Title: Dual Convolutional Neural Networks of Ensemble learning with Attention Mechanism for Classification Task using APTOS Diabetic Retinopathy Dataset

Abstract:

Diabetic Retinopathy (DR) which is a severe complication of diabetes, significantly impacts vision, potentially leading to blindness. The effective diagnosis of DR is crucial, and Convolutional Neural Networks(CNNs) have emerged as a prominent tool for this purpose. This paper introduces a specialized CNN model, Attention Inception ResNet with Straightforward Convolution, specifically designed for binary classification in DR. The model demonstrates robust diagnostic capabilities, achieving remarkable results on DR datasets with 97.69% accuracy, 0.953 loss, 97.77% F1-Score, 97.02% sensitivity, 98.18% specificity, 97.77% precision and recall and an AUC of 98%. This underlines the model's potential as an effective supplementary tool for DR images analysis.

Keywords: Deep Learning, ResNet, InceptionNet, Attention Mechanism, Diabetic Retinopathy Diagnosis

1. Introduction

Diabetes mellitus, often simply known as diabetes, is a set of metabolic disorders characterized by high blood sugar levels over a prolonged period. This occurs when the body is unable to produce enough insulin to regulate these elevated sugar levels [1]. Diabetes can lead to various health complications including diabetic retinopathy(DR), stroke, kidney failure, heart attacks. Diabetic retinopathy specifically occurs when high blood sugar levels damage the blood vessels in the retina leading to swelling and leakage [1].

For example, background retinopathy, the initial stage of DR, does not immediately affect vision but starts to damage the blood vessels in the eye [2]. These vessels can exhibit minor swelling, leak fluid and proteins, and bleed. As DR progresses to proliferative retinopathy, it extensively damages the retina, significantly increasing the risk of vision loss and potential blindness due to inadequate blood supply to the retina [3].

Given these severe and life-threatening impacts, early detection and accurate diagnosis of DR are crucial. However, diagnosis often requires a large number of personnel and experienced doctors, making it somewhat subjective and prone to misdiagnosis [2]. This also leads to a

waste of human resources. In areas with inadequate medical facilities or resources, diagnosis and treatment might be inaccessible. Hence, a large number of the deep learning based diagnosis systems were introduced, for instance, using such systems to analyze pixels and identify lesions reduces the risk of misdiagnosis [2]. Furthermore, it enhances diagnostic efficiency and accuracy, and lowers the cost of medical resources utilization, making treatment mode accessible and alleviating the effects of the disease.

In order to achieve the accurate diagnosis of diabetic retinopathy, this coursework had design a ensemble convolutional neuron network inspired by the structure of Inception and Residual network. This network distinguishes two level of the diabetic retinopathy which are normal retina and DR retina. The network was built with the idea of Inception-ResNet, attention mechanism, and straightforward convolutional blocks with several techniques to prevent overfitting, gradient vanish etc. Additionally, at the end of the model, Grad-CAM(Gradient-weighted Class Activation Mapping) was added to prompt the interpretability and visual analysis.

The rest of the article is structured in following way. Section 2 introduces a the brief contents of the relevant previous work within this research field. Section 3 provides the detail of the datasets that this work had been used, the specific methods utilized to process the data, the complete proposed method of this work, evaluation standards taken into account for model evaluation, and the hardware and software environment this work had been operating on. Section 4 provides all steps taken to achieve the model construction, the result of each experiment within the step, detailed analysis of the experiment and a range of comparison are included as well after introducing the experiment. Section 5 draws the conclusion of the whole coursework and presents the limitation and possible future work.

2. Related work

A numbers of works had been done in field of diagnosing the diabetic retina through various methods. The following part will introduce the relevant works within this area of research.

Nahiduzzaman et al. have developed a method for diagnosing diabetic retinopathy using a hybrid CNN-SVD model and the ELM algorithm, achieving high accuracy of 99.73% in binary class and 98.09% in multiclass classification on the APTOS-2019 dataset and 96.26% on the Messidor-2 dataset, noted for its efficiency and reduced computational complexity [1].

In the study of Saedd et al.[2], a two staged-fine-tuned CNN was employed to diagnose DR. Their method first embeds DR lesion structure in pre-trained CNN using lesiong Regions of interest(ROIs) and then adapt high-level layers for better discrimination of retinal fundus images. Though this approach, they got 99.72% and 99.54% accuracy on both Messidor and EyePACS datasets for binary classification. Though the work is limited to binary classification which can only accurately distinguish disease and non-disease situation instead of clarifying the level of severity.

Zang et al.'s DcarNet for multi-level DR classification achieved 95.7%, 85% and 71% accurac on OCT and OCTA data[3]. AL-Antary and Arafa's MSA-Net showed 87.5% accuracy, 90.5% sensitivity, and 78.7% specificity and F1-Score of 76.7% on APTOS dataset [4]. Khan et al.'s VGG-NiN model reached Micro-AUC and Macro-AUC of 0.95 and 0.84 on retinal images [5]. As for Mustafa et al., a multi-stream CNN with a boosting framework which inspired by pre-trained model structures had operated on Messidor-2 and EyePACS datasets which then achieved the experiment results of 95.58% accuracy [6]. Ali et al. had proposed a novel CNN model using ResNet50 and InceptionV3 for feature extraction and classification of DR which achieved an accuracy of 96.85%, sensitivity of 99.28%, specificity of 98.92%, precision of 96.46%, and an F1 score of 98.65% on MESSIDOR and IDRiD datasets [7].

Raiaan et al.'s model uses ResNet-10, a shallow CNN model for classification mission of DR and demonstrated an accuracy of 98.65% on combined datasets including APTOS, Messidors2, and IDRID [8]. Chen et al., had developed an approach for DR detection on integrated shallow CNNs, in which their model uses mutli-scale shallow CNNs to classify retinal images which resulted in accuracy around 80% and loss of 20% [9].

Farge et al.'s model using DenseNet169 and CBAM reached 97% accuracy, 97% sensitivity, 98.3% specificity, 0.9455 QWK for binary classification, and 82% accuracy, 0.888 QWK for severity grading [10].

And lastly, He et al., focused on addressing imbalanced data distribution in DR grading using CABNet which achieved 93.1% accuracy and 96.6% AUC, 92.9% Precision, 90.2% Recall and F1-Score of 91.5% on Messidor dataset [11].

These previous works in the field of DR diagnosis demonstrates a range of innovative approaches, with most achieving impressive results and performance metrics. However, there are limitations as well, such as a targeted focus on binary classification over multi-class

classification, and some models exhibiting lower performance indicators, This highlights certain gaps and areas for improvement in the overall efficiency of these model.

3. Material and Method

3.1 Dataset

3.1.1 Dataset Introduction

This coursework employed a singular dataset comprising images of both diabetic retinopathy and normal retinal scans, aimed at identifying diabetics through retinal symptoms.

The dataset encompasses a total of 3662 images, each representing either a normal retina or various stages of diabetic retinopathy. All images are formatted in a 224x224 pixel RGB format.

The dataset categorizes these images into five distinct classes, with one class representing normal retinas devoid of any symptoms, and the other four classes indicating different severity levels of diabetic retinopathy, where those four folders are Mild, Moderate, Proliferate and Severe. Images of diabetic reinopathy are displayed below in figure 1.

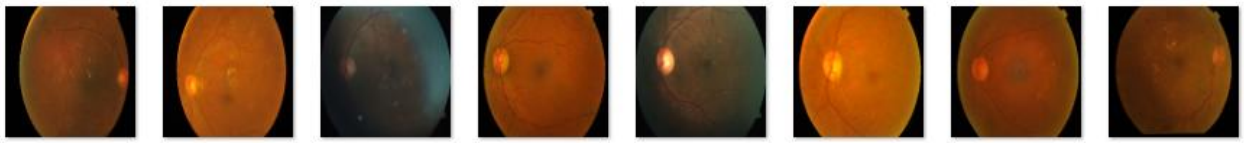

Figure 1: Diabetic Retinopathy Images

The datasets link is provided as follow: https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-2019-data

And this dataset is originally from APTOS 2019 Blindness Detection. Link will be provided below: https://www.kaggle.com/c/aptos2019-blindness-detection/data.

3.1.2 Dataset Pre-Split

Upon examining the dataset, it becomes evident that there is a noticeable similarity among images within the four severity datasets of diabetic retinopathy.

Following are images within the four folders:

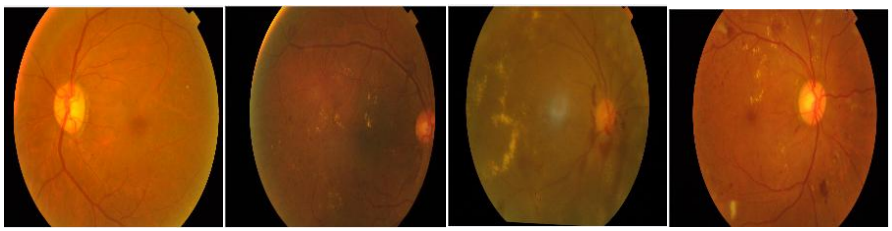

Figure 2: Images of Mild, Moderate, Proliferate and Severe.

Additionally, a significant imbalance was observed across the five categories, with the four representing varying degrees of severity having considerably fewer images compared to the folder for normal retinas. The exact number of each class are 370 images in Mild, 999 images in Moderate, 295 images in Proliferate, 193 images in Severe and 1805 images in Normal. There are ways of augmenting the data for example such as, Oversampling, Random Cropping, Noise Addition or even generate similar images based on generative models. But since the number of images in normal is way larger than the others, additionally, using augmentation to get augmented images from such a small number of them will most certainly results in low quality of images, repetition and low performance of model. This coursework is also resources limited since it is operating on single computer rather than across a large server which raises the costs as well, and therefore, utilizing generative networks method to generate similar images is not feasible.

However, by consolidating the four severity-related folders into a single category representing diabetic retinopathy, the disparity in image count between the diabetic and normal categories is reduced to a more manageable 40 images. Consequently, this project has opted to proceed with a binary classification approach for disease diagnosis and model training.

3.1.3 Train and Test Set Separation

After rearranging the dataset, a reasonable ratio for splitting data should be applied, and therefore, a small test was conducted using a Pre-trained InceptionV3 model to operate on two different split ratio in order to figure out an appropriate one. The results are showed in table 1 and figure 3.

Ratio	Acc	Loss	F1	Val_Acc	Val_Loss	Val_F1
6:2:2	0.99	0.03	0.99	0.98	0.07	0.98
7:1.5:1.5	0.99	0.04	0.99	0.98	0.07	0.98

Table 1: Ratio Comparison

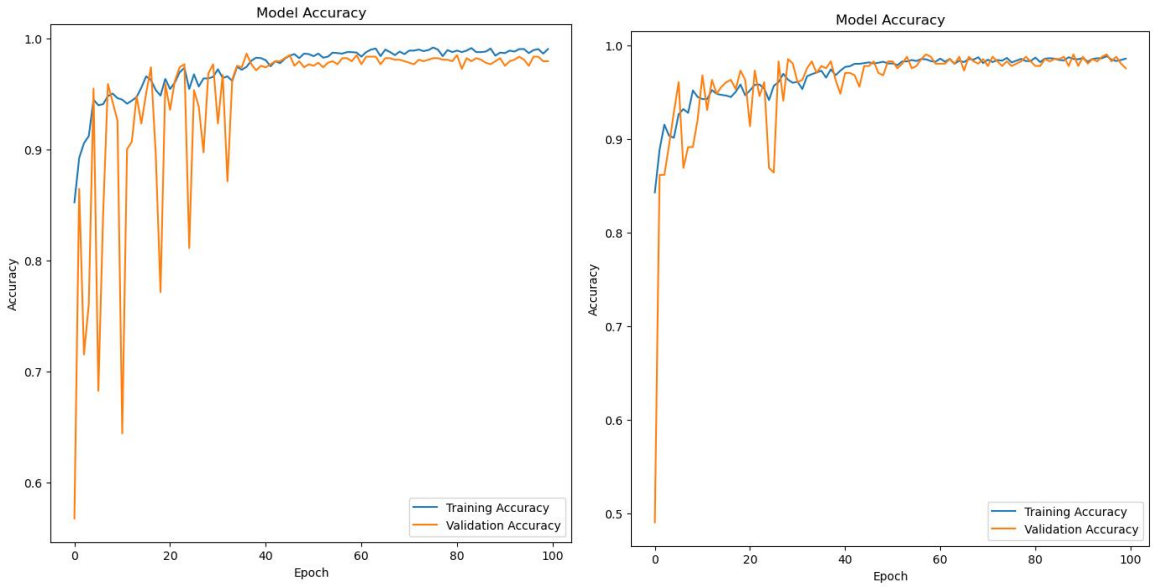

Figure 3: Performance of Ratio 6:2:2 and Ratio 7:1.5:1.5

According to the table 1 and figure 3, though the values showed resemblance, however the stability of second Ratio is far better than the first one. And thus, datasets were separated into a ratio of 85% for training and 15% for testing folder, the validation images will be separated from training dataset while building the model. Overall, the split ratio is around 7 : 1.5 : 1.5. The separation was done through the library of Python called shutil and random, which could complete the operation of files, for instance, randomly selecting a specific proportion of images in the targeted folder and move to another folder to achieve data separation.

3.1.4 Data Preprocessing

In this coursework, an initial screening of two image preprocessing techniques was conducted to address the impact of color and shape in retinal images on model training. Transformed images were retained for further application in the model's operational framework, leading to the final selection of one effective image processing approach.

3.1.4.1 Grayscale Images

Grayscale imaging is a crucial technique in digital image processing, involves converting colour images to shades of gray, effectively reducing the image to single intensity channel. This method emphasize variations in brightness over colour difference, making it particularly useful in various applications, especially in fields like medical imaging. In grayscale images, each pixel represents a shade of gray, ranging from black to white, with numerous shades of gray in between. By focusing on luminance rather than chromatic content, grayscale imaging is often used to extract specific characteristics from images such as shapes, texture and shapes [12], which are vital in many analytical and diagnostic applications.

As previously mentioned, there is a significant deceptive similarity across the dataset; for instance, the colors of normal and diabetic retinas display a high degree of resemblance. To prevent the model from focusing on these color similarities, which might result in poor differentiation between the two classes, and therefore, images were transformed into grayscale. The transition to grayscale was efficiently achieved using OpenCV.

Following figure 4 presents the results.

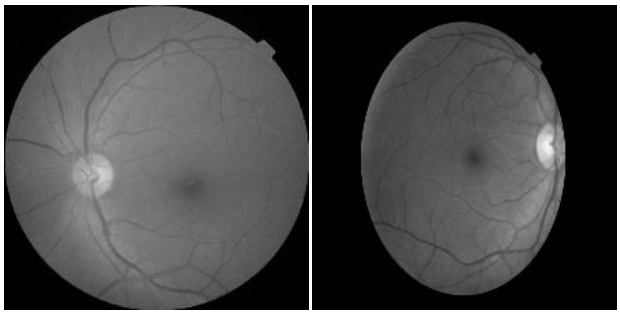

Figure 4: Grayscale Images

3.1.4.2 CLAHE Transition

CLAHE, or Contrast Limited Adaptive Histogram Equalization, is a method known for enhancing image contrast and sharpness. It operates by adaptively dividing the image into several small blocks, then performing histogram equalization on each of these blocks within certain contrast limits [13]. This approach improves the histogram distribution of the image, which is essential for enhancing detail and image quality.

Considering the fact that images of dataset were also having trouble on contrast, for instance, in figure 5, images are evidently uneven in light and shade, details are unclear. Therefore, by applying CLAHE method to adjust the image and save in another folder for training might solve the issue.

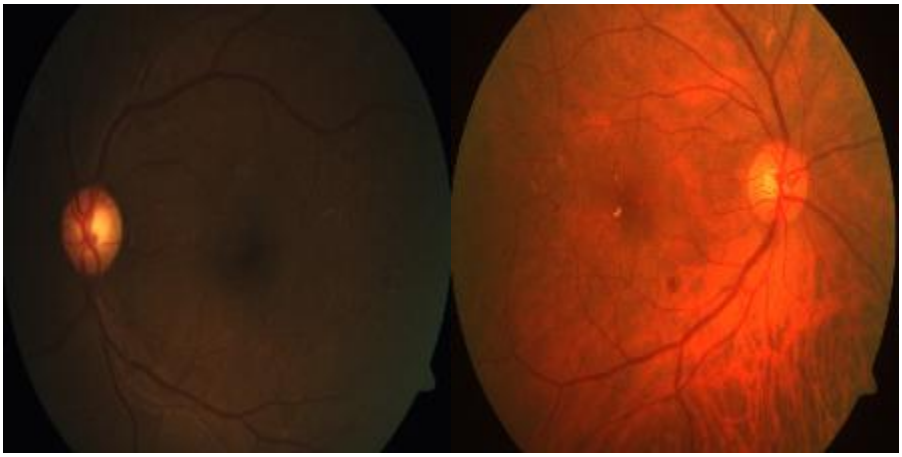

Figure 5: Uneven contrast and shade.

To achieve CLAHE, Opencv was utilized to implement CLAHE for image processing. To start with, images were converted to grayscale to reduce complexity by changing three channels of RGB to one, thus increased the efficiency. During the CLAHE object initialization, the contrast enhancement parameter ‘clipLimit’ was established at 9, ensuring it remained at a reasonable and manageable level.

