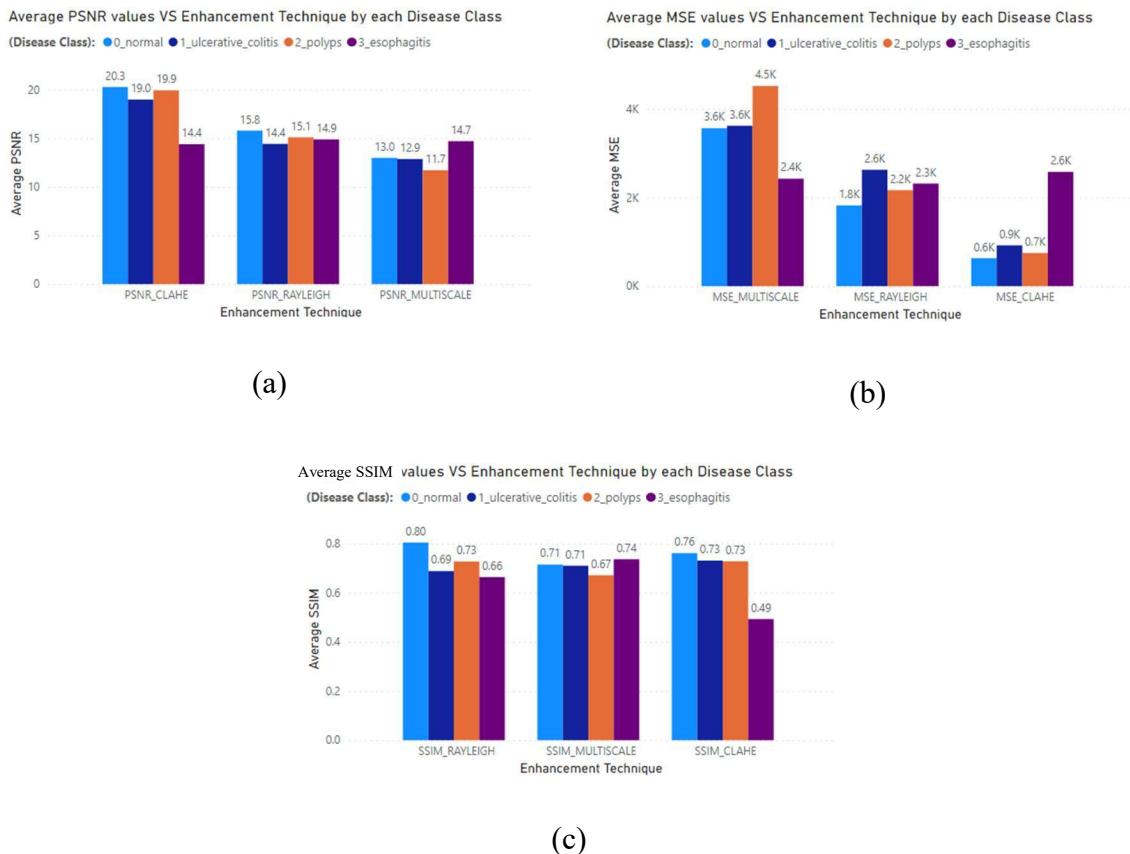


Figure 28: Image Processing Enhancement Results. (a) PSNR, (b) MSE, (c) SSIM.

On average across all classes, when comparing the PSNR of the enhancement techniques, it is observed that CLAHE has the highest PSNR value, followed by Rayleigh and finally Multiscale. From this it can be gauged that the CLAHE technique enhanced the quality of features in the image the best in terms of Power-to-Noise ratio. It is also observed that among all of the classes the, 0_normal class images have the highest PSNR whereas 3_esophagitis has the lowest.

With regards to SSIM, it can be observed that the values are quite similar to one another among all 3 techniques for classes 0_normal, 1_ulcerative_colitis and 2_polyps. The SSIM for 3_esophagitis is however comparatively lower in CLAHE as compared to other classes. This can be correlated back to Figure 27 above where we observe that the resulting CLAHE image has much more details in its image and is very different from the original image as compared to Rayleigh and Multiscale. On average across all classes, the SSIM results rank in the order of Rayleigh, Multiscale and CLAHE. Among all 3 enhancement techniques, it is observed that the SSIM for 0_normal is the highest as compared to the other classes. This indicates that the structure of the enhanced 0_normal class images were not as significantly altered as compared to the other classes as it does not have distinctive features of diseases that were enhanced.

In terms of MSE, it can be observed CLAHE has the lowest MSE on average across all classes, followed by Rayleigh and Multiscale respectively. The MSE for all classes within CLAHE is low with the values for 0_normal, 1_ulcerative_colitis and 2_polyps classes being more than 50% lower as compared to the other corresponding classes in other enhancement techniques. The MSE for 3_esophagitis in CLAHE however is much higher as compared to other classes. This shows that the CLAHE algorithm had performed the most significant changes in the 3_esophagitis images in terms of feature enhancement which can also be observed in Figure 27 above.

4.2 CNN SOLUTION DEVELOPMENT

The CNN solution development results analysis was performed in 4 phases. Phase 1 analysed and compared the results obtained from Stage 1 solution development only. Phase 2 on the other hand analysed and compared the results obtained from Stage 2 solution development only. Phase 3 analysed the performance of both Stage 1 and Stage 2 models when tested on dataset Set 5's Test and Validation sets. Lastly, in Phase 4 analysis, the models from Stage 1 and Stage

2 would be tested on the Test and Validation sets of Set 1-4 combined. The performances of the models were compared to one another.

The best model from Phase 1 analysis i.e., the best model from Stage 1 was proposed as the first alternative solution. The best model from Phase 2 analysis i.e., the best model from Stage 2 was proposed as the second alternative solution. Based on analysis of the performances of the models in Phase 3 and Phase 4, the final best generalized CNN solution was selected.

As mentioned in Section 3.3.4.8 the main evaluation metric used in the solution development is the average of validation and test set results. This is denoted as ValTestAvg (Validation Test Average) in the results.

4.2.1 Tagging System

Due to length limitations of the model names, each model is assigned a tag for identification and for results visualization. Each model architecture from both Stage 1 or 2 adhered to the following naming convention. The general format of tagging the models are as below:

##_bb_cc_dd_ee

The representations of each of the notations are detailed in Table 15 below:

Table 15: Model Tagging Naming Convention

Label	Representation	Range of Values
##	Alphabetical representation of each model group architecture. The sequence of alphabet representation follows the sequence of model group number.	<ul style="list-style-type: none"> - A - M. - A represents any models from Group 1 architecture, B represent models from Group 2 architecture etc. - It is important to note that Group Architecture 11A is represented by alphabet L - Architecture 12 is represented by M
bb	Corresponding group number.	<ul style="list-style-type: none"> - 1 – 12. - 1 – 10 represent Stage 1 models.

		<ul style="list-style-type: none"> - 11, 11A and 12 represent Stage 2 models. - Where 12 represents the ensemble model architecture - 11A represents Group 11 models trained on 100 epochs instead of 50.
cc	Model sub-group number. This number represents the variations of architectures designed within the group itself.	<ul style="list-style-type: none"> - 1 – 7 for Stage 1 models. - 1 – 10 for stage 2 models. - 12345678910 for Group 12 models i.e., ensemble method.
dd	Image processing technique implemented on the image.	<ul style="list-style-type: none"> - List of 4 values i.e., N, C, M, R and MIXED. - N – Normal without image processing - C – CLAHE - M – MULTISCALE - R – Rayleigh. - Group 12 architecture is represented by MIXED
ee	Fine Tuning / optimization technique implemented	<ul style="list-style-type: none"> - TA –Adam optimizer - LR – Dynamic Learning Rate - TA_LR – Adam optimizer + Dynamic Learning.

A detailed list of the model names and their corresponding tags are attached in Appendix D.

4.2.2 Phase 1

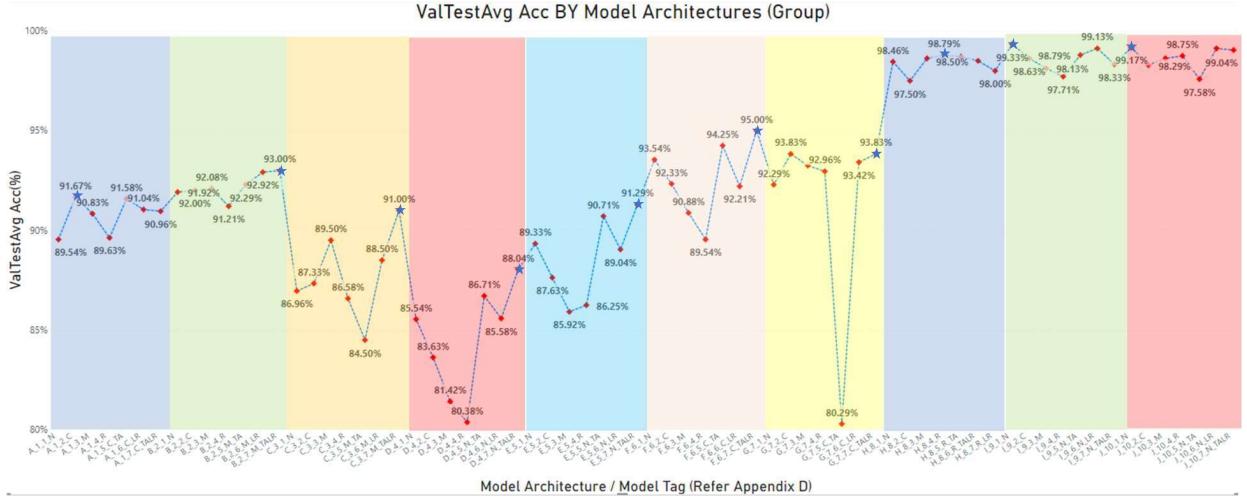


Figure 29: Phase 1 Results

The development of models in Stage 1 was broken down in a total of 10 groups of architectures. Each grouped is highlighted and labelled with a different colour scheme on Figure 29. Within each grouped, a total of 7 sub-groups were developed as explained in Section 3.3.4. In overall a total of 70 CNN models were trained, validated, and tested in Stage 1.

The best for each group is highlighted with blue coloured star in Figure 29. A simplified table of the best 10 from each group is depicted in Table 16 below. The top 3 models are highlighted in yellow.

Table 16: Best 10 Models from Stage 1.

modelName	ModelTag	Model Group Merged	Val Acc (%)	Test Acc (%)	ValTest Average (%)	Test F1Score (%)
01_BaseModel1_CLAHE	A_1_2_C	1	92.17	91.17	91.67	91.10
02_BaseModel2_MULTISCALE_tunedAdamLrScheduler	B_2_7_M_TA_LR	2	93.17	92.83	93.00	92.73
03_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	C_3_7_M_TA_LR	3	91.67	90.33	91.00	90.18
04_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	D_4_7_N_TA_LR	4	87.33	88.75	88.04	88.70
05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	E_5_7_N_TALR	5	91.17	91.42	91.29	91.31
06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	F_6_7_C_TALR	6	94.83	95.17	95.00	95.16
07_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	G_7_7_C_TALR	7	93.67	94.00	93.83	93.97
08_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	H_8_4_R	8	98.67	98.92	98.79	98.91
09_TransferLearningFineTuning_MobileNetV1_NORMAL	I_9_1_N	9	99.33	99.33	99.33	99.33
10_TransferLearningFineTuning_MobileNetV2_NORMAL	J_10_1_N	10	99.17	99.17	99.17	99.17

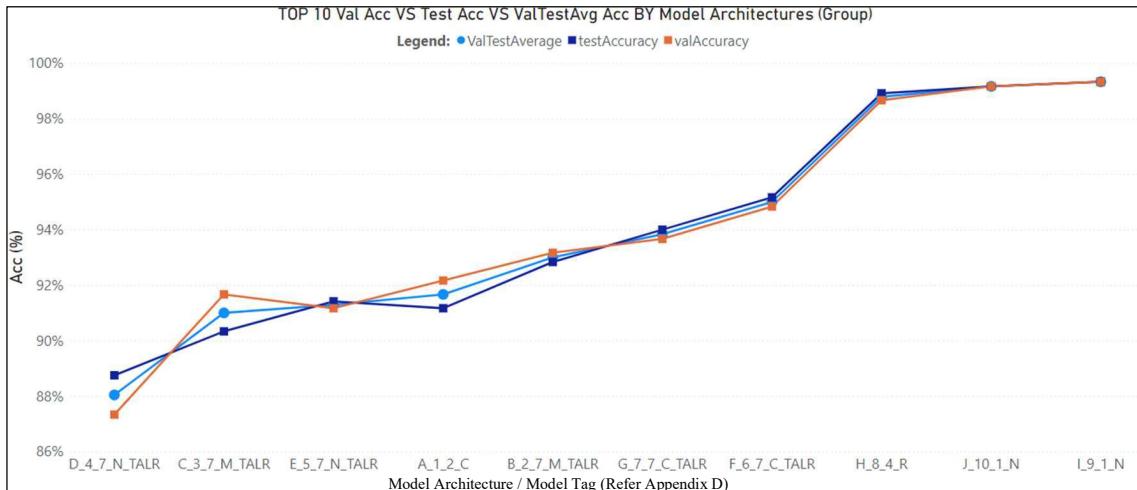


Figure 30: Comparison of Test Acc, Val Acc and ValTestAvg Acc for the Best 10 Models in Stage 1.

Figure 30 above compares the validation, test and ValTestAvg of the best 10 architectures from each class. In all 10 architectures, it can be observed that the ValTestAvg ranges between the test and validation accuracy. The range of values for the validation and test accuracy between all 10 architectures also do not deviate much from each other indicating that the results obtained for ValTestAvg as a primary metric is reliable and not skewed or affected by any outliers.

By analysing Figure 30, it is observed that in terms of model architecture type, the transfer learning fine tuning models performed the best amongst the 10 with models I_9_1_N, J_10_1_N and H_8_4_R placing at positions 1, 2 and 3 respectively. Self-developed models placed generally in the middle of placings as compared to the rest. Between the 2 self-developed models, models B_2_7_M_TALR (Group 2) performed better than A_1_2_C (Group 1). Generally, transfer learning feature extraction architectures performed the worst with exception of model F_6_7_C_TALR.

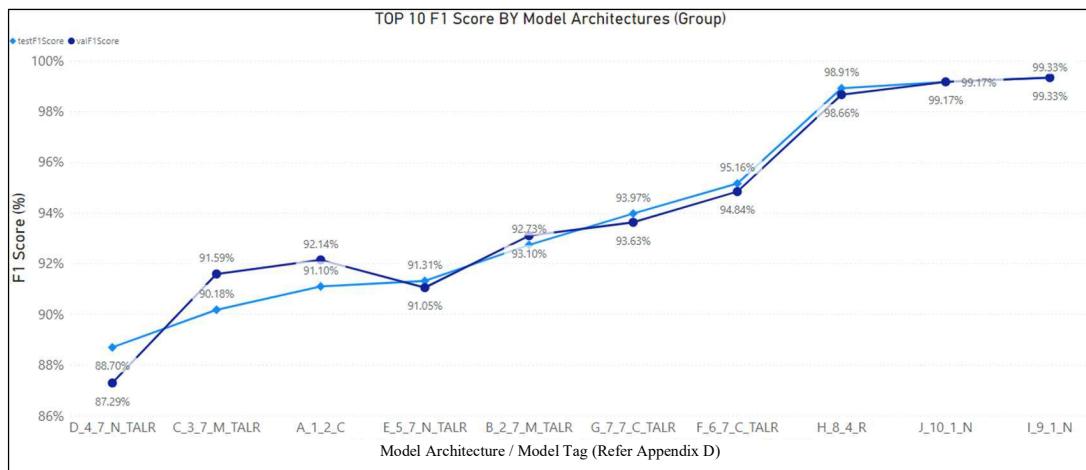


Figure 31: Test F1-Score of the Best 10 Models from Stage 1.

Figure 31 above depicts the F1 Score for validation and test sets of the best 10 models. From the trend observed, it can be seen that placings of the best 5 models remained the same with some slight changes in the sequence of the bottom 5 models. The same pattern is seen whereby transfer learning fine tuning models performed the best generally. It is also observed that MobileNet architectures generally perform better as compared to ResNet architecture. A similar trend of the F1 score chart also validates the results obtained using ValTestAvg.

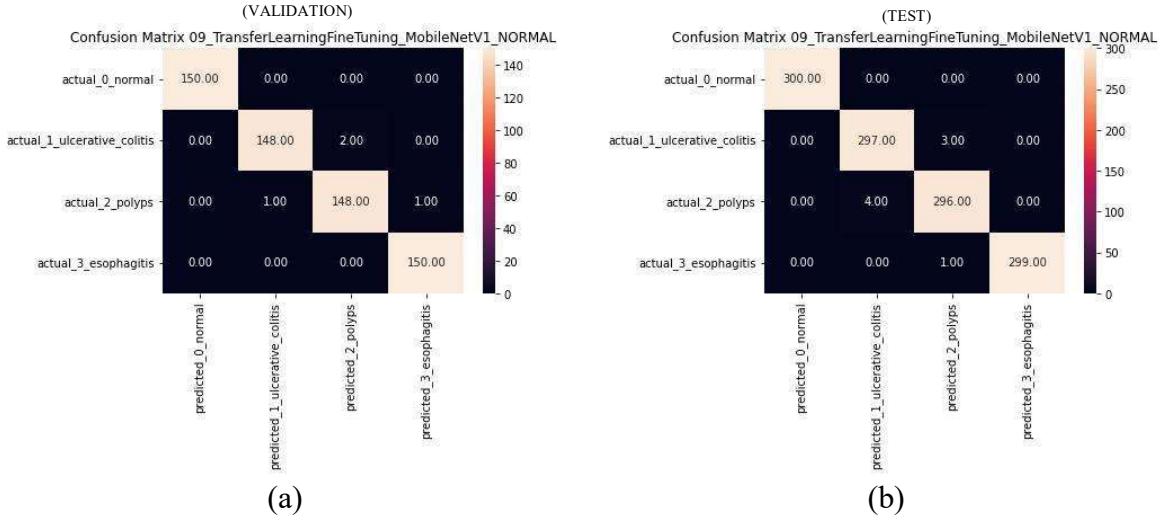


Figure 32: Confusion Matrix of the Best Model from Stage 1 (model I_9_1_N) (a) Validation Set, (b) Test Set.

Figure 32 above shows the confusion matrix of the best model from Stage 1 i.e., model I_9_1_N when tested on the Validation and Test set respectively.

When comparing between different transfer learning model architectures, it is observed that the MobileNet architecture in general performs better as compared to ResNet architectures. Figure 33 below depicts the average validation, test and ValTestAvg accuracy between all of the ResNet (28 models) based architectures and the MobileNet (28 Model) architectures in Stage 1 training.

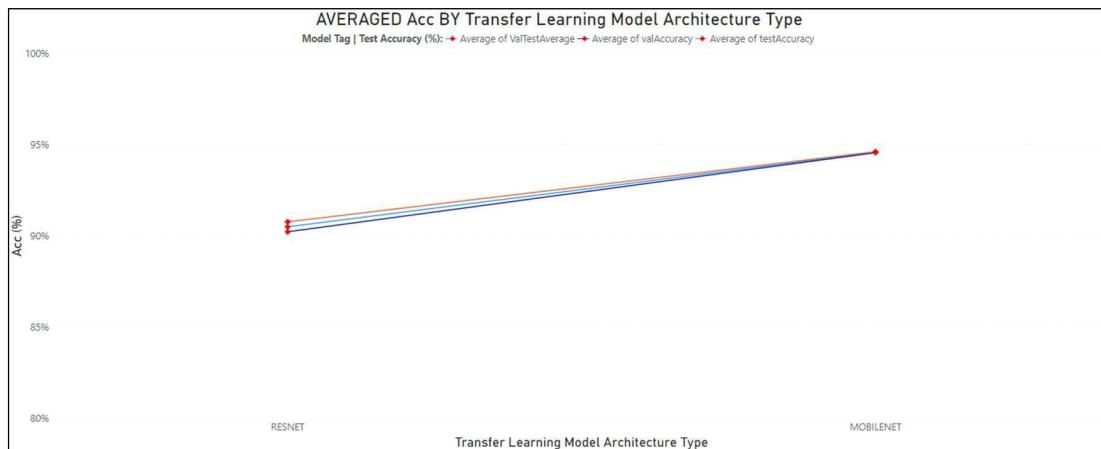


Figure 33: Comparison of Average Accuracies Between ResNet Architecture against MobileNet Architecture.

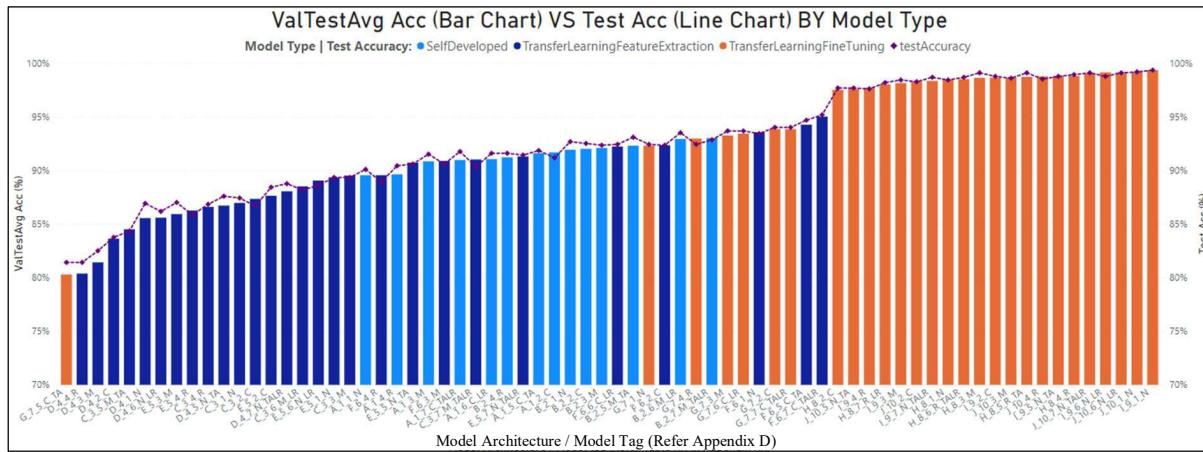


Figure 34: Comparison of ValTestAvg Acc and Test Acc of the Models from Stage 1. Categorized by Model Type According to Colour.

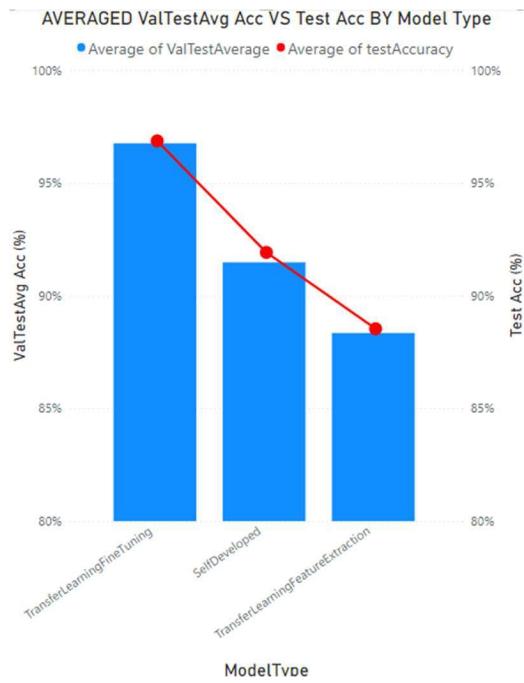


Figure 35: Averaged ValTestAvg Acc and Test Acc by Model Architecture Type.

