Analysing Wireless Capsule Endoscopy Images Using Deep Learning Frameworks to Classify Different GI Tract Diseases

Rupesh Kumar Dey
School of Computing and Technology
Asia Pacific University of Technology
& Innovation (APU)
Kuala Lumpur, Malaysia
TP061720@mail.apu.edu.my

Abstract— **GI Tract related diseases are one of the most prevalent in today's society. Studies have shown that continuous monitoring, early detection, and treatment of these diseases are imperative in improving patients' recovery rate. Wireless Capsule Endoscopy (WCE) is an innovative imaging technology that enables invasive imaging of the GI Tract. 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. The study aims to design and develop a generalized multiclass CNN classification algorithm to be used in CAD system for diagnosis of various GI tract diseases by analyzing WCE GI tract images with varying tract lining lesions. CNN classification-based solution framework encompassing various network architectures, image processing enhancement techniques and data augmentation methods are proposed. Three histogram stretching based enhancement techniques were introduced to enhance the quality of the raw image prior to performing classification. Data augmentation was performed as well. Different network architectures of self-developed architectures, transfer learning feature extraction, fine tuning and an ensemble of models were developed. The results were analyzed, 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.**

Keywords—GI Tract, CNN, AI, Medical, Neural Networks

I. INTRODUCTION

The WHO reported that approximately 765000 deaths were caused by stomach cancer. Colon and rectum cancer contributed to approximately 525000 deaths globally [1]. Studies from [2] reported 135430 new GI tract diseases occurrences since 2017,. 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 [3]. These statistics show how common, and severe GI tract diseases are globally. Clinical data shows that the 5-year survival rate of these diseases are between 23-27% with some diseases going as low as 4% [2]. The survival rate however rises exponentially to around 95% if these lesions are detected during early stages [4]. Early detection enables early treatment and prevention

Developments in Artificial Intelligence and Image Processing has unlocked new solutions for many medical diagnosis problems. Manual tasks can now be replaced with more automated and efficient methods. The general approach by reviewed literatures can be categorized into 2 main pathways which are Image Processing Feature Extraction + Machine Learning algorithms (referred to as traditional feature extraction) and Deep Learning. The inherent problem with the traditional feature extraction method is that they are very limited to the specific conditions they are built for and not as robust as they have to manually tailored for specific applications. An alternative solution are Deep Learning models, specifically Convolutional Neural Networks (CNN). These models are developed to automatically learn, analyse, and improve their classification performance from large amounts of data, making them dynamic and robust for many CAD related problems.

The rest of the paper is structured as follows. Section II defines the problem statement, Section III details the related works in the field, Section IV provides a brief overview of the dataset, Section V then details the project methodology, Section VI presents the findings and results, Section VII then provides a detailed analysis of the results followed by conclusion and future recommendations in Section VIII and IX.

II. PROBLEM STATEMENT

Most works for diagnosis of GI tract diseases diagnosis gravitate towards binary classification problems namely, by [5] and [1]. Binary classification-based solutions lack the refinement for practical applications. The application of transfer learning has also been gaining traction over the years. Though transfer learning assists in improving the CNN models' performance especially when there is lack of data, it can only do so to a certain extent. Taking and applying them directly for medical related CAD applications may not necessarily results in the optimal results. Retraining and fine-tuning these models with sufficient medical images are still required. Self-developed architectures and transfer learning feature extraction models are commonly proposed solutions. There is a lack of work pertaining on other techniques such as transfer learning fine tuning and ensemble model. Image enhancements are known to improve the quality of images. Many of the reviewed 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. Many augmentation techniques used are only geometric i.e., rotation, translation and flipping. There is a lack of techniques that manipulate the contrast, colour, and illumination of the image. Developing a model that is adapt to a single source of image, may result in the model not being adaptable to newly acquired images from different sources. the This study aims to address the abovementioned issues.

III. Related Works

A. *Self Developed Architectures*

[5] 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. Different batches of input data were used as input for the CNN to learn from. An aggregation of the classification results from a total of 5 networks were considered. The proposed solution was capable of achieving an overall 95% accuracy on static images and 93% on a live video dataset. [6] developed an 8-layer CNN model to detect bleeding images in the GI tract. 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 achieved an overall recall rate of 99.2%. [7] proposed using a single CNN architecture with different input data pre-processing to classify between 6 different motility states of the small intestine. Typical images are processed using single or 3 RGB channels. However, this study included 2 additional channels which are, the Laplacian of the image brightness channel L and, the Hessian of the image brightness represented by H, inspired by [8], [9]. The authors experimented with a late-fusion and an early-fusion workflow. The late-fusion architecture model considers 3 different networks that takes in input images of RGB, H and L separately. The early-fusion architecture concatenated the RGB, H and L channels at the input head and fed them into a 5-channel input CNN architecture. Results showed that late-fusion workflow produced better results. RGB channels expressed colour representations whereas L and H feature maps expressed structure and textural information about the image. [10] compared the performance of Feature Extraction + Machine Learning against CNN architectures to detect polyps. The SVM model was adopted for the feature extraction + machine learning approach using features of Histogram of Oriented Gradient (HOG) combined with hue Histogram features prior to feeding the features into the SVM model. Results showed that the CNN based solution performed better. Authors of [11] 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. [12] developed a 2 stage CNN network to detect perforations in the Tympanic Membrane of the ear, from medical endoscopic ear images. Stage 1 aimed at detecting the presence of a Tympanic Membrane. The 2nd stage performed binary classification to detect perforations. The stage 1 CNN achieved 98.7% accuracy and stage 2 CNN achieved 87.2% accuracy.

B. *Transfer Learning Architectures*

[1] proposed a GI tract bleeding image recognition model using transfer learning. The authors compared the performance of 2 architectures which were the MobileNet + Additional Layers against Additional Layers as a standalone CNN. The combination of the MobileNet + Additional Layers produced better results, achieving an accuracy of 99.3%. [13] compared the performance of 3 different state of the art networks of LeNet, AlexNet, GoogLeNet and VGG networks to detect bleeding from GI tract images. All of the proposed networks achieved an accuracy of more than 95%. The authors also compared the training time for each of these models. The VGG network architecture took the longest followed by GooLeNet, AlexNet and LeNet respectively. [14] performed a step wise Fine Tuning training scheme to train a Deep Learning network. The authors performed 2 stages of training, Stage 1 and Stage 2 using VGG-16, AlexNet and Inception-V3 networks. Stage 1 training tuned the network's weights to be familiar with medical pathology images and features. Stage 2 then trained these CNNs on a smaller dataset of malignant and benign pathology images to perform the final classification. Results showed that the 2 stage algorithm performed significantly better compared to the single stage training framework. [15] performed transfer learning Fine Tuning using GoogleNet, ResNet-50 and AlexNet networks, pretrained on the ImageNet dataset to classify between different types of polyps. Input image pre-processing steps were performed using averaging filters. The dataset used was the Kvasir dataset [16]. [17] developed a CNN based CAD system that analysed Narrow-Band Imaging (NBI) of the colon to detect Polyps. 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 CNN model achieved an accuracy of 90.1%, sensitivity of 96.3% and a specificity of 78.1%. [18] 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 authors used Genetic Algorithm (GA) together with the CNN to determine the best combination of features from the final layer as input into the SVM classifier. The solution was able to achieve 99.8% accuracy.

[19] leveraged on the transfer learning of GooLeNet architecture to classify and detect 6 different anatomical locations of the GI Tract by analysing Esophagogastroduodenoscopy imaging. The model achieved an overall AUC of 99% detection for the upper, middle, and lower stomach. [20] developed 3 different state of the art networks of VGG-16, Inception-V3 and InceptionResNet-V2 models using transfer learning fine tuning to classify NBI images into 2 different classes of early-gastric cancer (EGC) vs non-early gastric cancer. The proposed solution was also compared to traditional Feature Extraction + Machine Learning models. The 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. The authors also observed that unfreezing and training more layers of the base models led to better model performance. Results also showed that a bigger input image size produced much better results. The Feature Extraction + Machine Learning approach used LBP, CLBP, Gabor and GLCM features with an SVM model. [21] developed an automatic tumour detection framework from gastric pathology images using the ResNet-50 network architecture. 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 of the model ensemble. This manged to

increase the F1 score of the solution to 96% [20] explored and studied various CNN architectures and their training configurations to classify between 3 classes of gastritis. A variety of state-of-the-art network architectures of VGG-16, InceptionNet-V3, InceptionResNetV2 and ResNet-50 were used. [22] 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 state of the art models of Resnet-101 and Inception-V3. 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%. [23] compared the performance of VGG-16 architecture and ResNet-50 architecture in detecting lung cancer by analysing histopathological slides of images. The two networks were trained for a binary classification problem. 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.