Furthermore, parameter ‘titleGridSize’ was set as 10 * 10 which then the image will be split into a 10*10 grid. This division into smaller blocks allows each block to undergo independent histogram equalization, enabling CLAHE algorithm to enhance contrast locally within different areas of the image. The size of detail in the final image enhancement, with smaller blocks potentially leading to more precise contrast adjustments. Figure 6 shows the transformed images. By observing the results, it is clear that the lines of nerve were outlined, and therefore, this preprecessing was adopted for further model training which will be mentioned in section 4.

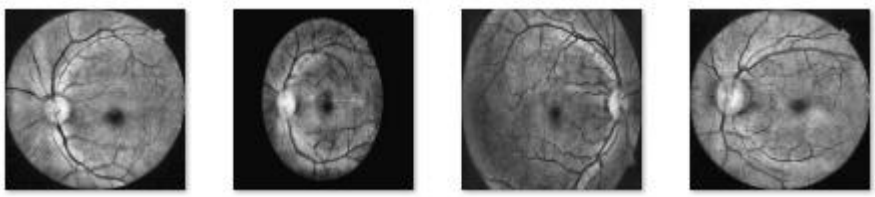

Figure 6: CLAHE images transformed

3.1.4.3 Basic data augmentation using ImageDataGenerator

Following the application of CLAHE, grayscale, edge detection and HSV to images, this phase involved further processing through techniques like shearing, rotation, width shifting, zooming, horizontal flipping and resizing etc. This approach generated varied versions of images within the same category, enhancing the diversity of dataset, aiding generalization of model while help preventing overfitting. Figure 7 showcases a comparison of processed samples.

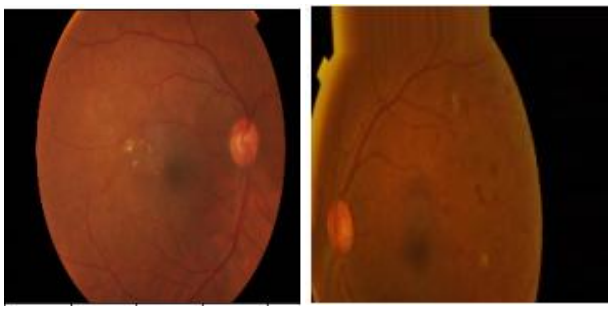

Figure 7: Before and After the transition

3.1.4.4 Partial Summary for data processing

In the initial stage of the model building, only basic augmentation and data separation were applied to build and test model from individual model to combination of models. Once the model was initially built, CLAHE and grayscale images were put to refine the model while eliminate the possibilities of potential negative impact of original images.

3.2 Proposed Model

The proposed model is an ensemble of a straightforward convolutional model and an Inception-ResNet model, each incorporating attention mechanisms. At the end of the model, Grad-CAM is also utilized for visualization purposes in this setup. Figure 9 and figure 10 illustrate the overview and detailed structure of the model, other internal structure such as Inception-Block, attention mechanisms are displayed in the ensemble stage in Experiment and results.

The first individual model is a straightforward convolutional neural network, design to effectively extract features from input data and emphasize important aspects through an integrated attention mechanism. Its architecture consists of three convolutional layers, each paired with a custom attention module, creating a balance between feature detection and focused processing.

At its core part, the model employs convolutional layers to identify and interpret diverse patterns and textures from the input. These layers progressively capture more abstract features as the data flows through the network. The attention mechanism is a key aspect of this model, follows each convolutional layer. It generates attention maps that highlight significant regions of the feature maps, directing the model's focus to the most informative parts of the input. This selective attention not only improves the precision of the model in recognizing relevant features but also enhances the overall interpretability of the processing.

The combination of attention in each layer instead of at the end of the model, allows it to not just perform standard feature extraction but also to adaptively prioritize parts of the data that are crucial for the subsequent analysis and decision making.

The second model is structured as an Inception-ResNet model, which combines Residual block and Inception block, each with added attention mechanisms. It is designed to handle complex input data effectively.

Within the second model, ResNet helps avoid common problems in deep networks such as gradient vanishing by using shortcuts to keep information flowing. Alongside, Inception blocks use several parallel branches to catch different features from the input data. This setup lets the model see and understand a wide range of patterns.

After going through these block, an attention mechanism comes at play. This part makes sure the model pays enough attention to the valuable parts of the input images. In the looping structure of the model, after each ResNet and Inception blcok, there's a step where model adds up the absorbed feature values. This help in keeping the essential information from each part of the network. The number of filters used in each ResNet block changes based on a set plan, making the model flexible in how it process features.

When all calculations and processes are finished inside these two models. These outputs which respectively represents different perspectives of the input data with model 1 offering a straightforward interpretation and model 2 providing a deeper, detailed analysis. Furthermore, the featured maps are merged using concatenation method. After that, it is passed to a dropout layer for regularization in order to reduce overfitting in complex model like this. The classification layer used softmax activation, ensuring the outputs are interpreted as probabilities, providing clear and actionable results.

After the training process, Grad-CAM revels the specific areas the model focuses on within the images. This method address the “black-box” issue commonly associated with neural networks by making decision-making process more transparent. Grad-CAM serves as a tool to enhance the transparency of CNN-Based models. It operates by highlighting the specific input regions crucial for the model’s predictions, providing insights into the significance of individual neurons in relation to a particular area of interest [14].

Overall, the ensemble model brings together the strength of both individual models, enhanced by dropout regularization. This approach allows for a comprehensive and reliable analysis of input data, leading to accurate predictions and classifications.

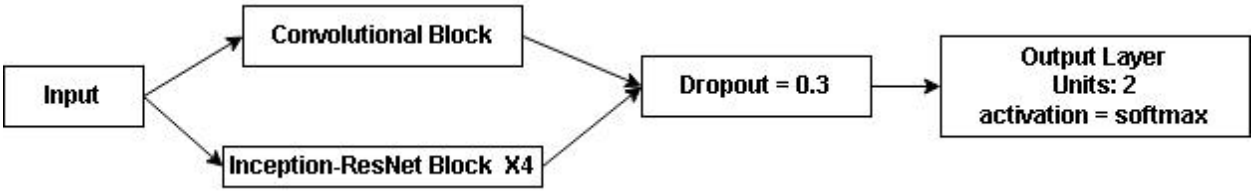

Figure 9: Proposed Model Structure overview.

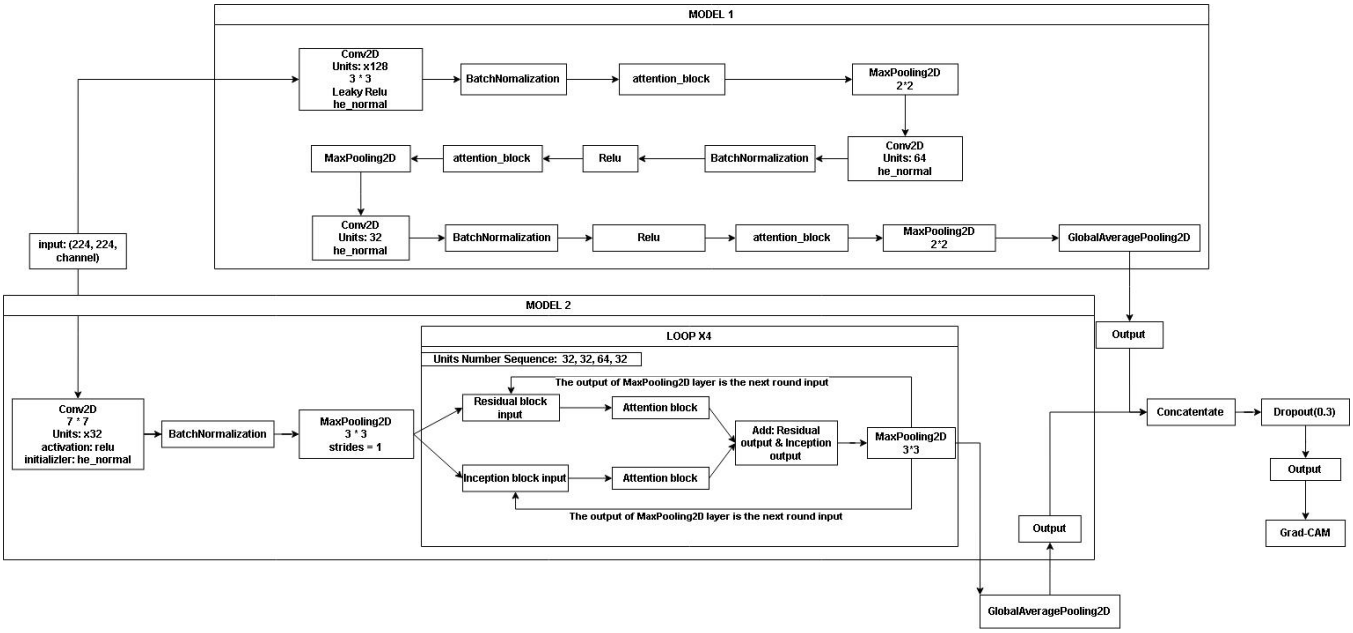

Figure 10: Detailed structure of model

During the compilation of the model, this coursework had also manually define a custom crossentropy loss function and custom accuracy metrics calculation based on the existing algorithm and equation which is expressed in (1) to (4) within the Evaluation Strategy.

3.3 Evaluation Strategy

It is necessary to evaluate models in during model training, and it is of importance that utilizing correct values and metrics to achieve this. And thus, following presents the evaluation metrics.

(1) Accuracy: Accuracy is a quite intuitive metric and a criteria for evaluating model performances, which displays the overall correctness of a classification model. It calculates the ratio of the correctly predictions among the total predictions, where T, P, N, F represents, true, positive, negative and false.

$$y_{pred\,round} = round(y_{pred}) \tag{1}$$

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} 1(y_{true}^{(i)} = y_{pred\_rounded}^{(i)}) \tag{2}$$

As for the custom defined equation, it calculates the proportion of correct predictions for the given predicted values ‘y_true’ and true values ‘y_true’. First it rounds the predictions to either 0 or 1 using round function. Then it compares these values. The accuracy is the average number of times the rounded predictions match true values. Where N represents the total number of samples, and the indicator function outputs 1 when the prediction equals the true value and 0 otherwise.

(2) Loss: The loss metric evaluates the gap between the predicted labels and the actual labels in a machine learning model. It serves as a gauge of the model's accuracy, with a lower loss indicating closer alignment between predictions and reality. This criterion is essential for refining and improving the model's performance.

$$y_{pred} = clip(y_{pred}, \in, 1 - \in) \tag{3}$$

$$\text{Crossentropy\_Loss\_Function} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{true,\,c}^{(i)} \log(y_{pred,\,c}^{(i)}) \tag{4}$$

In this case of DR classification, Crossentropy Loss is defined based on the idea given by [15]. It is a function based on the measuring probability distribution and actual distribution. The closer

the trained distribution is to the actual distribution, as indicated by a smaller discrepancy, the more accurately the model predicts and classifies [15].

The calculation of crossentropy requires to clip the function where $\in$ represents epsilon, a extreme small positive number for example 10^-9 or 10^-7 to prevent the potential issue during the calculation. The clip method ensures that the value of the prediction value is in the range of 0 to 1 which guaranteed the stability of the calculation. For crossentropy formula, where N is the total sample number, C is the category number, in this case, it is two. And $y_{\text{true, c}}^{(i)}$, $y_{\text{pred, c}}^{(i)}$ are respectively the true value and predicted value of sample i for class C, represents the probability.

(3) Precision: This metric measures the ratio of true positive cases to the total number of cases classified as positive by the model, serving as a key indicator of its accuracy in correctly identifying positive instances. In this case, this coursework are using Micro-average precision to calculate the value as two categories are quite balanced.

$$\text{Micro} - \text{average Precision} = \frac{\sum \text{True Positives(TP)}}{\sum \text{True Positives(TP)} + \sum \text{False Positives(FP)}} \quad (5)$$

(4) F1-Score: F1-Score evaluates a model's effectiveness by computing the harmonic mean between Precision and Recall, thus providing balanced measure of both false positive and false negative.

$$\text{F1 Score} = \frac{2 \times \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

(5) Confusion Matrix: This matrix displays the accurate number of True Positive, True Negative, False Positive and False Negative.

(6) ROC: The Receiver Operating Characteristic curve, illustrates the balance between a model's sensitivity (True Positive Rate(TPR) or Recall) and its False Positive Rate(FPR) across various threshold settings, effectively mapping the trade-offs between correctly identifying positive instances and avoiding false positives.

$$\text{Micro} - \text{average FPR} = \frac{\text{Micro} - \text{Average FP}}{\text{Micro} - \text{average FP} + \text{Micro} - \text{average TN}} \quad (7)$$

$$\text{Micro} - \text{average TPR} = \frac{\text{Micro} - \text{Average TP}}{\text{Micro} - \text{average TP} + \text{Micro} - \text{average FN}} \quad (8)$$

(7) ROC Curve: The Receiver Operating Characteristic curve is a graphical representation that illustrates a classification model's diagnostic ability by plotting the True Positive Rate (Sensitivity) against the False Positive Rate at various threshold settings. It helps in evaluating the trade-offs between sensitivity and specificity in the model. After calculating TPR and FPR at various thresholds, plot them on a 2D graph with FPR on the X-axis and TPR on the Y-axis to create the ROC curve.

(8) AUC: AUC(Area Under Curve) represents the probability that a randomly chosen positive instance is ranked higher than a negative one by the model, serving as a concise measure of its discrimination ability. A higher AUC, approaching 1, signifies better model performance.

(9) Recall / Sensitivity: Recall gauges a model's capacity to correctly identify actual positive cases, offering a clear view of its proficiency in detecting positives and reducing the instances of false negatives. In this context, Recall is evaluated using the Micro-average method.

$$\text{Micro} - \text{average Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}} \quad (9)$$

(10) Specificity: In contrast to Recall, measures the model's ability to accurately identify true negatives, thus demonstrating its skill in avoiding false positives and ensuring that negative cases are not mistakenly classified as positive.

$$\text{Micro} - \text{average Specificity} = \frac{\sum \text{TP}}{\sum \text{TN} + \sum \text{FN}} \quad (10)$$

(11) Precision-Recall Curve: A graphical representation showing the trade-off between Precision and Recall for a classification model at different thresholds, particularly valuable in assessing performance on imbalanced datasets.

(12) Classification Report: This report provides a detailed summary of a model's performance, including essential metrics like Precision, Recall, F1-Score, and Specificity for each class, offering a clear insight into the model's classification accuracy for different categories.

3.4. Environment Execution

The execution of the environment is provided in table 2 with the details of hardware and software.

Software	Framework	Tensorflow 2.9.0
	Language	Python
	Libraries and Application	Numerical Python, Keras, Matplotlib, Scikit-Learn, Tensorflow-Addons 0.18.0/0.19.0, Opencv, shutil
Hardware	Central processing unit(CPU)	Intel(R) Core(TM) i7-8750H CPU @ 2.2PGHz(12 CPUs), ~2.2GHz
	Graphic Processing Unit(GPU)	NVIDIA GeForce GTX 2060

Table 2: Environment and Technology

4. Experimental Results

This section presents a comprehensive process of the coursework.

The experiment consisted of four main stages.

- Initially, the primary architecture of the model was established, incorporating elements from inception and residual architectures, along with a basic three-layer convolutional structure for the preliminary design.
- Subsequently, the distinct models, such as ResNet, Inception and Convolutional block were integrated. Following this integration, an attention mechanism was incorporated into the combined model.
- In the third phase, the combined model, which includes convolution blocks, resnet-inception blocks with attention mechanisms, and in-built loss functions and accuracy metrics, underwent testing with three different data preprocessing methods to determine the most effective approach.

- In the final phase, the model was fine-tuned using the best-suited data preprocessing method. This phase also included the integration of manually defined loss functions and calculations for accuracy metrics to optimize performance.

4.1. Performance Results using the dataset

4.1.1 Model construction and Experiment Pre-Condition

During the first stage, three basic model structures was constructed, establishing the initial architecture for later ensemble model. Then they are all trained on the datasets with the ratio of 70% : 15% : 15% of train, validation and test. For data augmentation, which is referred to as 'Method 1', the ImageDataGenerator was utilized with the following settings: rescaling each image by a factor of 1/255, applying random within 20 degrees, width and height shifts up to 20%, shear transformations up to 20%, zooming up to 20%, enabling horizontal flipping, additionally, 'nearest' was used as the fill mode for new pixels introduced by these tranformations. All the analytic description of the model construction and results will be in Discussion part.

4.1.2 First Individual Model

The first model features a straightforward convolutional neural network. The networks begins with an input layer, designed to handle the specified input shape of the images. It then progresses through three convolutional layers with increasing filter sizes starting from 128 units and gradually reducing to 96 and 64 units, each with a 3 * 3 kernel size. After each convolutional layer, batch normalization is applied, followed by a MaxPooling2D layer to reduce the spatial dimensions of the feature maps.

The convolutional layers are apparently used for feature extraction, use of "leaky_relu" activation for the first layer helps preventing neuron death, using "relu" for the rest of the layers instead of "leaky_relu" is that while preventing neuron deaths, it also increase the amount of calculations which this device may not able to bear after ensemble, and therefore it is crucial to keep the model simple and effective as well at early stage.

At the end of the model, GlobalAveragePooling2D is applied, thereby summarizing features and reducing parameters.

The structure is provided below in figure 11.