Figure 34 above shows the ValTestAvg and test accuracy of all the developed 70 model architectures in Stage 1 categorized by the architectures type. The results are sorted in the manner of ascending ValTestAvg and from the plot it can be observed that transfer learning fine tuning models performed the best, followed by self-developed architectures and lastly, a majority of the lowest performing architectures are transfer learning feature extraction models. Figure 35 also depicts this as an average across all 70 models developed.

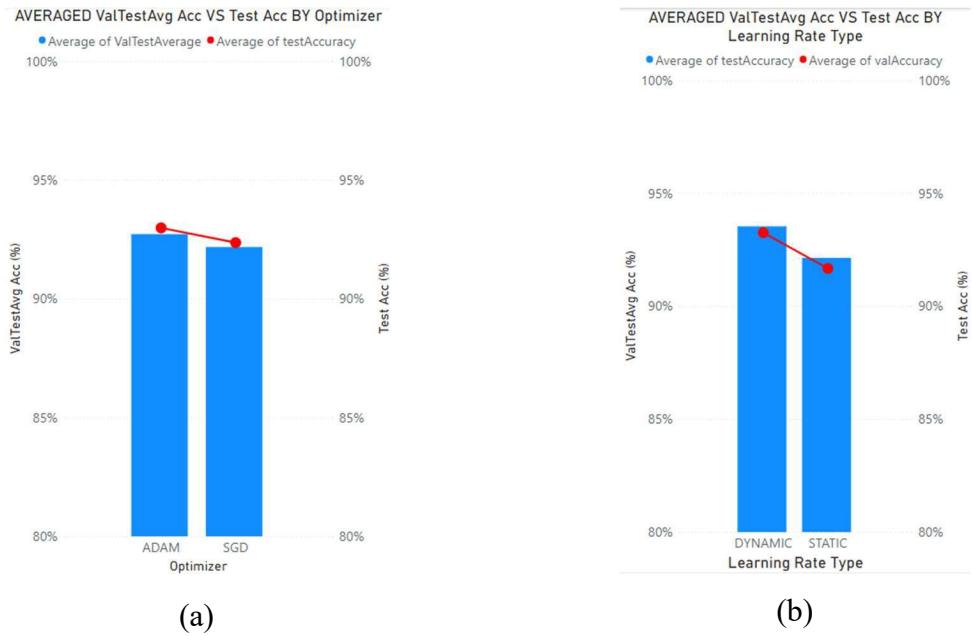


Figure 36: Averaged ValTestAvg Acc and Test Acc of Models in Stage 1 by (a) Optimizer, (b) Learning Rate Configuration.

Figure 36(a) on the other hand compares the ValTestAvg performance by different types of optimizers i.e., Stochastic Gradient Descent (SGD) vs Adam optimizers. The Adam optimizer was used on only 14 / 70 models developed which was significantly lesser compared to those trained with SGD. Figure 36(a) shows that on average, models trained with the Adam optimizer performed slightly better. Figure 36(b) compares the model architectures ValTestAvg average performance by different types of learning rate schemes. In this study only 2 types of learning rates schemes were implemented which were static and dynamic. Results showed that model architectures having a dynamic learning rate performed better as compared to the models that have static learning rate. These patterns are reflected on the list of best 10 stage 1 architectures that have 6/10 models having been trained with Adam and with a dynamic learning rate.

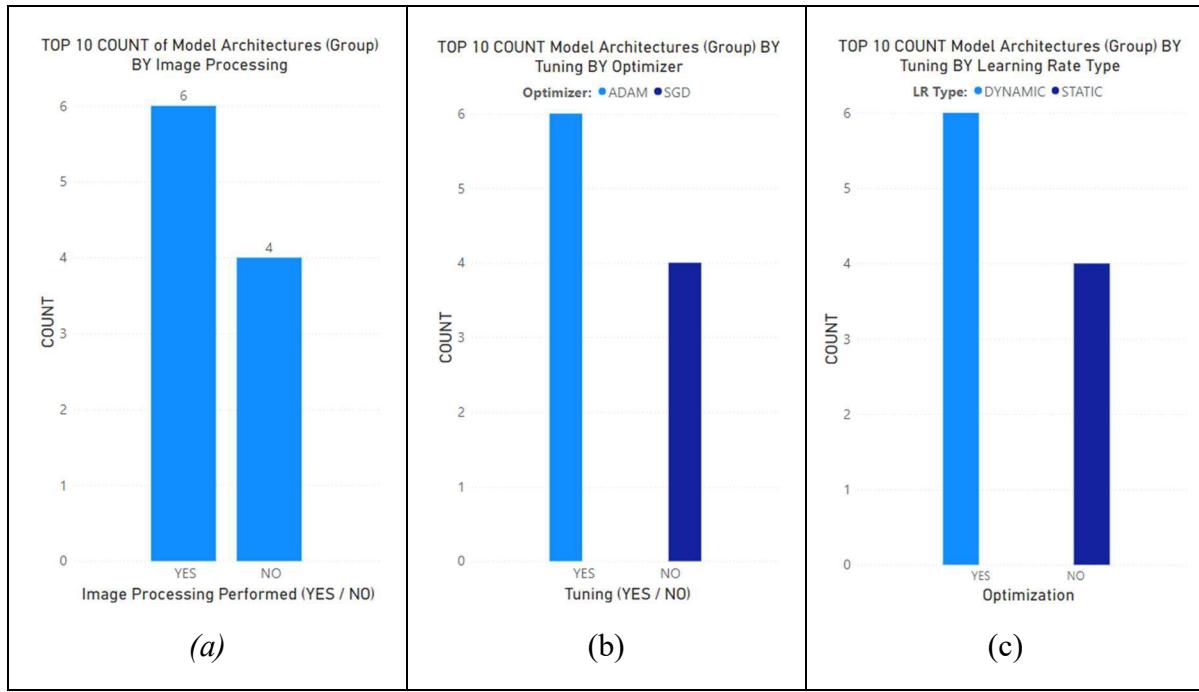
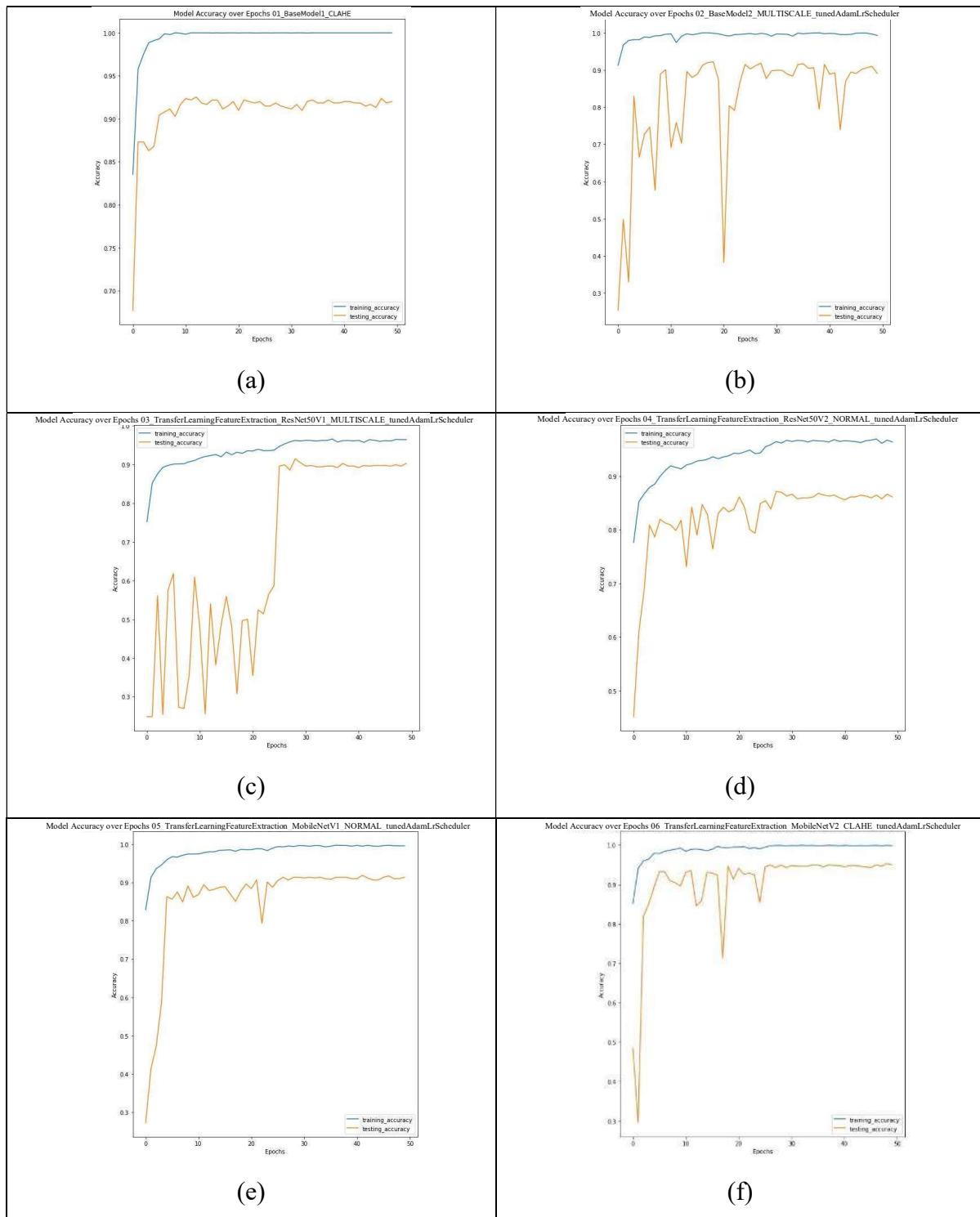


Figure 37: Count of Best 10 Models from Stage 1 by (a) Image Processing Enhancement Performed, (c) Model Optimizer, (c) Learning Rate Configuration

Figure 37 compares in overall the parameters among the best 10 architectures in Stage 1. It can be observed from Figure 37(b) and Figure 37(c) that a majority of the model architectures (6/10) have optimization being implemented, all of which have a combination of both Adam and dynamic learning rate. Hence, a combination of these 2 parameters improved a lot of the models' performance significantly as compared to using them individually. Figure 37(a) shows that a majority (6 /10) of the model architectures were developed using the datasets with image processing enhancement whereby the breakdown is 3 CLAHE, 2 Multiscale and 1 Rayleigh.



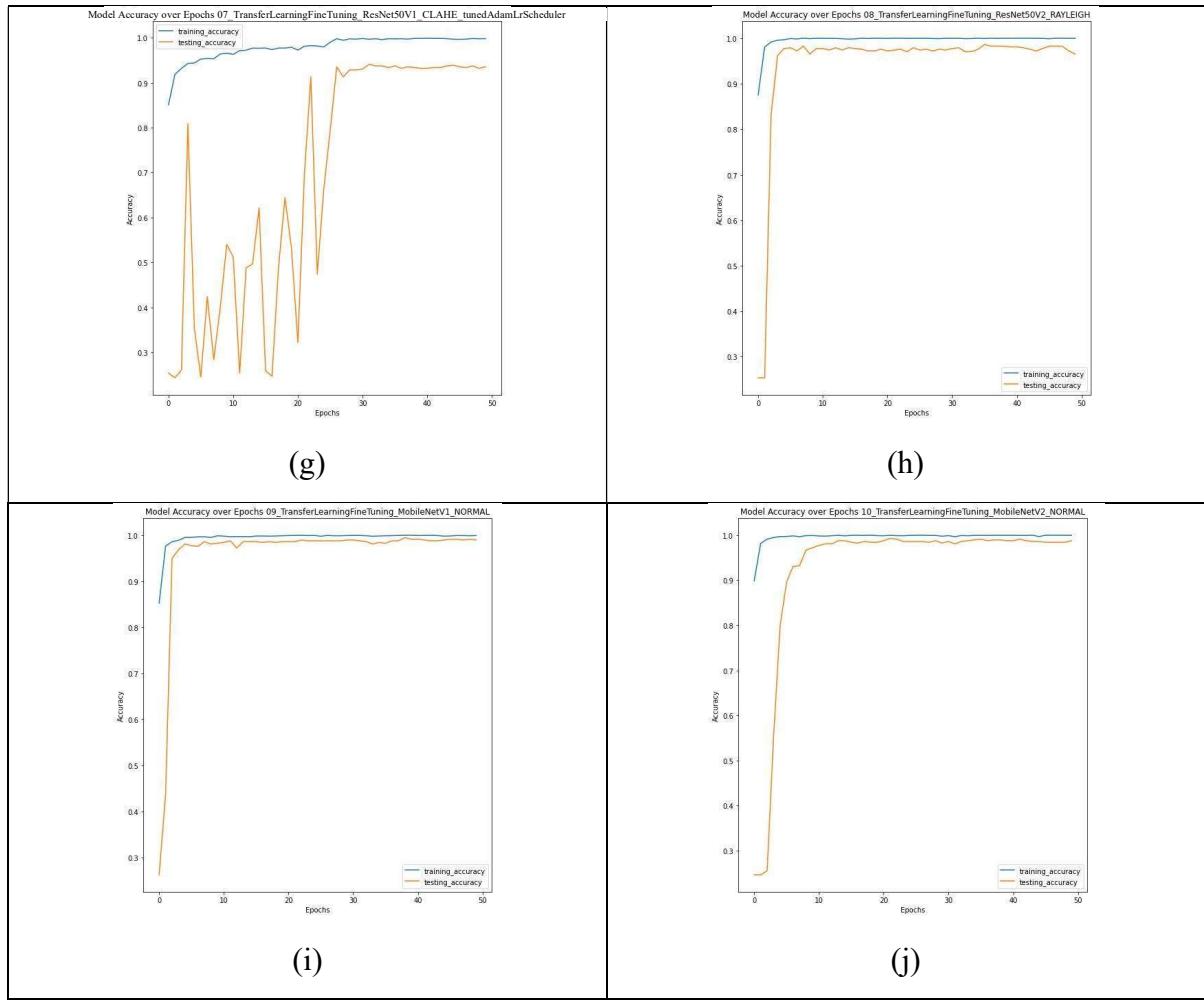


Figure 38: Training Curve of the Best 10 Models from Stage 1.

Figure 38 above depicts the training curve of the best 10 models from stage 1. It can be observed that for all models, the training curve progression is stable though different model groups / architectures have varying steepness of training curves indicating different convergence rate for each model. A steeper training curve indicates that the learning during training converges faster. In contrast, the pattern of the validation curve varies significantly from one model to the other. During the initial 30 epochs of training, it can be observed that models' validation curve fluctuates significantly as opposed to the 30th to 50th epoch range where the validation curve of the models' start to stabilize as training progresses and the gap between the training and validation accuracy curve starts to reduce. This gave an indication that the models were progressively learning from the data and performed better after each epoch. A majority of the models have relatively staggered validation curves except for the best 3 models from stage 1 which are from groups 8, 9 and 10 as seen in Figure 38(h) to Figure 38(j). The validation curves for these models are very smooth with a very narrow gap between training and validation curves

throughout the training process which also reflects the reason why these 3 models performed the best.

4.2.3 Phase 2

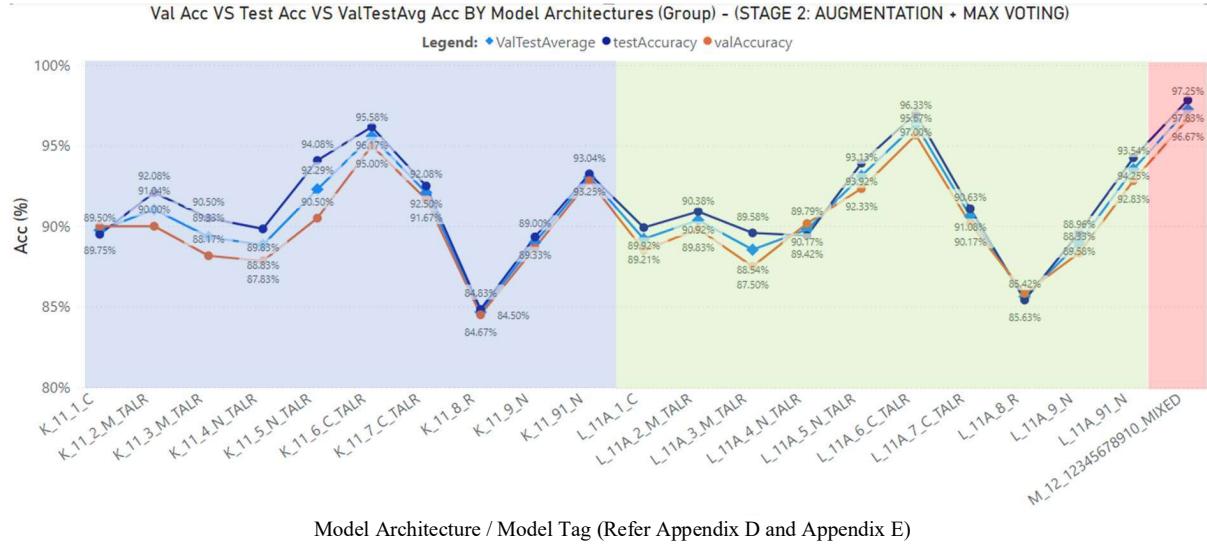


Figure 39: Phase 2 Results

Figure 39 above, compares the validation, test and ValTestAvg accuracies of the 21 models developed in Stage 2. The Group 11 models are categorized under the blue coloured region, 11A under the green coloured region and Group 12 with the red highlighted region. Comparing between groups 11 and 11A, the best 3 models are from K_11_6_C_TALR, K_11_5_N_TALR and K_11_7_C_TALR arranged in descending ValTestAvg order. It can be observed that the 10 models from group 11 and 11A share the same trend of results.

In contrast to Stage 1 results, the top 2 performing model in Stage 2 now come under transfer learning feature extraction category instead of transfer learning fine tuning. However, the top 2 architectures still come under the MobileNet architecture with the positions being MobileNet-V1 followed by Mobilenet-V2 which was in reverse to what was observed in Stage 1. The 3rd architecture still falls under transfer learning fine tuning architecture but is now the ResNet50-V2 architecture instead of the ResNet50-V1 architecture. The self-developed models on the other hand performed in moderation as compared to the rest having achieved quite satisfactory results.

When comparing between model groups 11 against 11A, it can be observed that by increasing the epochs from 50 to 100, that on average half of the models improved and the others deteriorated. However, the variations in results between them are minor. This showed that the development of the model training were matured by 50 epochs. Training the models further for a total of 100 epochs did not significantly improve the models' performance and also took up twice as much the time which did not justify its implementation. Given the circumstances in results, it was then decided that the ensemble model architecture (Group 12) would utilize the model checkpoints from Group 11.

In overall, when comparing models from Group 11, 11A and 12, the best performing model however among all is the ensemble model, M_12_12345678910_MIXED from Group 12 that combined all the best 10 models from group 11. It achieved both higher test and ValTestAvg results with slightly lower validation results when compared to model L_11A_6_C_TALR by a marginal 0.33% only.

Moreover, Model 12 is best generalized as it takes into account the majority of voting by the 10 models as the final prediction. By taking a collective consideration of the results of the 10 models from Group 11, it can be seen that final results improved to finalize the best performing model for Stage 2.

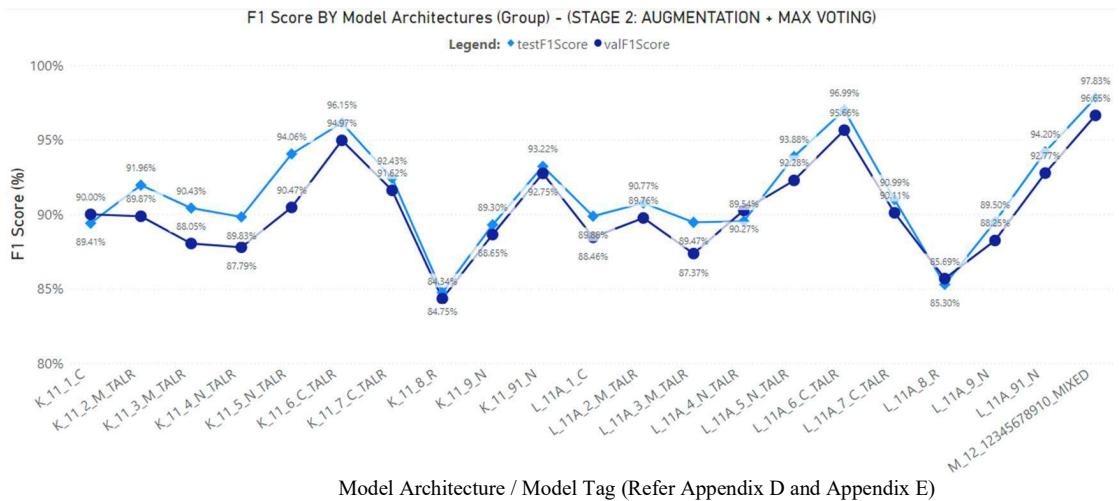


Figure 40: Test F1-Score of the Best 10 Models from Stage 2.

The same pattern of results is also observed in terms of F1 score.

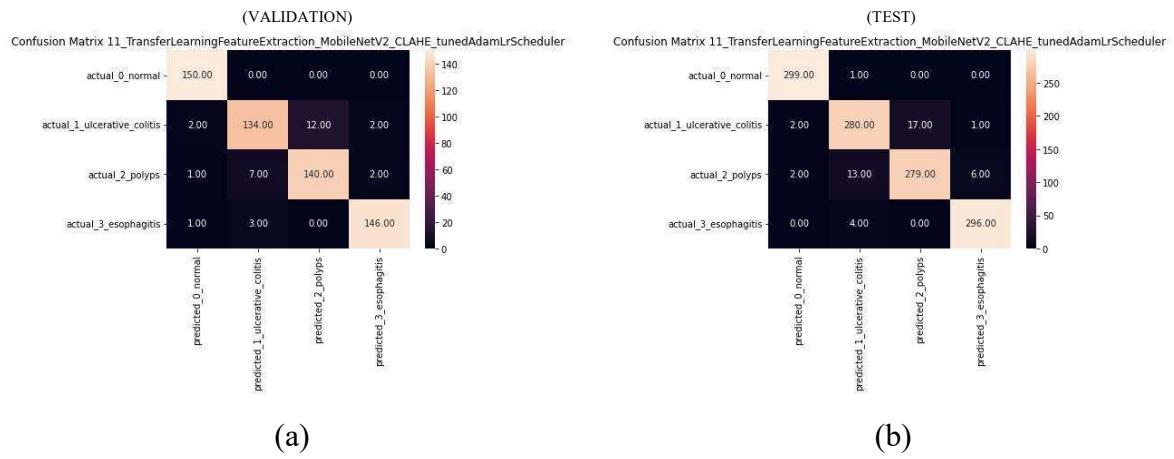


Figure 41: Confusion Matrix of the Best Individual Model from Stage 2 (Model K_11_6_C_TALR) (a) Validation Set, (b) Test Set.

Figure 41 above shows the confusion matrix of the best individual model from Stage 2 i.e., model K_11_6_C_TALR when tested on the Validation and Test set respectively.