C. Data Augmentation

To counter imbalanced data and to increase the dataset size, [11] implemented augmentation techniques of crop, flip, rotation, and smoothing using gaussian filters. [5] and [13] 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. [24] applied geometric augmentations to raw MRI images of the small bowel using flipping and shifting. [1] applied data augmentation technique of flipping. The authors however noted that augmenting and creating too many synthetic images may lead to data redundancy [25], [26]. [15] used techniques of flipping, zooming, shifting and rotation on the Kvasir dataset [16]. [19] augmented images using rotation. [21] utilized affine transformation using a combination of rotation, scaling, and horizontal and vertical mirroring to augment image data of gastric pathology images. [12] applied geometrical augmentations of sheer range, rotation and horizontal flipping to create synthetic Tympanic Membrane images. [15] performed augmentation of flipping, zooming shifting and rotation. [20] applied geometric augmentation techniques of rotation, width shifting, height shifting, zooming, horizontal flip, vertical flip, and scale normalization.

D. Image Processing

[5] performed histogram equalization and colour palette reduction using minimum variance quantization on WCE GI Tract images. [18] implemented a sequence of image processing techniques on the input images. 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 to denoise the image. [15] applied averaging filter to GI tract images to enhance features. [27] 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). [28] applied Rayleigh CLAHE algorithm on retinal images. The authors performed the histogram stretching on the Intensity (I) channel using the HSI colour model instead of RGB.

IV. Dataset

Several criteria were set as basic requirements to choose the dataset. The dataset had to be a multiclass dataset, comprised of images from different sources i.e., the images were taken from different cameras and patients, the size of the images had to be at least 300 x 300 pixels and above, and RGB images were used for this study. A minimum number of 1000 images was set as a basic required quantity for each class.

The dataset was sourced from Kaggle titled WCE Curated Colon Disease Dataset Deep Learning [29]. The dataset is comprised of WCE images from 3 sources which are [16], [30], [31]. 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

V. Methodology

The overall project high level framework is defined as Fig. 1 below.

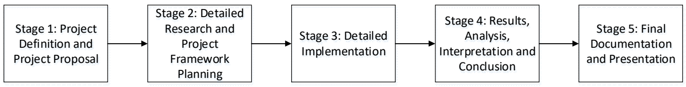

Stage 1: Project Definition and Project Proposal
Stage 2: Detailed Research and Project Framework Planning
Stage 3: Detailed Implementation
Stage 4: Results, Analysis, Interpretation and Conclusion
Stage 5: Final Documentation and Presentation

Fig. 1 Project High Level Framework.

Fig. 2 depicts the overall implementation process flow for this project and is broken down in 7 stages.

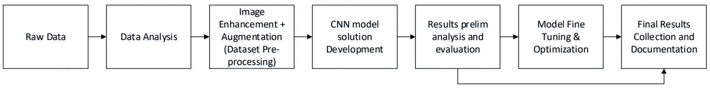

Raw Data
Data Analysis
Image Enhancement + Augmentation (Dataset Pre-processing)
CNN model solution Development
Results prelim analysis and evaluation
Model Fine Tuning & Optimization
Final Results Collection and Documentation

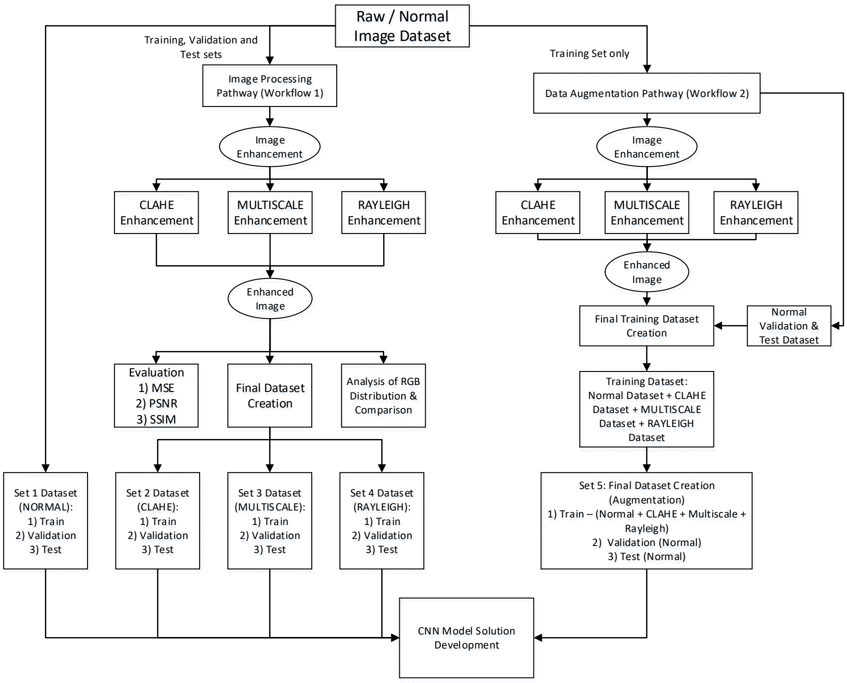
Fig. 2 Implementation Process Flow

Implementation began with the sourcing and 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. 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. Cross stage model performance comparison was also performed.

A. Image Enhancement and Augmentation (Dataset Pre-processing)

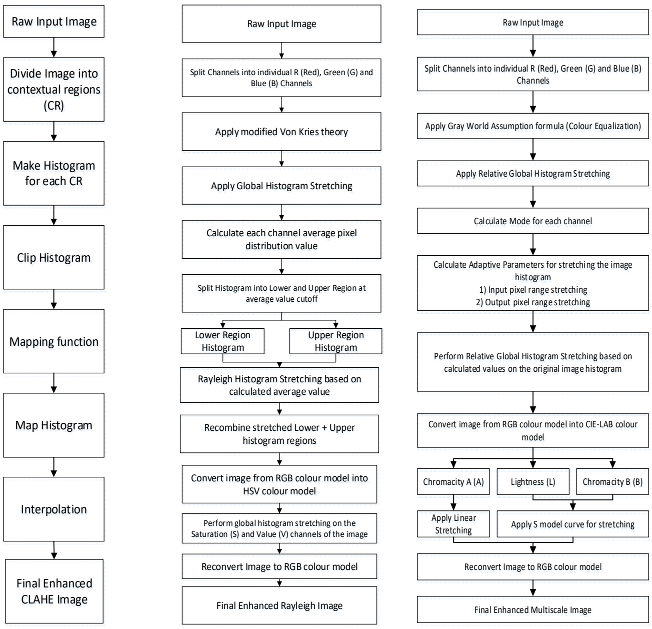
Image processing stage is broken down into two major workflows, one is in the form of performing dataset pre-processing and two is in the form of data augmentation. In workflow 1, three different image processing algorithms were used to pre-process the raw images. For reproducibility and for ease of use during training, 3 additional sets of datasets were created. Each dataset would contain training, validation, and test sets. 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 were left untouched. The 3 different image processing algorithms were applied to the training images and added to the original data. This 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 resulted in a total of 5 datasets (original + 3 image processed datasets + 1 Augmentation dataset) that will be used for developing the CNN models. The overall process is simplified as Fig. 3 below.

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. The process flow of each of these image processing techniques can be seen in Fig. 4.



Raw / Normal Image Dataset
Training, Validation and Test sets
Training Set only
Image Processing Pathway (Workflow 1)
Data Augmentation Pathway (Workflow 2)
Image Enhancement
CLAHE Enhancement
MULTISCALE Enhancement
RAYLEIGH Enhancement
Enhanced Image
Image Enhancement
CLAHE Enhancement
MULTISCALE Enhancement
RAYLEIGH Enhancement
Enhanced Image
Final Training Dataset Creation
Normal Validation & Test Dataset
Evaluation 1) MSE 2) PSNR 3) SSIM
Final Dataset Creation
Analysis of RGB Distribution & Comparison
Training Dataset: Normal Dataset + CLAHE Dataset + MULTISCALE Dataset + RAYLEIGH Dataset
Set 1 Dataset (NORMAL): 1) Train 2) Validation 3) Test
Set 2 Dataset (CLAHE): 1) Train 2) Validation 3) Test
Set 3 Dataset (MULTISCALE): 1) Train 2) Validation 3) Test
Set 4 Dataset (RAYLEIGH): 1) Train 2) Validation 3) Test
Set 5: Final Dataset Creation (Augmentation) 1) Train – (Normal + CLAHE + Multiscale + Rayleigh) 2) Validation (Normal) 3) Test (Normal)
CNN Model Solution Development