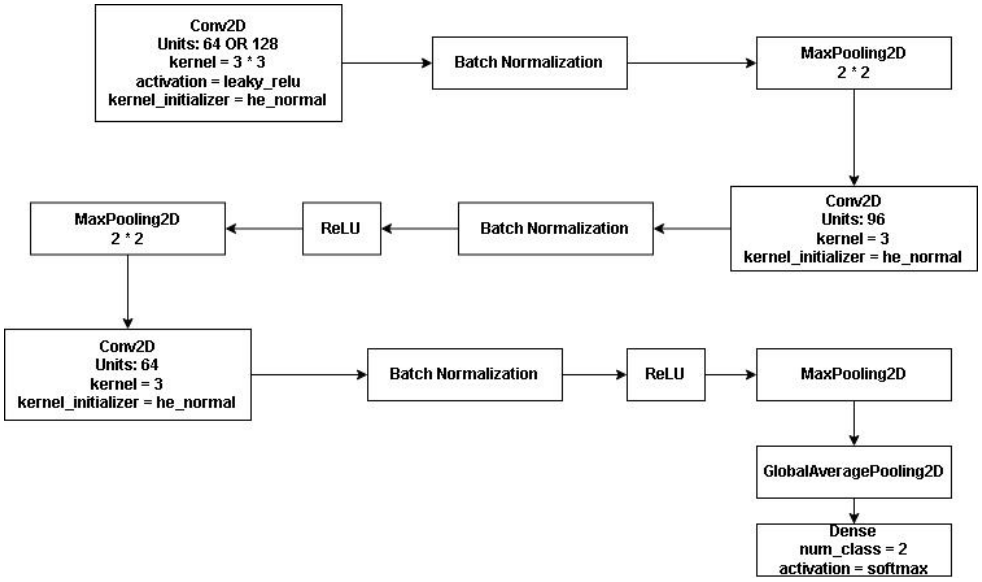

Figure 11: Convolutional Block

This model was tested on the dataset with the 16 batch size, input shape of 224 * 224 * 3, additionally with data augmentation method 1. Evaluation metrics are presented below from figure 12 to figure 16.

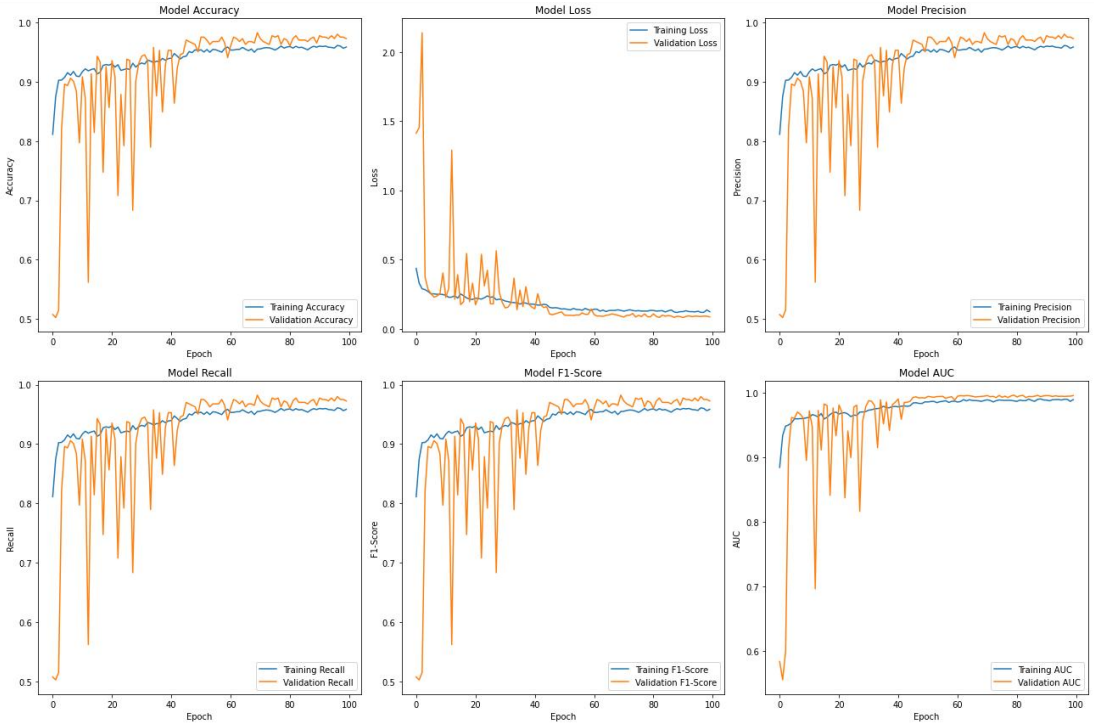

Figure 12: Simple Convolution Block Accuracy, Loss, Precision, Recall, F1-Score and AUC

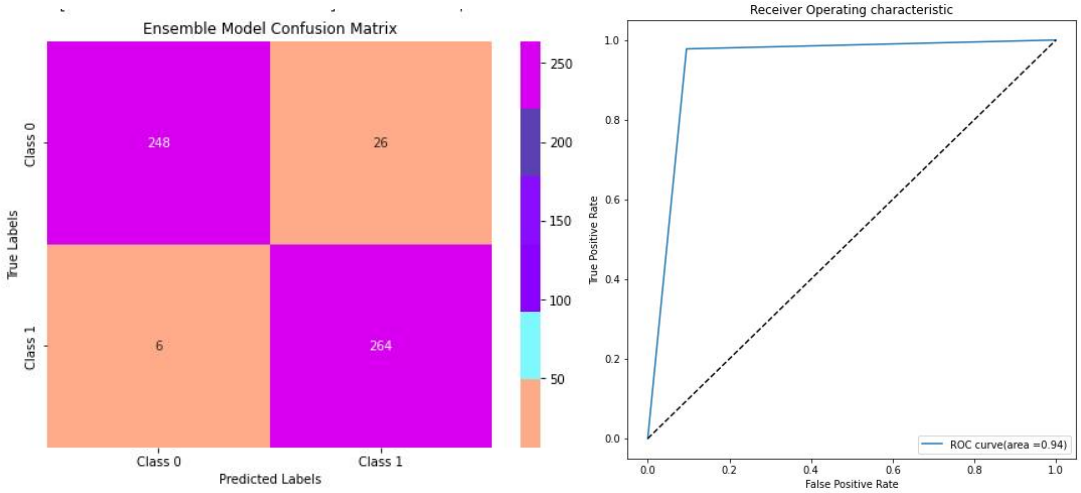

Figure 13: Confusion Matrix and ROC Curve with AUC value(Class 0 for diabetic, Class 1 for normal)

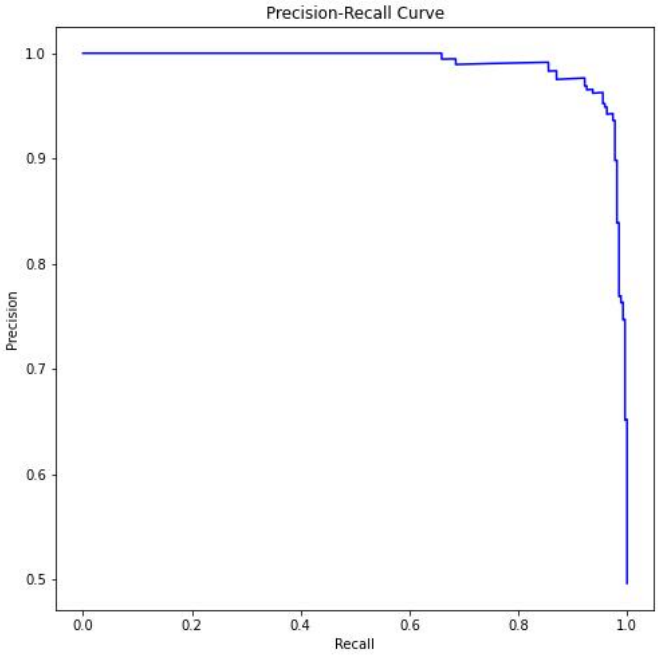

Figure 14: Precision-Recall Curve

```
sensitivity: 0.9777777777777777
specificity: 0.9051094890510949
```

Figure 15: Sensitivity

```
17/17 [==============================] - 0s 19ms/step
              precision    recall  f1-score   support

           0       0.98      0.91      0.94       274
           1       0.91      0.98      0.94       270

    accuracy                           0.94       544
   macro avg       0.94      0.94      0.94       544
weighted avg       0.94      0.94      0.94       544
```

Figure 16: Sensitivity, Specificity and Classification Report

4.1.3 Inception Model and Residual Model

These two models were built to test the compatibility of inception and residual structure on this specific dataset as well as providing insights to building the ensemble model with this certain structure.

The inception model described in figure 17 incorporates three distinct branches, each equipped with different neuron counts to capture various features from input images. The first branch uses smaller filters for fine details, the second has medium-sized filters for intermediate features, and the third employs the largest filters for broader features. The multi-branch architecture, enhanced by a filter factor, allows for flexible unit number adjustments, ensuring a balance between specificity and generality. This structure is deliberately designed to be simple and adaptable, primarily to assess its compatibility with the DR dataset, thus ensuring that the model is efficiently tailored for the ensemble model.

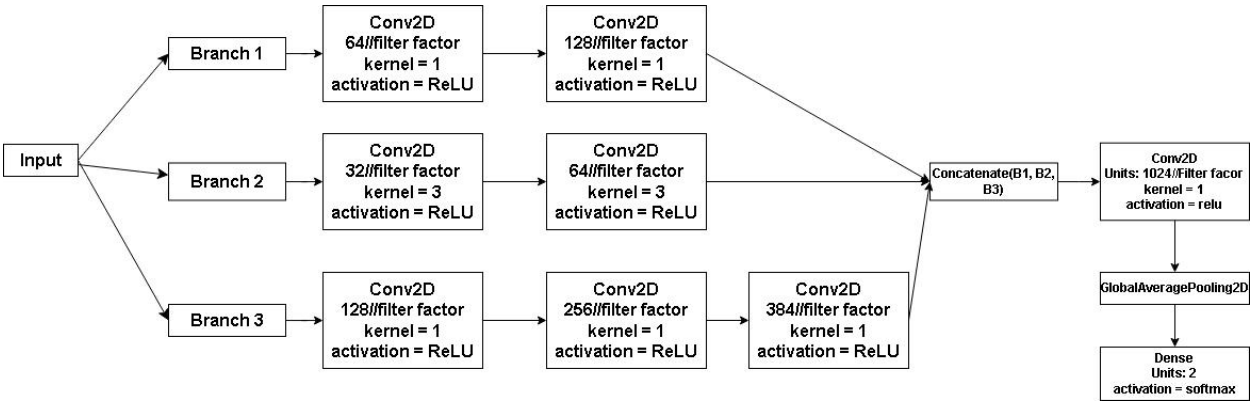

Figure 17: Initial Inception Block

Below from figure 18 to figure 24 displays the testing results for each metric of the above model.

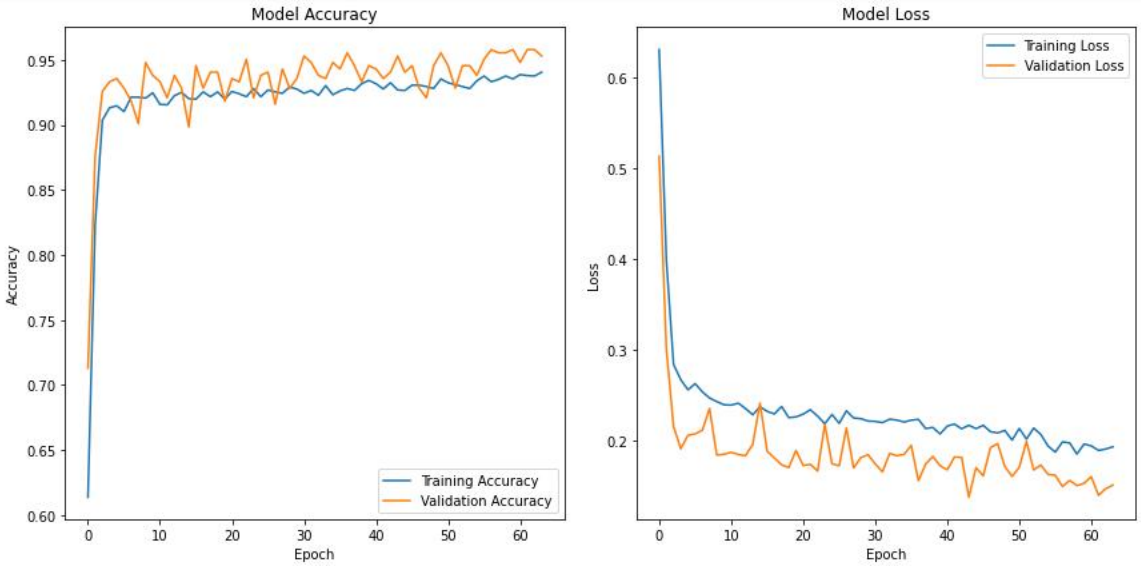

Figure 18: Accuracy, Loss, of the Inception model

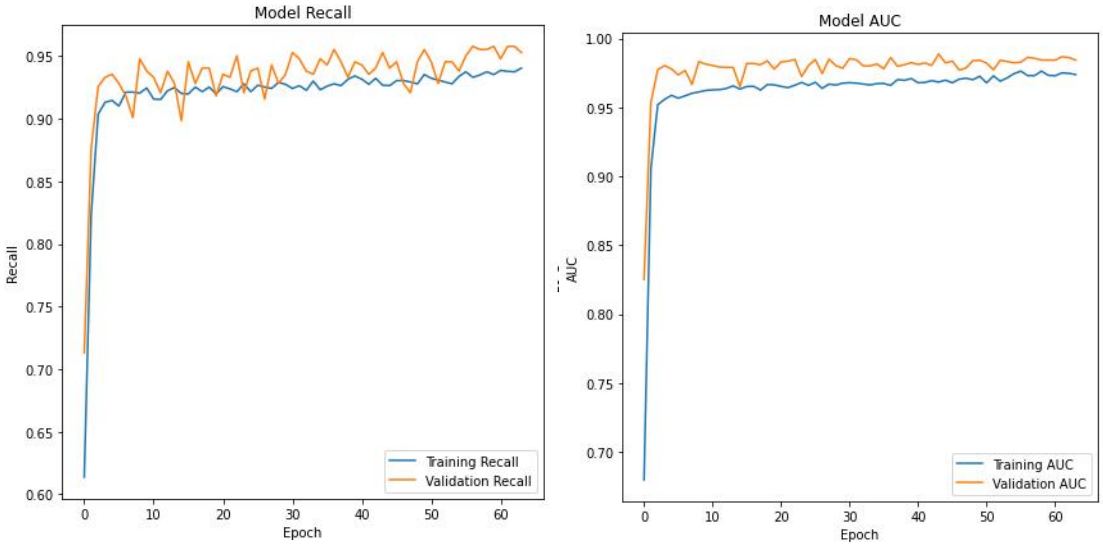

Figure 19: Precision and AUC Trend of Inception Model

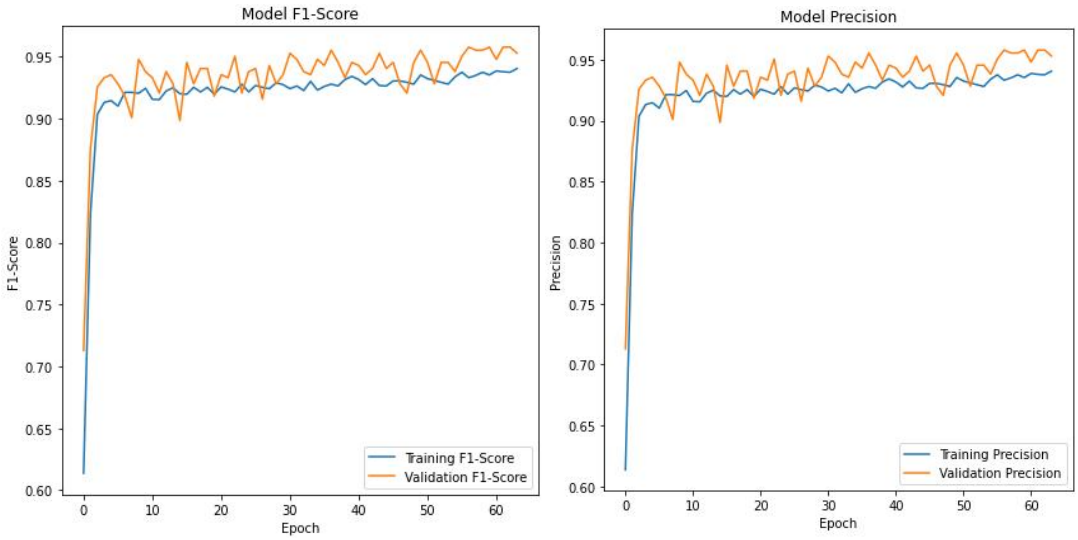

Figure 20: F1-Score, Precision Trend of Inception Model

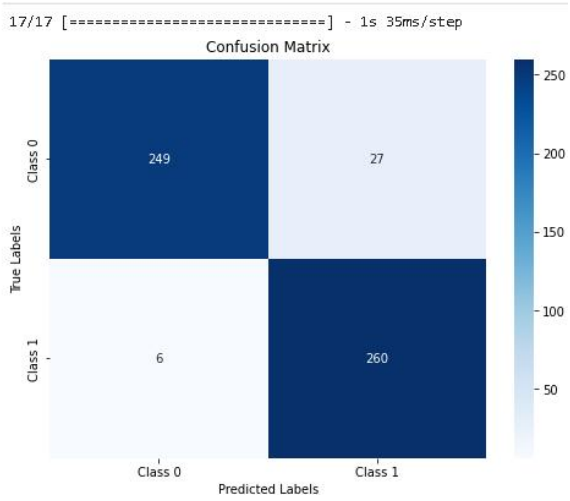

Figure 21: Confusion Matrix of Inception Model

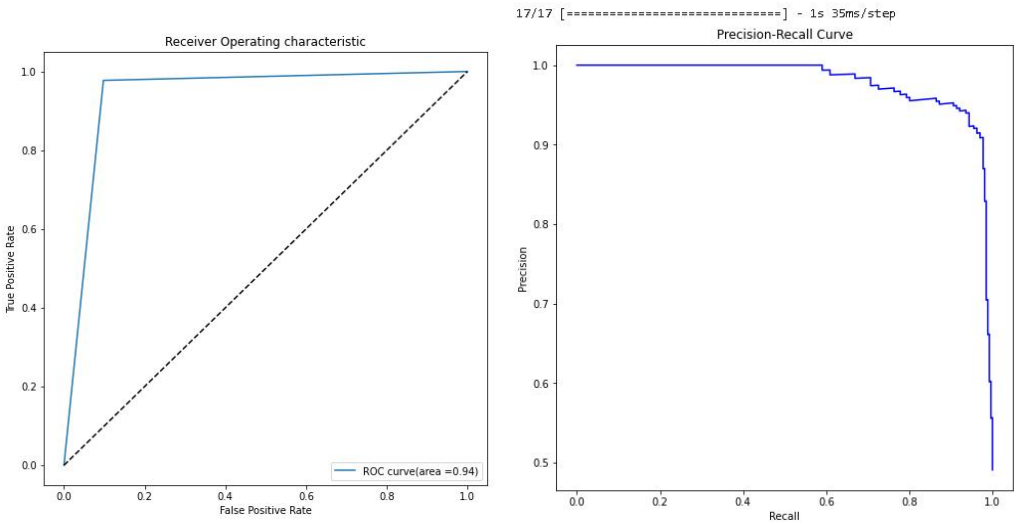

Figure 22: ROC Curve and Precision-Recall Curve of Inception Model

```
sensitivity: 0.9774436090225563
specificity: 0.9021739130434783
```