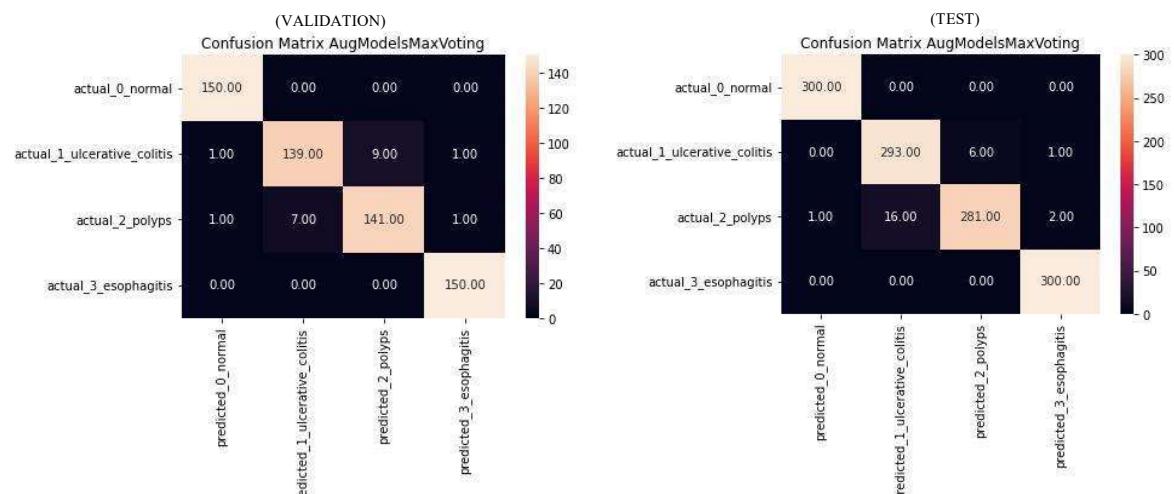
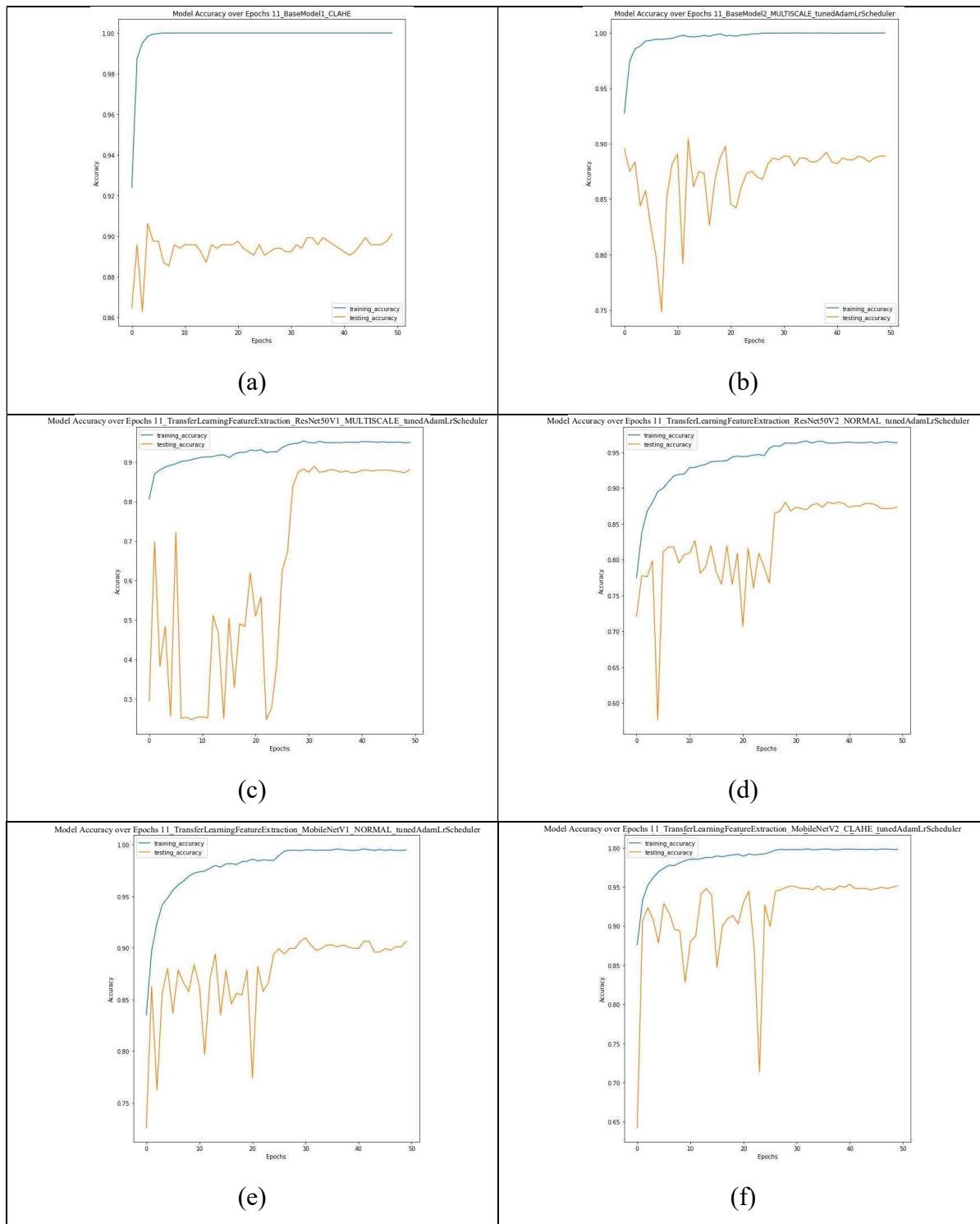


Figure 42: Confusion Matrix of the Ensemble Model from Stage 2 (Model M_12_12345678910_MIXED) (a) Validation Set, (b) Test Set.

Figure 42 above shows the confusion matrix of the ensemble model from Stage 2 i.e., model M_12_12345678910_MIXED when tested on the Validation and Test set respectively.



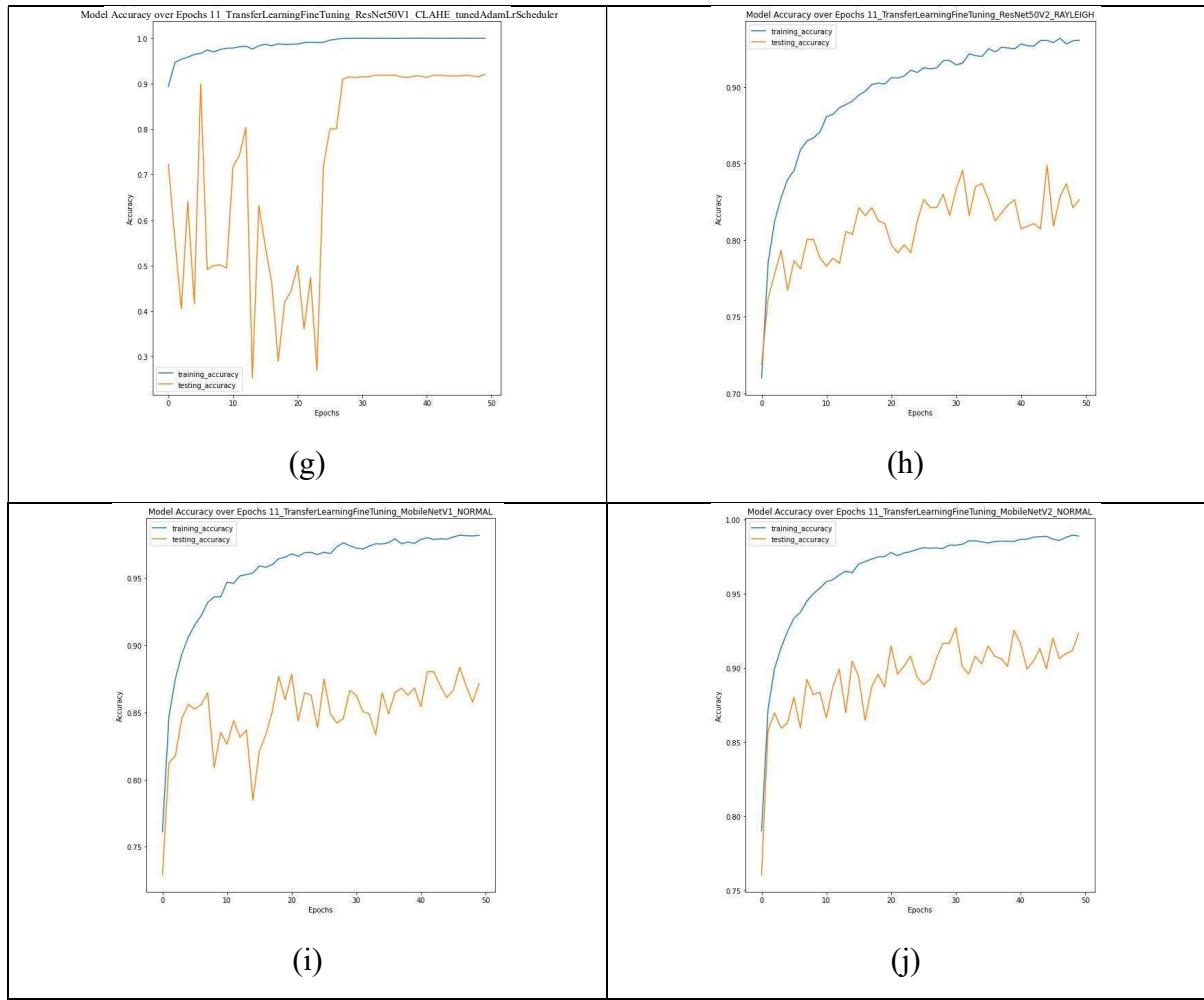


Figure 43: Training Curve of the 10 Models from Group 11, Stage 2.

Figure 43 above depicts the learning curve of the best 10 models from Stage 2. Similar to Stage 1 it can be observed that the training model curve is relatively smooth for all the models and that different models have different gradients in its training curve i.e., different convergence rate. The same pattern as Stage 1 is observed with regards to the validation learning curve albeit the fluctuations in the validation curves are more evident. However, the performances of the models slowly improves and stabilizes as training epoch progresses indicating that the models' learning improved after each epoch.

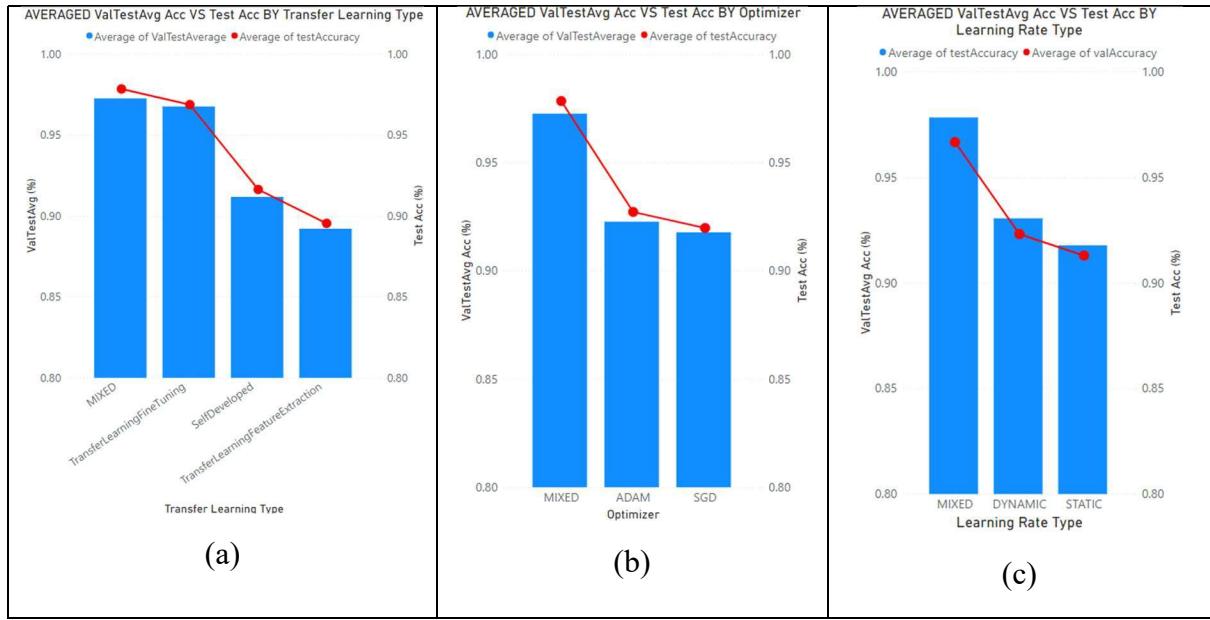


Figure 44: Averaged ValTestAvg Acc and Test Acc of Models in Stage 2 by (a) Transfer Learning Type, (b) Optimizer, (c) Learning Rate Configuration

Figure 44 compares the ValTestAvg performances of the total 91 models developed through Stage 1 and Stage 2 in terms of different model training parameters. MIXED indicates the mixed model parameters in the ensemble method. By looking at Figure 44, on average across all Stage 1 and 2 models it can be observed that the ensemble model performed the best when compared to the rest. It is noticed that the gap of results between TransferLearningFineTuning and SelfDeveloped models in Figure 44(a) are considerably large indicating that the training scheme using ensemble and transfer learning fine tuning are much better as compared to self-developed and transfer learning feature extraction models. With regards to optimizer in Figure 44(b), it can be observed that in overall the ensemble / mixed technique performed the best followed by ADAM and lastly SGD. In terms of learning rate in Figure 44(c), it can be observed that the ensemble method performed the best as well followed by dynamic and static learning rate respectively.

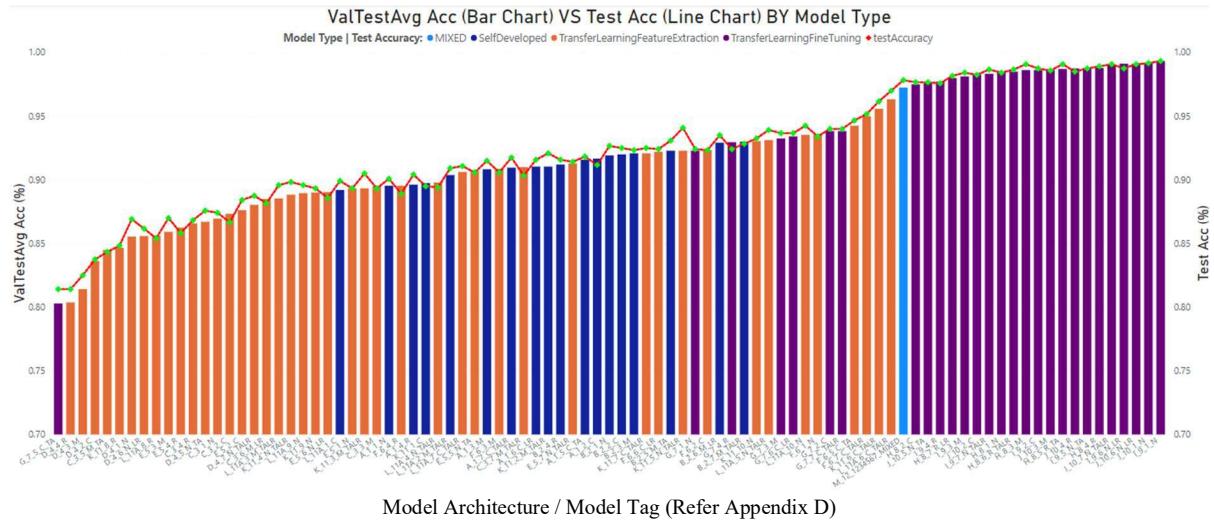


Figure 45: Comparison of ValTestAvg Acc and Test Acc of the Models from Stage 1 and Stage 2. Categorized by Model Type According to Colour.

When looking at the results in overall, we can see that the majority of number of best models are dominated primarily by transfer learning fine tuning and the ensemble models in terms of ValTestAvg. Feature extraction models generally perform poorly as compared to the rest.

4.2.4 Phase 3

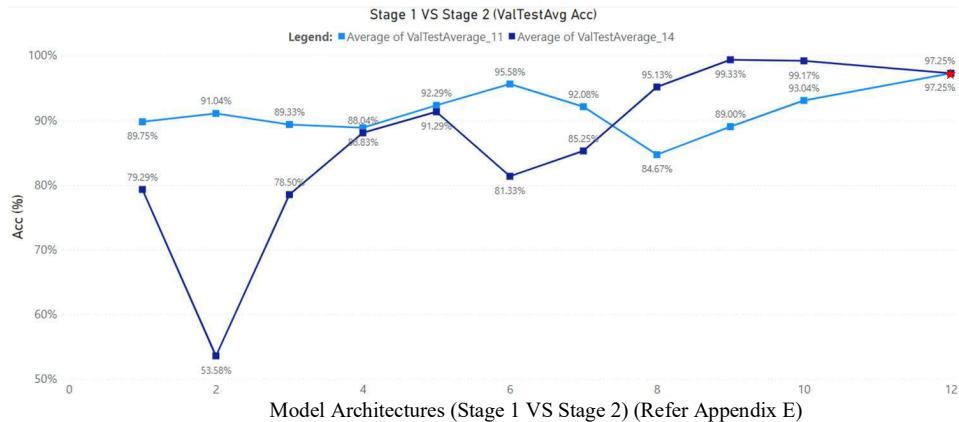


Figure 46: Phase 3 Results.

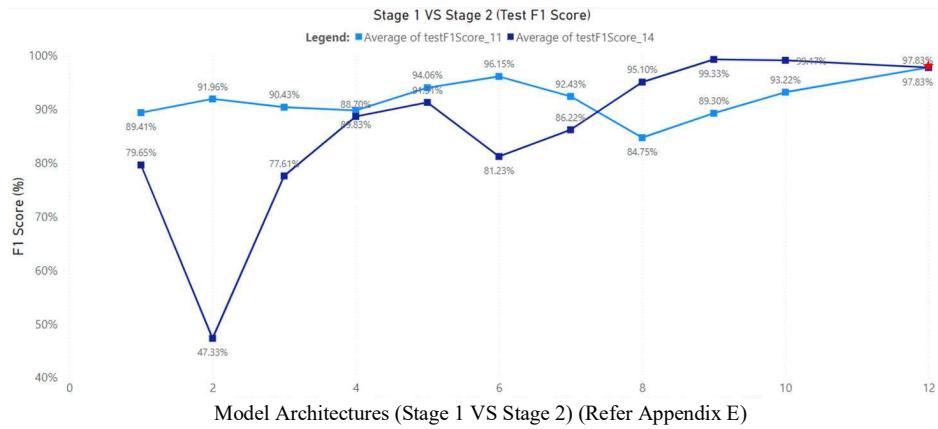


Figure 47: Comparison of Test F1-Score for Models in Stage 1 against Stage 2 Models and Ensemble Model in Phase 3.

In Phase 3, the 10 best models from Stage 1 were tested on the test and validation set of dataset Set 5. These models were not retrained but instead tested on a different set of validation and test set from their original dataset. Their ValTestAvg and test F1-score results were compared to the 11 models from Group 11 and 12.

Figure 46 compares the ValTestAvg of models from Stage 1 against Stage 2. The final data point of model group 12 (marked in red coloured star) displays the final results of the ensemble model for comparison against the other 20 models developed from group 1 – 11. The models are grouped according to their architectures denoted as modelGroupMerged i.e., BaseModel1 from Stage 1 compared against BaseModel1 architecture from Stage 2.

It can be observed that in majority, Stage 2 models outperformed their corresponding Stage 1 model architectures except for models in modelGroupedMerged categories 8 (H_8_4_R), 9 (I_9_1_N), 10 (J_10_1_N) which correspond to the top 3 models from Stage 1. Figure 47 which depicts the corresponding F1 score comparison also shows the same trend of results.

The model from group 12 model performed relatively well as compared to the rest of its counterparts in Stage 1 and Stage 2 except against the top 2 models from stage 1. In overall, the placings in terms of model ValTestAvg performance rank in the manner of I_9_1_N (99.33%), J_10_1_N (99.17%) and finally M_12_12345678910_MIXED (97.25%) at third placing. It is important to note that the ValTestAvg results for models I_9_1_N and J_10_1_N are same to

that in Stage 1 due to the fact that these 2 models were trained, validated, and tested on normal images, without image processing enhancement in Stage 1.

4.2.5 Phase 4

In Phase 4, the 10 best models from Stage 1 and 2 and the ensemble model were tested on the test and validation sets of dataset Set 1, 2, 3 and 4, combined. This totalled to an amount of 2400 validation images and 4800 test images. This is to simulate the situation where multiple images at scale are fed into the system for predictions. These models were not retrained but instead tested on a consolidated dataset with a variety of images. Their ValTestAvg and test F1-score results were compared to one another.

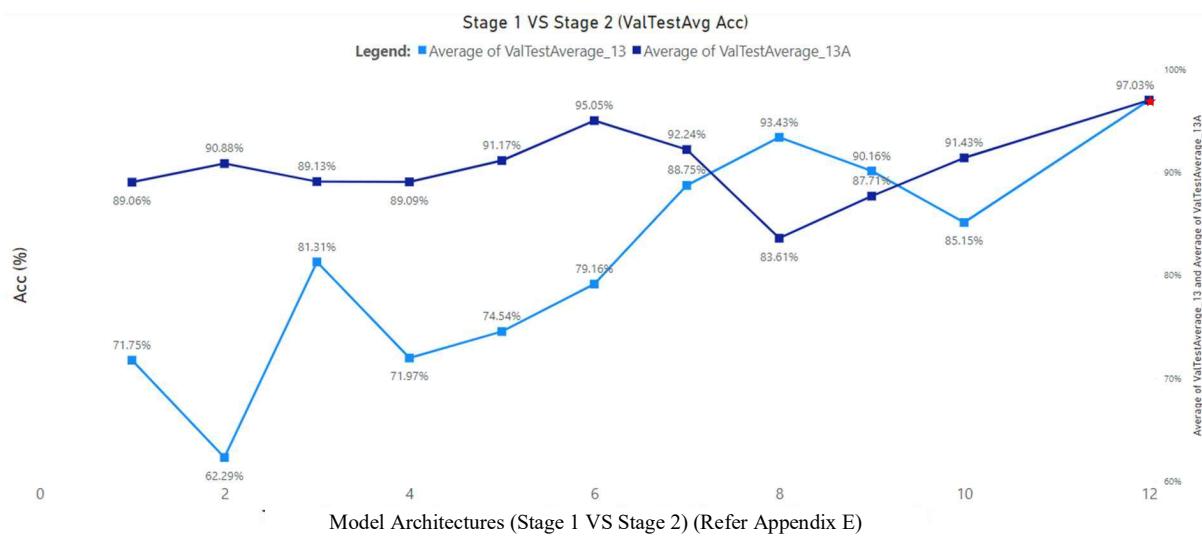


Figure 48: Phase 4 Results.

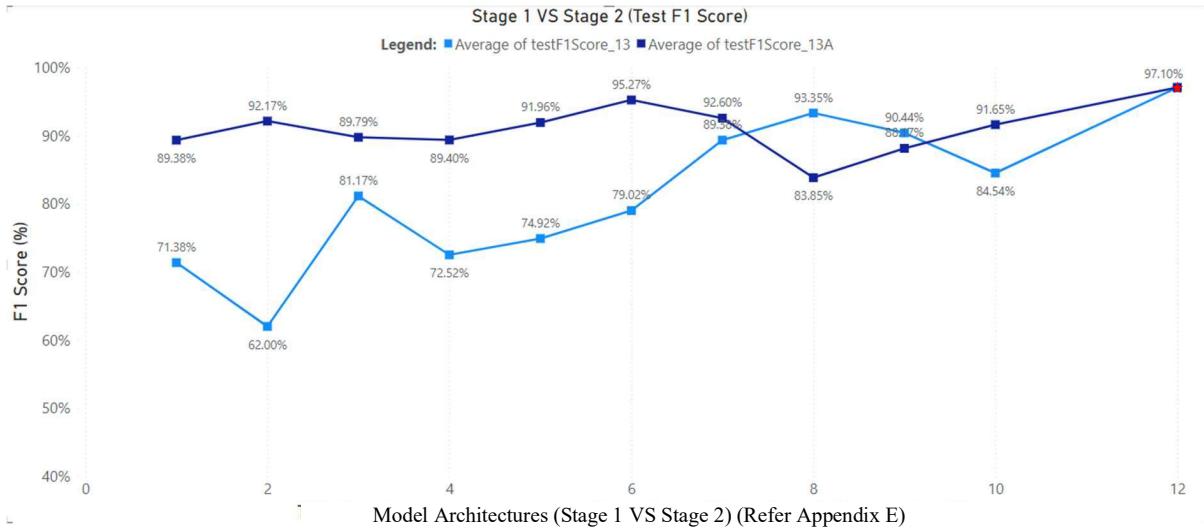


Figure 49: Comparison of Test F1-Score for Models in Stage 1 against Stage 2 Models and Ensemble Model in Phase 4

Figure 48 compares the ValTestAvg of models from Stage 1 against Stage 2. The final data point of model group 12 (marked in red coloured star) displays the final results of the ensemble model for comparison against the other 20 models developed from group 1 – 11. Similar to Phase 3, the models are grouped according to their architectures denoted as modelGroupMerged i.e., BaseModel1 from Stage 1 against BaseModel1 architecture from Stage 2.