Fig. 3 Detailed Process flow of Image Enhancement + Augmentation



Raw Input Image
Divide Image into contextual regions (CR)
Make Histogram for each CR
Clip Histogram
Mapping function
Map Histogram
Interpolation
Final Enhanced CLAHE Image
Raw Input Image
Split Channels into individual R (Red), Green (G) and Blue (B) Channels
Apply modified Von Kries theory
Apply Global Histogram Stretching
Calculate each channel average pixel distribution value
Split Histogram into Lower and Upper Region at average value cutoff
Lower Region Histogram
Upper Region Histogram
Rayleigh Histogram Stretching based on calculated average value
Recombine stretched Lower + Upper histogram regions
Convert image from RGB colour model into HSV colour model
Perform global histogram stretching on the Saturation (S) and Value (V) channels of the image
Reconvert Image to RGB colour model
Final Enhanced Rayleigh Image
Raw Input Image
Split Channels into individual R (Red), Green (G) and Blue (B) Channels
Apply Gray World Assumption formula (Colour Equalization)
Apply Relative Global Histogram Stretching
Calculate Mode for each channel
Calculate Adaptive Parameters for stretching the image histogram 1) Input pixel range stretching 2) Output pixel range stretching
Perform Relative Global Histogram Stretching based on calculated values on the original image histogram
Convert image from RGB colour model into CIE-LAB colour model
Chromacity A (A)
Lightness (L)
Chromacity B (B)
Apply Linear Stretching
Apply S model curve for stretching
Reconvert Image to RGB colour model
Final Enhanced Multiscale Image

(a) CLAHE (b) RAYLEIGH (c) MULTISCALE

Fig. 4 Image Processing Enhancement Algorithm Process Workflow

A summary of the datasets created post image processing is shown in Table 1 below

Table 1 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)

B. *CNN Solution Development*

A high-level process flow of the entire CNN solution development phase is shown in Fig. 5 below. A detailed process flow for Stage 1 and Stage 2 can be referred to in Fig. 6 and Fig. 7

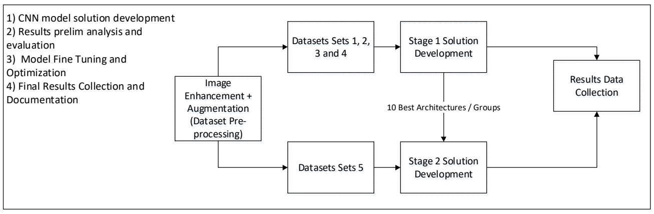

1) CNN model solution development
2) Results prelim analysis and evaluation
3) Model Fine Tuning and Optimization
4) Final Results Collection and Documentation
Image Enhancement + Augmentation (Dataset Pre-processing)
Datasets Sets 1, 2, 3 and 4
Stage 1 Solution Development
10 Best Architectures / Groups
Datasets Sets 5
Stage 2 Solution Development
Results Data Collection

Fig. 5 Detailed Process flow of CNN Solution Development

The entire CNN solution development process was broken down into 2 main stages of Stage 1 and Stage 2. 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. A total of 70 CNN models, categorized into 10 architectures (groups) were developed. For each architecture, the model was trained and tested on datasets 1 to 4. The best model out of the 4 will be further optimized with different learning rates, optimizer, and a combination of both to create an additional 3 models for each group.

Self-developed CNN architectures are architectures that were developed and trained from the ground-up. Two out of the ten groups constitute self-developed architectures. Transfer learning models on the other hand used state of the art CNN models of 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. A total of 4 groups constituted transfer learning feature extraction and the remaining 4 constitute transfer learning fine tuning. 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. The aim of this mode of training is to induce variability in the range of images that was used to train the models to enhance the model's generalization capability. 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.