Figure 23: Specificity and Sensitivity

```
17/17 [==============================] - 1s 35ms/step
              precision    recall  f1-score   support

           0       0.98      0.90      0.94       276
           1       0.91      0.98      0.94       266

    accuracy                           0.94       542
   macro avg       0.94      0.94      0.94       542
weighted avg       0.94      0.94      0.94       542
```

Figure 24: Classification Report

The ResNet-style neural begins with a initial convolutional layer with 64 filters, followed by batch normalization and max pooling, setting the stage for feature extraction. The core consists residual blocks, where each with two main convolutional layers. These blocks employ a residual connection to merge the block's input and output. An optional MaxPooling Step follows each block for dimension reduction. The model concludes with a GlobalAveragePooling2D layer, compress the dimension of the feature maps, and a dense layer with softmax activation.

Figure 25 presents the structure of the residual blcok.

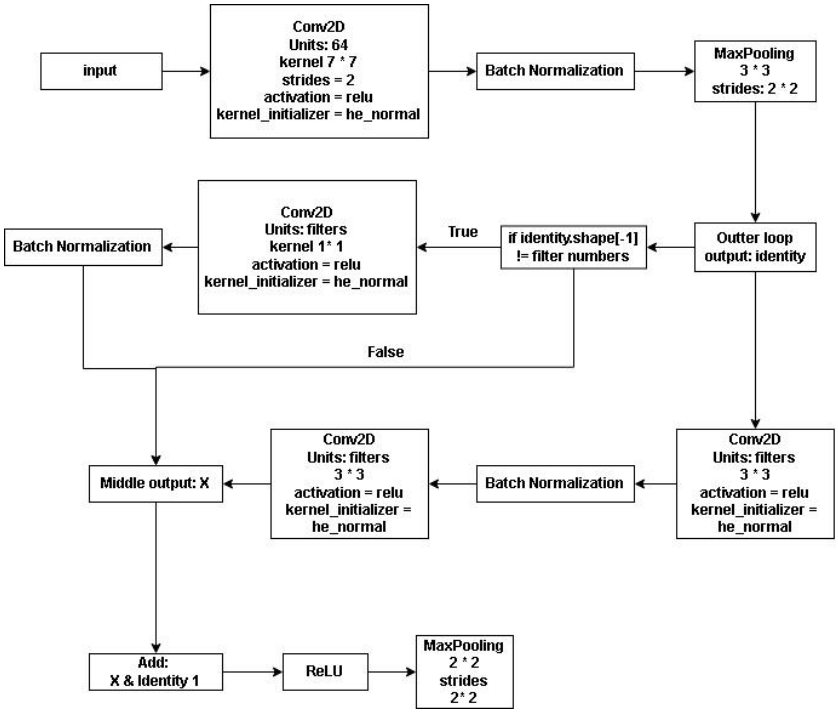

Figure 25: Initial Residual Block

Furthermore, the test results of the residual block is presented from figure 26 to figure 32

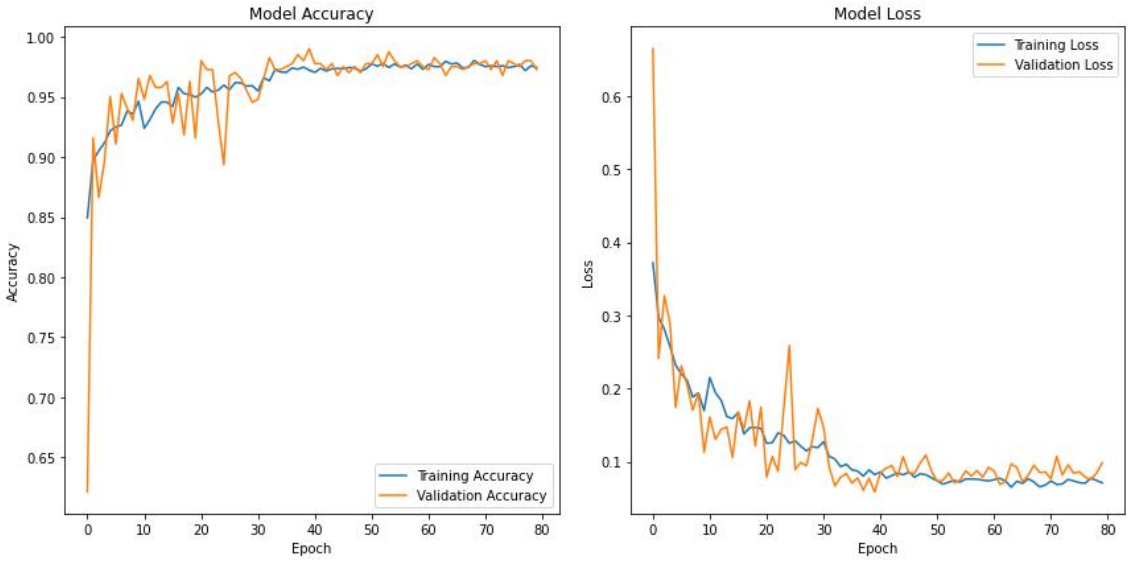

Figure 26: Accuracy and Loss of the Residual Model

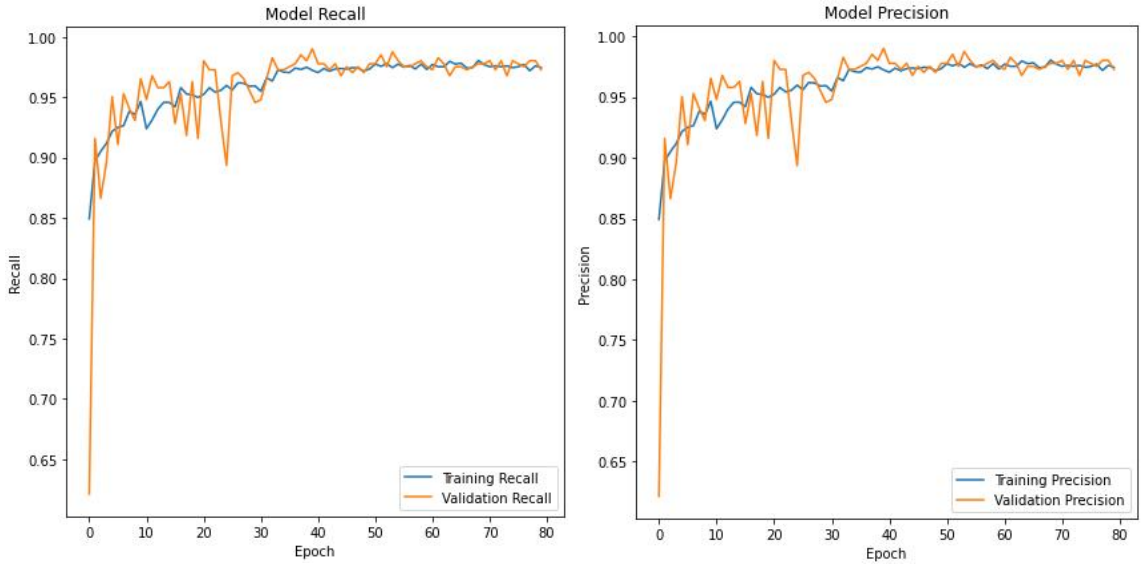

Figure 27: Recall and Precision Trend of Residual Model

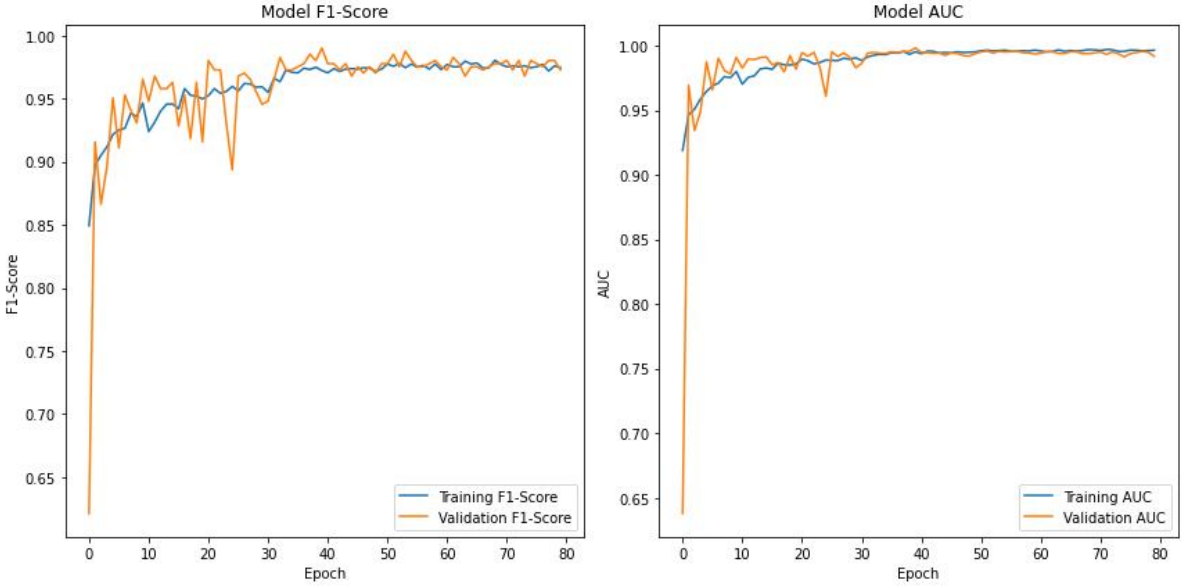

Figure 28: F1-Score and AUC Curve of Residual Model.

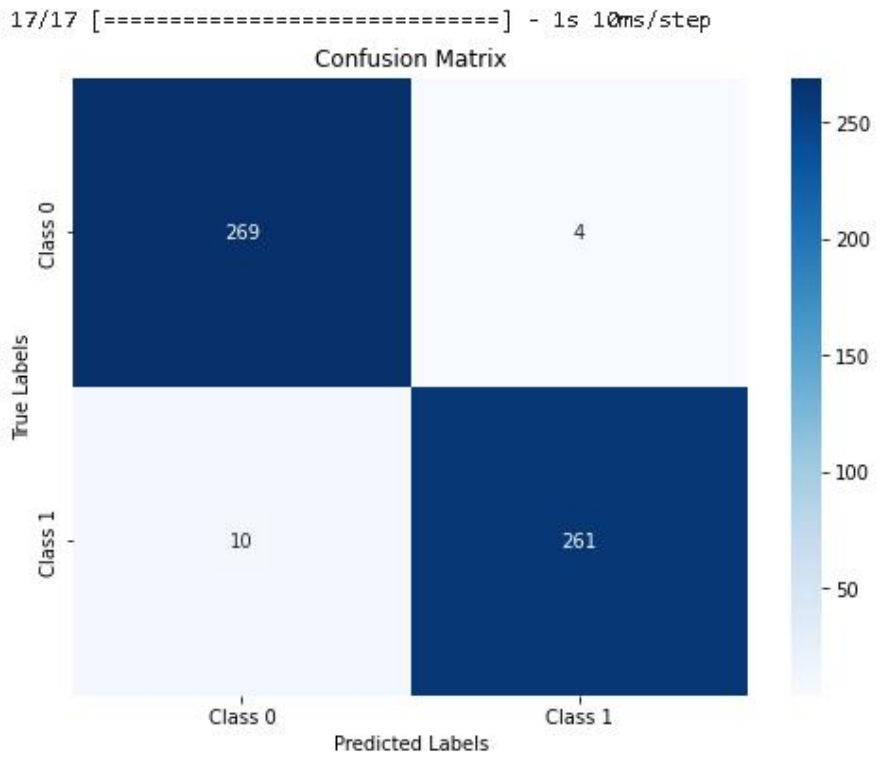

Figure 29: Confusion Matrix of Residual Model

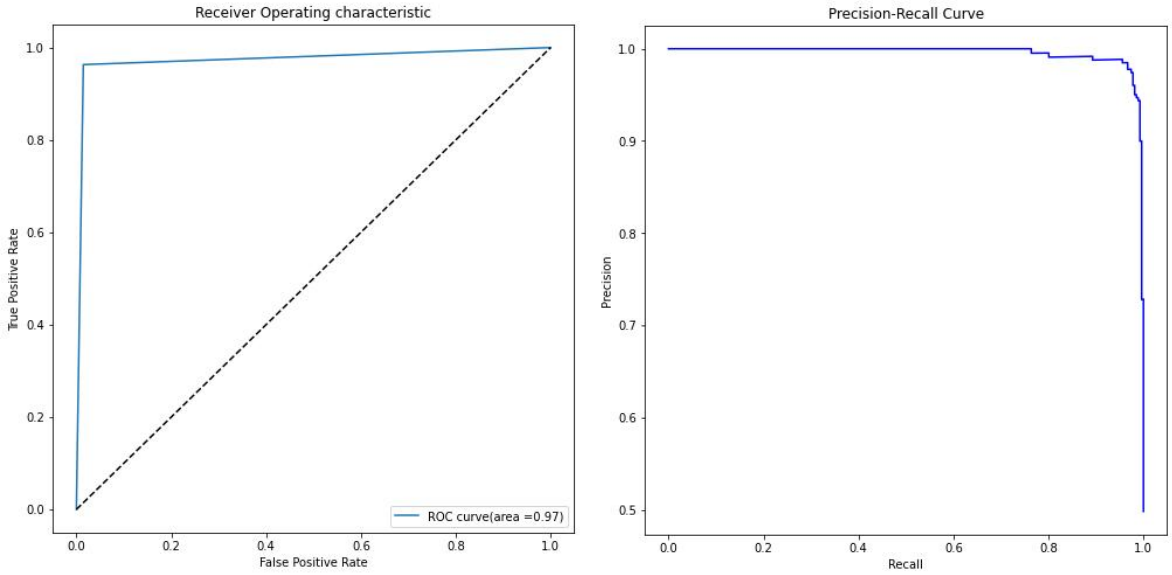

Figure 30: ROC Curve and Precision Recall Curve of Residual block

```
sensitivity: 0.9630996309963099
specificity: 0.9853479853479854
```

Figure 31: Specificity and Sensitivity

```
17/17 [==============================] - 0s 10ms/step
              precision    recall  f1-score   support

           0       0.96      0.99      0.97       273
           1       0.98      0.96      0.97       271

    accuracy                           0.97       544
   macro avg       0.97      0.97      0.97       544
weighted avg       0.97      0.97      0.97       544
```

Figure 32: Classification Report

Based on the former results individual tests, these individual models are put to combination tests, with Inception and ResNet as one singular, and Convolutoinal Block as another one.

4.1.4 Inception-ResNet Model

Based on the Inception and Residual Structure, Inception-ResNet was defined.

The model begins with an input layer, taking data of specified “input_shape”. It then proceeds to an initial convolutional stage that plays the role of a filtering and feature extraction with 32 filters of size 7 * 7 to get as many features as possible for later process. This followed by batch normalization and max pooling layer, setting the stage for further data process.

Inside the loop lies the Residual and Inception Block.