It can be observed that in majority here as well that Stage 2 models outperformed their corresponding Stage 1 model architectures except for models in modelGroupedMerged categories 8 (H_8_4_R) and 9 (I_9_1_N) only which correspond to the top 2 models from Stage 1. This was also observed in Phase 3 results analysis. Figure 49 which depicts the corresponding F1 score comparison also shows the same trend of results.

The model from group 12 however, performed the best amongst the rest followed by models from group 6 and 8 respectively. The results in the order from first to third are M_12_12345678910_MIXED (97.03%), K_6_7_C_TALR (95.05%) and finally H_8_4_R (93.43%). Models I & J that performed the best in Phase 3 analysis performed well as well in overall, but not as good as the top 3 models in Phase 4. In this phase of analysis, it can be observed that the top 3 models that performed the best were those that were trained on some type of image processed enhancement data giving the model the generalization capability to handle variety of data. This is especially in the case of the ensemble method.

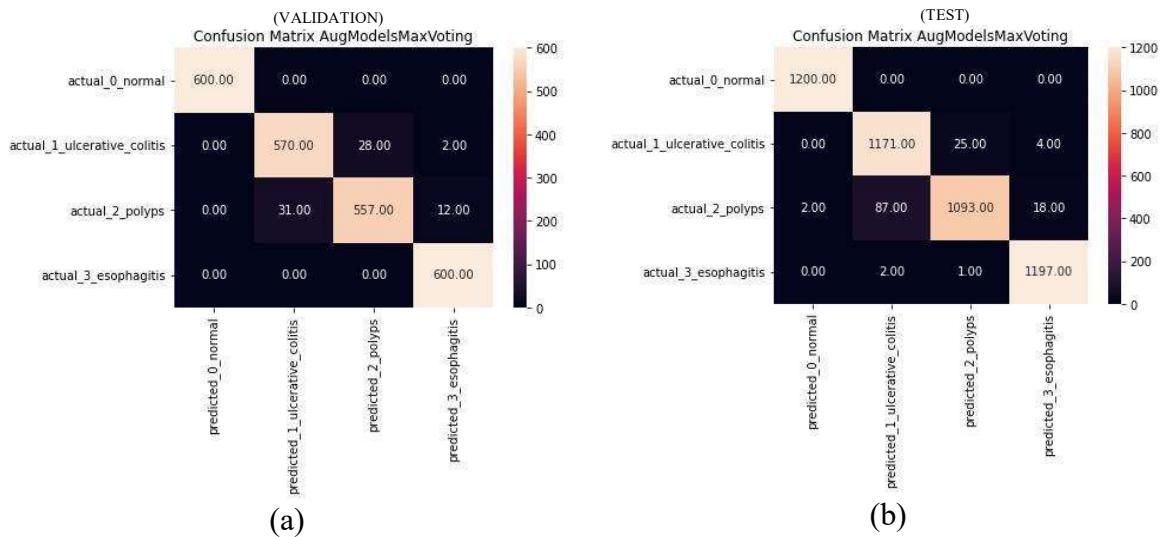


Figure 50: Confusion Matrix of the Ensemble Model from Phase 4 Analysis (Model M_12_12345678910_MIXED) (a) Validation Set, (b) Test Set.

Figure 50 above shows the confusion matrix of the ensemble model from Phase 4 testing i.e., model M_12_12345678910_MIXED when tested on the Validation and Test set of datasets Sets 1, 2, 3 and 4 combined.

CHAPTER 5

DISCUSSION

5.1 Image Processing Enhancement as a Data Pre-Processing Step

The three image processing enhancement techniques proposed in this study were initially developed for underwater image enhancement. However, after detailed analysis, it was realized that the working principals of the algorithm made it generalized for other applications as well. The initial raw dataset was pre-processed with 3 different image processing techniques that not only enhanced simple texture features in the image but also enhanced other aspects such as colour, contrast, and illumination. This is an extension of the work done by (Ayyaz et al., 2022; Hmoud Al-Adhaileh et al., 2021; Pannu et al., 2020). These enhancement techniques apply complex and dynamic image processing algorithms that considers various image parameters extensively. The algorithms applied also cater to the varying pixel distribution of different images in the dataset. (Ayyaz et al., 2022) and (Hmoud Al-Adhaileh et al., 2021) only implemented a single image processing enhancement technique in the training pipeline as a pre-processing step in their proposed solution. The methodology proposed in this study however takes a more detailed approach in using image processing enhancement to train the CNN models. This was done by using the different image enhancement techniques to individually create separate training, validation, and testing datasets. This allowed a more comprehensive solution development by segregating solution development into 2 separate stages of Stage 1 and 2. It also allowed a more detailed analysis of results. In Stage 1, emphasis was put on training the CNN models with different image processing techniques within each group and the performances of the CNN models from within each group were evaluated accordingly. Results showed that some architectures performed better with image processing enhancement whilst others did not. However, across the board, image processing-based models performed with good results. From a qualitative analysis standpoint, from Section 4.1.1 it can be observed that CLAHE enhanced the features in the images the best. The results obtained from Section 4.1.2 also supported this observation. Out of the total 10 best models from Stage 1 that were trained on image processed based datasets, the most commonly used image processing technique was CLAHE. The implementation of image processing enhancement helped answer the second research question defined in Section 1.6. Moreover, the image processing techniques proposed also served as the foundation for performing image data augmentation which is discussed in Section 5.2 below.

5.2 Performing Data Augmentation

By leveraging on the image processing enhancement techniques, an alternative method of image data augmentation was performed in this study. A lot of reviewed literatures typically only implemented geometric based augmentation. Geometric augmentation does help to a certain extent in creating synthetic data especially when used for increasing the size of a small dataset for the CNN models to learn from. This is seen in the studies implemented by (Hmoud Al-Adhaileh et al., 2021; X. Liu et al., 2018; Rustam et al., 2021). However, the focus on these studies were solely on using augmentation to increase the size of the dataset and not on using augmentation to create variety in input data for the model to learn from. Implementing geometric augmentation only on an already sufficiently size dataset may not be as effective especially on WCE images of the GI tract that are symmetrical. Geometric augmentation solely, will only enable the models to learn features of images in different orientations and positions. This may not completely reflect the real-life scenario in scaled-up applications where medical images typically come from different sources of cameras with varying colour, texture, contrast, and illumination. The goal of image augmentation in this study was instead focused on creating a variety in the input data. Hence, this study employed data augmentation that was centred around image colour, texture, contrast, and illumination manipulation. This expanded the spectrum of complexity and variety in image parameters for the models to learn from, enabling them to be more generalized. By using synthetic images that were of these different features, the knowledge gained by the CNN architectures were enriched enabling them to better recognize and capture variations of the same disease in real life images. (Pannu et al., 2020) and (Takiyama et al., 2018) did employ contrast and colour-based augmentation in their solution development. However, these 2 studies focused only on binary classification instead of multiclass classification. Moreover, none of literatures reviewed performed a comparison study of the performance of models trained with and without augmentation as done in Section 4.2. Augmentation solution-based development was done in Stage 2 as detailed in Section 3.3.3. From the results observed in Phase 3 and 4 in Sections 4.2.4 and Section 4.2.5, the augmented-data-trained-models from Stage 2 outperformed the best 10 models in stage 1 in majority. This reinforced that CNN models that were trained with a larger variety of data, i.e., augmented data are more robust. It can be observed progressively the significance of data augmentation through Phase 1 to Phase 4 results analysis detailed in Section 4.2.1 to Section 4.2.5. The implementation of data augmentation helped answer the third research question defined in Section 1.6.

5.3 Network Architectures

A majority of the literatures reviewed focused primarily on self-developed and transfer learning feature extraction models as opposed to transfer learning fine tuning. A total of 6 out of 10 literatures related to transfer learning reviewed were feature extraction-based models as opposed to 3 out of 10 literatures that worked with fine tuning models. Only 2 out of 16 literatures related to GI Tract disease detection reviewed based their solution on an ensemble of models which are papers (B. Liu et al., 2018; Pannu et al., 2020). This literatures however did not performance a comparative study of the ensemble model against other types of model architecture / solution development. As presented in Section 4.2, the superiority of the transfer learning fine tuning models as opposed to its feature extraction counterpart and self-developed models was evident.

Transfer learning models are trained for very general purposes and their application is primarily aimed at being used to tackle the issue of a lack of data whilst still achieving satisfactory results. The common method of applying transfer learning is by transfer learning feature extraction. However, transfer learning feature extraction only enables the model to learn to a certain extent as a majority of layers within the CNN are frozen. This limits the learning capability of the CNN to only the last few layers that were added. In order to truly leverage on the power of transfer learning, it is always best to take the architecture and tune it by training it on new data specific to the application it is intended for. This is where transfer learning fine tuning comes in. Fine tuning enables portions of the network or the entire network to re-learn the new information. However, in most cases transfer learning fine tuning only unlocks a portion of the base model's layer due to computational limitations in retraining the entire model, and to retain some of the valuable knowledge learned previously learned by the base network. Similar to transfer learning feature extraction, additional layers are added to the base network which will also learn new features of the data. The last few layers on the latter end of the base network are typically unfrozen for re-training as the initial layers of the base network are focused on extracting high level features which may not require such adjustments. However, the final layers are focused on extracting finer details in the image which require tuning. Applying transfer learning fine tuning enables an already smart and experienced model to be more specialized in its specific application. It is more capable of handling new data especially medical images related data thus explaining the superiority of transfer learning fine tuning models as opposed to feature extraction in the results section.

Ensemble method was implemented as an experimentation inspired by (B. Liu et al., 2018; Pannu et al., 2020). An ensemble of models consider the predictions of many predictive models and averages or max votes their results. The motivation behind the implementation of this type of model configuration is from the initial analysis of results in Phase 1 and Phase 2 where it was observed that no single model could perform well on all validation and test sets of the total 5 datasets. There was no single generalized model that was all rounded. The development of the ensemble model took inspiration of the concept of teamwork which resonates within machine learning and automation applications as well. In an ensemble of modes, each individual model has their own strengths and weaknesses. In this study, the best 10 models from Stage 2 development i.e., Group 11 were taken, and an ensemble of these models were built. The models were chosen from Stage 2 solution development instead of Stage 1 due to the fact that the Stage 2 models were trained on the augmented dataset, Set 5 which had a bigger variety of images making these models more generalized. Moreover, the comparative results showed in Section 4.2.4 and Section 4.2.5 showed that generally the models from Stage 2 performed better. After developing the ensemble of models, the results obtained was compared against the best 20 models from Stage 1 and 2 on the variety of datasets as done in Section 4.2.4 and Section 4.2.5. It can be observed that ensemble model performed very well across the board, especially in Phase 4 results. The ensemble model basically democratizes the results by taking into account the decisions by many individual models and takes the final decision as a best among all. By doing so, the model decision that is made is more reassuring and it also creates a generalized based solution for the problem which is lacking in the industry. The implementation of data augmentation helped answer the first research question defined in Section 1.6.

5.4 Evaluation Process

To evaluate the performance of the models, both the validation and test set accuracies were taken into consideration. An average of these 2 metrics were calculated and used as the main evaluation parameter. The testing accuracy was not used solely as the main metric for evaluation because during initial results analysis, it was observed that a significant number of models shared the same test accuracy and F1-score value. This was also observed for models within the same group, which made it difficult to select the best 10 models from each group. There were also difficulties in making comparisons between the 10 best models of different groups within the same stage that shared the same test accuracy. Hence, the weightage of results from the validation accuracy was also brought into consideration and the average of both the validation and test accuracies were calculated. Moreover, it is important to note that the

validation and test set used during testing contained 1200 test images for each testing dataset set and 600 validation images for each validation dataset set, amounting to a total of 1800 images which in overall is a sufficient amount of data to properly test and evaluate the performance of the models. The validation accuracy displayed during progressive epoch training was not used as the evaluation metric as only batches of the data were used to produce that value and may not reflect the performance of the models as a whole accurately. The F1-score metric was considered as a secondary metric to validate the ValTestAvg accuracy results obtained. The F1-score considered the harmonic mean of two more important metrics which are precision and recall. The F1-score is typically used as the main metric for evaluation in circumstances where the distribution of data quantity is imbalanced between each class. However, this was not the case in this study where the data was equally distributed between each class making accuracy the better suited candidate as the primary evaluation metric.

Results analysis was broken down into 4 phases. Phase 1 and 2 focused on analysing the performance of the models solely within the same stage of model development i.e., on the same type of test-validation image data the models were trained on. However, Phase 3 and 4 analysis performed a comparative performance evaluation of the models from Stage 1 and 2 as a whole whereby the models were tested with both images within the same stage of development and the other stage. This simulated introducing foreign and unseen types of data to test the models and to analyse their performance and generalization capability. Only during the analysis in Phase 3 and 4 can the robustness of the augmentation and ensemble models be observed as a whole when compared to Stage 1 models, as the comparison of results was done apple-to-apple.

5.5 Comparing Image Size

The models from Group 1 and 2 are both self-developed models with different input image size. On average, the comparison between models in group 1 against group 2 showed that the models in group 2 performed better. From this observation, it can be deduced that a larger input image size is preferred, as larger images have richer and more detailed information within it for interpretation by the CNN. However, it is important to note that the size of the image should be balanced such that it does not compromise with the time and computational complexity of training and running predictions.

5.6 Optimization

There are multiple network parameters to consider when tuning the design of a Convolutional Neural Network. The two chosen parameters in this study were the optimizer and the learning rate which are considerably the 2 biggest architecture components for a CNN model's learning. Optimizers are algorithms that controls the learning of the network. Essentially, the optimizer of the network dictates how the weights and biases within the network are updated for each iteration of training. Learning rate on the other hand is a hyperparameter that controls the amount of change in the weights and biases of the networks in response to the calculated error for each iteration. In simple terms, it controls how fast or slow the model learns. It is used hand in hand with the optimizer. Tuning the learning rate to an optimum value is imperative as very small learning rate may result in a very long and unnecessary training. However, too large a learning rate may result in weights and biases being updated too fast, causing the model to overshoot the global optimum point resulting in an unstable training process. By default, in TensorFlow, the learning rate of Stochastic Gradient Descent based optimizers is 0.01 whereas the default learning rate for Adam optimizer is 0.001.

Almost all of the literatures reviewed in Section 2.2 implemented their solution using SGD as the optimizer and with a static learning rate, i.e., the learning rate of the network remained constant throughout the model training. A static learning rate however restricts the model's ability in controlling how fast it learns each epoch. This study expanded the tuning of the network parameters by taking into consideration the Adam optimizer and the dynamic learning rate on top of the commonly used SGD and static learning rate as detailed in Section 3.3.4.7 above.

Adam was selected as the alternative optimizer as it is a common alternative to SGD and is also famously used in the field for developing CNN classification-based problems. The dynamic learning rate was implemented by using the LrScheduler in some of the models' training. The inspiration to implement the LrScheduler came about when it was observed that the learning curve of the models especially the validation curve were too unstable. This was a clear indication that the model's learning rate was too large causing the model to overshoot from the global optimum. By implementing the LrScheduler the models were allowed to go through a transient state first with a larger learning rate which then stabilized as the learning rate of the model progressively diminishes by half after every epoch enabling the model to more accurately

reach the global optimum. Moreover, a combination of these 2 parameters were also considered to observe if a synergistic effect of both parameters provided the model with better performance.

Taking the training curve of the ResNet50-V1 transfer learning feature extraction model from Group 3 for example, the best performing model was that with the combined Adam + LrScheduler. Comparing between the standard SGD model, Adam only, LrScheduler only and the Adam + LrScheduler models from Group 3 in Figure 51 below, it can be observed that models implemented with the LrScheduler had relatively more stable training curves as opposed to those without. This is particularly more evident with the Adam + LrScheduler models.

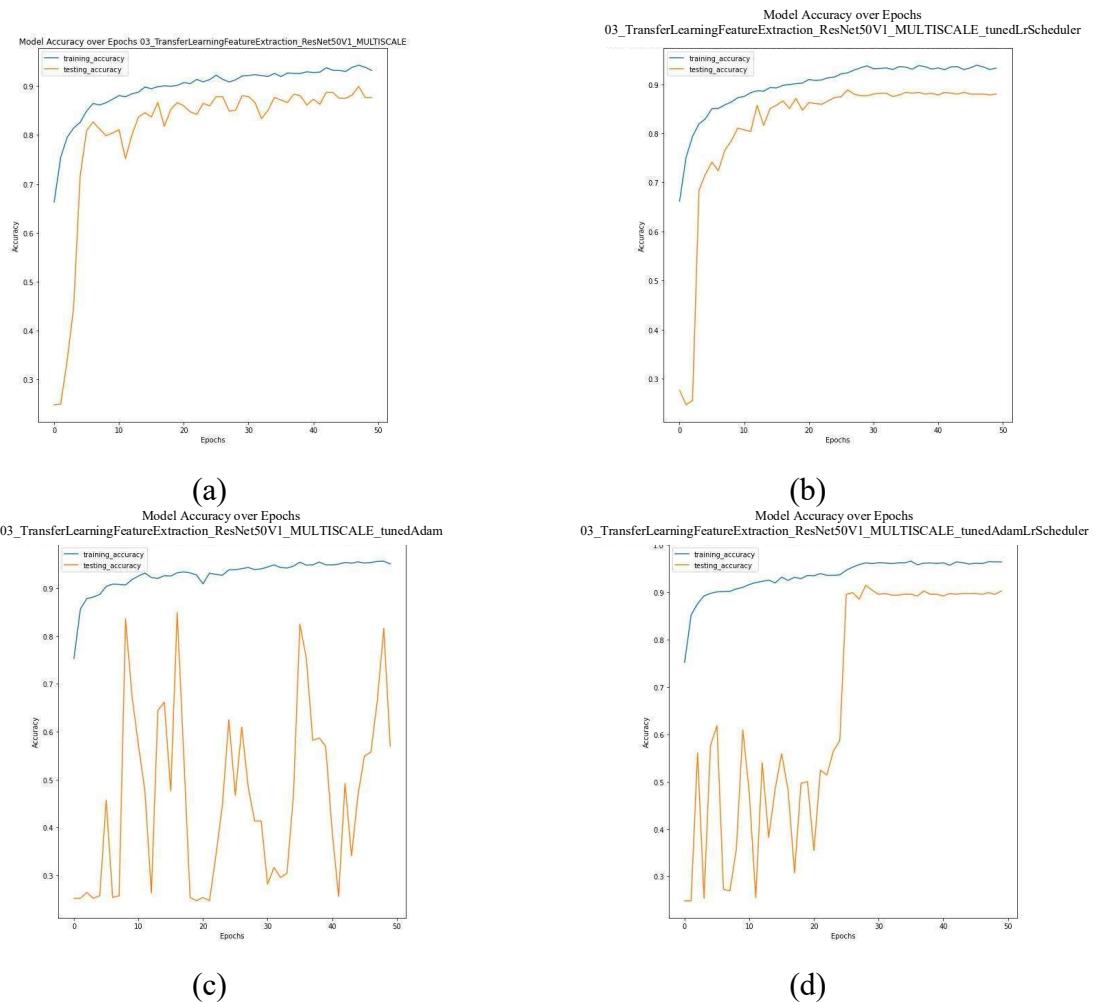


Figure 51: Comparison of Model 03_TransferLearningFeatureExtraction_ResNet50V1's Training Curve Before and After Optimization (a) Without Optimization, (b) With Dynamic Learning Rate (LrScheduler), (c) With Adam Optimizer, (d) With Adam Optimizer and Dynamic Learning Rate (LrScheduler).

It can also be clearly observed that out of the 10 models in stage 1, 6 out of the 10 groups' best model were those tuned with Adam + LrScheduler combined. This showed that tuning the models with individual parameters did not contribute much to improving the models' performance as opposed to combining them together to have a synergistic effect.

CHAPTER 6

CONCLUSION

6.1 Conclusion of Study

The aim of the study was to design and develop a generalized multiclass CNN classification model for detecting and classifying various GI tract diseases from WCE GI tract images. This was further refined into 3 objectives as defined in Section 1.8. In order to achieve the objectives outlined, the study had undertaken several major steps as defined below:

1. Performed a detailed literature review of the subject domain and related works. This was done to better understand the subject domain, the latest related works being done with regards to solution development and lastly as a source for ideas for developing the solution. This was defined in CHAPTER 2 above.
2. A raw, suitably sized, equally balanced, and reliable dataset was sourced from Kaggle. The raw data contained WCE images from 4 different classes obtained from different sources i.e., taken from different cameras.
3. An end-to-end detailed methodology was then defined to develop the CNN solution. The methodology was defined in CHAPTER 3.
4. 3 different image processing enhancements techniques were adopted, implemented on the dataset and analysed. The image enhancement techniques focused on more complex image manipulation such as colour, texture, contrast and illumination. From this, 4 different datasets were created i.e., datasets Set 1 to Set 4 as defined in Section 3.3.3 to be used for the solution development.
5. Leveraging on the image processing enhanced images, an additional augmentation dataset, Set 5 was created as defined in Section 3.3.3 to simulate a variety of data used for training the CNN models. This augmentation technique was centred around colour contrast and illumination augmentation as opposed to commonly used geometric augmentation. The augmentation dataset was created with varying the types of images that the CNN models learned in order to improve its generalization capability.
6. A total of 91 CNN models were developed and this was broken down into 2 stages of Stage 1 and Stage 2. The CNN solution development encompassed various network architectures, transfer learning techniques, mode of solutions, network architecture

parameters and the 5 sets of datasets prepared. A detailed process flow of this was detailed in Section 3.3.4.

7. The results were collected, interpreted, compared, and critically analysed to produce insights detailed in CHAPTER 4 and CHAPTER 5. Results analysis was performed in 4 phases to extensively review the performances of the models developed in Stage 1 and Stage 2 against different types of data.

Based on the analyses performed in Phase 1 and Phase 2, the I_9_1_N (09_TransferLearningFineTuning_MobileNetV1_NORMAL) model was the best performing model in Stage 1, the ensemble model i.e., model M_12_12345678910_MIXED (12_AugVotingModels) was the best performing model for Stage 2. The best performing model in Phase 3 analysis was I_9_1_N model as well. The best performing model in Phase 4 analysis was the M_12_12345678910_MIXED model. In overall, it was observed that the Stage 2 models performed better against Stage 1 models. The results concluded that performing image processing does indeed help in improving the model's performance. However, its effects are dependent on the type of network architecture used. Moreover, results showed that training the CNN architectures with a wide variety of augmented data as done in Stage 2 does indeed improve the model's generalization capability. This is evident especially in Phase 3 and phase 4 analysis. The best performing in overall weighing in on performance and generalization capability is the ensemble model from Group 12, as it performed the best considerably across all phases of analysis. Every aspect of the solution development was justified and well documented in this document as part of the main documentation or as part of the Appendix. Moreover, the research questions and problem statements defined in this study were answered clearly in CHAPTER 4 and CHAPTER 5. Therefore, it can be concluded that the aim and objectives of the project have been met.

6.2 Limitations and Future Works

There are several areas however with room for improvements and future works that were identified.