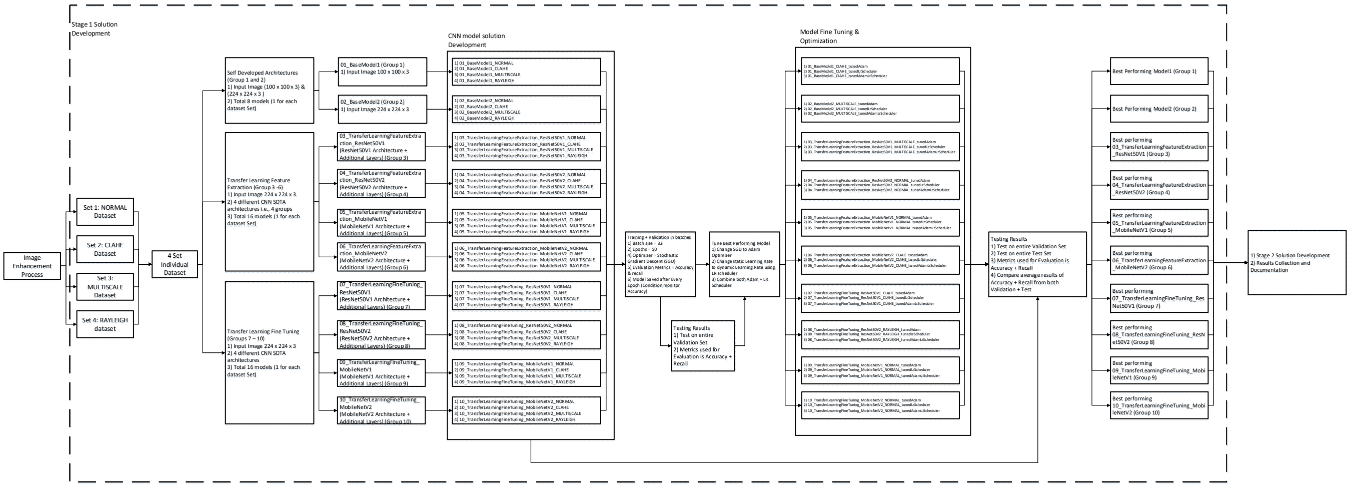

Stage 1 Solution Development
Image Enhancement Process
Set 1: NORMAL Dataset
Set 2: CLAHE Dataset
Set 3: MULTISCALE Dataset
Set 4: RAYLEIGH dataset
4 Set Individual Dataset
Self Developed Architectures (Group 1 and 2) 1) Input Image (100 x 100 x 3) & (224 x 224 x 3) 2) Total 8 models (1 for each dataset Set)
Transfer Learning Feature Extraction (Group 3 -6) 1) Input Image 224 x 224 x 3 2) 4 different CNN SOTA architectures i.e., 4 groups 3) Total 16 models (1 for each dataset Set)
Transfer Learning Fine Tuning (Groups 7 – 10) 1) Input Image 224 x 224 x 3 2) 4 different CNN SOTA architectures 3) Total 16 models (1 for each dataset Set)
01_BaseModel1 (Group 1) 1) Input Image 100 x 100 x 3
02_BaseModel2 (Group 2) 1) Input Image 224 x 224 x 3
03_TransferLearningFeatureExtraction_ResNet50V1 (ResNet50V1 Architecture + Additional Layers) (Group 3)
04_TransferLearningFeatureExtraction_ResNet50V2 (ResNet50V2 Architecture + Additional Layers) (Group 4)
05_TransferLearningFeatureExtraction_MobileNetV1 (MobileNetV1 Architecture + Additional Layers) (Group 5)
06_TransferLearningFeatureExtraction_MobileNetV2 (MobileNetV2 Architecture + Additional Layers) (Group 6)
07_TransferLearningFineTuning_ResNet50V1 (ResNet50V1 Architecture + Additional Layers) (Group 7)
08_TransferLearningFineTuning_ResNet50V2 (ResNet50V2 Architecture + Additional Layers) (Group 8)
09_TransferLearningFineTuning_MobileNetV1 (MobileNetV1 Architecture + Additional Layers) (Group 9)
10_TransferLearningFineTuning_MobileNetV2 (MobileNetV2 Architecture + Additional Layers) (Group 10)
CNN model solution Development
1) 01_BaseModel1_NORMAL 2) 01_BaseModel1_CLAHE 3) 01_BaseModel1_MULTISCALE 4) 01_BaseModel1_RAYLEIGH
1) 02_BaseModel2_NORMAL 2) 02_BaseModel2_CLAHE 3) 02_BaseModel2_MULTISCALE 4) 02_BaseModel2_RAYLEIGH
1) 03_TransferLearningFeatureExtraction_ResNet50V1_NORMAL 2) 03_TransferLearningFeatureExtraction_ResNet50V1_CLAHE 3) 03_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE 4) 03_TransferLearningFeatureExtraction_ResNet50V1_RAYLEIGH
1) 04_TransferLearningFeatureExtraction_ResNet50V2_NORMAL 2) 04_TransferLearningFeatureExtraction_ResNet50V2_CLAHE 3) 04_TransferLearningFeatureExtraction_ResNet50V2_MULTISCALE 4) 04_TransferLearningFeatureExtraction_ResNet50V2_RAYLEIGH
1) 05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL 2) 05_TransferLearningFeatureExtraction_MobileNetV1_CLAHE 3) 05_TransferLearningFeatureExtraction_MobileNetV1_MULTISCALE 4) 05_TransferLearningFeatureExtraction_MobileNetV1_RAYLEIGH
1) 06_TransferLearningFeatureExtraction_MobileNetV2_NORMAL 2) 06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE 3) 06_TransferLearningFeatureExtraction_MobileNetV2_MULTISCALE 4) 06_TransferLearningFeatureExtraction_MobileNetV2_RAYLEIGH
1) 07_TransferLearningFineTuning_ResNet50V1_NORMAL 2) 07_TransferLearningFineTuning_ResNet50V1_CLAHE 3) 07_TransferLearningFineTuning_ResNet50V1_MULTISCALE 4) 07_TransferLearningFineTuning_ResNet50V1_RAYLEIGH
1) 08_TransferLearningFineTuning_ResNet50V2_NORMAL 2) 08_TransferLearningFineTuning_ResNet50V2_CLAHE 3) 08_TransferLearningFineTuning_ResNet50V2_MULTISCALE 4) 08_TransferLearningFineTuning_ResNet50V2_RAYLEIGH
1) 09_TransferLearningFineTuning_MobileNetV1_NORMAL 2) 09_TransferLearningFineTuning_MobileNetV1_CLAHE 3) 09_TransferLearningFineTuning_MobileNetV1_MULTISCALE 4) 09_TransferLearningFineTuning_MobileNetV1_RAYLEIGH
1) 10_TransferLearningFineTuning_MobileNetV2_NORMAL 2) 10_TransferLearningFineTuning_MobileNetV2_CLAHE 3) 10_TransferLearningFineTuning_MobileNetV2_MULTISCALE 4) 10_TransferLearningFineTuning_MobileNetV2_RAYLEIGH
Training + Validation in batches 1) Batch size = 32 2) Epochs = 50 4) Optimizer = Stochastic Gradient Descent (SGD) 5) Evaluation Metrics = Accuracy & recall 6) Model Saved after Every Epoch (Condition monitor Accuracy)
Testing Results 1) Test on entire Validation Set 2) Metrics used for Evaluation is Accuracy + Recall
Tune Best Performing Model 1) Change SGD to Adam Optimizer 2) Change static Learning Rate to dynamic Learning Rate using LR scheduler 3) Combine both Adam + LR Scheduler
Model Fine Tuning & Optimization
1) 01_BaseModel1_CLAHE_tunedAdam 2) 01_BaseModel1_CLAHE_tunedLrScheduler 3) 01_BaseModel1_CLAHE_tunedAdamLrScheduler
1) 02_BaseModel2_MULTISCALE_tunedAdam 2) 02_BaseModel2_MULTISCALE_tunedLrScheduler 3) 02_BaseModel2_MULTISCALE_tunedAdamLrScheduler
1) 03_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdam 2) 03_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedLrScheduler 3) 03_TransferLearningFeatureExtraction_ResNet50V1_MULTISCALE_tunedAdamLrScheduler
1) 04_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdam 2) 04_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedLrScheduler 3) 04_TransferLearningFeatureExtraction_ResNet50V2_NORMAL_tunedAdamLrScheduler
1) 05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdam 2) 05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedLrScheduler 3) 05_TransferLearningFeatureExtraction_MobileNetV1_NORMAL_tunedAdamLrScheduler
1) 06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdam 2) 06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedLrScheduler 3) 06_TransferLearningFeatureExtraction_MobileNetV2_CLAHE_tunedAdamLrScheduler
1) 07_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdam 2) 07_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedLrScheduler 3) 07_TransferLearningFineTuning_ResNet50V1_CLAHE_tunedAdamLrScheduler
1) 08_TransferLearningFineTuning_ResNet50V2_RAYLEIGH_tunedAdam 2) 08_TransferLearningFineTuning_ResNet50V2_RAYLEIGH_tunedLrScheduler 3) 08_TransferLearningFineTuning_ResNet50V2_RAYLEIGH_tunedAdamLrScheduler
1) 09_TransferLearningFineTuning_MobileNetV1_NORMAL_tunedAdam 2) 09_TransferLearningFineTuning_MobileNetV1_NORMAL_tunedLrScheduler 3) 09_TransferLearningFineTuning_MobileNetV1_NORMAL_tunedAdamLrScheduler
1) 10_TransferLearningFineTuning_MobileNetV2_NORMAL_tunedAdam 2) 10_TransferLearningFineTuning_MobileNetV2_NORMAL_tunedLrScheduler 3) 10_TransferLearningFineTuning_MobileNetV2_NORMAL_tunedAdamLrScheduler
Testing Results 1) Test on entire Validation Set 2) Test on entire Test Set 3) Metrics used for Evaluation is Accuracy + Recall 4) Compare average results of Accuracy + Recall from both Validation + Test
Best Performing Model1 (Group 1)
Best Performing Model2 (Group 2)
Best performing 03_TransferLearningFeatureExtraction_ResNet50V1 (Group 3)
Best performing 04_TransferLearningFeatureExtraction_ResNet50V2 (Group 4)
Best performing 05_TransferLearningFeatureExtraction_MobileNetV1 (Group 5)
Best performing 06_TransferLearningFeatureExtraction_MobileNetV2 (Group 6)
Best performing 07_TransferLearningFineTuning_ResNet50V1 (Group 7)
Best performing 08_TransferLearningFineTuning_ResNet50V2 (Group 8)
Best performing 09_TransferLearningFineTuning_MobileNetV1 (Group 9)
Best performing 10_TransferLearningFineTuning_MobileNetV2 (Group 10)
1) Stage 2 Solution Development 2) Results Collection and Documentation