For the residual block, it starts with convolutional layers using number of filters in the array [32, 32, 64, 32]. These layers are equipped with ReLU activation function and batch normalization. The residual connection in these blocks help to avoid the vanishing gradient issues when the network will get deeper after concatenation by providing a direct path for gradient during backpropagation.

As for the parallel block of Inception structure, it consists three branches with various convolutional setups. The first branch uses 1 * 1 convolutions with 64 filters, the second branch has a series of 1 * 1 followed by 3 * 3 convolutions(64 then 32 filters), and the third branch extends this idea further with 128 and 32 filters. Batch normalization is applied in each branch.

After processing through the inception block, the outputs of all branches are concatenated and then matched in filter size with the output of the corresponding residual block. This is done through 1 * 1 convolution. Then the outputs of the residual and Inception Block are then combined using an “Add()” operation. This ensures that features extracted by both the ResNet and Inception methodologies are integrated. A max pooling layer is applied for further spatial reduction. The model concludes with a GlobalAveragePooling2D layer and softmax activation.

The following figure 33 to figure 35 displays the architecture of this model.

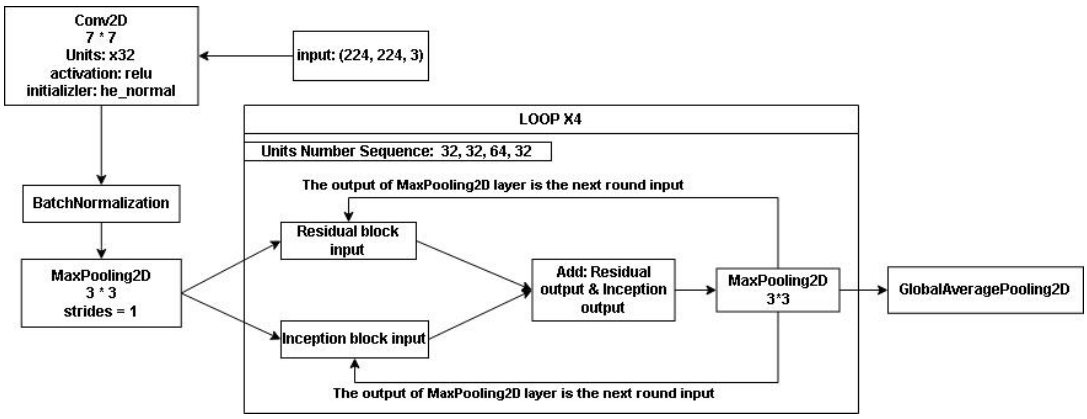

Figure 33: Inception-ResNet Architecture

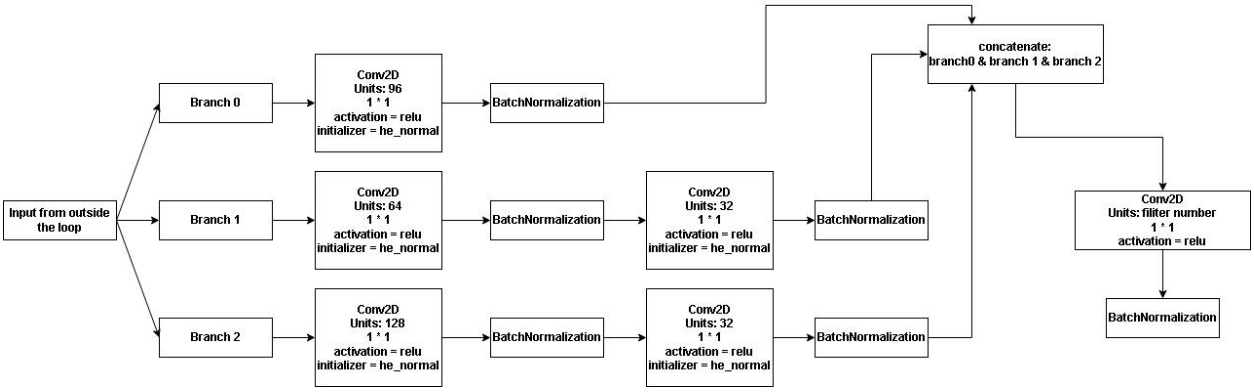

Figure 34: Internal Structure of the Inception Block

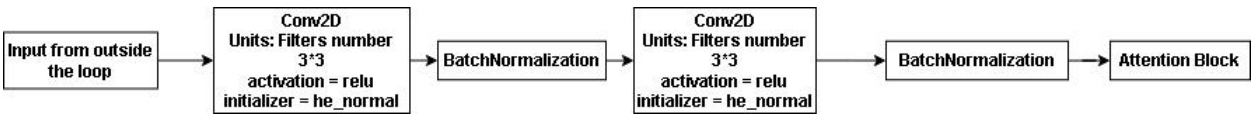

Figure 35: Internal Structure of Residual Block

The following figure 36 to figure 42 displays the results for the Inception-ResNet on the dataset

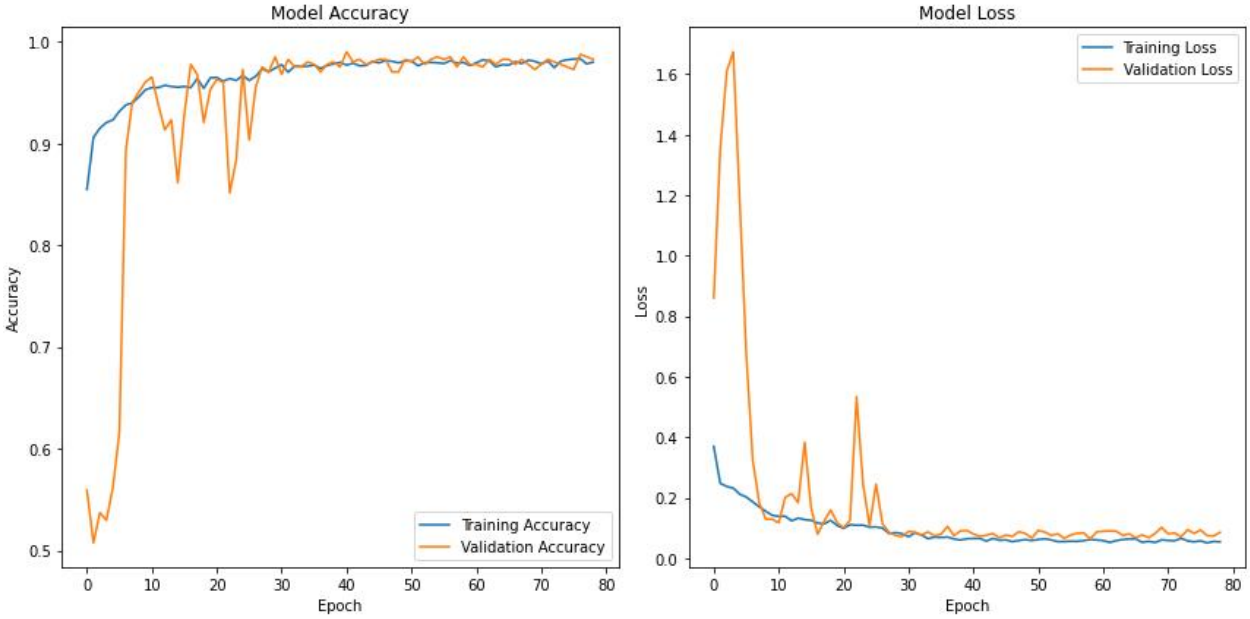

Figure 36: Accuracy and Loss

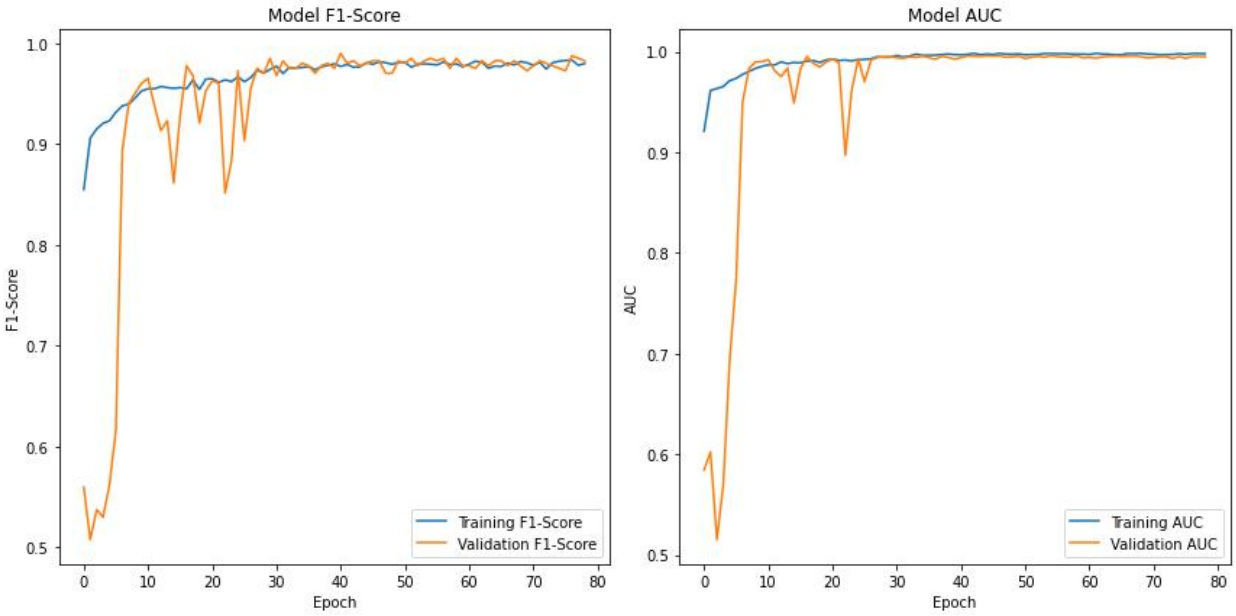

Figure 37: F1-Score and AUC Trend

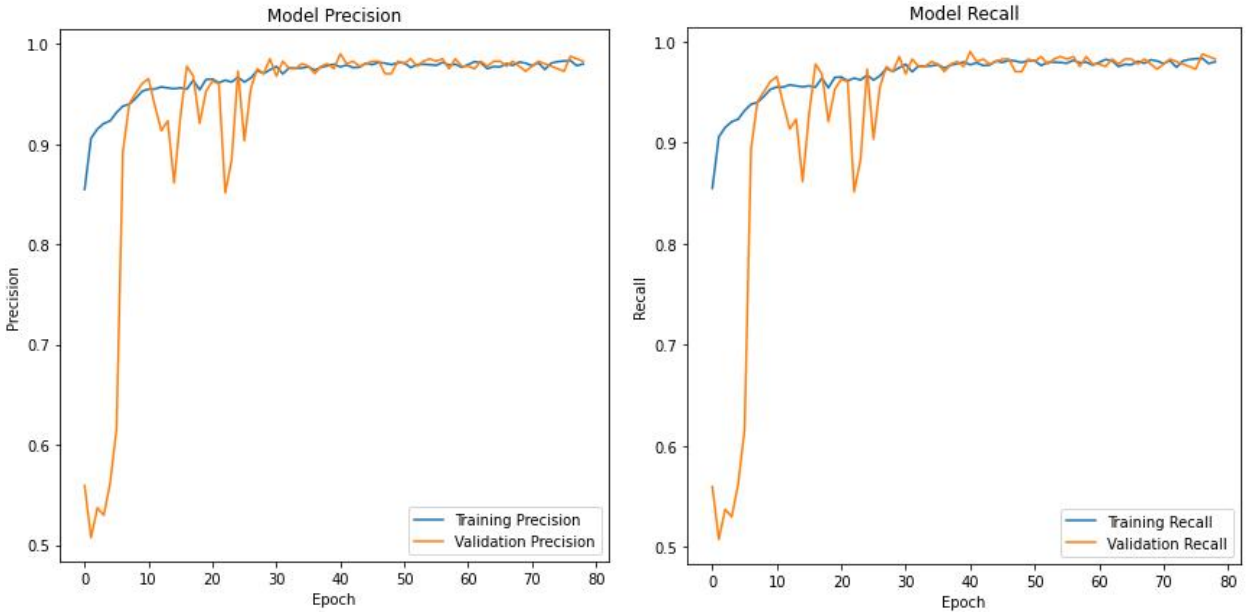

Figure 38: Precision and Recall Curve

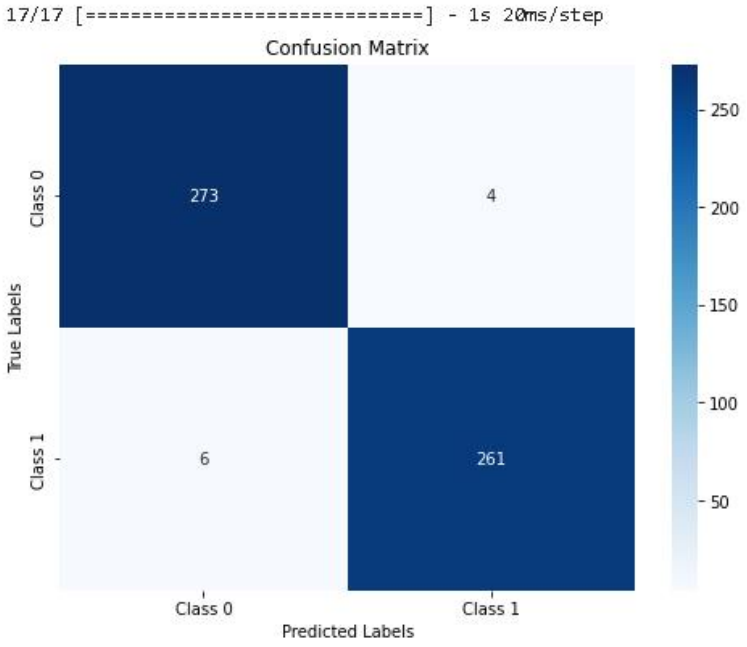

Figure 39: Confusion Matrix

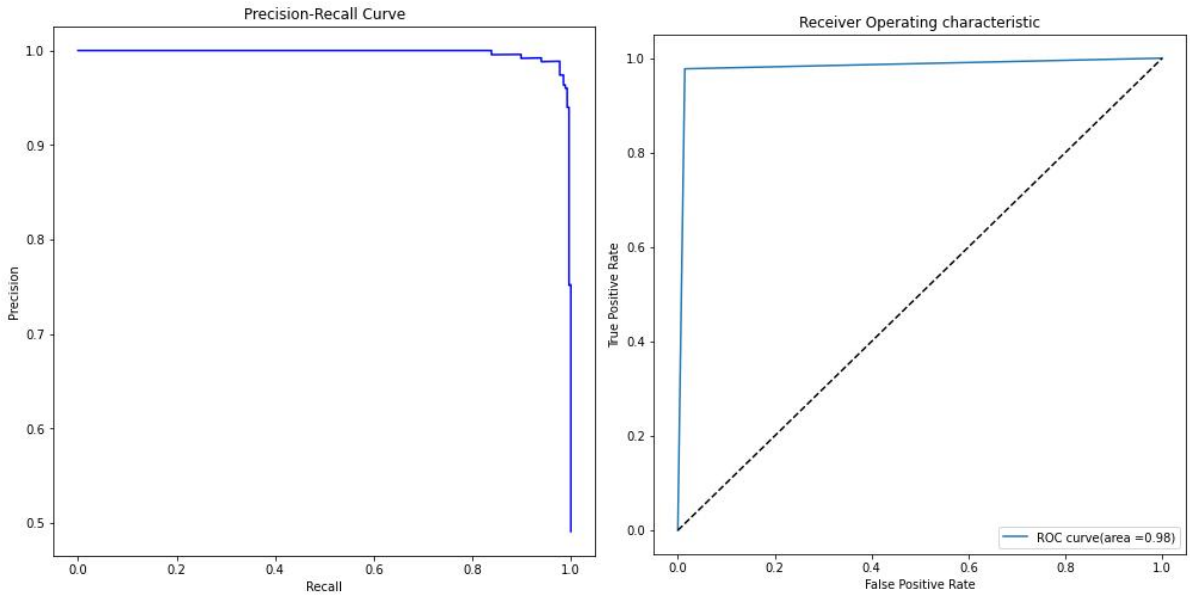

Figure 40: Precision-Recall Curve and ROC Curve

```
sensitivity: 0.9775280898876404
specificity: 0.9855595667870036
```

Figure 41: Specificity and Sensitivity

```
17/17 [==============================] - 0s 20ms/step
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       277
           1       0.98      0.98      0.98       267

    accuracy                           0.98       544
   macro avg       0.98      0.98      0.98       544
weighted avg       0.98      0.98      0.98       544
```

FIgure 42: Classification report

4.1.4 Attention Mechanism

After completed the implementation of the two basic models of ensemble, attention mechanism is implemented. A simple attention mechanism similar to spatial attention is created. It works by highlighting important parts and ignoring less important parts of the input. This is done using a convolutional layer that makes an attention map. After being adjusted by a sigmoid function, this map is used to emphasize important features in the input, helping the model to focus better and

be more accurate. The attention mechanism will be following after each layer and loop of convolutional block and Inception-ResNet block, therefore, considering the computing ability of this device, a simple and effective mechanism will suffice.

Structure is in figure 43.