The first area is to automate the entire developed methodology and CNN solution into a Machine Learning pipeline that can be packaged, deployed, and tested in real life applications.

Another potential area of improvement is to increase the number of classes or diseases to detect. This would certainly help the feasibility of such a system in the medical industry. Presently, the number of diseases that are capable of being detected are 3 which are Polyps, Ulcerative Colitis and Esophagitis. Given the availability of more annotated images, this can be further expanded into a greater number of classes of diseases for the CNN model to detect.

The present study had only implemented and compared 3 types of image processing enhancement techniques to improve the input image quality. The potential of using image processing enhancement is evident and hence more research should be poured into exploring and studying even more techniques especially those that are prominent in the medical imaging domain. Image restoration techniques could be a potential alternative though a more detailed implementation would be required.

Detecting these lesions and diseases shows the impressive capability of Deep Learning CNN models. Extending detection to localisation of these lesions within an image using object detection models such as YOLO, Single Shot Detection and FasterRNN would certainly be next step of progression for such a CAD system.

Given additional time and computational resources, achieving the recommended future works above will certainly be possible.

REFERENCES

- Abbasi, W. A., & Minhas, F. U. A. A. (2016). Issues in performance evaluation for host-pathogen protein interaction prediction. *Journal of Bioinformatics and Computational Biology*, 14(3), 1–17. <https://doi.org/10.1142/S0219720016500116>
- Amro. (2022). *CLAHE (Contrast Limited Adaptive Histogram Equalization)*. GitHub. http://amroamroamro.github.io/mexopencv/opencv/clah_demo_gui.html
- Apostolopoulos, I. D., & Mpesiana, T. A. (2020). Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2), 635–640. <https://doi.org/10.1007/s13246-020-00865-4>
- Ayyaz, M. S., Lali, M. I. U., Hussain, M., Rauf, H. T., Alouffi, B., Alyami, H., & Wasti, S. (2022). Hybrid deep learning model for endoscopic lesion detection and classification using endoscopy videos. *Diagnostics*, 12(1). <https://doi.org/10.3390/diagnostics12010043>
- Bardina, M., Volodchenko, I., & Sukhov, M. (2020). Application Development for Determining the Author of a Picture Based on Neural Network Technologies. *Proceedings - 2020 International Russian Automation Conference, RusAutoCon 2020, September*, 1042–1046. <https://doi.org/10.1109/RusAutoCon49822.2020.9208200>
- Chaturvedi, S. S., Gupta, K., & Prasad, P. S. (2021). Skin lesion analyser: an efficient seven-way multi-class skin cancer classification using mobilenet. *Advances in Intelligent Systems and Computing*, 1141, 165–176. https://doi.org/10.1007/978-981-15-3383-9_15
- Chen, P. J., Lin, M. C., Lai, M. J., Lin, J. C., Lu, H. H. S., & Tseng, V. S. (2018). Accurate Classification of Diminutive Colorectal Polyps Using Computer-Aided Analysis. *Gastroenterology*, 154(3), 568–575. <https://doi.org/10.1053/j.gastro.2017.10.010>
- Conference, I. I. (2017). *CONVOLUTIONAL NEURAL NETWORKS FOR INTESTINAL HEMORRHAGE DETECTION IN WIRELESS CAPSULE ENDOSCOPY IMAGES* Key Laboratory of Complex Systems Modeling and Simulation , School of Computer Science and Technology , Hangzhou Dianzi University School of Communicati. July.
- Corchs, S., & Schettini, R. (2010). Underwater image processing: State of the art of restoration and image enhancement methods. *Eurasip Journal on Advances in Signal Processing*, 2010. <https://doi.org/10.1155/2010/746052>
- Da Nóbrega, R. V. M., Peixoto, S. A., Da Silva, S. P. P., & Filho, P. P. R. (2018). Lung Nodule Classification via Deep Transfer Learning in CT Lung Images. *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, 2018-June, 244–249. <https://doi.org/10.1109/CBMS.2018.00050>
- Demir, A., Yilmaz, F., & Kose, O. (2019). Early detection of skin cancer using deep learning architectures: Resnet-101 and inception-v3. *TIPTEKNO 2019 - Tip Teknolojileri Kongresi*, 2019-Janua(February 2020), 2–6.

<https://doi.org/10.1109/TIPTEKNO47231.2019.8972045>

- Du, W., Rao, N., Liu, D., Jiang, H., Luo, C., Li, Z., Gan, T., & Zeng, B. (2019). Review on the Applications of Deep Learning in the Analysis of Gastrointestinal Endoscopy Images. *IEEE Access*, 7, 142053–142069. <https://doi.org/10.1109/ACCESS.2019.2944676>
- Ellahyani, A., El Jaafari, I., & Charfi, S. (2021). Computer-aided diagnosis system for ulcer detection in wireless capsule endoscopy images. *Journal of Physics: Conference Series*, 1743(1), 6–10. <https://doi.org/10.1088/1742-6596/1743/1/012016>
- Emam, A. Z., Ali, Y. A., & Ben Ismail, M. M. (2015). Adaptive features extraction for Capsule Endoscopy (CE) video summarization. *Proceedings - International Conference on Computer Vision and Image Analysis Applications, ICCVIA 2015, October*. <https://doi.org/10.1109/ICCVIA.2015.7351879>
- Fei-Fei, L., Deng, J., & Li, K. (2010). ImageNet: Constructing a large-scale image database. *Journal of Vision*, 9(8), 1037–1037. <https://doi.org/10.1167/9.8.1037>
- Ganz, M., Yang, X., & Slabaugh, G. (2012). Automatic segmentation of polyps in colonoscopic narrow-band imaging data. *IEEE Transactions on Biomedical Engineering*, 59(8), 2144–2151. <https://doi.org/10.1109/TBME.2012.2195314>
- Ghani, A. S. A., & Isa, N. A. M. (2015). Underwater image quality enhancement through composition of dual-intensity images and Rayleigh-stretching. *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin, 2015-Febru*(February), 219–220. <https://doi.org/10.1109/ICCE-Berlin.2014.7034265>
- Halalli, B., & Makandar, A. (2018). Computer Aided Diagnosis - Medical Image Analysis Techniques. *Breast Imaging*. <https://doi.org/10.5772/intechopen.69792>
- He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, J. (2015). Deep Residual Learning for Image Recognition. *ArXiv*. <https://doi.org/10.48550/ARXIV.1512.03385>
- He, J. Y., Wu, X., Jiang, Y. G., Peng, Q., & Jain, R. (2018). Hookworm Detection in Wireless Capsule Endoscopy Images with Deep Learning. *IEEE Transactions on Image Processing*, 27(5), 2379–2392. <https://doi.org/10.1109/TIP.2018.2801119>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9908 LNCS, 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
- Hitam, M. S., Awalludin, E. A., Jawahir Hj Wan Yussof, W. N., & Bachok, Z. (2013). Mixture contrast limited adaptive histogram equalization for underwater image enhancement. *International Conference on Computer Applications Technology, ICCAT 2013*. <https://doi.org/10.1109/ICCAT.2013.6522017>
- Hmoud Al-Adhaileh, M., Mohammed Senan, E., Alsaade, F. W., Aldhyani, T. H. H., Alsharif, N., Abdullah Alqarni, A., Uddin, M. I., Alzahrani, M. Y., Alzain, E. D., & Jadhav, M. E. (2021). Deep Learning Algorithms for Detection and Classification of Gastrointestinal Diseases. *Complexity*, 2021. <https://doi.org/10.1155/2021/6170416>

- Hornbuckle, W. E., Simpson, K. W., & Tennant, B. C. (2008). *Gastrointestinal Function SALIVARY SECRETIONS FUNCTION. I*, 413–457.
- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.
- Huang, D., Wang, Y., Song, W., Sequeira, J., & Mavromatis, S. (2018). Shallow-Water Image Enhancement Using Relative Global Histogram Stretching Based on Adaptive Parameter Acquisition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10704 LNCS, 453–465. https://doi.org/10.1007/978-3-319-73603-7_37
- Institute, N. C. (2022). *Gastrointestinal tract*. National Institutes of Health. <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/gastrointestinal-tract>
- Iqbal, K., Odetayo, M., James, A., Salam, R. A., & Talib, A. Z. H. (2010). Enhancing the low quality images using Unsupervised Colour Correction Method. *2010 IEEE International Conference on Systems, Man and Cybernetics*, 1703–1709. <https://doi.org/10.1109/ICSMC.2010.5642311>
- Jia, X., & Meng, M. Q. H. (2016). A deep convolutional neural network for bleeding detection in Wireless Capsule Endoscopy images. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2016-Octob*, 639–642. <https://doi.org/10.1109/EMBC.2016.7590783>
- Jintasuttisak, T., & Intajag, S. (2014). Color retinal image enhancement by Rayleigh contrast-limited adaptive histogram equalization. *International Conference on Control, Automation and Systems, Iccas*, 692–697. <https://doi.org/10.1109/ICCAS.2014.6987868>
- Karkanis, S. A., Iakovidis, D. K., Maroulis, D. E., Karras, D. A., & Tzivras, M. (2003). Computer-Aided Tumor Detection in Endoscopic Video Using Color Wavelet Features. *IEEE Transactions on Information Technology in Biomedicine*, 7(3), 141–152. <https://doi.org/10.1109/TITB.2003.813794>
- Kumar, N., Gupta, M., Gupta, D., & Tiwari, S. (2021). Novel deep transfer learning model for COVID-19 patient detection using X-ray chest images. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-021-03306-6>
- Lee, J., Pant, S. R., & Lee, H. (2015). An Adaptive Histogram Equalization Based Local Technique for Contrast Preserving Image Enhancement. *International Journal of Fuzzy Logic and Intelligent Systems*, 15, 35–44. <https://doi.org/10.5391/IJFIS.2015.15.1.35>
- Lee, J. Y., Choi, S. H., & Chung, J. W. (2019). Automated classification of the tympanic membrane using a convolutional neural network. *Applied Sciences (Switzerland)*, 9(9). <https://doi.org/10.3390/app9091827>
- Li, B., & Meng, M. Q. H. (2012). Tumor recognition in wireless capsule endoscopy images using textural features and SVM-based feature selection. *IEEE Transactions on Information Technology in Biomedicine*, 16(3), 323–329. <https://doi.org/10.1109/TITB.2012.2185807>

- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- Liu, B., Yao, K., Huang, M., Zhang, J., Li, Y., & Li, R. (2018). Gastric Pathology Image Recognition Based on Deep Residual Networks. *Proceedings - International Computer Software and Applications Conference*, 2, 408–412. <https://doi.org/10.1109/COMPSAC.2018.10267>
- Liu, X., Wang, C., Bai, J., & Liao, G. (2020). Fine-tuning Pre-trained Convolutional Neural Networks for Gastric Precancerous Disease Classification on Magnification Narrow-band Imaging Images. *Neurocomputing*, 392(xxxx), 253–267. <https://doi.org/10.1016/j.neucom.2018.10.100>
- Liu, X., Wang, C., Hu, Y., Zeng, Z., Bai, J., & Liao, G. (2018). Transfer Learning with Convolutional Neural Network for Early Gastric Cancer Classification on Magnifying Narrow-Band Imaging Images. *Proceedings - International Conference on Image Processing, ICIP*, 1388–1392. <https://doi.org/10.1109/ICIP.2018.8451067>
- MONTALBO, F. J. (2022). *WCE Curated Colon Disease Dataset Deep Learning*. Kaggle. <https://www.kaggle.com/datasets/francismon/curated-colon-dataset-for-deep-learning>
- Montalbo, F. J. P. (2022). Diagnosing gastrointestinal diseases from endoscopy images through a multi-fused CNN with auxiliary layers, alpha dropouts, and a fusion residual block. *Biomedical Signal Processing and Control*, 76, 103683. <https://doi.org/https://doi.org/10.1016/j.bspc.2022.103683>
- Namatēvs, I. (2018). Deep Convolutional Neural Networks: Structure, Feature Extraction and Training. *Information Technology and Management Science*, 20(1), 40–47. <https://doi.org/10.1515/itms-2017-0007>
- National Health Service (NHS). (2022). *Ulcerative colitis*. National Health Service (NHS). <https://www.nhs.uk/conditions.ulcerative-colitis/#:~:text=Ulcerative%20colitis%20is%20a%20long,can%20bleed%20and%20produce%20pus>.
- Pannu, H. S., Ahuja, S., Dang, N., Soni, S., & Malhi, A. K. (2020). Deep learning based image classification for intestinal hemorrhage. *Multimedia Tools and Applications*, 79(29–30), 21941–21966. <https://doi.org/10.1007/s11042-020-08905-7>
- Pei, M., Wu, X., Guo, Y., & Fujita, H. (2017). Small bowel motility assessment based on fully convolutional networks and long short-term memory. *Knowledge-Based Systems*, 121, 163–172. <https://doi.org/10.1016/j.knosys.2017.01.023>
- Pogorelov, K., Randel, K. R., Griwodz, C., Eskeland, S. L., De Lange, T., Johansen, D., Spampinato, C., Dang-Nguyen, D. T., Lux, M., Schmidt, P. T., Riegler, M., & Halvorsen, P. (2017). Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. *Proceedings of the 8th ACM Multimedia Systems Conference, MMSys 2017*, 164–169. <https://doi.org/10.1145/3083187.3083212>
- Qu, J., Hiruta, N., Terai, K., Nosato, H., Murakawa, M., & Sakanashi, H. (2018). Gastric

Pathology Image Classification Using Stepwise Fine-Tuning for Deep Neural Networks. *Journal of Healthcare Engineering*, 2018. <https://doi.org/10.1155/2018/8961781>

Rustum, F., Siddique, M. A., Siddiqui, H. U. R., Ullah, S., Mehmood, A., Ashraf, I., & Choi, G. S. (2021). Wireless Capsule Endoscopy Bleeding Images Classification Using CNN Based Model. *IEEE Access*, 9, 33675–33688. <https://doi.org/10.1109/ACCESS.2021.3061592>

Salem, N., Malik, H., & Shams, A. (2019). Medical image enhancement based on histogram algorithms. *Procedia Computer Science*, 163, 300–311. <https://doi.org/10.1016/j.procs.2019.12.112>

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. <https://doi.org/10.1109/CVPR.2018.00474>

Saric, M., Russo, M., Stella, M., & Sikora, M. (2019). CNN-based Method for Lung Cancer Detection in Whole Slide Histopathology Images. *2019 4th International Conference on Smart and Sustainable Technologies, SpliTech 2019*, 14–17. <https://doi.org/10.23919/SpliTech.2019.8783041>

Seguí, S., Drozdzal, M., Pascual, G., Radeva, P., Malagelada, C., Azpiroz, F., & Vitrià, J. (2016). Generic feature learning for wireless capsule endoscopy analysis. *Computers in Biology and Medicine*, 79, 163–172. <https://doi.org/10.1016/j.combiomed.2016.10.011>

Segui, Santí, Drozdzal, M., Vilarino, F., Malagelada, C., Azpiroz, F., Radeva, P., & Vitria, J. (2012). Categorization and segmentation of intestinal content frames for wireless capsule endoscopy. *IEEE Transactions on Information Technology in Biomedicine*, 16(6), 1341–1352. <https://doi.org/10.1109/TITB.2012.2221472>

Segui, Santi, Drozdzal, M., Zaytseva, E., Malagelada, C., Azpiroz, F., Radeva, P., & Vitria, J. (2014). Detection of wrinkle frames in endoluminal videos using betweenness centrality measures for images. *IEEE Journal of Biomedical and Health Informatics*, 18(6), 1831–1838. <https://doi.org/10.1109/JBHI.2014.2304179>

Sharif, M., Attique Khan, M., Rashid, M., Yasmin, M., Afza, F., & Tanik, U. J. (2021). Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. *Journal of Experimental and Theoretical Artificial Intelligence*, 33(4), 577–599. <https://doi.org/10.1080/0952813X.2019.1572657>

Shin, Y., & Balasingham, I. (2017). Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 3277–3280. <https://doi.org/10.1109/EMBC.2017.8037556>

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>

Siegel, R. L., Miller, K. D., Fedewa, S. A., Ahnen, D. J., Meester, R. G. S., Barzi, A., & Jemal, A. (2017). Colorectal cancer statistics, 2017. *CA: A Cancer Journal for Clinicians*, 67(3), 177–193. <https://doi.org/10.3322/caac.21395>

- Silva, J., Histace, A., Romain, O., Dray, X., & Granado, B. (2014). Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer. *International Journal of Computer Assisted Radiology and Surgery*, 9(2), 283–293. <https://doi.org/10.1007/s11548-013-0926-3>
- Take, I., Shi, Q., & Zhong, Y. (2015). Progress with each passing day : role of endoscopy in early gastric cancer. *Translational Gastrointestinal Cancer*, 4(6), 423–428. <https://doi.org/10.3978/j.issn.2224-4778.2015.09.04>
- Takiyama, H., Ozawa, T., Ishihara, S., Fujishiro, M., Shichijo, S., Nomura, S., Miura, M., & Tada, T. (2018). Automatic anatomical classification of esophagogastroduodenoscopy images using deep convolutional neural networks. *Scientific Reports*, 8(1), 1–8. <https://doi.org/10.1038/s41598-018-25842-6>
- Team, T. A. C. S. M. and E. (2020). About Colorectal Cancer; What Is Colorectal Cancer ? *American Cancer Society*, 1–15. <https://www.cancer.org/cancer/colon-rectal-cancer/about/what-is-colorectal-cancer.html#references>
- Tuba, E., Tuba, M., & Jovanovic, R. (2017). An algorithm for automated segmentation for bleeding detection in endoscopic images. *Proceedings of the International Joint Conference on Neural Networks, 2017-May*, 4579–4586. <https://doi.org/10.1109/IJCNN.2017.7966437>
- Van Hieu, N., & Hien, N. L. H. (2020). Recognition of plant species using deep convolutional feature extraction. *International Journal on Emerging Technologies*, 11(3), 904–910.
- Wang, S., Xia, X., Ye, L., & Yang, B. (2021). Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals*, 11(3), 1–23. <https://doi.org/10.3390/met11030388>
- Wang, W., Li, Y., Zou, T., Wang, X., You, J., & Luo, Y. (2020). A novel image classification approach via dense-mobilenet models. *Mobile Information Systems*, 2020. <https://doi.org/10.1155/2020/7602384>
- Xie, S., & Tu, Z. (2017). Holistically-Nested Edge Detection. *International Journal of Computer Vision*, 125(1–3), 3–18. <https://doi.org/10.1007/s11263-017-1004-z>
- Yao, Y., & Doretto, G. (2010). Boosting for transfer learning with multiple sources. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, December*, 1855–1862. <https://doi.org/10.1109/CVPR.2010.5539857>
- Zhou, H., & Sun, Q. (2020). Research on Principle and Application of Convolutional Neural Networks. *IOP Conference Series: Earth and Environmental Science*, 440(4). <https://doi.org/10.1088/1755-1315/440/4/042055>

APPENDIX A: LITERATURE REVIEW COMPARISON MATRIX

Author	Developed Algorithm / Work Contribution	Methodology / Key Contribution Highlights	Analysis and Results
Pannu, H. S., Ahuja, S., Dang, N., Soni, S., & Malhi, A. K. (2020). Deep learning based image classification for intestinal hemorrhage. <i>Multimedia Tools and Applications</i> , 79(29–30), 21941–21966. https://doi.org/10.1007/s11042-020-08905-7 (Deep Learning Model - 01)	- Self developed CNN architecture to classify Bleeding and Non Bleeding images. - Proposed ensemble of 5 CNN algorithms. Final prediction results finalized using aggregation from the 5 CNNs.	Architecture: Self Developed architecture (CNN). Input Image size: 100 x 100 x 3 Num classes: 2 Optimizer: Stochastic Gradient Descent (SGD) Learning Rate: 0.01 Momentum & Decay: 0.9 Momentum Batch size & Epochs: 128 batch size, 20 epochs Image Enhancement: Histogram Equalization + minimum variance quantization. Data Augmentation: Geometric Augmentation (Rotation), Colour, Contrast and Texture Based Augmentation (Luminance Channel stretching, blurring and Poisson Noise)	Static Image Dataset: 95% Accuracy Real Life Video Dataset: 93% Accuracy
Jia, X., & Meng, M. Q. H. (2016). A deep convolutional neural network for bleeding detection in Wireless Capsule Endoscopy images. <i>Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2016-Octob</i> , 639–642. https://doi.org/10.1109/EMBC.2016.7590783 (Deep Learning Model - 06)	- Self developed CNN architecture to classify Bleeding and Non Bleeding images. - Used a combination of CNN for feature extraction and used SVM as the final classifier. Input from CNN is fed into SVM model.	'Architecture: Self Developed architecture (CNN + SVM). Input Image size: 240 x 240 x 3 Num classes: 2 Optimizer: Stochastic Gradient Descent (SGD) Learning Rate: 0.001 Momentum & Decay: 0.9 Momentum, 0.004 Decay Batch size & Epochs: 100 batch size, 5000 epochs Image Enhancement: N/A Data Augmentation: N/A	99.2% Recall.
Pei, M., Wu, X., Guo, Y., & Fujita, H. (2017). Small bowel motility assessment based on fully convolutional networks and long short-term memory. <i>Knowledge-Based Systems</i> , 121, 163–172. https://doi.org/10.1016/j.knosys.2017.01.023 (Deep Learning Model - 03)	- Worked on developing CNN + LSTM network to estimate the size of the small bowel to determine if the patient has small bowel disease. - Used CNN as the feature extraction model. Used LSTM to analyse frames of features extracted from CNN to predict the final bowel diameter.	'Architecture: Self Developed architecture (CNN + LSTM). Input Image size: 20 x 20 x 1 Output: Diameter of small bowel (Regression) Optimizer: Stochastic Gradient Descent (SGD) Learning Rate: 0.01 Momentum & Decay: 0.9 Momentum, 0.0005 Decay Batch size & Epochs: 64 batch size, 600 epochs Image Enhancement: N/A Data Augmentation: Geometric Augmentations (Flipping and Shifting)	4.472 mean MSE.
Huang, D., Wang, Y., Song, W., Sequeira, J., & Mavromatis, S. (2018). Shallow-Water Image Enhancement Using Relative Global Histogram Stretching Based on Adaptive Parameter Acquisition. <i>Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)</i> , 10704 LNCS, 453–465. https://doi.org/10.1007/978-3-319-73603-7_37 (Deep Learning Model - 04)	- Proposed a self developed CNN architecture to classify different states of small intestine motility. - Proposed adding Hessian and Laplacian colour channels as input into the model. - Proposed 2 methods of incorporating the additional channels via early fusion and late fusion. - Early fusion involves merging channels as 5 (RGB + H +L) into a single CNN network. - Late fusion involves using 3 different CNNs for each RGB, H and L and merging their feature maps at the end of the CNN prior to performing classification.	'Architecture: Self Developed architecture (CNN) Input Image size: 100 x 100 x N (1, 3 or 5) Num classes: 6 Optimizer: Stochastic Gradient Descent (SGD) Learning Rate: 0.1 (reduces every 100000 iterations by 10 times) Momentum & Decay: N/A Batch size & Epochs: 128 batch size, 400000 epochs Image Enhancement: N/A Data Augmentation: N/A	99.2% Recall 99.9% Precision 99.55% F1-Score