Fig. 6 Detailed Stage 1 Model Development

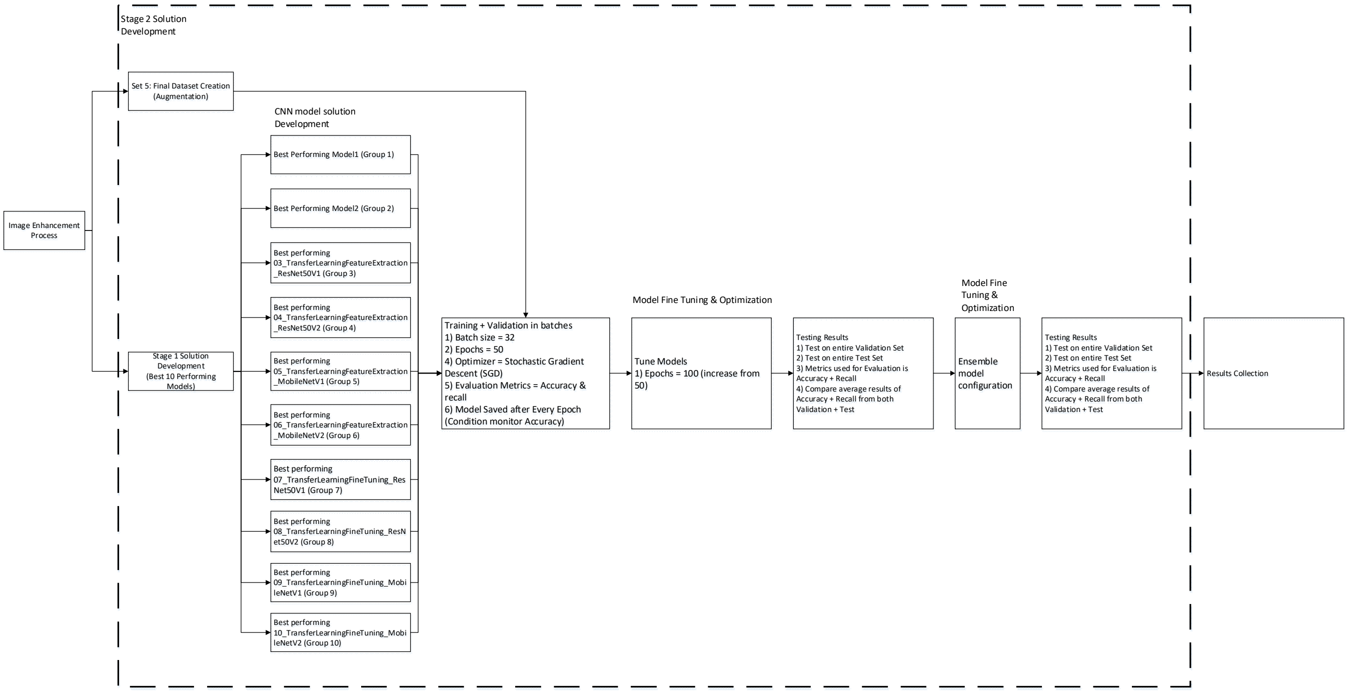

Stage 2 Solution Development
Image Enhancement Process
Set 5: Final Dataset Creation (Augmentation)
Stage 1 Solution Development (Best 10 Performing Models)
CNN model solution Development
Best Performing Model1 (Group 1)
Best Performing Model2 (Group 2)
Best performing 03_TransferLearningFeatureExtraction_ResNet50V1 (Group 3)
Best performing 04_TransferLearningFeatureExtraction_ResNet50V2 (Group 4)
Best performing 05_TransferLearningFeatureExtraction_MobileNetV1 (Group 5)
Best performing 06_TransferLearningFeatureExtraction_MobileNetV2 (Group 6)
Best performing 07_TransferLearningFineTuning_ResNet50V1 (Group 7)
Best performing 08_TransferLearningFineTuning_ResNet50V2 (Group 8)
Best performing 09_TransferLearningFineTuning_MobileNetV1 (Group 9)
Best performing 10_TransferLearningFineTuning_MobileNetV2 (Group 10)
Training + Validation in batches
1) Batch size = 32
2) Epochs = 50
4) Optimizer = Stochastic Gradient Descent (SGD)
5) Evaluation Metrics = Accuracy & recall
6) Model Saved after Every Epoch (Condition monitor Accuracy)
Model Fine Tuning & Optimization
Tune Models
1) Epochs = 100 (increase from 50)
Testing Results
1) Test on entire Validation Set
2) Test on entire Test Set
3) Metrics used for Evaluation is Accuracy + Recall
4) Compare average results of Accuracy + Recall from both Validation + Test
Model Fine Tuning & Optimization
Ensemble model configuration
Testing Results
1) Test on entire Validation Set
2) Test on entire Test Set
3) Metrics used for Evaluation is Accuracy + Recall
4) Compare average results of Accuracy + Recall from both Validation + Test
Results Collection

Fig. 7 Detailed Stage 2 Model Development

C. Individual CNN Architectures

The architecture of the developed models in Stage 1 and 2 are shown as below in Fig. 8 and Fig. 9. For transfer learning models, the pre-trained ImageNet models were 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. The architectures for the transfer learning models (feature extraction and fine tuning) are shown in Fig. 10

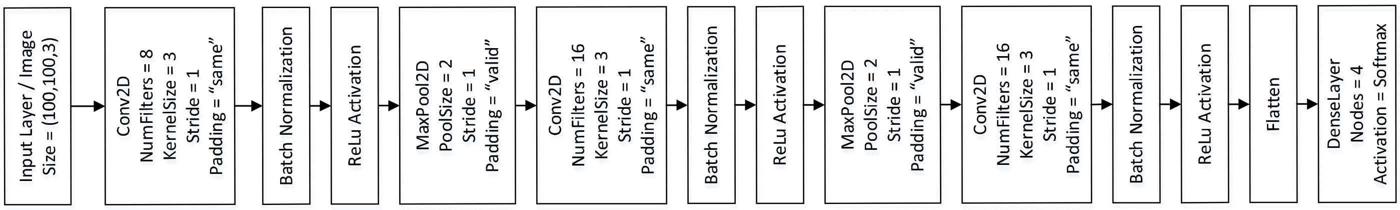


Fig. 8 Architecture of 01_Base_model1 (Group 1)

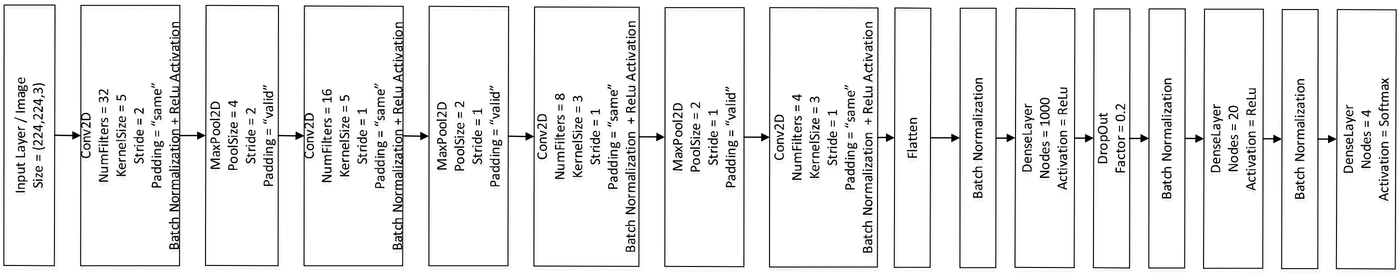


Fig. 9 Architecture of 02_baseModel2 (Group 2)

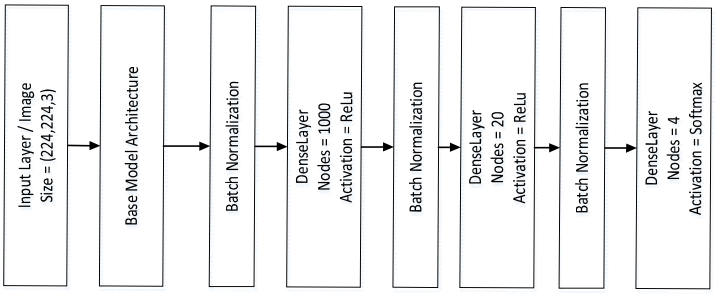


Fig. 10 Architecture of Transfer Learning Models

D. *Stage Developed 2 Models*

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. 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. These group of models were denoted as Group 11A. Finally, an ensemble comprised of the optimized 10 models from Stage 2 was built whereby max voting is used to finalize the results. This final model is denoted as Group 12.

E. *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. 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 towards the end. Finally, a combination the Adam optimizer and dynamic learning rate was implemented to observe how the synergistic effects of both these parameters.

F. *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. 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 validation set. The results produced would then be used as a determinant on the best model for each group to further optimize. Optimization would then produce 3 additional models. These 3 additional models would be tested on the validation set as well. The complete 7 architectures would additionally be tested on the 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.

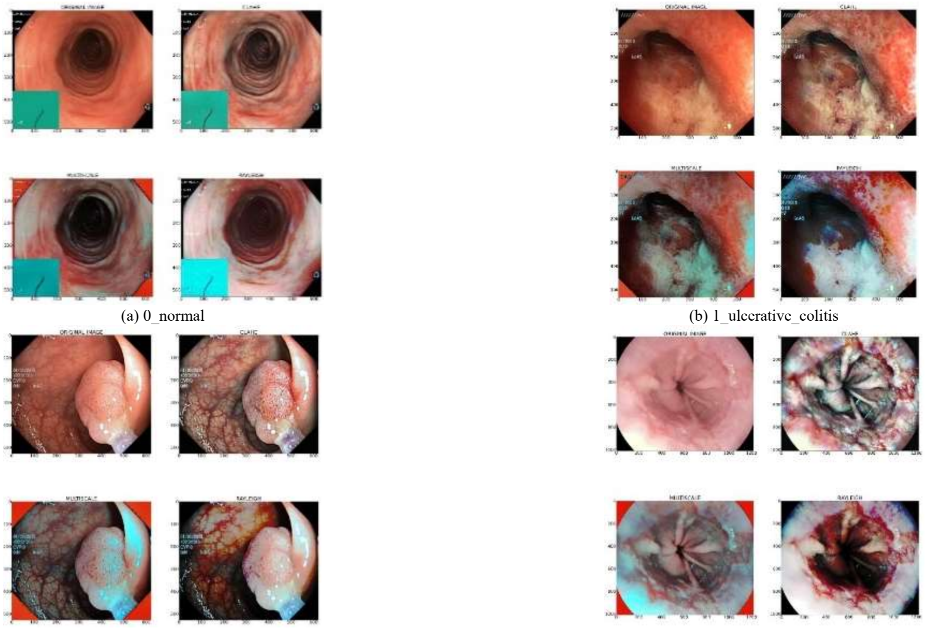
G. *Tools and Frameworks*

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. 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. Training of the models were 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.

VI. Analysis of Results

A. *Image Processing*

Fig. 11 below depicts the comparison of images for each class before and after undergoing image processing enhancement. Visually by inspecting Fig. 11 below it can be observed that the original images' features were not very distinct. By undergoing the 3 different image enhancement techniques, the features i.e., the edges and contrast of the image improved significantly post-processing.



(a) 0_normal

(b) 1_ulcerative_colitis

(c) 2_polyps

(d) 3_esophagitis

Fig. 11 Comparison of Image Before and After Undergoing Image Processing Enhancement for each class.

B. 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 and 2 adhered to the following naming convention. The general format of tagging the models is defined in (1):

$$\boldsymbol{\#\#\_bb\_cc\_dd\_ee} \qquad (1)$$

The representations of each of the notations are detailed in Table 2 below:

Table 2 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.	- 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.	- 1 – 12. - 1 – 10 represent Stage 1 models. - 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.	- 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.	- 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	- TA –Adam optimizer - LR – Dynamic Learning Rate - TA_LR – Adam optimizer + Dynamic Learning.