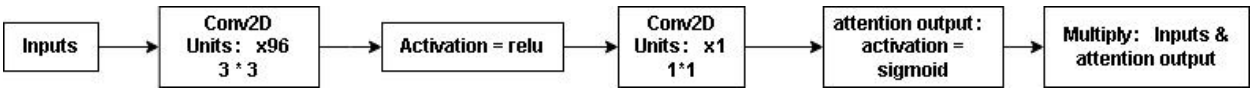

Figure 43: Attention mechanism

4.1.5 Convolutional Block with Attention

This stage, the singular model is combined with attention mechanism. For convolutional block, attention block follows behind each batch normalization layer after convolutional layer but before the max pooling layer, so as to create focus point before the dimension shrinks and after the feature is extracted.

The model structure is displayed below which is similar to the final structure, with difference in hyperparameters. Following figure 44 displays the structure.

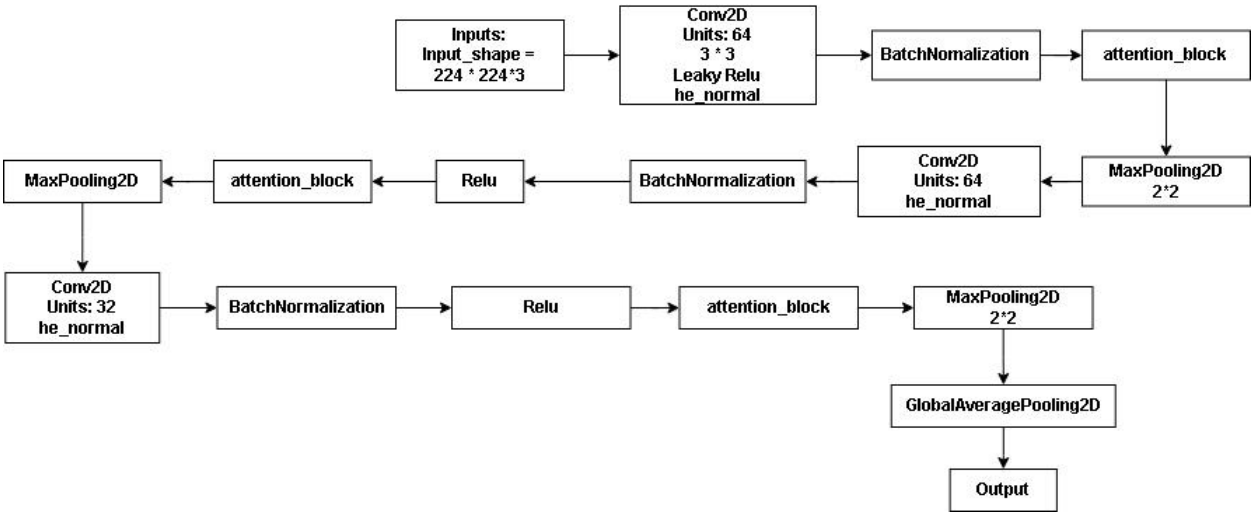

Figure 44: Convolutional Layer with Attention Mechanism

From figure 45 to figure 51 displays the test results.

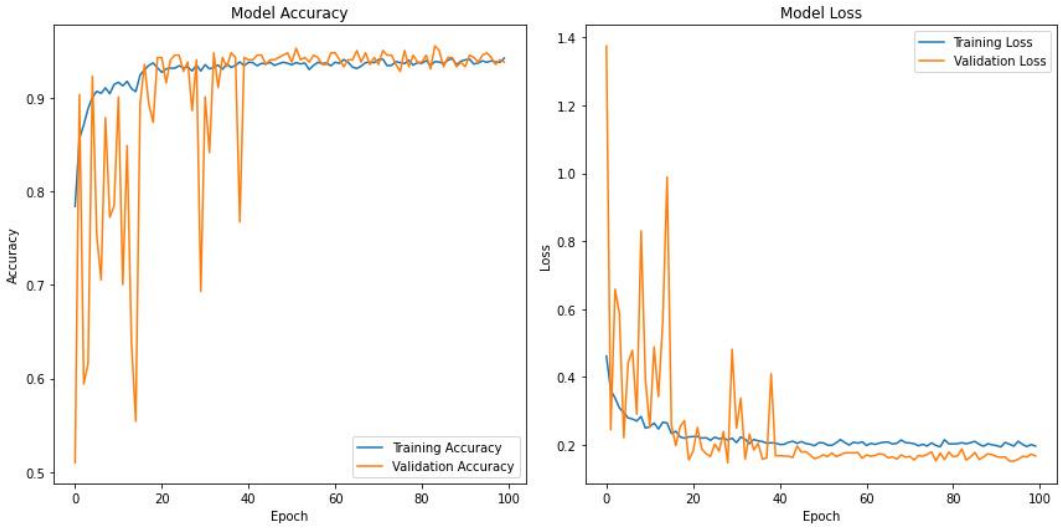

Figure 45: Accuracy and Loss

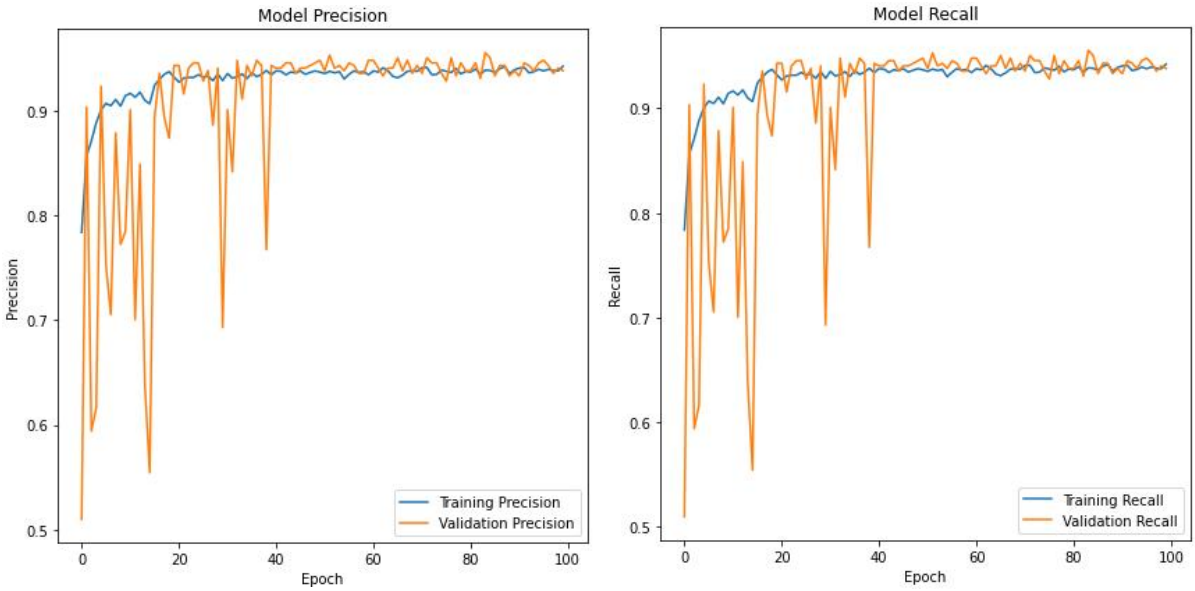

Figure 46:Precision and Recall

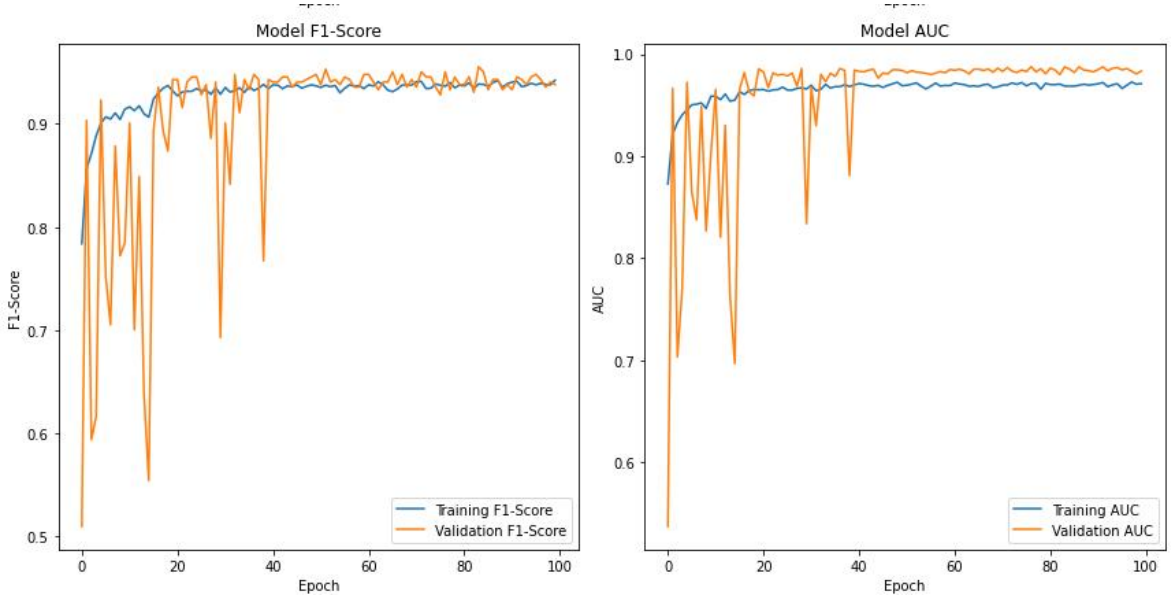

Figure 47: F1-Score and AUC

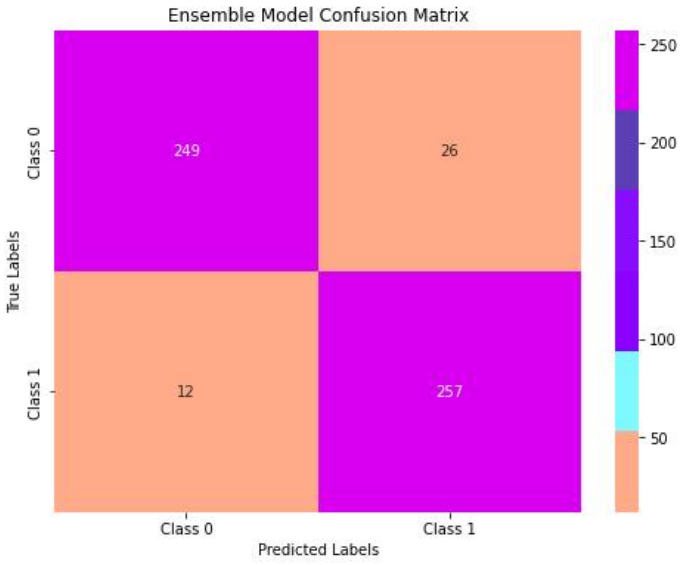

Figure 48: Confusion Matrix

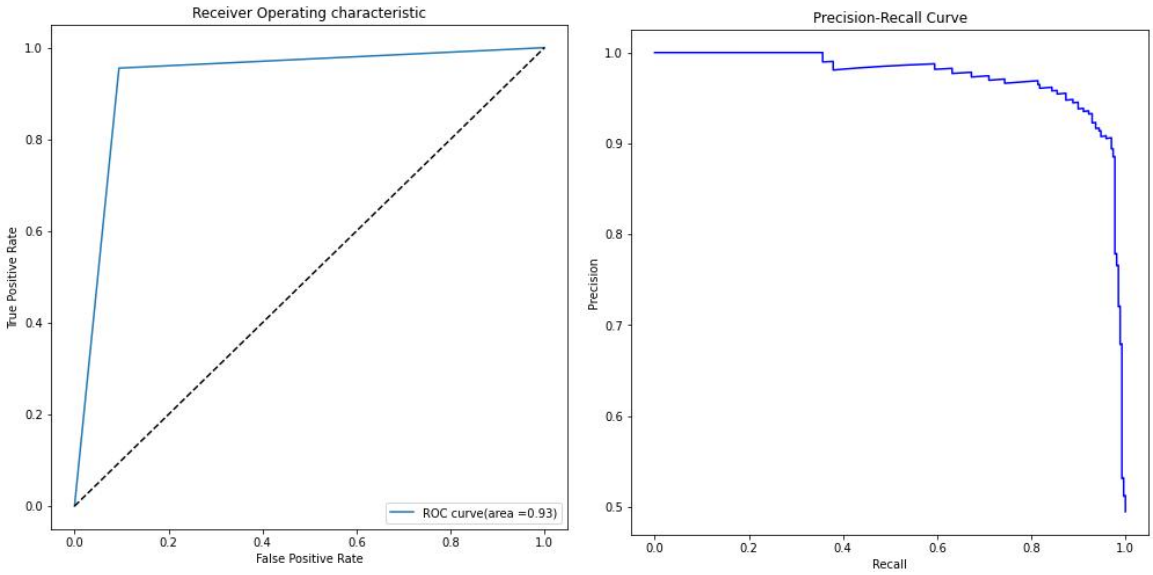

Figure 49: ROC Curve and Precision Recall Curve

```
sensitivity: 0.9553903345724907
specificity: 0.9054545454545455
```

Figure 50: Sensitivity and Specificity

	precision	recall	f1-score	support
0	0.95	0.91	0.93	275
1	0.91	0.96	0.93	269
accuracy			0.93	544
macro avg	0.93	0.93	0.93	544
weighted avg	0.93	0.93	0.93	544

Figure 51: Classification Report

4.1.6 Ensemble Model Version 1

After testing the attention mechanism on the simple convolutional block, the ensemble model is now built with the attention added to the previous Inception-ResNet model and concatenate with convolutional block. The following figures demonstrate the structure of ensemble model.

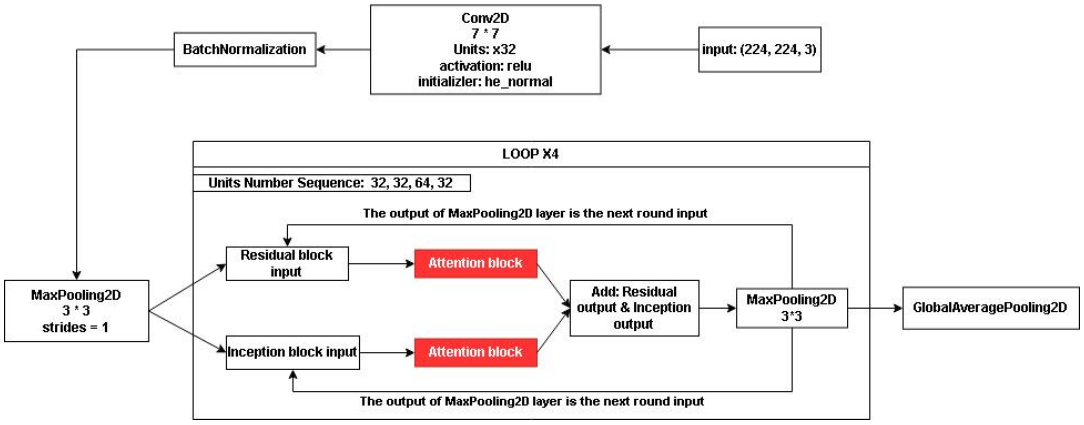

Figure 52: Inception-ResNet with Attention Integrated

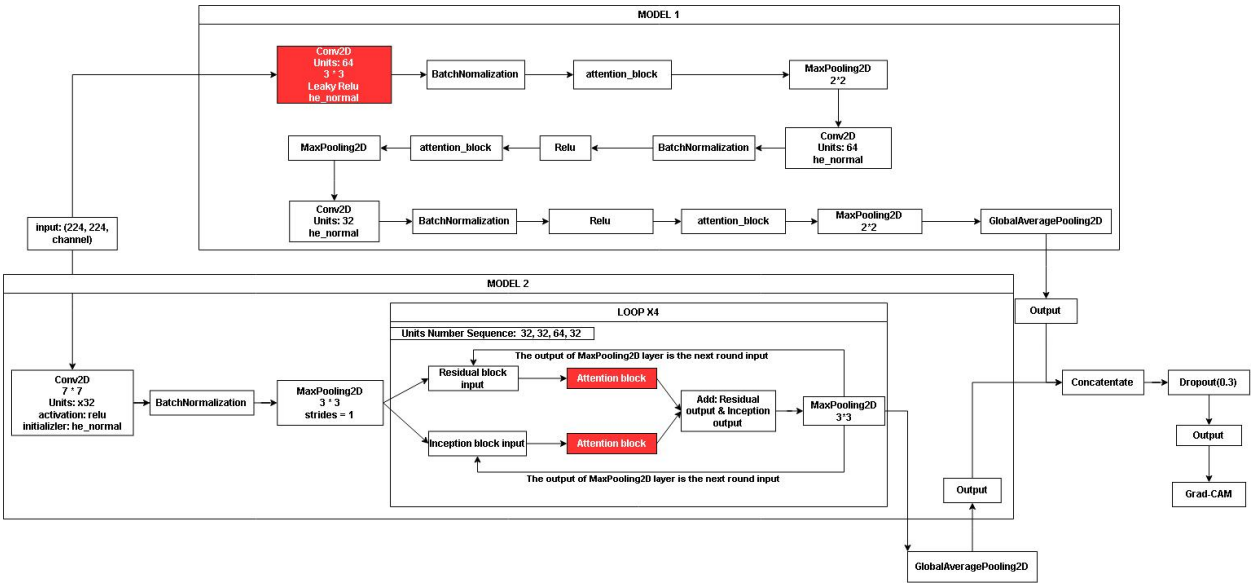

Figure 53: Ensemble Model Version 1

Besides the highlighted parts which illustrate difference, hyperparameters have also distinct from the final which will be fine tuned later. Additionally, results of the ensemble model version 1 is provided below.