Author	Developed Algorithm / Work Contribution	Methodology / Key Contribution Highlights	Analysis and Results
<p>Shin, Y., & Balasingham, I. (2017). Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 3277–3280.</p> <p>(Deep Learning Model – 07)</p>	<ul style="list-style-type: none"> - Compared performance of CNN vs Feature Extraction + Machine Learning models in detecting polyps from colonoscopy images. - Used a combination of colour and shape features with an SVM model. - Compared the performance of CNN using RGB images vs grayscale images. 	<p>Architecture: Self Developed architecture (CNN) Input Image size: 128 x 128 x 3 Num classes: 2 Optimizer: Adamax Learning Rate: 0.002 Momentum & Decay: N/A Batch size & Epochs: 200 epochs Image Enhancement: N/A Data Augmentation: N/A</p>	<p>a) Feature Extraction + SVM: 84.15% Accuracy 81.18% Specificity 86.73 Sensitivity</p> <p>b) CNN (gray) 61.48% Accuracy 32.35% Specificity 86.73% Sensitivity</p> <p>c) CNN RGB 92.08% Accuracy 91.76% Specificity 90.82% Sensitivity</p>
<p>He, J. Y., Wu, X., Jiang, Y. G., Peng, Q., & Jain, R. (2018). Hookworm Detection in Wireless Capsule Endoscopy Images with Deep Learning. IEEE Transactions on Image Processing, 27(5), 2379–2392.</p> <p>(Deep Learning Model – 08)</p>	<ul style="list-style-type: none"> - Proposed a CNN network for Hookworm detection - Proposed a 2 stage CNN network known as edge detection network and hookworm classification network integrated under a single pipeline. - Compared the performance of the CNN to Feature Extraction + Machine Learning method. 	<p>Architecture: Self Developed architecture (2 CNN) Input Image size: 227 x 227 x 3 Num classes: 2 Optimizer: N/A Learning Rate: 0.001 Momentum & Decay: 0.9 Momentum, 0.8 Gamma Batch size & Epochs: 20 Batch Size, 20 epochs Image Enhancement: N/A Data Augmentation: Geometric Augmentation (Crop, Flip, Rotation), Texture Augmentation (Smoothing with Gaussian filters)</p>	90.9% Sensitivity
<p>Lee, J. Y., Choi, S. H., & Chung, J. W. (2019). Automated classification of the tympanic membrane using a convolutional neural network. Applied Sciences (Switzerland), 9(9). (Deep Learning Model – 05)</p>	<ul style="list-style-type: none"> - Developed a 2 stage CNN network for perforation detection in the Tympanic Membrane of the ear. - Stage 1 network detected the presence of Tympanic Membrane in the image frame. Stage 2 classifies if the membrane is perforated or not. 	<p>Architecture: Self Developed architecture (2 stage CNN) Input Image size: 224 x 224 x 3 Num classes: 2 Optimizer: Stochastic Gradient Descent (SGD) Learning Rate: 0.0001 Momentum & Decay: 0.9 Momentum Batch size & Epochs: 32 Batch Size, 400 epochs Image Enhancement: N/A Data Augmentation: Geometric Augmentation (Sheer Range, Rotation and Horizontal flipping)</p>	Stage 1 Network: 98.7% Accuracy Stage 2 Network: 87.2% Accuracy
<p>Rustum, F., Siddique, M. A., Siddiqui, H. U. R., Ullah, S., Mehmood, A., Ashraf, I., & Choi, G. S. (2021). Wireless Capsule Endoscopy Bleeding Images Classification Using CNN Based Model. IEEE Access, 9, 33675–33688.</p> <p>(Furqan Rustam Paper)</p>	<ul style="list-style-type: none"> - Proposed transfer learning model to classify if the GI tract images are bleeding or not. - Used MobileNet as the base model as a feature extraction network added with 3 additional layers. - Compared the performance of using the standalone feature extraction MobileNet vs MobileNet + additional layers. 	<p>Architecture: Transfer Learning Feature Extraction (MobileNet) Input Image size: 150 x 150 x 3 Num classes: 2 Optimizer: Adam Learning Rate: N/A Momentum & Decay: N/A Batch size & Epochs: N/A Image Enhancement: N/A Data Augmentation: Geometric Augmentation (Flipping)</p>	99.3% Accuracy 100% Precision 99.4% Recall 99.7% F1 Score

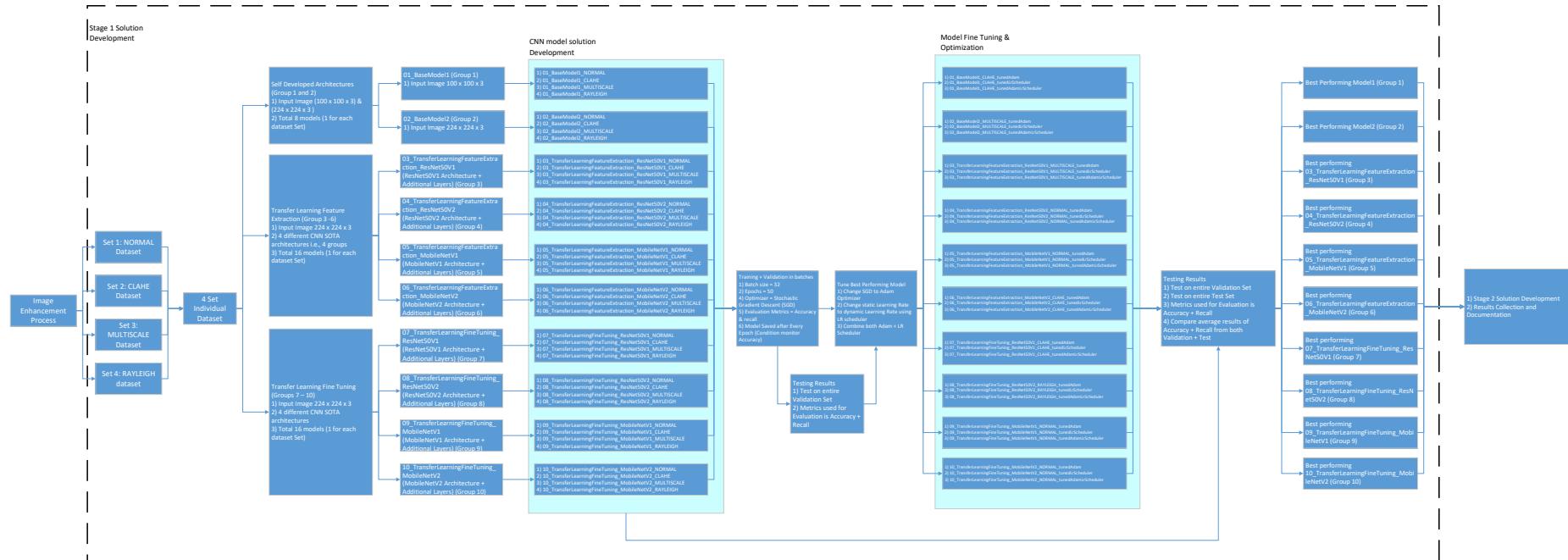
Author	Developed Algorithm / Work Contribution	Methodology / Key Contribution Highlights	Analysis and Results
Conference, I. I. (2017). CONVOLUTIONAL NEURAL NETWORKS FOR INTESTINAL HEMORRHAGE DETECTION IN WIRELESS CAPSULE ENDOSCOPY IMAGES Key Laboratory of Complex Systems Modeling and Simulation , School of Computer Science and Technology , Hangzhou Dianzi University School of Communicati. July. (Deep Learning Model Transfer Learning - 05)	<ul style="list-style-type: none"> - Studied and compared the performance of 3 transfer learning networks of LeNet, AlexNet, GooleNet and VGG network to detect bleeding images of the GI tract. - LeNet was developed from scratch without transfer learning. Others used transfer learning as feature extraction and fine tuning. 	Architecture: Transfer Learning Feature Extraction + Fine Tuning (LeNet, GooleNet, AlexNet, VGG) Input Image size: 32 x 32 x 3 (LeNet), 227 x 227 x 3 (AlexNet), 224 x 224 x 3 (GooleNet and VGG) Num classes: 2 Optimizer: SGD Learning Rate: 0.0001 (LeNet), 0.01 (Others) Momentum & Decay: 0.9 Momentum, 0.0005 Weight Decay Batch size & Epochs: 32 Batch Size, 30000 epochs. Image Enhancement: N/A Data Augmentation: Geometric Augmentation (Rotation), Colour, Contrast and Texture Based Augmentation (Luminance Channel stretching, blurring and Poisson Noise)	Original Dataset vs Augmentation Dataset a) LeNet 97.44% Precision vs 79.81% Precision 96.20% Recall vs 97.21% Recall b) AlexNet 98.72% Precision vs 98.06% Precision 98.72% Recall vs 99.02% Recall c) GooleNet 100% Precision vs 96.33% Precision 98.73% Recall vs 99.10% Recall d) VGG 98.72% Precision vs 98.65% Precision 98.72% Recall vs 99.10 Recall
Qu, J., Hiruta, N., Terai, K., Nosato, H., Murakawa, M., & Sakanashi, H. (2018). Gastric Pathology Image Classification Using Stepwise Fine-Tuning for Deep Neural Networks. Journal of Healthcare Engineering, 2018. https://doi.org/10.1155/2018/8961781 (Deep Learning Model Transfer Learning - 08)	<ul style="list-style-type: none"> - Performed stepwise fine tuning training on transfer learning networks of AlexNet, Inception-V3, and VGG-16 Networks to determine if pathology images of a cell is benign or malignant. - Broke down training into 2 separate stages. - Stage 1 trained the network to generalize on enabling the networks to recognize pathological cell images of stroma and epithelium cells. - Stage 2 refined the training of the network to enable it to distinguish between Benign and Malignant cells. 	Architecture: Transfer Learning Fine Tuning (Inception-V3, AlexNet, VGG) Input Image size: 256 x 256 x 3 Num classes: 2 Optimizer: N/A Learning Rate: N/A Momentum & Decay: N/A Batch size & Epochs: N/A Image Enhancement: N/A Data Augmentation: N/A	a) VGG-16 88.1% Accuracy 96.3% AUC 86.9% Precision 89.8% Recall b) AlexNet 83.7% Accuracy 92% AUC 84% Precision 84% Recall c) Inception-V3 86.2% Accuracy 93.4% AUC 86% Precision 87% Recall
Hmoud Al-Adhaileh, M., Mohammed Senan, E., Alsaade, F. W., Aldhyani, T. H. H., Alsharif, N., Abdullah Alqarni, A., Uddin, M. I., Alzahrani, M. Y., Alzain, E. D., & Jadhav, M. E. (2021). Deep Learning Algorithms for Detection and Classification of Gastrointestinal Diseases. Complexity, 2021. https://doi.org/10.1155/2021/6170416 (Deep Learning Model Transfer Learning - 02)	<ul style="list-style-type: none"> - Performed transfer learning fine tuning using GooleNet, ResNet-50 and AlexNet to classify between different types of Polyps. - The final 3 layers of the CNN networks were replaced with fully connected layers. 	Architecture: Transfer Learning Fine Tuning (GooleNet, ResNet-50, AlexNet) Input Image size: 227 x 227 x 3 (AlexNet), 224 x 224 x 3 (Others) Num classes: 5 Optimizer: Adam Learning Rate: 0.0003 (GoogleNet), 0.0001 (Others). LR changes dynamically each epoch. Momentum & Decay: Dynamically changes by pre-defined equation Batch size & Epochs: 128 (AlexNet), 20 (Others) Image Enhancement: Texture Enhancement (Averaging Filter) Data Augmentation: Geometric Augmentation (Flipping, Zooming, Shifting and Rotation)	a) GoogleNet 96.7% Accuracy 99.99% AUC 96.6% Sensitivity 99% Specificity b) ResNet-50 95% Accuracy 99.69% AUC 94.8% Sensitivity 98.9% Specificity c) AlexNet 97% Accuracy 99.98% AUC 96.8% Sensitivity 99.2% Specificity

Author	Developed Algorithm / Work Contribution	Methodology / Key Contribution Highlights	Analysis and Results
<p>Chen, P. J., Lin, M. C., Lai, M. J., Lin, J. C., Lu, H. H. S., & Tseng, V. S. (2018). Accurate Classification of Diminutive Colorectal Polyps Using Computer-Aided Analysis. <i>Gastroenterology</i>, 154(3), 568–575. (Deep Learning Model Transfer Learning - 03)</p>	<ul style="list-style-type: none"> - Proposed transfer learning feature extraction using Inception-V3 model to analyse narrow band imaging to detect polyps. - Replaced the top layer of the network only. - Results were compared to analysis results by trained professionals. 	<p>Architecture: Transfer Learning Feature Extraction (Inception-V3) Input Image size: N/A Num classes: 3 Optimizer: N/A Learning Rate: N/A Momentum & Decay: N/A Batch size & Epochs: N/A Image Enhancement: N/A Data Augmentation: N/A</p>	90.1% Accuracy 96.3% Sensitivity 78.1% Specificity
<p>Ayyaz, M. S., Lali, M. I. U., Hussain, M., Rauf, H. T., Alouffi, B., Alyami, H., & Wasti, S. (2022). Hybrid deep learning model for endoscopic lesion detection and classification using endoscopy videos. <i>Diagnostics</i>, 12(1). (Deep Learning Model Transfer Learning - 01)</p>	<ul style="list-style-type: none"> - Proposed a hybrid approach leveraging on Deep Learning and Machine Learning to classify 5 different GI tract related diseases. - Network architectures used were VGG-19 and AlexNet (simultaneously) as feature extraction networks to produce a feature array that is fused between the 2 networks. . - Selected features from the output array are selected using Genetic Algorithm. - Selected features are used as inputs in SVM, Bagged Tree, Cosine KNN, KNN, Naive Bayes and Subspace Discriminant Model to perform final classification. 	<p>Architecture: Transfer Learning Feature Extraction (VGG-19 and AlexNet) + Machine Learning Input Image size: 256 x 256 x 3 Num classes: 5 Optimizer: N/A Learning Rate: N/A Momentum & Decay: N/A Batch size & Epochs: N/A Image Enhancement: Texture Based (Subtracting a processed mask from the original image + median filtering) Data Augmentation: N/A</p>	Fine Tree 90.3% Cosine KNN 99.3% Bagged Tree 98.8% Linear SVM 98.6% Coarse Tree 75.9% Cubic SVM 99.8% Naïve Bayes 96.2% Coarse KNN 90.7%
<p>Takiyama, H., Ozawa, T., Ishihara, S., Fujishiro, M., Shichijo, S., Nomura, S., Miura, M., & Tada, T. (2018). Automatic anatomical classification of esophagogastroduodenoscopy images using deep convolutional neural networks. <i>Scientific Reports</i>, 8(1), 1–8. (Deep Learning Model Transfer Learning - 04)</p>	<ul style="list-style-type: none"> -Used the GooLeNet architecture to classify between 6 different classes of anatomical locations in the GI tract. - Goal was to classify different locations of gastrointestinal diseases. 	<p>Architecture: Transfer Learning Feature Extraction (GooLeNet) Input Image size: 244 x 244 x 3 Num classes: 6 Optimizer: Adam Learning Rate: 0.0002 Momentum & Decay: N/A Batch size & Epochs:N/A Image Enhancement: N/A Data Augmentation: Geometric Augmentation (Rotation)</p>	99% AUC

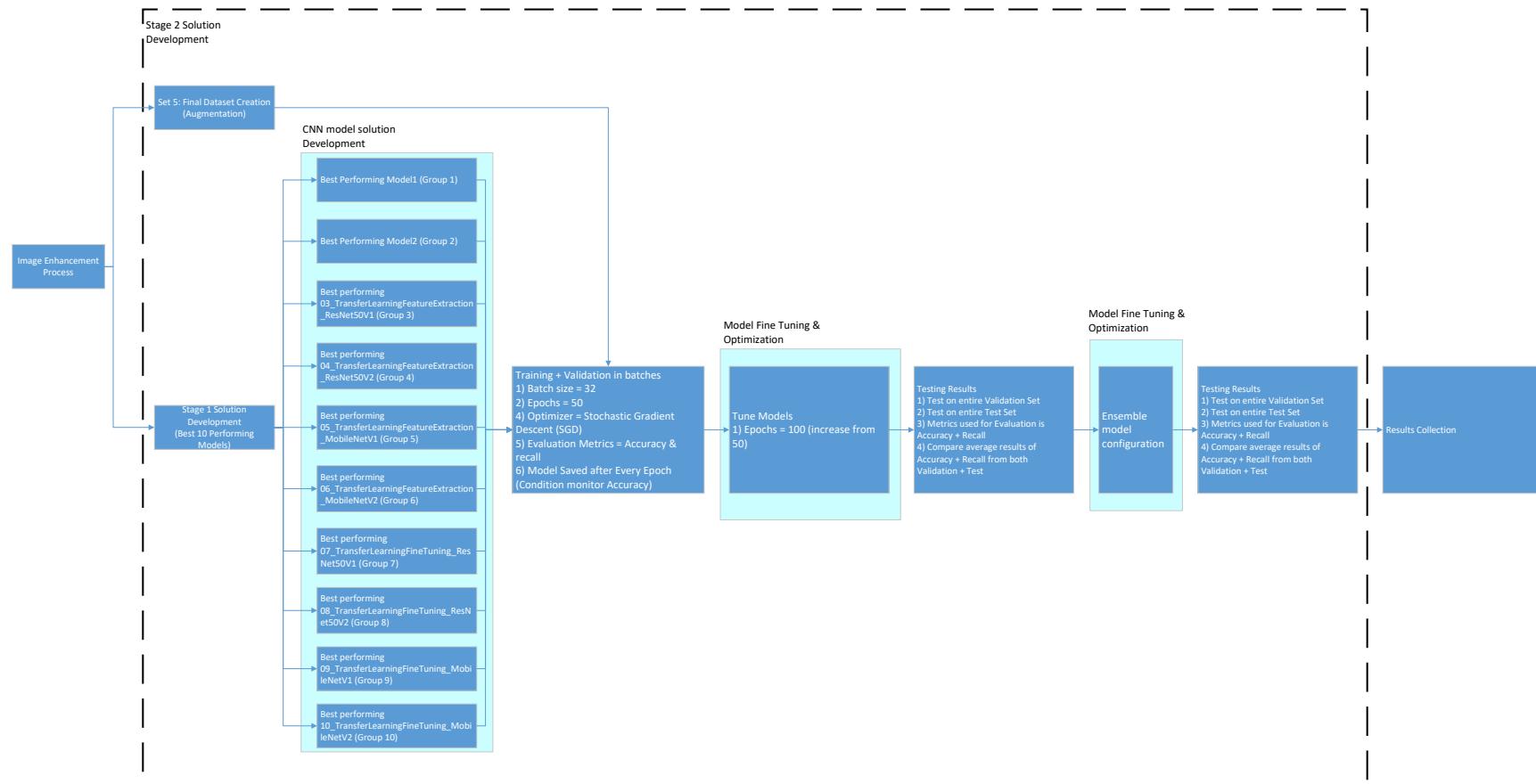
Author	Developed Algorithm / Work Contribution	Methodology / Key Contribution Highlights	Analysis and Results
Liu, X., Wang, C., Hu, Y., Zeng, Z., Bai, J., & Liao, G. (2018). Transfer Learning with Convolutional Neural Network for Early Gastric Cancer Classification on Magnifying Narrow-Band Imaging Images. Proceedings - International Conference on Image Processing, ICIP, 1388–1392. (Deep Learning Model Transfer Learning - 06)	<ul style="list-style-type: none"> - Developed 3 state of the art networks of VGG-16, InceptionV3, and InceptionResNetV2 using transfer learning fine tuning to classify between 2 classes of early gastric cancer and non early gastric cancer. - Analysed MNBI images. Studied the effects of different training dataset size, types of CNN used, number of CNN layers and the effect of input images size on the models classification performance. - Compared the performance of the CNN models to Feature Extraction + Machine Learning models. 	<p>Architecture: Transfer Learning Fine Tuning (VGG-16, InceptionV3, InceptionResNetV2)</p> <p>Input Image size: 80 x 80 x 3 - 299 x 299 x 3</p> <p>Num classes: 2</p> <p>Optimizer: SGD</p> <p>Learning Rate: 0.0001</p> <p>Momentum & Decay: 0.8 Momentum</p> <p>Batch size & Epochs: 32 Batch Size, 30 Epochs</p> <p>Image Enhancement: N/A</p> <p>Data Augmentation: Geometric Augmentation (rotation, width shifting, height shifting, zooming, horizontal flip, vertical flip and scale normalization)</p>	<p>a) InceptionV3 98.5% Accuracy 98.1% Sensitivity 98.9% Specificity</p> <p>b) VGG-16 97.6% Accuracy 95.8% Sensitivity 99.2% Specificity</p> <p>c) InceptionResNetV2 97.7% Accuracy 97.4% Sensitivity 98% Specificity.</p>
Liu, B., Yao, K., Huang, M., Zhang, J., Li, Y., & Li, R. (2018). Gastric Pathology Image Recognition Based on Deep Residual Networks. Proceedings - International Computer Software and Applications Conference, 2, 408–412. (Deep Learning Model Transfer Learning - 07)	<ul style="list-style-type: none"> - Developed automatic CNN model for tumour detection. - Used ResNet50 network as feature extraction. - Used ensemble method of ResNet50 network trained on dataset split into 5 different portions. 	<p>Architecture: Transfer Learning Feature Extraction Ensemble (ResNet50)</p> <p>Input Image size: 224 x 224 x 3</p> <p>Num classes: 2</p> <p>Optimizer: SGD</p> <p>Learning Rate: 0.01</p> <p>Momentum & Decay: 0.8 Momentum</p> <p>Batch size & Epochs: 32 Batch Size, 50000 Epochs</p> <p>Image Enhancement: N/A</p> <p>Data Augmentation: Geometric Augmentation (Affine transformation of rotation, scaling, horizontal and vertical flipping)</p>	<p>Without Ensemble: 95.5% F1 score with Ensemble: 96% F1 score.</p>
Liu, X., Wang, C., Bai, J., & Liao, G. (2020). Fine-tuning Pre-trained Convolutional Neural Networks for Gastric Precancerous Disease Classification on Magnification Narrow-band Imaging Images. Neurocomputing, 392(xxxx), 253–267. (Deep Learning Model Transfer Learning - 09)	<ul style="list-style-type: none"> - Developed CNN architectures to classify 3 classes of pre-cancerous lesions of GI tract using VGG-16, InceptionNetV3, InceptionResNetV2 and ResNet50 - Authors performed 3 experiments <ul style="list-style-type: none"> a) Exp 1: Performance of transfer learning models as feature extractors b) A combination of ResNet50 + SVM as classifier vs Manual Feature Extraction + SVM c) Compared ResNet50 vs VGG16 without transfer learning. 	<p>Architecture: Transfer Learning Feature Extraction (ResNet50, VGG-16, InceptionV3 and InceptionResNetV2) + Machine Learning Model</p> <p>Input Image size: 224 x 224 x 3</p> <p>Num classes: 3</p> <p>Optimizer: SGD</p> <p>Learning Rate: 0.0001</p> <p>Momentum & Decay: 0.9 Momentum</p> <p>Batch size & Epochs: 32 Batch Size, 50 Epochs</p> <p>Image Enhancement: N/A</p> <p>Data Augmentation: N/A</p>	<p>Experiment 1 ResNet50: 95% VGG16: 89%</p> <p>Experiment2: ResNet50 + SVM: 90% Feature Extraction + SVM: 50-70%</p> <p>Experiment3 ResNet50: 25% VGG16: 34%</p>

Author	Developed Algorithm / Work Contribution	Methodology / Key Contribution Highlights	Analysis and Results
Demir, A., Yilmaz, F., & Kose, O. (2019). Early detection of skin cancer using deep learning architectures: Resnet-101 and inception-v3. TIPTEKNO 2019 - Tip Teknolojileri Kongresi, 2019-Janua(February 2020), 2–6. (Deep Learning Model Transfer Learning - 10 (non GI))	<ul style="list-style-type: none"> - Proposed transfer learning on skin cancer images for early stage cancer detection - Used ResNet-101 and Inception-V3 models as feature extraction models. 	Architecture: Transfer Learning Feature Extraction (ResNet101 & InceptionV3) Input Image size: 224 x 224 x 3 Num classes: 2 Optimizer: N/A Learning Rate: 0.001 Momentum & Decay: N/A Batch size & Epochs: 50 Epochs Image Enhancement: N/A Data Augmentation: N/A	ResNet101: 84.09% Accuracy InceptionV3: 87.42% Accuracy
Saric, M., Russo, M., Stella, M., & Sikora, M. (2019). CNN-based Method for Lung Cancer Detection in Whole Slide Histopathology Images. 2019 4th International Conference on Smart and Sustainable Technologies, SpliTecH 2019, 14–17. (Deep Learning Model Transfer Learning - 11 (non GI))	<ul style="list-style-type: none"> - Compared performance of VGG-16 and ResNet-50 architecture to detect lung cancer by analysing histopathological slides of images. 	Architecture: Transfer Learning Feature Extraction (ResNet50 & VGG16) Input Image size: 256 x 256 x 1 Num classes: 2 Optimizer: Stochastic Gradient Descent (SGD) Learning Rate: 0.0001 Momentum & Decay: N/A Batch size & Epochs: 16 Batch Size, 17 Epochs Image Enhancement: N/A Data Augmentation: N/A	Top 1% ResNet50: 75.2% Accuracy VGG16: 70.5% Accuracy Top 5% ResNet50: 93% Accuracy VGG16: 91.2% Accuracy
Salem, N., Malik, H., & Shams, A. (2019). Medical image enhancement based on histogram algorithms. Procedia Computer Science, 163, 300–311. (image processing - 02)	<ul style="list-style-type: none"> - Studied and compare 4 different histogram based contrast enhancement algorithms of: a) Global Histogram Equalization (HE) b) Cumulative Histogram Equalization (CHE) c) Quadrant Dynamic Histogram Equalization (QDHE) d) Contrast Limited Adaptive Histogram equalization (CLAHE) - Performance of algorithms evaluated using PSNR, MSE and standard deviation 	N/A	<ul style="list-style-type: none"> - CLAHE best recommended for images with non-uniform intensity distribution across the image. - CLAHE proposed best enhancement technique for retina whilst QDHE was proposed as the best for brain, endometrium, breast, and knee images.
Jintasuttisak, T., & Intajag, S. (2014). Color retinal image enhancement by Rayleigh contrast-limited adaptive histogram equalization. International Conference on Control, Automation and Systems, Iccas, 692–697. (image processing - 05)	<ul style="list-style-type: none"> - Applied Rayleigh CLAHE algorithm on retinal images to enhance contrast. - Stretched the image histogram of "I" channel in HSI colour model instead of RGB to control the brightness of the image - Output images were analysed visually and via Correlation Coefficient (CC). 	N/A	<ul style="list-style-type: none"> - Visual inspection showed that the enhanced results had improved image quality with good overall contrast and appearance. - From a quantitative standpoint, the proposed algorithm offers high values of CC i.e., the algorithm is capable of preserving the chromatic information of the image.