C. 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 analysed and compared the results obtained from Stage 2 solution development.. 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. Based on analysis of the performances of the models in Phase 1 to 4 the final best generalized CNN solution was selected.

1) Phase 1 Results Analysis

The development of models in Stage 1 was broken down in a total of 10 groups of architectures. Each group is highlighted and labelled with a different colour scheme on Fig. 12. Within each group, a total of 7 sub-groups were developed. The best for each group is highlighted with blue coloured star in Fig. 12. A simplified table of the best 10 from each group is depicted in Table 3 below.

Table 3 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_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

Fig. 13 below 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. There is not much fluctuation between the 3 results metric indicating the reliability of the ValTestAvg results. By analysing Fig. 13, 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. Generally, transfer learning feature extraction architectures performed the worst with exception of model F_6_7_C_TALR.

Fig. 14 below depicts the F1 Score for validation and test sets of the best 10 models. 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. 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.

Fig. 21 below 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.

2) Phase 2 Results Analysis

Fig. 15 below, 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. Comparing between model groups 11 against 11A, it can be observed that increasing the epochs from 50 to 100 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. The same pattern of results is also observed in terms of F1 score.

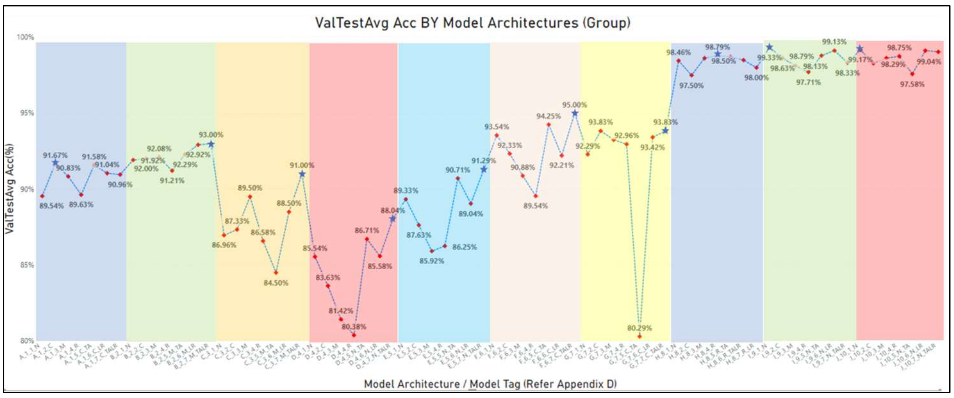


Fig. 12 Phase 1 Results

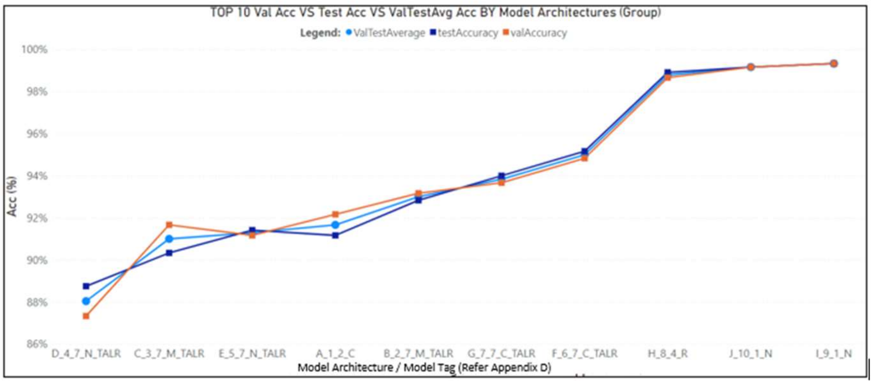


Fig. 13 Comparison of Test Acc, Val Acc and ValTestAvg Acc for the Best 10 Models in Stage 1.

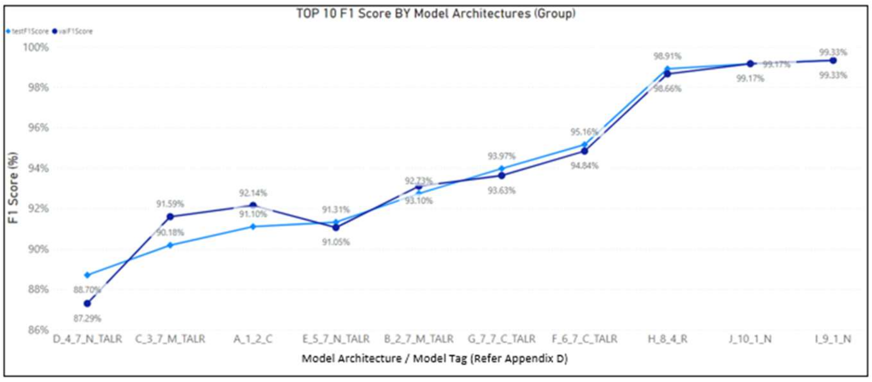


Fig. 14 Test F1-Score of the Best 10 Models from Stage 1.

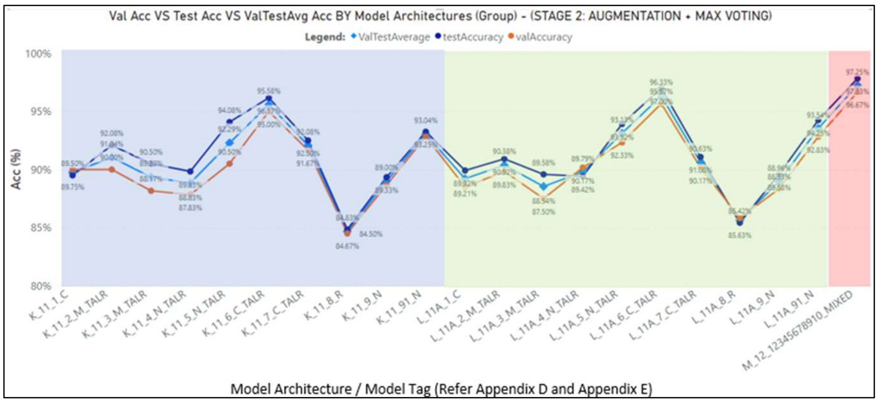


Fig. 15 Phase 2 Results

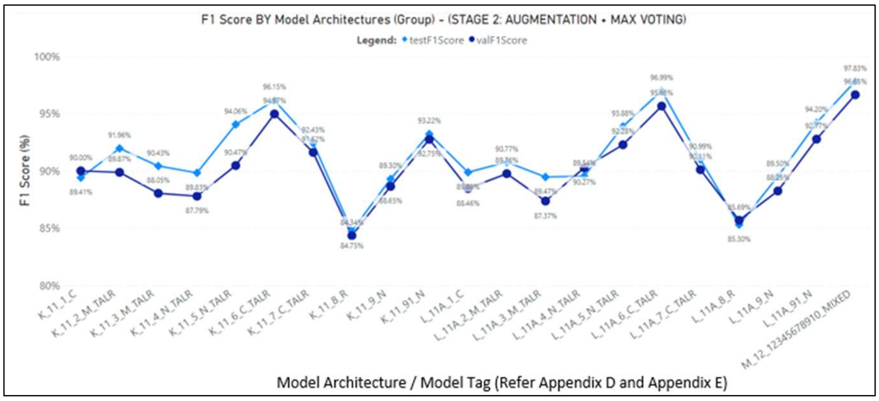


Fig. 16 Test F1-Score of the Best 10 Models from Stage 2.

3) Phase 3 Results Analysis

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. Fig. 17 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. Fig. 18 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) Phase 4 Results Analysis

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.

Fig. 19 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. Fig. 20 which depicts the corresponding F1 score comparison also shows the same trend of results.

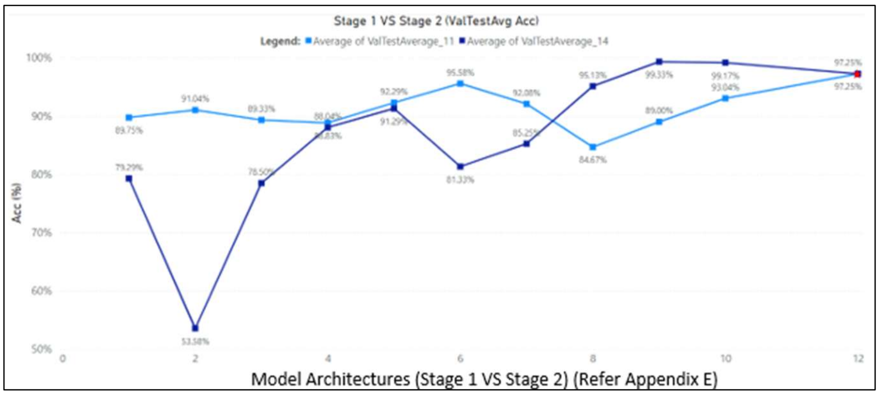


Fig. 17 Phase 3 Results.