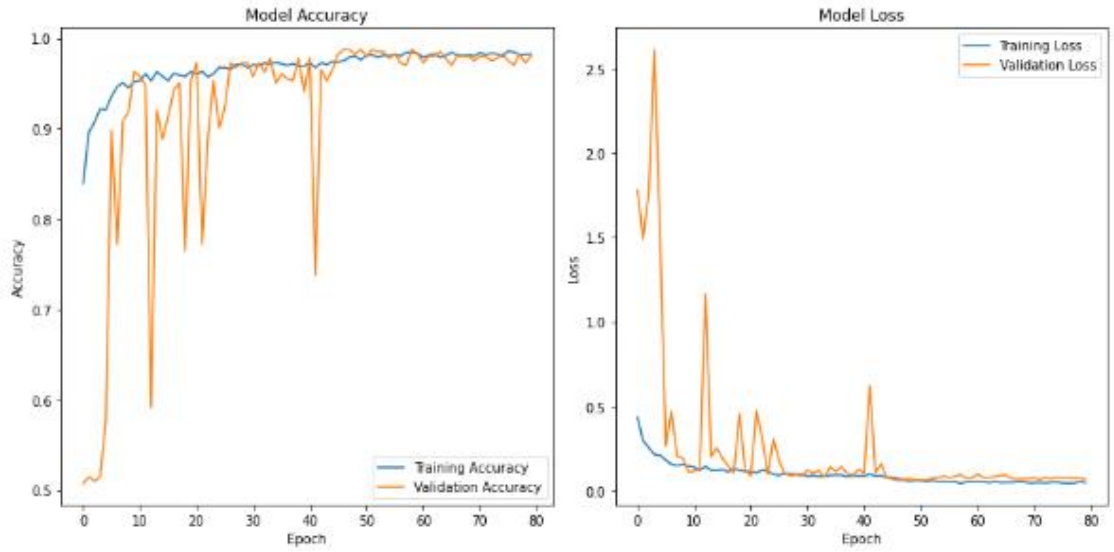

Figure 54: Ensemble Accuracy and Loss

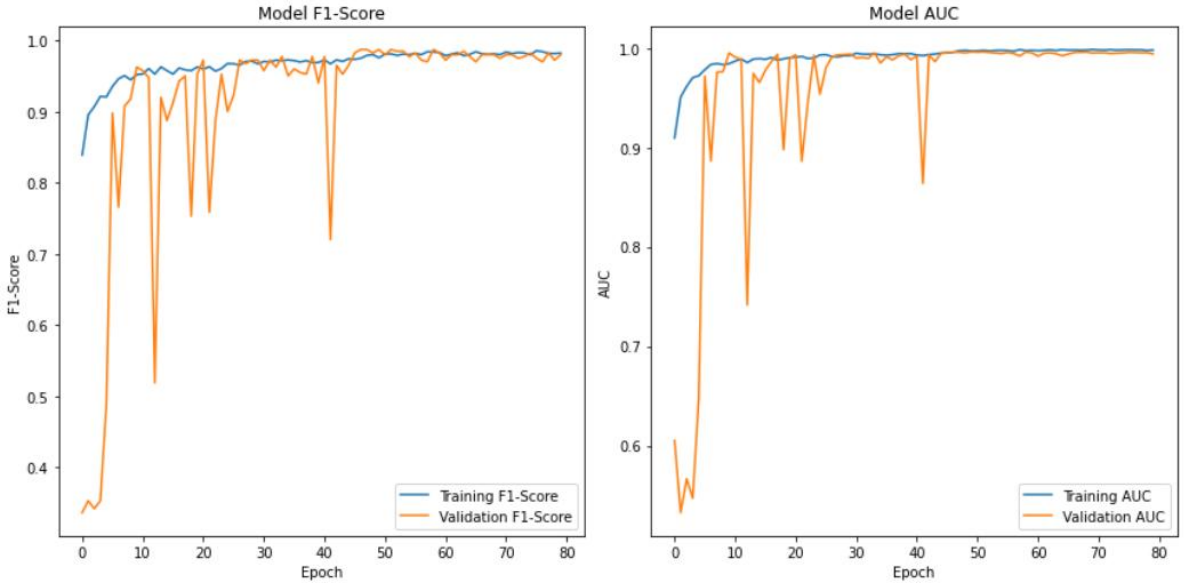

Figure 55: F1-Score and AUC Curve

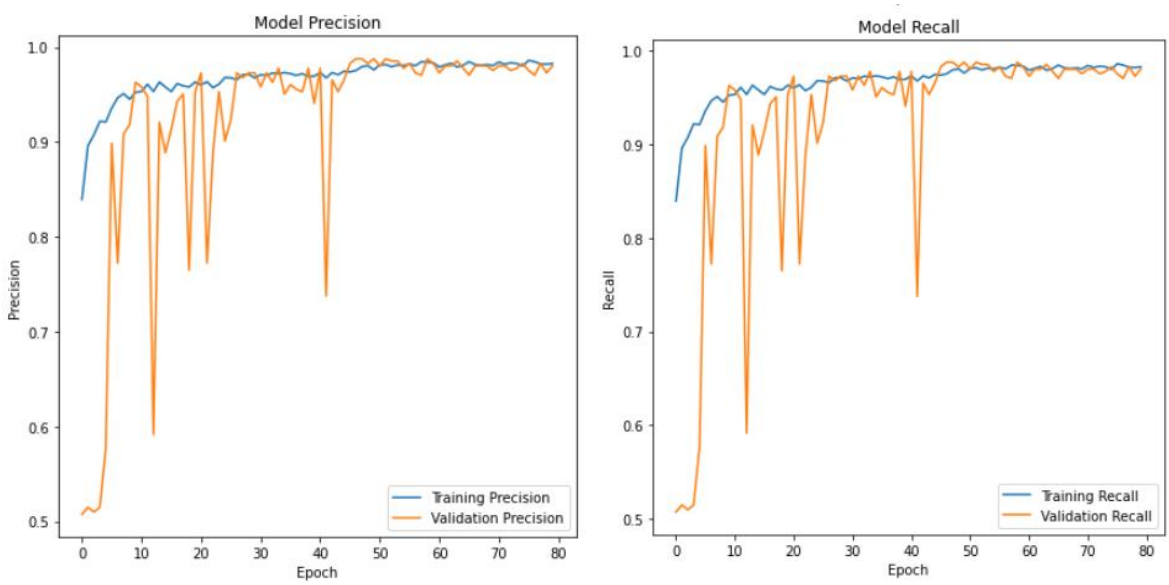

Figure 56: Precision and Recall Curve

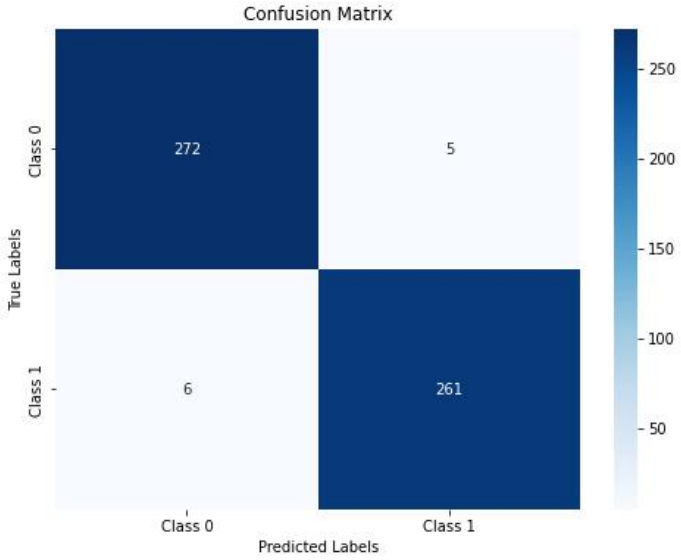

Figure 57: Ensemble Model Version 1 Confusion Matrix

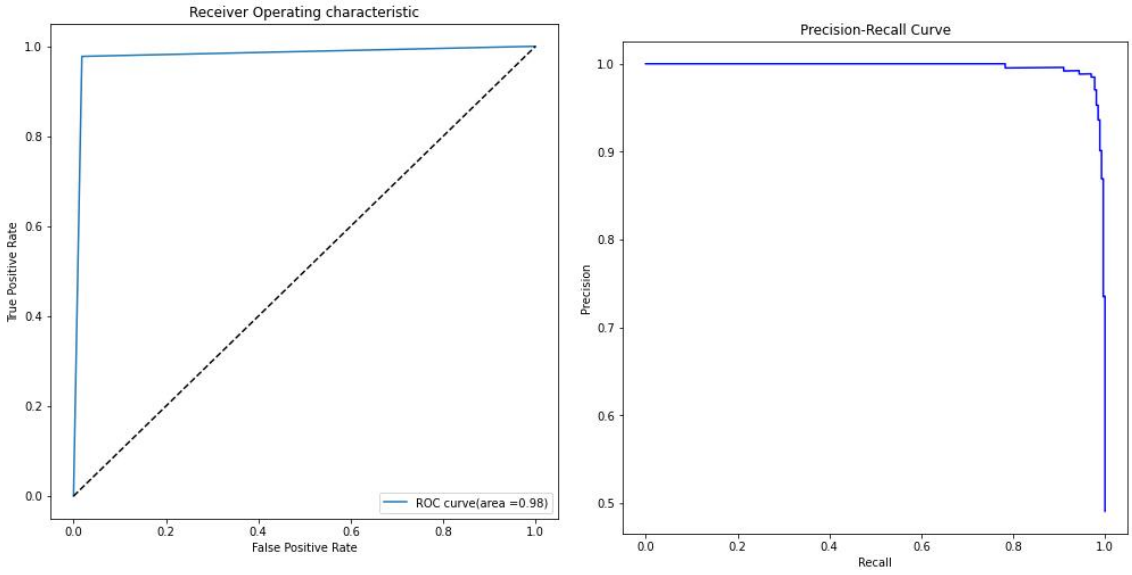

Figure 58: Ensemble model ROC and Precision-Recall

```
sensitivity: 0.9775280898876404
specificity: 0.9819494584837545
```

Figure 59: Sensitivity and Specificity

```
17/17 [==============================] - 1s 35ms/step
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       277
           1       0.98      0.98      0.98       267

    accuracy                           0.98       544
   macro avg       0.98      0.98      0.98       544
weighted avg       0.98      0.98      0.98       544
```

Figure 60: Classification Report

4.1.7 Preprocessing tests

Since the model is constructed completely, two more data preprocessing tests were applied on Grayscale images and CLAHE transition images as mentioned previously. And the following figures demonstrate the results on these two derivation datasets.

Due to the equation and algorithm of the Micro-F1-Scroe, Micro-Precision and Micro-Recall, the results will be similar or same as accuracy, and thus, accuracy and loss are the main focus in the test.

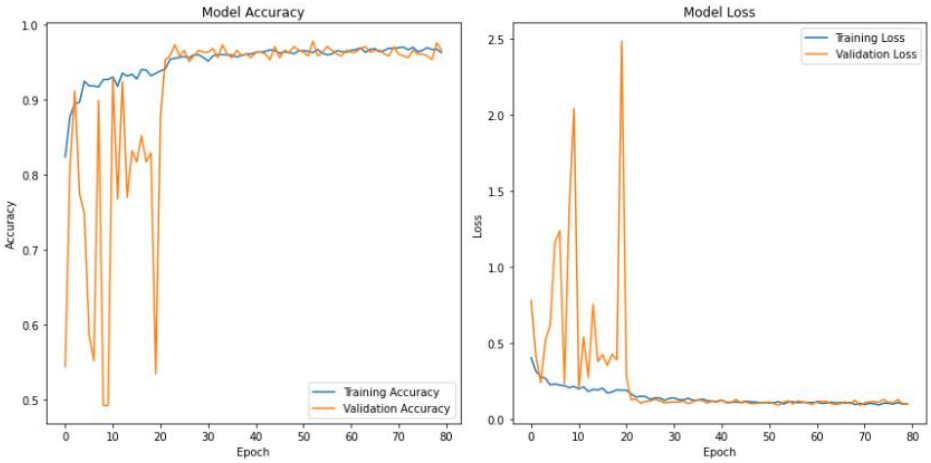

Figure 61: CLAHE Ensemble Test Results

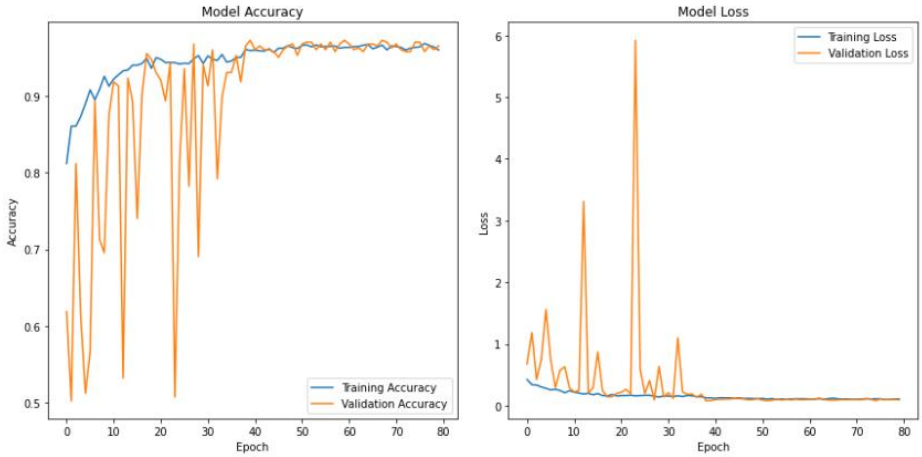

Figure 62: Grayscale Images Test Results

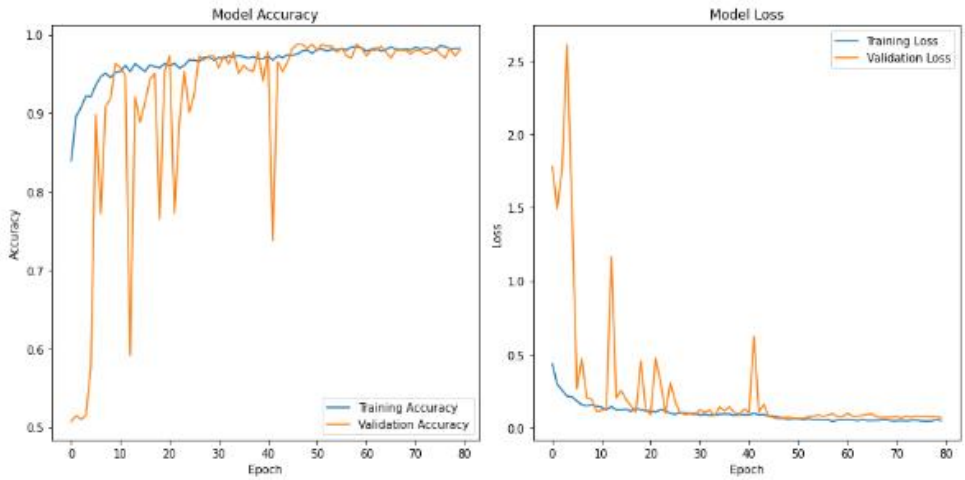

Figure 63: No Image colour transfer Test Results

4.1.8 Fine Tuning and Visualization

According to the results of different image transition, it is consider that model perform the best and image is more compatible when no transition of the overall attributes of the original images. Therefore, the ensemble model entered the final hyperparameter adjustments including layers, neuron number, batch size, image input sizes, automatic learning rate adjustments and early stop to store the model at its best epoch.

The structure of first version of ensemble model is shown in figure 53, and detailed structure of each block is illustrated from figure 64 to figure 66.