APPENDIX B: DETAILED PROCESS FLOW OF CNN MODEL DEVELOPMENT IN STAGE 1



APPENDIX C: DETAILED PROCESS FLOW OF CNN MODEL DEVELOPMENT IN STAGE 2



APPENDIX D: DEVELOPED MODEL DETAILS AND RESULTS

modelName	ModelTag	ModelTagAlpha	modelGroup	ModelSubGroup	ModelSuffix	ModelType	augmentation	Dataset Set	ImageProcessingTechnique	Optimization	optimizer	lrType	epochs	val Accuracy	test Accuracy	ValTest Average	testF1 Score
01_BaseModel1_NORMAL	A_1_1_N	A	1	1	NORMAL	SelfDeveloped	FALSE	1	NONE	SGD	STATIC	50	89.00%	90.08%	89.54%	90.02%	
01_BaseModel1_CLAHE	A_1_2_C	A	1	2	CLAHE	SelfDeveloped	FALSE	2	CLAHE	NO	SGD	STATIC	50	92.17%	91.17%	91.67%	91.10%
01_BaseModel1_MULTISCALE	A_1_3_M	A	1	3	MULTISCALE	SelfDeveloped	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	90.17%	91.50%	90.83%	91.44%
01_BaseModel1_RAYLEIGH	A_1_4_R	A	1	4	RAYLEIGH	SelfDeveloped	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	88.83%	90.42%	89.63%	90.32%
01_BaseModel1_CLAHE_tunedAdam	A_1_5_C_TA	A	1	5	CLAHE_tunedAdam	SelfDeveloped	FALSE	2	CLAHE	YES	ADAM	STATIC	50	91.33%	91.83%	91.58%	91.76%
01_BaseModel1_CLAHE_tunedLrScheduler	A_1_6_C_LR	A	1	6	CLAHE_tunedLrScheduler	SelfDeveloped	FALSE	2	CLAHE	YES	SGD	DYNAMIC	50	90.50%	91.58%	91.04%	91.52%
01_BaseModel1_CLAHE_tunedAdamLrScheduler	A_1_7_C_TALR	A	1	7	CLAHE_tunedAdamLrScheduler	SelfDeveloped	FALSE	2	CLAHE	YES	ADAM	DYNAMIC	50	90.17%	91.75%	90.95%	91.66%
02_BaseModel2_NORMAL	B_2_1_N	B	2	1	NORMAL	SelfDeveloped	FALSE	1	CLAHE	NO	SGD	STATIC	50	91.17%	92.67%	91.92%	92.64%
02_BaseModel2_CLAHE	B_2_2_C	B	2	2	CLAHE	SelfDeveloped	FALSE	2	CLAHE	NO	SGD	STATIC	50	91.33%	92.30%	91.54%	92.34%
02_BaseModel2_MULTISCALE	B_2_3_M	B	2	3	MULTISCALE	SelfDeveloped	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	91.83%	92.33%	92.08%	92.28%
02_BaseModel2_RAYLEIGH	B_2_4_R	B	2	4	RAYLEIGH	SelfDeveloped	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	90.83%	91.58%	91.11%	91.53%
02_BaseModel2_MULTISCALE_tunedAdam	B_2_5_M_TA	B	2	5	MULTISCALE_tunedAdam	SelfDeveloped	FALSE	3	MULTISCALE	YES	ADAM	STATIC	50	91.50%	93.08%	92.29%	93.00%
02_BaseModel2_MULTISCALE_tunedLrScheduler	B_2_6_M_LR	B	2	6	MULTISCALE_tunedLrScheduler	SelfDeveloped	FALSE	3	MULTISCALE	YES	SGD	DYNAMIC	50	92.33%	93.50%	92.92%	93.42%
02_BaseModel2_MULTISCALE_tunedAdamLrScheduler	B_2_7_M_TALR	B	2	7	MULTISCALE_tunedAdamLrScheduler	SelfDeveloped	FALSE	3	MULTISCALE	YES	ADAM	DYNAMIC	50	93.17%	92.83%	93.00%	92.73%
03_TransferLearningFeatureExtraction_ResNetS0V1_NORMAL	C_3_1_N	C	3	1	NORMAL	TransferLearningFeatureExtraction	FALSE	1	NONE	SGD	STATIC	50	86.50%	87.42%	86.96%	87.31%	
03_TransferLearningFeatureExtraction_ResNetS0V1_CLAHE	C_3_2_C	C	3	2	CLAHE	TransferLearningFeatureExtraction	FALSE	2	CLAHE	NO	SGD	STATIC	50	88.00%	86.67%	87.33%	86.52%
03_TransferLearningFeatureExtraction_ResNetS0V1_MULTISCALE	C_3_3_M	C	3	3	MULTISCALE	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	89.67%	89.33%	89.50%	89.15%
03_TransferLearningFeatureExtraction_ResNetS0V1_RAYLEIGH	C_3_4_R	C	3	4	RAYLEIGH	TransferLearningFeatureExtraction	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	86.33%	86.83%	86.58%	86.70%
03_TransferLearningFeatureExtraction_ResNetS0V1_RAYLEIGH_tunedAdam	C_3_5_M_TA	C	3	5	MULTISCALE_tunedAdam	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	YES	ADAM	STATIC	50	84.67%	84.33%	84.50%	83.91%
03_TransferLearningFeatureExtraction_ResNetS0V1_MULTISCALE_tunedLrScheduler	C_3_6_M_LR	C	3	6	MULTISCALE_tunedLrScheduler	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	YES	SGD	DYNAMIC	50	88.83%	88.17%	88.50%	87.86%
03_TransferLearningFeatureExtraction_ResNetS0V1_MULTISCALE_tunedAdamLrScheduler	C_3_7_M_TALR	C	3	7	MULTISCALE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	YES	ADAM	DYNAMIC	50	91.00%	90.85%	91.00%	90.85%
04_TransferLearningFeatureExtraction_ResNetS0V2_NORMAL	D_4_1_N	D	4	1	NORMAL	TransferLearningFeatureExtraction	FALSE	1	NONE	SGD	STATIC	50	84.33%	85.92%	84.54%	85.68%	
04_TransferLearningFeatureExtraction_ResNetS0V2_CLAHE	D_4_2_C	D	4	2	CLAHE	TransferLearningFeatureExtraction	FALSE	2	CLAHE	NO	SGD	STATIC	50	83.50%	83.75%	83.42%	83.65%
04_TransferLearningFeatureExtraction_ResNetS0V2_MULTISCALE	D_4_3_M	D	4	3	MULTISCALE	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	80.33%	82.50%	81.42%	82.21%
04_TransferLearningFeatureExtraction_ResNetS0V2_RAYLEIGH	D_4_4_R	D	4	4	RAYLEIGH	TransferLearningFeatureExtraction	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	79.33%	81.42%	80.38%	81.37%
04_TransferLearningFeatureExtraction_ResNetS0V2_NORMAL_tunedAdam	D_4_5_N_TA	D	4	5	NORMAL_tunedAdam	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	ADAM	STATIC	50	85.83%	87.58%	86.71%	87.44%
04_TransferLearningFeatureExtraction_ResNetS0V2_NORMAL_tunedLrScheduler	D_4_6_N_LR	D	4	6	NORMAL_tunedLrScheduler	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	SGD	DYNAMIC	50	85.00%	86.17%	85.58%	86.11%
04_TransferLearningFeatureExtraction_ResNetS0V2_NORMAL_tunedAdamLrScheduler	D_4_7_N_TALR	D	4	7	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	ADAM	DYNAMIC	50	87.33%	88.75%	88.04%	88.70%
05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL	E_5_1_N	E	5	1	NORMAL	TransferLearningFeatureExtraction	FALSE	1	NONE	SGD	STATIC	50	89.33%	89.33%	89.33%	89.24%	
05_TransferLearningFeatureExtraction_MobileNetV1_CLAHE	E_5_2_C	E	5	2	CLAHE	TransferLearningFeatureExtraction	FALSE	2	CLAHE	NO	SGD	STATIC	50	86.83%	88.42%	87.63%	88.32%
05_TransferLearningFeatureExtraction_MobileNetV1_MULTISCALE	E_5_3_M	E	5	3	MULTISCALE	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	84.83%	87.00%	85.92%	86.81%
05_TransferLearningFeatureExtraction_MobileNetV1_RAYLEIGH	E_5_4_R	E	5	4	RAYLEIGH	TransferLearningFeatureExtraction	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	86.67%	85.83%	86.25%	85.74%
05_TransferLearningFeatureExtraction_MobileNetV1_tunedAdam	E_5_5_N_TA	E	5	5	NORMAL_tunedAdam	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	ADAM	STATIC	50	90.83%	90.71%	90.71%	90.46%
05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedLrScheduler	E_5_6_N_LR	E	5	6	NORMAL_tunedLrScheduler	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	SGD	DYNAMIC	50	89.50%	89.84%	89.45%	89.45%
05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	E_5_7_N_TALR	E	5	7	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	ADAM	DYNAMIC	50	91.17%	91.42%	91.29%	91.31%
06_TransferLearningFeatureExtraction_MobileNetV2_NORMAL	F_6_1_N	F	6	1	NORMAL	TransferLearningFeatureExtraction	FALSE	1	NONE	SGD	STATIC	50	92.67%	93.42%	93.43%	93.38%	
06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE	F_6_2_C	F	6	2	CLAHE	TransferLearningFeatureExtraction	FALSE	2	CLAHE	NO	SGD	STATIC	50	92.33%	92.33%	92.33%	92.30%
06_TransferLearningFeatureExtraction_MobileNetV2_MULTISCALE	F_6_3_M	F	6	3	MULTISCALE	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	91.17%	90.58%	90.88%	90.63%
06_TransferLearningFeatureExtraction_MobileNetV2_RAYLEIGH	F_6_4_R	F	6	4	RAYLEIGH	TransferLearningFeatureExtraction	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	90.17%	88.92%	89.54%	88.87%
06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdam	F_6_5_C_TA	F	6	5	CLAHE_tunedAdam	TransferLearningFeatureExtraction	FALSE	2	CLAHE	YES	ADAM	STATIC	50	93.83%	94.67%	94.25%	94.64%
06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedLrScheduler	F_6_6_C_LR	F	6	6	CLAHE_tunedLrScheduler	TransferLearningFeatureExtraction	FALSE	2	CLAHE	YES	SGD	DYNAMIC	50	92.00%	92.42%	92.21%	92.38%
06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	F_6_7_C_TALR	F	6	7	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	2	CLAHE	YES	ADAM	DYNAMIC	50	94.83%	95.17%	95.00%	95.16%
07_TransferLearningFeatureTuning_ResNetS0V1_NORMAL	G_7_1_N	G	7	1	NORMAL	TransferLearningFeatureTuning	FALSE	1	NONE	NO	SGD	STATIC	50	92.17%	92.42%	92.29%	92.35%
07_TransferLearningFeatureTuning_ResNetS0V1_CLAHE	G_7_2_C	G	7	2	CLAHE	TransferLearningFeatureTuning	FALSE	2	CLAHE	NO	SGD	STATIC	50	93.67%	94.00%	93.83%	93.95%
07_TransferLearningFeatureTuning_ResNetS0V1_MULTISCALE	G_7_3_M	G	7	3	MULTISCALE	TransferLearningFeatureTuning	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	92.83%	93.67%	93.25%	93.62%
07_TransferLearningFeatureTuning_ResNetS0V1_RAYLEIGH	G_7_4_R	G	7	4	RAYLEIGH	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	93.20%	93.25%	93.20%	93.25%
07_TransferLearningFeatureTuning_ResNetS0V1_CLAHE_tunedAdam	G_7_5_C_TA	G	7	5	CLAHE_tunedAdam	TransferLearningFeatureTuning	FALSE	2	CLAHE	YES	ADAM	STATIC	50	79.37%	81.42%	80.29%	81.65%
07_TransferLearningFeatureTuning_ResNetS0V1_CLAHE_tunedLrScheduler	G_7_6_C_LR	G	7	6	CLAHE_tunedLrScheduler	TransferLearningFeatureTuning	FALSE	2	CLAHE	YES	SGD	DYNAMIC	50	83.17%	93.67%	93.42%	93.62%
07_TransferLearningFeatureTuning_ResNetS0V1_CLAHE_tunedAdamLrScheduler	G_7_7_C_TALR	G	7	7	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureTuning	FALSE	2	CLAHE	YES	ADAM	DYNAMIC	50	92.67%	94.00%	93.83%	93.97%
08_TransferLearningFeatureTuning_ResNetS0V2_NORMAL	H_8_1_N	H	8	1	NORMAL	TransferLearningFeatureTuning	FALSE	1	NONE	NO	SGD	STATIC	50	98.50%	98.42%	98.46%	98.41%
08_TransferLearningFeatureTuning_ResNetS0V2_CLAHE	H_8_2_C	H	8	2	CLAHE	TransferLearningFeatureTuning	FALSE	2	CLAHE	NO	SGD	STATIC	50	97.33%	97.67%	97.50%	97.65%
08_TransferLearningFeatureTuning_ResNetS0V2_MULTISCALE	H_8_3_M	H	8	3	MULTISCALE	TransferLearningFeatureTuning	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	98.17%	99.08%	98.63%	99.08%
08_TransferLearningFeatureTuning_ResNetS0V2_RAYLEIGH	H_8_4_R	H	8	4	RAYLEIGH	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	98.67%	98.92%	98.79%	98.91%
08_TransferLearningFeatureTuning_ResNetS0V2_RAYLEIGH_tunedAdam	H_8_5_R_TA	H	8	5	RAYLEIGH_tunedAdam	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	YES	ADAM	STATIC	50	98.33%	99.08%	98.71%	99.08%
08_TransferLearningFeatureTuning_ResNetS0V2_RAYLEIGH_tunedLrScheduler	H_8_7_R_LR	H	8	7	RAYLEIGH_tunedLrScheduler	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	YES	SGD	DYNAMIC	50	97.83%	98.17%	98.00%	98.16%
08_TransferLearningFeatureTuning_ResNetS0V2_RAYLEIGH_tunedAdamLrScheduler	H_8_6_R_TALR	H	8	8	RAYLEIGH_tunedAdamLrScheduler	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	YES	ADAM	DYNAMIC	50	98.33%	98.67%	98.50%	98.66%
09_TransferLearningFeatureTuning_MobileNetV1_NORMAL	I_9_1_N	I	9	1	NORMAL	TransferLearningFeatureTuning	FALSE	1	NONE	NO	SGD	STATIC	50	99.33%	99.33%	99.33%	99.33%
09_TransferLearningFeatureTuning_MobileNetV1_CLAHE	I_9_2_C	I	9	2	CLAHE	TransferLearningFeatureTuning	FALSE	2	CLAHE	NO	SGD	STATIC	50	98.50%	98.45%	98.45%	98.45%
09_TransferLearningFeatureTuning_MobileNetV1_MULTISCALE	I_9_3_M	I	9	3	MULTISCALE	TransferLearningFeatureTuning	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	97.83%	98.42%	98.42%	98.42%
09_TransferLearningFeatureTuning_MobileNetV1_RAYLEIGH	I_9_4_R	I	9	4	RAYLEIGH	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	97.83%	97.58%	97.71%	97.58%
09_TransferLearningFeatureTuning_MobileNetV1_NORMAL_tunedAdam	I_9_5_N_TA	I	9	5	NORMAL_tunedAdam	TransferLearningFeatureTuning	FALSE	1	NONE	YES	ADAM	STATIC	50	98.83%	98.75%	98.76%	98.75%
09_TransferLearningFeatureTuning_MobileNetV1_NORMAL_tunedLrScheduler	I_9_6_N_LR	I	9	6	NORMAL_tunedLrScheduler	TransferLearningFeatureTuning	FALSE	1	NONE	YES	SGD	DYNAMIC	50	99.50%	98.75%	99.13%	98.75%
09_TransferLearningFeatureTuning_MobileNetV1_NORMAL_tunedAdamLrScheduler	I_9_7_N_TALR	I	9	7	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureTuning	FALSE	1	NONE	YES	ADAM	DYNAMIC	50	98.00%	98.67%	98.33%	98.66%
10_TransferLearningFeatureTuning_MobileNetV2_NORMAL	J_10_1_N	J	10	1	NORMAL	TransferLearningFeatureTuning	FALSE	1	NONE	NO	SGD	STATIC	50	99.17%	99.17%	99.17%	99.17%
10_TransferLearningFeatureTuning_MobileNetV2_CLAHE	J_10_2_C	J	10	2	CLAHE	TransferLearningFeatureTuning	FALSE	2	CLAHE	NO	SGD	STATIC	50	98.33%	98.25%	98.29%	98.24%
10_TransferLearningFeatureTuning_MobileNetV2_MULTISCALE	J_10_3_M	J	10	3	MULTISCALE	TransferLearningFeatureTuning	FALSE	3	MULTISCALE	NO	SGD	STATIC	50	98.67%	98.58%	98.63%	98.58%
10_TransferLearningFeatureTuning_MobileNetV2_RAYLEIGH	J_10_4_R	J	10	4	RAYLEIGH	TransferLearningFeatureTuning	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	99.00%	98.50%	98.75%	98.50%
10_TransferLearningFeatureTuning_MobileNetV2_NORMAL_tunedAdam	J_10_5_N_TA	J	10	5	NORMAL_tunedAdam	TransferLearningFeatureTuning	FALSE	1	NONE	YES	ADAM	STATIC	50	97.50%	97.67%	97.58%	97.6