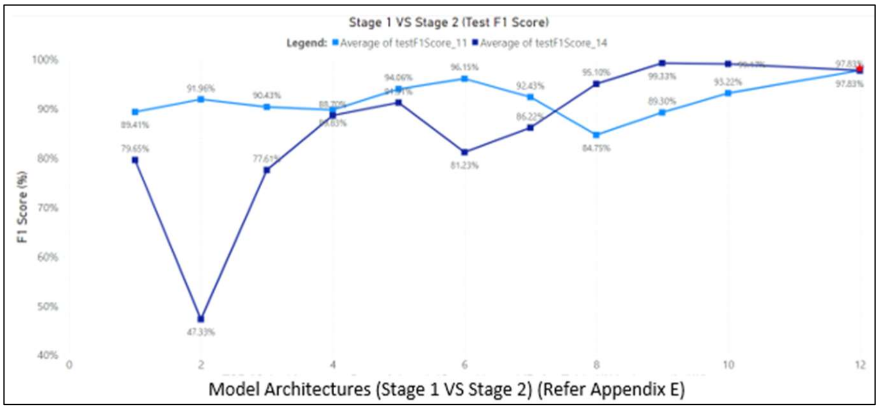


Fig. 18 Comparison of Test F1-Score for Models in Stage 1 against Stage 2 Models and Ensemble Model in Phase 3.

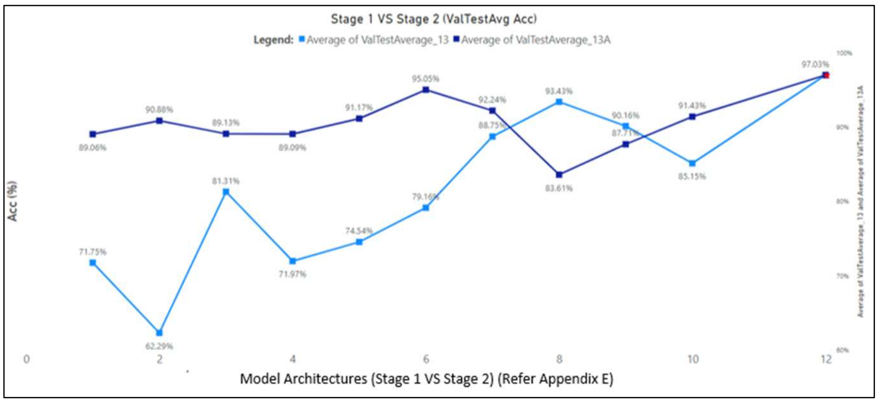


Fig. 19 Phase 4 Results

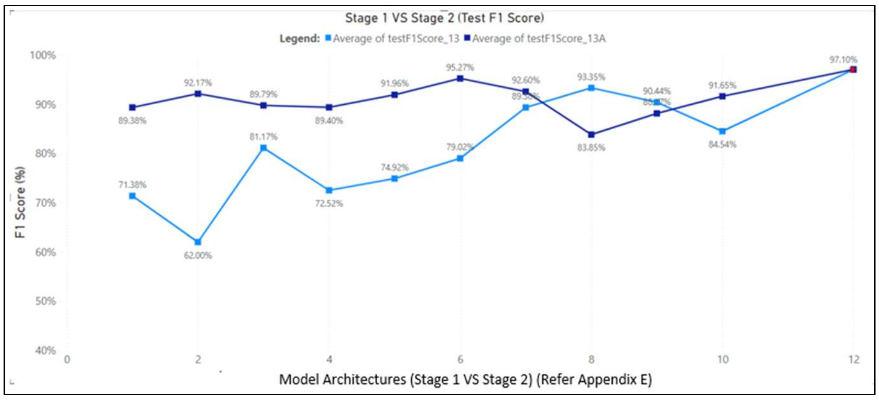
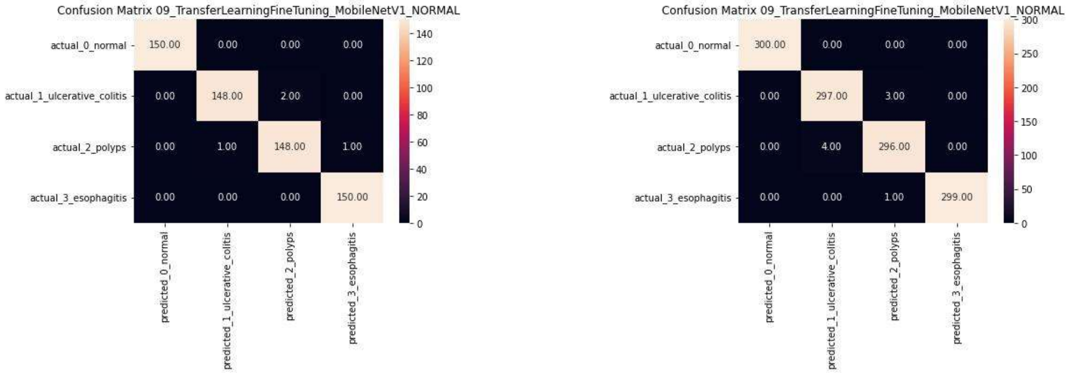


Fig. 20 Comparison of Test F1-Score for Models in Stage 1 against Stage 2 Models and Ensemble Model in Phase 4

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.

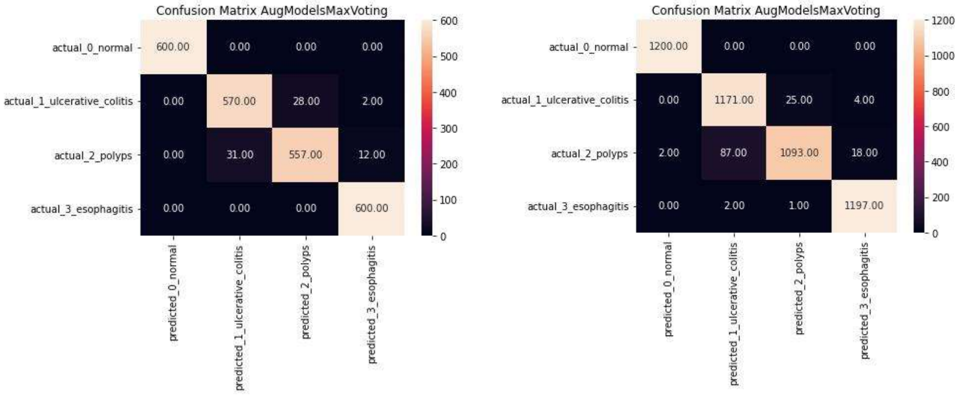
Fig. 22 below shows the confusion matrix of the ensemble model from Phase 4 testing i.e., model M_12_12345678910_MIXED when tested on the Validation and Test set of datasets Sets 1, 2, 3 and 4 combined.



(a) Validation

(b) Test

Fig. 21 Confusion Matrix of the Best Model from Stage 1 (model I_9_1_N)



(a) Validation

(b) Test

Fig. 22 Confusion Matrix of the Ensemble Model from Phase 4 Analysis (Model M_12_12345678910_MIXED)

VII. Discussion

A. *Image Processing Enhancement as a Data Pre-Processing Step*

The three image processing enhancement techniques proposed in this study focused on enhancing simple texture features in the image but also enhanced other aspects such as colour, contrast, and illumination. These enhancement techniques apply complex and dynamic image processing algorithms that considers various image parameters extensively. The methodology proposed in this study 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 creating a more comprehensive solution development by segregating solution development into 2 separate stages of Stage 1 and 2. Across the board, image processing-based models performed with good results. Moreover, the image processing techniques proposed also served as the foundation for performing image data augmentation.

B. *Performing Data Augmentation*

A lot of reviewed literatures typically only implemented geometric based augmentation. The goal of image augmentation in this study was focused on creating a variety in the input data instead of just increasing the size of the dataset. This study employed data augmentation that was centred around image colour, texture, contrast, and illumination manipulation, expanding the spectrum of complexity and variety in image parameters for the models to learn from. [5] and [19] 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. From the results observed in Phase 3 and 4 in Section VI, 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.

C. Network Architectures

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 better to take the architecture and tune it by training it on new domain specific data. 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 whilst retaining 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. 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. The final layers are focused on extracting finer details in the image which require tuning. It is more robust in 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 [5], [21]. 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 leaning 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 more generalized. Moreover, the comparative results showed in Section VI 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.. It can be observed that ensemble model performed very well across the board, especially in Phase 4 analysis. The ensemble model democratizes the results by considering the decisions by many individual models. By doing so, the model decision that is made is more reassuring and it also creates a generalized based solution.

D. Evaluation Process

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.

E. 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. predictions.

F. 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 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 A majority of literatures implemented their solution using SGD with a static learning rate. 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. 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. A combination of these 2 parameters were also considered to observe if a synergistic effect of both parameters provided the model with better performance.

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.

VIII. Conclusion

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. The study had undertaken several major steps as defined below:

- Performed a detailed literature review of the subject domain and related works to better understand the subject domain.
- A raw, suitably sized, equally balanced, and reliable dataset was sourced. The raw data contained WCE images from 4 different classes obtained from different sources i.e., taken from different cameras.
- An end-to-end detailed methodology was defined to develop the CNN solution.
- 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.
- An additional augmentation dataset, Set 5 was created. This augmentation technique was centred around colour contrast and illumination augmentation as opposed to commonly used geometric augmentation.
- 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 and network architecture parameters.
- The results were collected, interpreted, compared, and critically analysed. 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.

IX. Future Recommendations

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. 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. 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

[1] F. Rustam *et al.*, "Wireless Capsule Endoscopy Bleeding Images Classification Using CNN Based Model," *IEEE Access*, vol. 9, pp. 33675–33688, 2021, doi: 10.1109/ACCESS.2021.3061592.

[2] R. L. Siegel *et al.*, "Colorectal cancer statistics, 2017," *CA. Cancer J. Clin.*, vol. 67, no. 3, pp. 177–193, 2017, doi: 10.3322/caac.21395.

[3] T. A. C. S. M. and E. Team, "About Colorectal Cancer; What Is Colorectal Cancer ?," *Am. Cancer Soc.*, pp. 1–15, 2020, [Online]. Available: https://www.cancer.org/cancer/colon-rectal-cancer/about/what-is-colorectal-cancer.html#references.

[4] M. Sharif, M. Attique Khan, M. Rashid, M. Yasmin, F. Afza, and U. J. Tanik, "Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images," *J. Exp. Theor. Artif. Intell.*, vol. 33, no. 4, pp. 577–599, 2021, doi: 10.1080/0952813X.2019.1572657.

[5] H. S. Pannu, S. Ahuja, N. Dang, S. Soni, and A. K. Malhi, "Deep learning based image classification for intestinal hemorrhage," *Multimed. Tools Appl.*, vol. 79, no. 29–30, pp. 21941–21966, 2020, doi: 10.1007/s11042-020-08905-7.

[6] X. Jia and M. Q. H. Meng, "A deep convolutional neural network for bleeding detection in Wireless Capsule Endoscopy images," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2016-Octob, pp. 639–642, 2016, doi: 10.1109/EMBC.2016.7590783.

[7] S. Seguí *et al.*, "Generic feature learning for wireless capsule endoscopy analysis," *Comput. Biol. Med.*, vol. 79, pp. 163–172, 2016, doi: 10.1016/j.compbiomed.2016.10.011.

[8] S. Segui *et al.*, "Categorization and segmentation of intestinal content frames for wireless capsule endoscopy," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1341–1352, 2012, doi: 10.1109/TITB.2012.2221472.

[9] S. Segui *et al.*, "Detection of wrinkle frames in endoluminal videos using betweenness centrality measures for images," *IEEE J. Biomed. Heal. Informatics*, vol. 18, no. 6, pp. 1831–1838, 2014, doi: 10.1109/JBHI.2014.2304179.

[10] Y. Shin and I. Balasingham, “Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 3277–3280, 2017, doi: 10.1109/EMBC.2017.8037556.

[11] J. Y. He, X. Wu, Y. G. Jiang, Q. Peng, and R. Jain, “Hookworm Detection in Wireless Capsule Endoscopy Images with Deep Learning,” *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2379–2392, 2018, doi: 10.1109/TIP.2018.2801119.

[12] J. Y. Lee, S. H. Choi, and J. W. Chung, “Automated classification of the tympanic membrane using a convolutional neural network,” *Appl. Sci.*, vol. 9, no. 9, 2019, doi: 10.3390/app9091827.

[13] I. I. Conference, “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,” no. July, 2017.

[14] J. Qu, N. Hiruta, K. Terai, H. Nosato, M. Murakawa, and H. Sakanashi, “Gastric Pathology Image Classification Using Stepwise Fine-Tuning for Deep Neural Networks,” *J. Healthc. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/8961781.

[15] M. Hmoud Al-Adhaileh *et al.*, “Deep Learning Algorithms for Detection and Classification of Gastrointestinal Diseases,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6170416.

[16] J. Bernal *et al.*, “Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results from the MICCAI 2015 Endoscopic Vision Challenge,” *IEEE Trans. Med. Imaging*, vol. 36, no. 6, pp. 1231–1249, 2017, doi: 10.1109/TMI.2017.2664042.

[17] P. J. Chen, M. C. Lin, M. J. Lai, J. C. Lin, H. H. S. Lu, and V. S. Tseng, “Accurate Classification of Diminutive Colorectal Polyps Using Computer-Aided Analysis,” *Gastroenterology*, vol. 154, no. 3, pp. 568–575, 2018, doi: 10.1053/j.gastro.2017.10.010.

[18] M. S. Ayyaz *et al.*, “Hybrid deep learning model for endoscopic lesion detection and classification using endoscopy videos,” *Diagnostics*, vol. 12, no. 1, 2022, doi: 10.3390/diagnostics12010043.

[19] H. Takiyama *et al.*, “Automatic anatomical classification of esophagogastroduodenoscopy images using deep convolutional neural networks,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–8, 2018, doi: 10.1038/s41598-018-25842-6.

[20] X. Liu, C. Wang, Y. Hu, Z. Zeng, J. Bai, and G. Liao, “Transfer Learning with Convolutional Neural Network for Early Gastric Cancer Classification on Magnifiying Narrow-Band Imaging Images,” *Proc. - Int. Conf. Image Process. ICIP*, pp. 1388–1392, 2018, doi: 10.1109/ICIP.2018.8451067.

[21] B. Liu, K. Yao, M. Huang, J. Zhang, Y. Li, and R. Li, “Gastric Pathology Image Recognition Based on Deep Residual Networks,” *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, pp. 408–412, 2018, doi: 10.1109/COMPSAC.2018.10267.

[22] A. Demir, F. Yilmaz, and O. Kose, “Early detection of skin cancer using deep learning architectures: Resnet-101 and inception-v3,” *TIPTEKNO 2019 - Tip Teknol. Kongresi*, vol. 2019-Janua, no. February 2020, pp. 2–6, 2019, doi: 10.1109/TIPTEKNO47231.2019.8972045.

[23] M. Saric, M. Russo, M. Stella, and M. Sikora, “CNN-based Method for Lung Cancer Detection in Whole Slide Histopathology Images,” *2019 4th Int. Conf. Smart Sustain. Technol. Split. 2019*, pp. 14–17, 2019, doi: 10.23919/SpliTech.2019.8783041.

[24] M. Pei, X. Wu, Y. Guo, and H. Fujita, “Small bowel motility assessment based on fully convolutional networks and long short-term memory,” *Knowledge-Based Syst.*, vol. 121, pp. 163–172, 2017, doi: 10.1016/j.knosys.2017.01.023.

[25] W. A. Abbasi and F. U. A. A. Minhas, “Issues in performance evaluation for host-pathogen protein interaction prediction,” *J. Bioinform. Comput. Biol.*, vol. 14, no. 3, pp. 1–17, 2016, doi: 10.1142/S0219720016500116.

[26] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[27] N. Salem, H. Malik, and A. Shams, “Medical image enhancement based on histogram algorithms,” *Procedia Comput. Sci.*, vol. 163, pp. 300–311, 2019, doi: 10.1016/j.procs.2019.12.112.

[28] T. Jintasuttisak and S. Intajag, “Color retinal image enhancement by Rayleigh contrast-limited adaptive histogram equalization,” *Int. Conf. Control. Autom. Syst.*, no. Iccas, pp. 692–697, 2014, doi: 10.1109/ICCAS.2014.6987868.

[29] F. J. MONTALBO, “WCE Curated Colon Disease Dataset Deep Learning,” *Kaggle*, 2022. https://www.kaggle.com/datasets/francismon/curated-colon-dataset-for-deep-learning (accessed Apr. 10, 2022).

[30] F. J. P. Montalbo, “Diagnosing gastrointestinal diseases from endoscopy images through a multi-fused CNN with auxiliary layers, alpha dropouts, and a fusion residual block,” *Biomed. Signal Process. Control*, vol. 76, p. 103683, 2022, doi: https://doi.org/10.1016/j.bspc.2022.103683.

[31] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, “Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer.,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 9, no. 2, pp. 283–293, Mar. 2014, doi: 10.1007/s11548-013-0926-3.