APPENDIX D: DEVELOPED MODEL DETAILS AND RESULTS

modelName	ModelTag	ModelTagAlpha	modelGroup	ModelSubGroup	ModelSuffix	ModelType	augmentation	Dataset Set	ImageProcessingTechnique	Optimization	optimizer	lrType	epochs	val Accuracy	test Accuracy	Val/Test Average	testF1 Score
11_BaseModel1_CLAHE	K_11_1_C	K	11	1	CLAHE	SelfDeveloped	TRUE	2	CLAHE	NO	SGD	STATIC	50	90.00%	89.50%	89.75%	94.1%
11_BaseModel2_MULTISCALE_tunedAdamLrScheduler	K_11_2_M_TALR	K	11	2	MULTISCALE_tunedAdamLrScheduler	SelfDeveloped	TRUE	3	MULTISCALE	YES	ADAM	DYNAMIC	50	90.00%	92.08%	91.04%	91.96%
11_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	K_11_3_M_TALR	K	11	3	MULTISCALE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	3	MULTISCALE	YES	ADAM	DYNAMIC	50	88.17%	90.50%	89.33%	90.43%
11_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	K_11_4_N_TALR	K	11	4	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	1	NONE	YES	ADAM	DYNAMIC	50	87.83%	89.83%	88.83%	89.83%
11_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	K_11_5_N_TALR	K	11	5	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	1	NONE	YES	ADAM	DYNAMIC	50	90.50%	94.08%	92.29%	94.06%
11_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	K_11_6_C_TALR	K	11	6	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	2	CLAHE	YES	ADAM	DYNAMIC	50	95.00%	96.17%	95.58%	96.15%
11_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	K_11_7_C_TALR	K	11	7	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	2	CLAHE	YES	ADAM	DYNAMIC	50	91.67%	92.50%	92.08%	92.43%
11_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	K_11_8_R_RAYLEIGH	K	11	8	RAYLEIGH	TransferLearningFeatureExtraction	TRUE	4	RAYLEIGH	NO	SGD	STATIC	50	84.50%	84.83%	84.67%	84.5%
11_TransferLearningFineTuning_MobileNetV1_NORMAL	K_11_9_N_NORMAL	K	11	9	NORMAL	TransferLearningFeatureExtraction	TRUE	1	NONE	NO	SGD	STATIC	50	88.00%	89.93%	88.93%	90.3%
11_TransferLearningFineTuning_MobileNetV2_NORMAL	K_11_10_N_NORMAL	K	11	10	NORMAL	TransferLearningFeatureExtraction	TRUE	1	NONE	NO	SGD	STATIC	50	92.83%	93.25%	93.04%	93.22%
11A_BaseModel1_CLAHE	L_11A_1_C	11A	11A	1	CLAHE	SelfDeveloped	TRUE	2	CLAHE	NO	SGD	STATIC	100	88.50%	89.92%	89.21%	89.88%
11A_BaseModel2_MULTISCALE_tunedAdamLrScheduler	L_11A_2_M_TALR	11A	11A	2	MULTISCALE_tunedAdamLrScheduler	SelfDeveloped	TRUE	3	MULTISCALE	YES	ADAM	DYNAMIC	100	89.83%	90.92%	90.38%	90.77%
11A_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	L_11A_3_M_TALR	11A	11A	3	MULTISCALE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	3	MULTISCALE	YES	ADAM	DYNAMIC	100	87.50%	89.58%	88.54%	89.47%
11A_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	L_11A_4_N_TALR	11A	11A	4	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	1	NONE	YES	ADAM	DYNAMIC	100	90.17%	89.42%	89.79%	89.54%
11A_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	L_11A_5_N_TALR	11A	11A	5	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	1	NONE	YES	ADAM	DYNAMIC	100	92.33%	93.92%	93.13%	93.88%
11A_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	L_11A_6_C_TALR	11A	11A	6	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	2	CLAHE	YES	ADAM	DYNAMIC	100	95.67%	97.00%	96.33%	96.99%
11A_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	L_11A_7_C_TALR	11A	11A	7	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	TRUE	2	CLAHE	YES	ADAM	DYNAMIC	100	90.17%	91.08%	90.63%	90.99%
11A_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	L_11A_8_R_RAYLEIGH	11A	11A	8	RAYLEIGH	TransferLearningFeatureExtraction	TRUE	4	RAYLEIGH	NO	SGD	STATIC	100	85.83%	85.42%	85.63%	85.30%
11A_TransferLearningFineTuning_MobileNetV1_NORMAL	L_11A_9_N_NORMAL	11A	11A	9	NORMAL	TransferLearningFeatureExtraction	TRUE	1	NONE	NO	SGD	STATIC	100	88.33%	89.58%	88.96%	89.50%
11A_TransferLearningFineTuning_MobileNetV2_NORMAL	L_11A_10_N_NORMAL	11A	11A	10	NORMAL	TransferLearningFeatureExtraction	TRUE	1	NONE	NO	SGD	STATIC	100	92.83%	94.25%	93.54%	94.20%
12_Avg VotingModel_MIXED	M_12_1234567_MIXED	M	12	1234567	MIXED	MIXED	TRUE	MIXED	MIXED	NO	SGD	STATIC	50	96.67%	97.33%	97.25%	97.83%
14_BaseModel1_CLAHE	N_14_1_C	N	14	1	CLAHE	SelfDeveloped	FALSE	2	CLAHE	NO	SGD	STATIC	50	79.25%	78.15%	79.25%	79.5%
14_BaseModel2_MULTISCALE_tunedAdamLrScheduler	N_14_2_M_TALR	N	14	2	MULTISCALE_tunedAdamLrScheduler	SelfDeveloped	FALSE	3	MULTISCALE	YES	ADAM	DYNAMIC	50	84.00%	53.14%	74.8%	73.31%
14_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	N_14_3_M_TALR	N	14	3	MULTISCALE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	3	MULTISCALE	YES	ADAM	DYNAMIC	50	79.17%	77.83%	78.50%	77.61%
14_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	N_14_4_N_TALR	N	14	4	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	ADAM	DYNAMIC	50	87.33%	88.75%	88.04%	88.70%
14_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	N_14_5_N_TALR	N	14	5	NORMAL_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	1	NONE	YES	ADAM	DYNAMIC	50	91.17%	91.42%	91.29%	91.31%
14_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	N_14_6_C_TALR	N	14	6	CLAHE_tunedAdamLrScheduler	TransferLearningFeatureExtraction	FALSE	2	CLAHE	YES	ADAM	DYNAMIC	50	81.50%	81.17%	81.33%	81.23%
14_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	N_14_7_C_TALR	N	14	7	CLAHE_tunedAdamLrScheduler	TransferLearningFineTuning	FALSE	2	CLAHE	YES	ADAM	DYNAMIC	50	84.33%	86.17%	85.25%	86.22%
14_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	N_14_8_R_RAYLEIGH	N	14	8	RAYLEIGH	TransferLearningFineTuning	FALSE	4	RAYLEIGH	NO	SGD	STATIC	50	95.17%	95.08%	95.13%	95.10%
14_TransferLearningFineTuning_MobileNetV1_NORMAL	N_14_10_N_NORMAL	N	14	10	NORMAL	TransferLearningFineTuning	FALSE	1	NONE	NO	SGD	STATIC	50	99.33%	99.33%	99.33%	99.33%
14_TransferLearningFineTuning_MobileNetV2_NORMAL	N_14_9_N_NORMAL	N	14	9	NORMAL	TransferLearningFineTuning	FALSE	1	NONE	NO	SGD	STATIC	50	99.17%	99.17%	99.17%	99.17%

APPENDIX E: COMAPARISON OF THE BEST 10 MODELS IN STAGE 1 AND STAGE 2 AND THE ENSEMBLE MODEL

modelName	ModelTag	modelGroupMerged	valAccuracy	testAccuracy	ValTestAverage	testF1Score
01_BaseModel1_CLAHE	A_1_2_C	1	92.17%	91.17%	91.67%	91.10%
02_BaseModel2_MULTISCALE_tunedAdamLrScheduler	B_2_7_M_TALR	2	93.17%	92.83%	93.00%	92.73%
03_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	C_3_7_M_TALR	3	91.67%	90.33%	91.00%	90.18%
04_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	D_4_7_N_TALR	4	87.33%	88.75%	88.04%	88.70%
05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	E_5_7_N_TALR	5	91.17%	91.42%	91.29%	91.31%
06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	F_6_7_C_TALR	6	94.83%	95.17%	95.00%	95.16%
07_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	G_7_7_C_TALR	7	93.67%	94.00%	93.83%	93.97%
08_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	H_8_4_R	8	98.67%	98.92%	98.79%	98.91%
09_TransferLearningFineTuning_MobileNetV1_NORMAL	I_9_1_N	9	99.33%	99.33%	99.33%	99.33%
10_TransferLearningFineTuning_MobileNetV2_NORMAL	J_10_1_N	10	99.17%	99.17%	99.17%	99.17%
11_BaseModel1_CLAHE	K_11_1_C	1	90.00%	89.50%	89.75%	89.41%
11_TransferLearningFineTuning_MobileNetV2_NORMAL	K_11_10_N	10	92.83%	93.25%	93.04%	93.22%
11_BaseModel2_MULTISCALE_tunedAdamLrScheduler	K_11_2_M_TALR	2	90.00%	92.08%	91.04%	91.96%
11_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	K_11_3_M_TALR	3	88.17%	90.50%	89.33%	90.43%
11_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	K_11_4_N_TALR	4	87.83%	89.83%	88.83%	89.83%
11_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	K_11_5_N_TALR	5	90.50%	94.08%	92.29%	94.06%
11_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	K_11_6_C_TALR	6	95.00%	96.17%	95.58%	96.15%
11_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	K_11_7_C_TALR	7	91.67%	92.50%	92.08%	92.43%
11_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	K_11_8_R	8	84.50%	84.83%	84.67%	84.75%
11_TransferLearningFineTuning_MobileNetV1_NORMAL	K_11_9_N	9	88.67%	89.33%	89.00%	89.30%
11A_BaseModel1_CLAHE	L_11A_1_C	1	88.50%	89.92%	89.21%	89.88%
11A_TransferLearningFineTuning_MobileNetV2_NORMAL	L_11A_10_N	10	92.83%	94.25%	93.54%	94.20%
11A_BaseModel2_MULTISCALE_tunedAdamLrScheduler	L_11A_2_M_TALR	2	89.83%	90.92%	90.38%	90.77%
11A_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler	L_11A_3_M_TALR	3	87.50%	89.58%	88.54%	89.47%
11A_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler	L_11A_4_N_TALR	4	90.17%	89.42%	89.79%	89.54%
11A_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler	L_11A_5_N_TALR	5	92.33%	93.92%	93.13%	93.88%
11A_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler	L_11A_6_C_TALR	6	95.67%	97.00%	96.33%	96.99%
11A_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler	L_11A_7_C_TALR	7	90.17%	91.08%	90.63%	90.99%
11A_TransferLearningFineTuning_ResNet50V2_RAYLEIGH	L_11A_8_R	8	85.83%	85.42%	85.63%	85.30%
11A_TransferLearningFineTuning_MobileNetV1_NORMAL	L_11A_9_N	9	88.33%	89.58%	88.96%	89.50%
12_AugVotingModels	M_12_1234567_MIXED	12	96.67%	97.83%	97.25%	97.83%

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

APPENDIX F - PROJECT TIMELINE

COLOR LEGEND:

Activity Timeline

Activity Timeline Project Milestone



ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

APPENDIX F - PROJECT TIMELINE

COLOR LEGEND:

Activity Timeline

Project Milestone



ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

APPENDIX F - PROJECT TIMELINE

COLOR LEGEND:

Activity Timeline

Project Milestone



Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the course of the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session, and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: Rupesh Kumar Dey**Date:** 4th June 2022 **Meeting No:** 1**Project title:**

**ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING
FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES**

Supervisor's name: Dr Vazeerudeen**Supervisor's signature:** Vazeer**Items for discussion (noted by student before mandatory supervisory meeting):**

1. Introducing the updated topic for the project.
2. Clarify and discuss on the possible problem statements identified in the topic of study to proceed on.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Student provided explanation on the study topic desired to be undertaken and reasoning on choosing the topic.
2. Student proposed several problem statements to work on based on initial exploratory research.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. To proceed with chosen topic.
2. To streamline problems to focus on in the proposed study.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Student's name: Rupesh Kumar Dey

Date: 4th July 2022 Meeting No:2

Project title:

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

Supervisor's name: Dr Vazeerudeen

Supervisor's signature:



Items for discussion (noted by student before mandatory supervisory meeting):

1. Propose dataset to use for the project.
2. Discussion on overall project timeline planned and milestones.
3. Discussion on streamlined problem statements, aim and objectives.
4. Provide summary on literatures reviewed and the possible solutions that can be used to define the overall project methodology.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Proposed dataset is suitable for the project.
2. Proposed project timeline and planned milestones are suitable.
3. Streamlined problem statements and aim are suitable for the project. Several comments made to amend objectives given.
4. Overall literatures being reviewed are in line with the problem statement and the objectives of the project.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. Proceed to perform exploratory data analysis on the chosen dataset and image processing enhancement.
2. To make amendments to objectives
3. To finalise overall proposed project methodology to present in next meeting.
4. To focus on literatures that can be used as reference for proposed solutions.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Student's name: Rupesh Kumar Dey

Date: 29th July 2022 Meeting No:3

Project title:

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

Supervisor's name: Dr Vazerudeen

Supervisor's signature: 

Items for discussion (noted by student before mandatory supervisory meeting):

1. Discussion on literature review.
2. Discussion on proposed project methodology.
3. Outcomes of exploratory data analysis and image processing results.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Further discussion on literatures reviewed and focused literatures presented.
2. Overall proposed project methodology is acceptable. Minor comments given on the methodology flow.
3. Outcomes of exploratory data analysis and image processing results are adequate. To proceed with model development. Proposed methodology of using image processing to augment data is also suitable.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. To start on thesis writing on the introduction, problem statements literature review and also on methodology.
2. To start writing thesis section for results on exploratory data analysis and image processing.
3. To proceed with solution / model development

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Student's name: Rupesh Kumar Dey

Date: 15th August 2022 Meeting No: 4

Project title:

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

Supervisor's name: Dr Vazerudeen

Supervisor's signature: 

Items for discussion (noted by student before mandatory supervisory meeting):

1. Present findings for stage 1 model development.
2. Present progress on stage 2 model development
3. Discuss on proposed testing and evaluation methodology.
4. Presented progress on project thesis.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Progress on model development for stage 1 and progress for stage 2 is good. To continue and proceed. Some comments given on project folder structure to ensure neatness of project folder.
2. Proposed testing and evaluation methodology is suitable. To ensure that justification is provided in thesis for the evaluation method.
3. Progress on project thesis is sufficient. To continue working on thesis. Comment given to move large project methodology workflow figure into appendix.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. Continue working on thesis. Append thesis accordingly based on comments given.
2. To clean up project folder.
3. To continue with stage 2 model development.
4. Proceed with testing and evaluation.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Student's name: Rupesh Kumar Dey

Date: 15th Sept 2022 **Meeting No:** 5

Project title:

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

Supervisor's name: Dr Vazeerudeen

Supervisor's signature:

Vazeer

Items for discussion (noted by student before mandatory supervisory meeting):

1. Provide overview of results for project.
2. Provided overview on current thesis writing progress.
3. Provided overview of structure of discussion of results.
4. Clarify on required attachments.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Results obtained from testing is adequate and meets project objectives.
2. Progress on thesis writing is good. To continue and complete final components of the thesis. Ensure that formatting follows APU thesis writing formatting requirements.
3. Points presented for discussion section is adequate. Some comments given on the organization of points for better clarity and flow of presentation.
4. To ensure that summary of literatures reviewed, project schedule, meeting logs and ethics approval form is included in the appendices.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. To complete discussion, conclusion, and final formatting.
2. To share drafted final thesis for overall high-level review
3. To consolidate required appendices and include in final draft.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Student's name: Rupesh Kumar Dey

Date: 30th Sept 2022 **Meeting No:** 6

Project title:

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

Supervisor's name: Dr Vazeerudeen

Supervisor's signature:

Vazeer

Items for discussion (noted by student before mandatory supervisory meeting):

1. Discuss on comments on the final thesis draft
2. Discuss on final presentation content requirements
3. Clarify on final comments and clarifications.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Overall thesis structure is good. To furnish with required forms and attachments and get required signatures.
Prepare for final submission.
2. Discussed on points, flow and information required for final project presentation. Several pointers given. To prepare slides accordingly.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. Prepare for final presentation.
2. Prepare thesis for final submission.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

Project Log Sheet – Supervisory Session

Notes on use of the project log sheet:

1. This log sheet is designed for meetings of more than 15 minutes duration, of which there must be at minimum SIX (6) during the course of the project (SIX mandatory supervisory sessions).
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session, and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant section of the form.
5. The student should leave a copy (after the session) of the Project Log Sheet with the supervisor and to the administrator at the academic counter. A copy is retained by the student to be filed in the project file.
6. It is recommended that students bring along log sheets of previous meetings together with the project file during each supervisory session.
7. The log sheet is an important deliverable for the project and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the final year documentation, with sheets dated and numbered consecutively.

Student's name: Rupesh Kumar Dey**Date:** 30th Sept 2022 Meeting No:7**Project title:**

**ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING
FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES**

Supervisor's name: Mr Hamam Mokayed **Supervisor's signature:** Hamam Mokayed**Items for discussion (noted by student before mandatory supervisory meeting):**

1. Introducing the final project topic being undertaken, problem statements and objectives.
2. Introducing methodology undertaken.
3. Present high-level findings, discussion, and conclusion
4. Get feedback regarding work done and thesis. To share drafted thesis for comments.

Record of discussion (noted by student during mandatory supervisory meeting):

1. Student provided explanation on the final project work done.
2. Student shared drafted thesis. Several comments given regarding the structure of the thesis.

Action List (to be attempted or completed by student by the next mandatory supervisory meeting):

1. To update thesis based on comments provided.

Note: A student should make an appointment to meet his or her supervisor (via the consultation system) at least ONE (1) week prior to a mandatory supervisor session – please see document on project timelines. In the event a supervisor could not be booked for consultation, the project manager should be informed ONE (1) week prior to the session so that a meeting can be subsequently arranged.

APPENDIX H

Office Record	Receipt – APU Fast-Track Ethical Approval Student name: Rupesh Kumar Dey Student number: TP061720 Received by: Date: 01 Oct 2022
---------------	--

APU/APIIT FAST-TRACK ETHICAL APPROVAL FORM (STUDENTS)

Tick one box (level of study):

- POSTGRADUATE (PhD / MPhil / Masters)
 UNDERGRADUATE (Bachelor's degree)
 FOUNDATION / DIPLOMA / Other categories

Tick one box (purpose of approval):

- Thesis / Dissertation / FYP project
 Module assignment
 Other: _____

Title of Programme on which enrolled: Master of Science in Artificial Intelligence

Tick one box: Full-Time Study or Part-Time Study

ANALYSING WIRELESS CAPSULE ENDOSCOPY IMAGES USING DEEP LEARNING

Title of project / assignment: FRAMEWORKS TO CLASSIFY DIFFERENT GI TRACT DISEASES

Name of student researcher: Rupesh Kumar Dey

Name of supervisor / lecturer: Dr Vazeerudeen AbdulHameed

Student Researchers- please note that certain professional organisations have ethical guidelines that you may need to consult when completing this form.

Supervisors/Module Lecturers - please seek guidance from the Chair of the School Research Ethics Committee if you are uncertain about any ethical issue arising from this application.

		YES	NO	N/A
1	Will you describe the main procedures to participants in advance, so that they are informed about what to expect?			/
2	Will you tell participants that their participation is voluntary?			/
3	Will you obtain written consent for participation?			/
4	If the research is observational, will you ask participants for their consent to being observed?			/
5	Will you tell participants that they may withdraw from the research at any time and for any reason?			/
6	With questionnaires and interviews will you give participants the option of omitting questions they do not want to answer?			/
7	Will you tell participants that their data will be treated with full confidentiality and that, if published, it will not be identifiable as theirs?			/
8	Will you give participants the opportunity to be debriefed i.e., to find out more about the study and its results?			/

If you have ticked No to any of Q1-8, you should complete the full Ethics Approval Form.

		YES	NO	N/A
9	Will your project/assignment deliberately mislead participants in any way?			/
10	Is there any realistic risk of any participants experiencing either physical or psychological distress or discomfort?			/
11	Is the nature of the research such that contentious or sensitive issues might be involved? This includes research which could induce psychological stress, anxiety or humiliation, or cause more than minimal pain.			/
12	Does your research involve the use of sensitive materials?			/

	Eg, records of personal or sensitive confidential information,		
13	Does your research require external agency approval?		/
14	Does your research use hazardous or controlled substance?		/
15	Does your research require you to visit participants in their home or non-public space?		/
16	Does your research use genetically modified organisms?		/
17	Does your research investigate illegal activities or behaviours?		/
18	Does your research involve discussion or collection of information on potentially sensitive, embarrassing or distressing topics, administrative or secure data? This includes research involving respondents through internet where visual images are used, and where sensitive issues are discussed		/
19	Does your research involve invasive or potentially intrusive procedures?		/
20	Does your research involve administration of substances?		/
21	Will your research be involved in the collection/ processing of human tissue samples		/
22	Will your participants be receiving financial compensation for participating in your research?		/
23	Will your research data be used in the future after the conclusion of your project?		/
24	Will your research involve in processing sensitive data belonging to an organisation/person?		/
25	Will your research be collecting photographs, videos, and audio recordings of the participants?		/
26	Will the participants' personal particulars be known to any third party?		/
27	Will the participants' data confidentiality be made known to the public?		/
28	Will the research be conducted where the safety of the researchers maybe in question?		/
29	Will be the research be conducted outside of the UK and/or Malaysia?		/
30	Will your research involve human participants at premises other than those of the University?		/

If you have ticked Yes to any of Q9 – 30, you should complete the full Ethics Approval Form. In relation to question 10 this should include details of what you will tell participants to do if they should experience any problems (e.g., who they can contact for help). You may also need to consider risk assessment issues.

		YES	NO	N/A							
31	Does your project/assignment involve work with animals?			/							
32	<p>Do participants fall into any of the following special groups?</p> <p>Note that you may also need to obtain satisfactory clearance from the</p> <table border="1"> <tr><td>Children (under 18 years of age)</td></tr> <tr><td>People with communication or learning difficulties</td></tr> <tr><td>Patients</td></tr> <tr><td>People in custody</td></tr> <tr><td>People who could be regarded as vulnerable or lack capacity to make decision for themselves</td></tr> <tr><td>People engaged in illegal activities (eg drug taking)</td></tr> <tr><td>Groups of people whose relationship among each other allow one to have influence over the other such as: Carers and patients with chronic conditions; teachers and their students; prison authorities and prisoners;</td></tr> </table>	Children (under 18 years of age)	People with communication or learning difficulties	Patients	People in custody	People who could be regarded as vulnerable or lack capacity to make decision for themselves	People engaged in illegal activities (eg drug taking)	Groups of people whose relationship among each other allow one to have influence over the other such as: Carers and patients with chronic conditions; teachers and their students; prison authorities and prisoners;			/
Children (under 18 years of age)											
People with communication or learning difficulties											
Patients											
People in custody											
People who could be regarded as vulnerable or lack capacity to make decision for themselves											
People engaged in illegal activities (eg drug taking)											
Groups of people whose relationship among each other allow one to have influence over the other such as: Carers and patients with chronic conditions; teachers and their students; prison authorities and prisoners;											

	relevant authorities	employers and employees Deceased person's body parts or other human tissues including bodily fluids (e.g. blood, saliva). groups where permission of a gatekeeper is normally required for initial access to members. Human participants who are off-campus APU staff or students who wish to carry out investigations involving human participants at premises other than those of the University		
33	Does the project/assignment involve external funding or external collaboration where the funding body or external collaborative partner requires the University to provide evidence that the project/assignment had been subject to ethical scrutiny?			/

If you have ticked Yes to any Q31-33, you should complete the full Ethics Approval Form. There is an obligation on student and supervisor to bring to the attention of the APU School Research Ethics Committee any issues with ethical implications not clearly covered by the above checklist.

STUDENT RESEARCHER

Provide in the boxes below (plus any other appended details) information required in support of your application. THEN SIGN THE FORM.

Please Tick Boxes

I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU School Research Ethics Committee.	/
I am aware of APU liability policy and will make the necessary arrangement for insurance coverage of all researchers and participants of the project/assignment.	/
Give a brief description of participants, procedure of recruitment and procedure of data collection (methods, tests used etc) in up to 150 words.	
<p>Raw data is obtained from publicly available dataset in Kaggle (Link: https://www.kaggle.com/datasets/francismon/curated-colon-dataset-for-deep-learning). No external participants or recruitment is involved in the course of the study.</p>	
I also confirm that:	
i) All key documents e.g. consent form, information sheet, questionnaire/interview, and all material such as emails and posters for the purpose of recruitment of participants are appended to this application.	/
Or	
ii) Any key documents e.g. consent form, information sheet, questionnaire/interview schedules which need to be finalised following initial investigations will be submitted for approval by the project/assignment supervisor/module lecturer before they are used in primary data collection.	/

Signed:

 (Student Researcher)

Print Name: Rupesh Kumar Dey Date: 28 Sept 2022

Within this document, any variation to the items considered which affects ethical issues of the stated research will require submission of a revised research plan and research methodology details; as a consequence, new ethical consent may need to be sought.

The completed form (and any attachments) should be submitted for consideration by your Supervisor/Module Lecturer

**SUPERVISOR/MODULE LECTURER
PLEASE CONFIRM THE FOLLOWING:**

Please Tick Box

I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU School Research Ethics Committee	<input checked="" type="checkbox"/>
I have checked and approved the key documents required for this proposal (e.g. consent form, information sheet, questionnaire, interview schedule)	<input checked="" type="checkbox"/>

SUPERVISOR AND SECOND ACADEMIC SIGNATORY

STATEMENT OF ETHICAL APPROVAL (please delete as appropriate)

- 1) THIS PROJECT/ASSIGNMENT HAS BEEN CONSIDERED USING AGREED APU PROCEDURES AND IS NOW APPROVED
- 2) THIS PROJECT/ASSIGNMENT HAS BEEN APPROVED IN PRINCIPLE AS INVOLVING NO SIGNIFICANT ETHICAL IMPLICATIONS, BUT FINAL APPROVAL FOR DATA COLLECTION IS SUBJECT TO THE SUBMISSION OF KEY DOCUMENTS FOR APPROVAL BY SUPERVISOR (see Appendix A)


Signed... Print Name... Date...
(Supervisor/Lecturer) Dr. Vazeerudeen 01 Oct 2022

Signed... Print Name... Date...
(Second Academic Signatory)

Office Record	Receipt – Appendix A (APU Fast-Track Ethics Form)
Date Received:	Student name: Rupesh Kumar Dey
Received by:	Student number: TP061720

**APPENDIX A
AUTHORISATION FOR USE OF KEY DOCUMENTS**

Completion of Appendix A is required when for good reasons key documents are not available when a fast track application is approved by the supervisor/module lecturer and second academic signatory.

I have now checked and approved all the key documents associated with this proposal e.g. consent form, information sheet, questionnaire, interview schedule

Title of project/assignment.....

.....

Name of student researcher

Student ID: Intake:

Signed..... Print Name..... Date.....
(Supervisor/Lecturer)