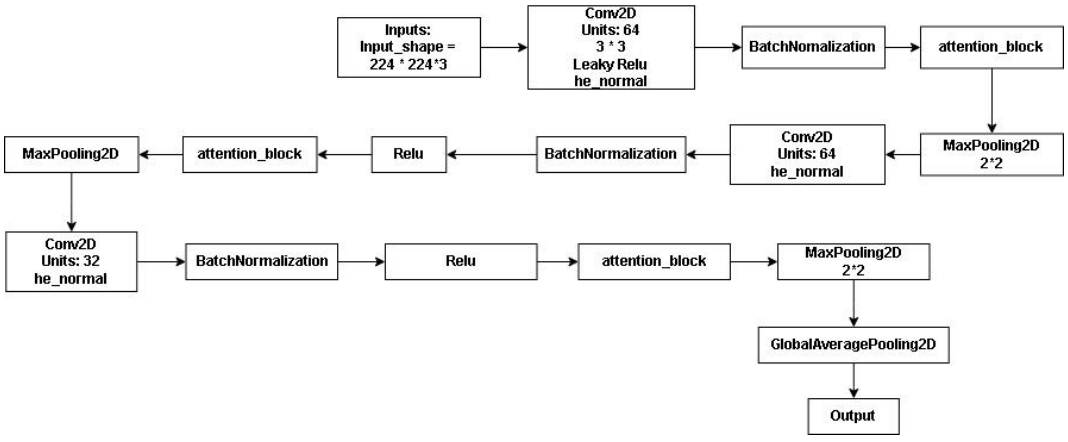

Figure 64: Convolutional Block of Ensemble Version 1

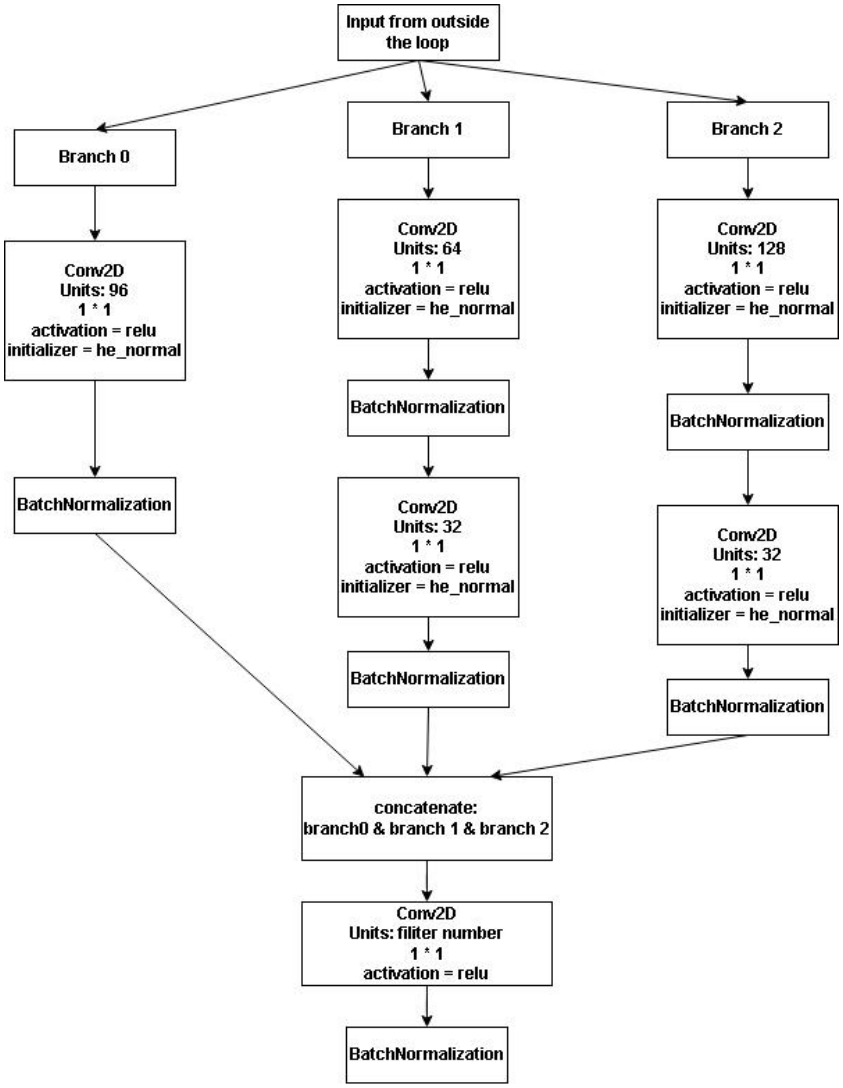

Figure 65: Inception Block of Ensemble Version 1

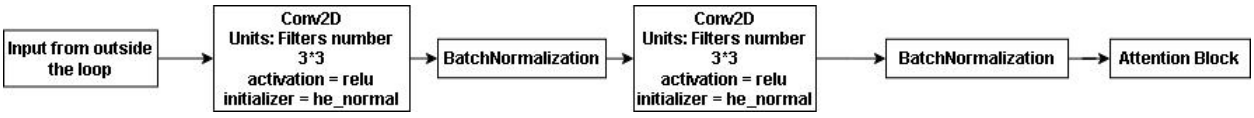

Figure 66: ResNet Block of Ensemble Version 1

Through using multiple combinations of hyperparameters, and adding customized loss function and accuracy, the ensemble model had been optimized to the proposed version, evaluation matrix were gradually changed from figure 67 to figure 70.

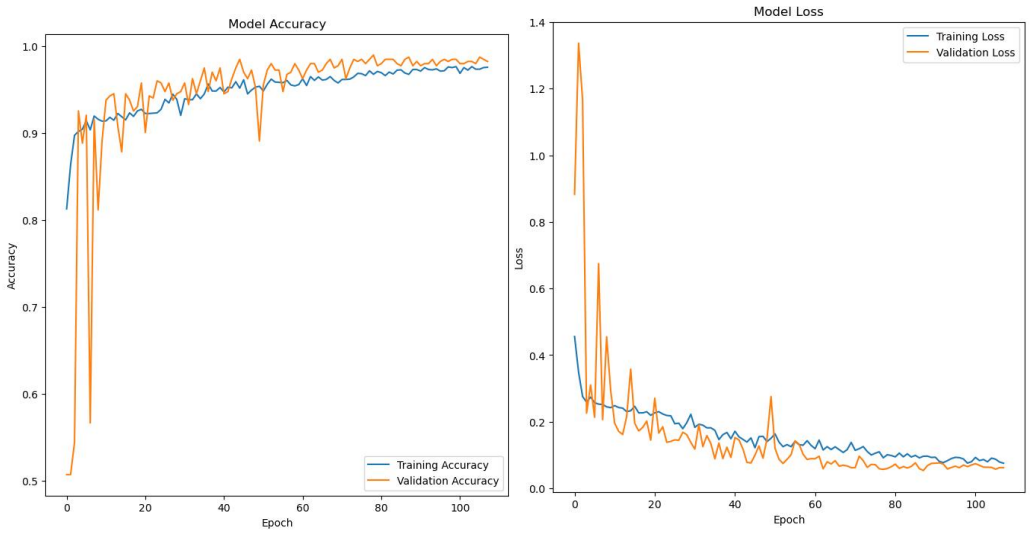

Figure 67: One more convolutonal Layer after ensemble, Dropout = 0.4, Batch_Size = 16, Inbuilt Loss and Accuracy

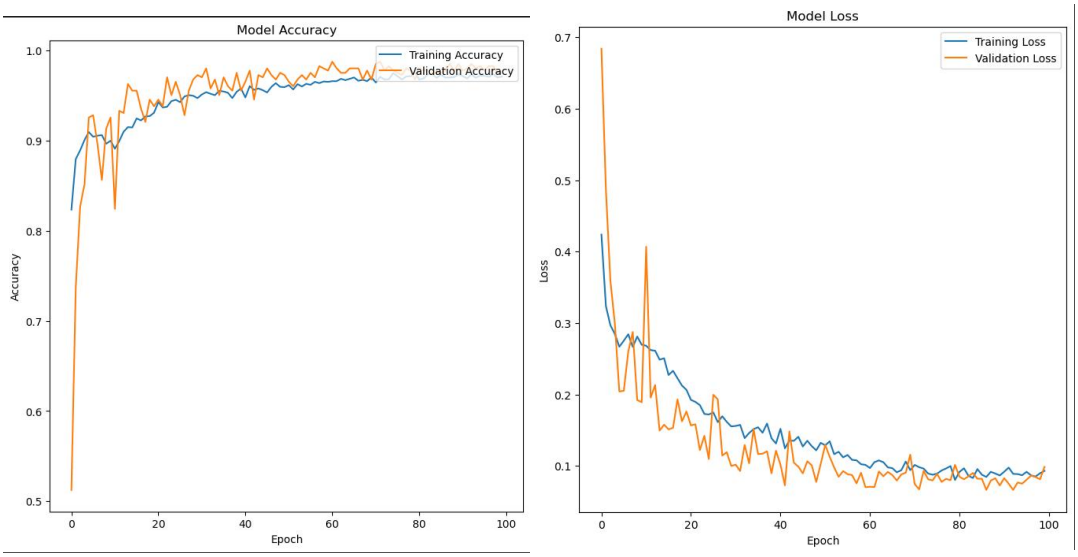

Figure 68: Batch Size = 8, Dropout = 0.5, Inbuilt Loss and Accuracy

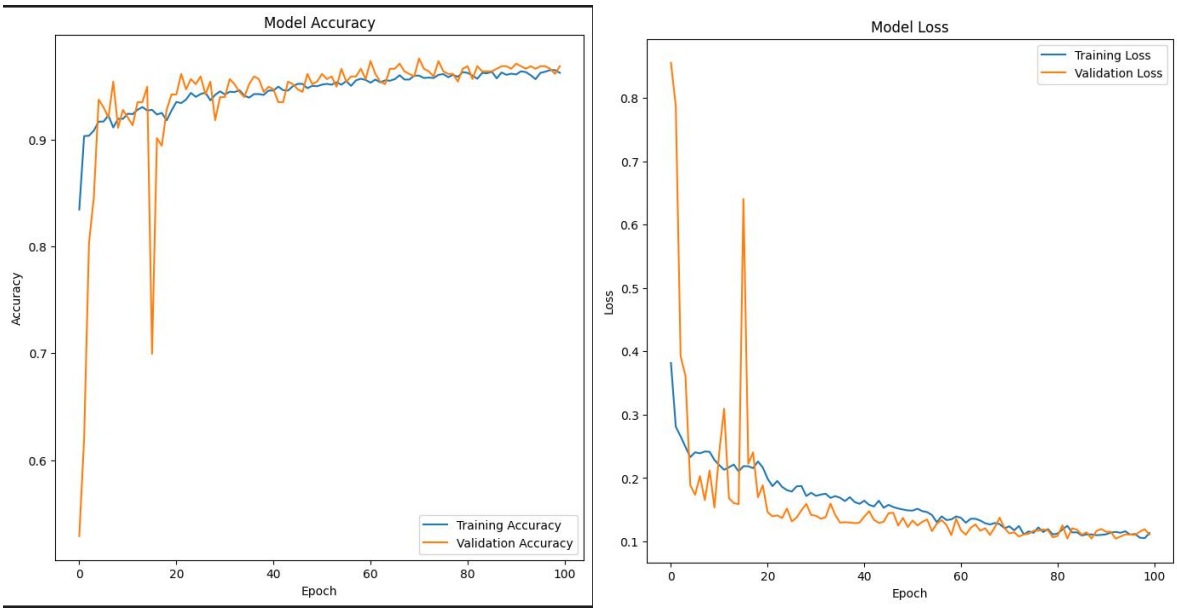

Figure 69: Batch_Size = 16, Dropout = 0.3, Customized Loss and Accuracy

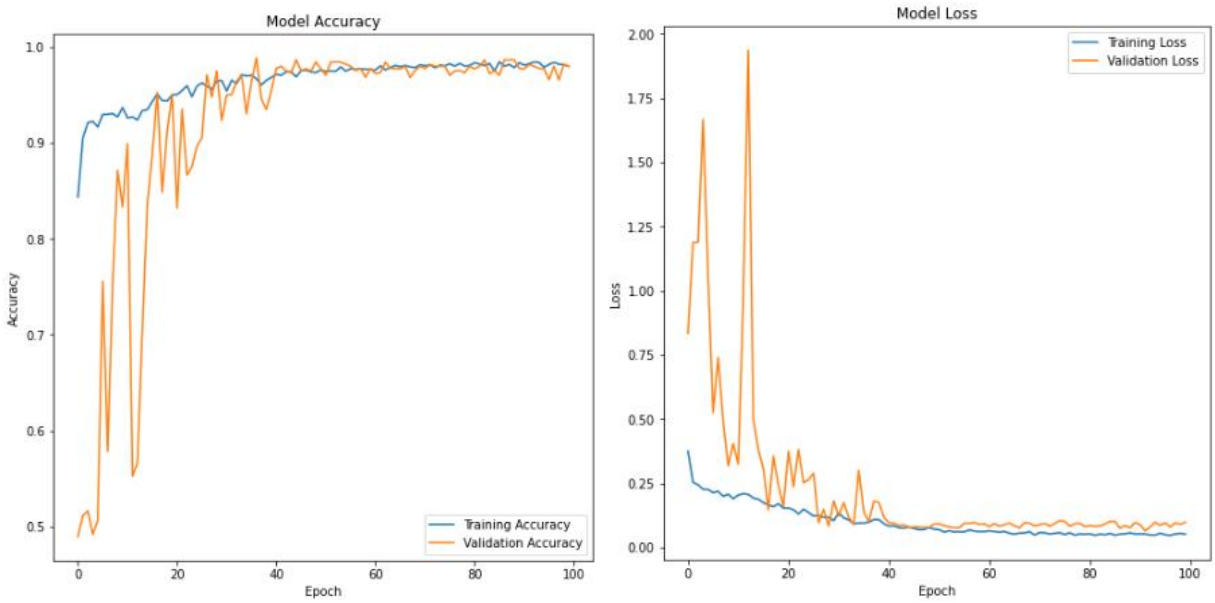

Figure 70: Batch_Size = 64, Dropout = 0.3, Customized Loss and Accuracy

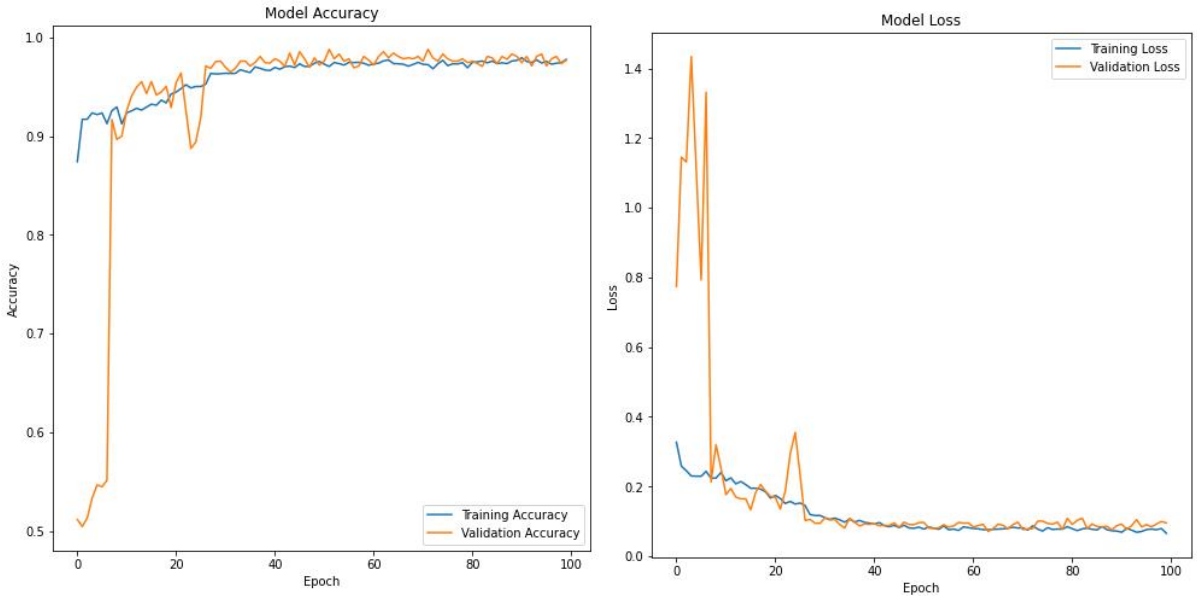

Figure 71: Proposed Version,Batch Size = 32, Dropout = 0.3, Customized Loss and Accuracy

With the validation accuracy around 98% and validation loss around 0.095, the confusion matrix, F1-Score, Precision and Recall were at a quite ideal level, all of which are presented from figure 72 to 77.

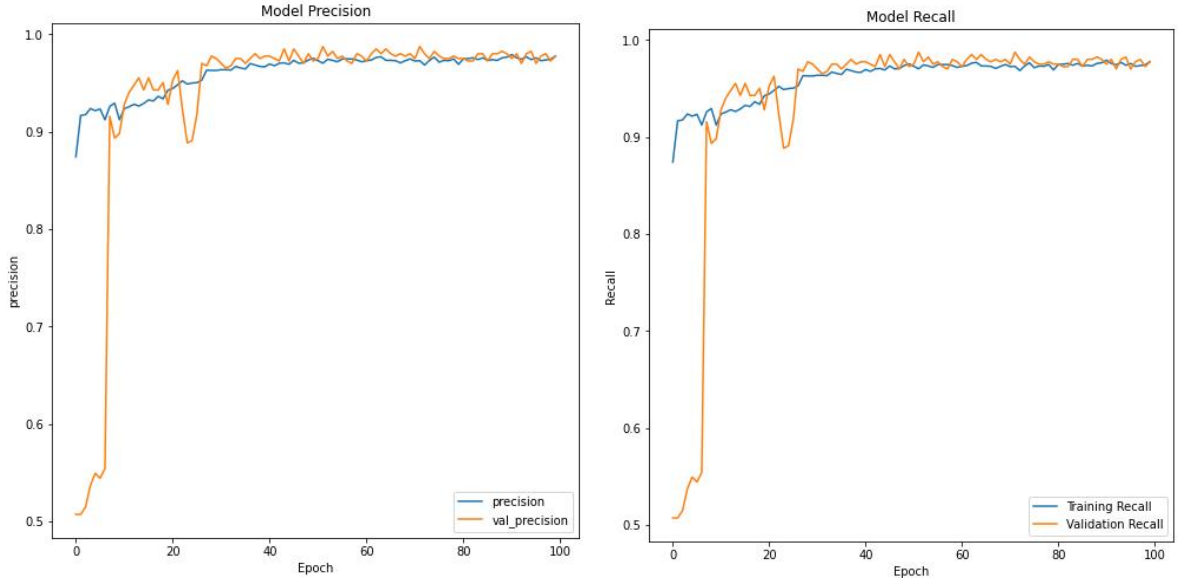

Figure 72: Proposed Precision and Recall, Customized Loss and Accuracy

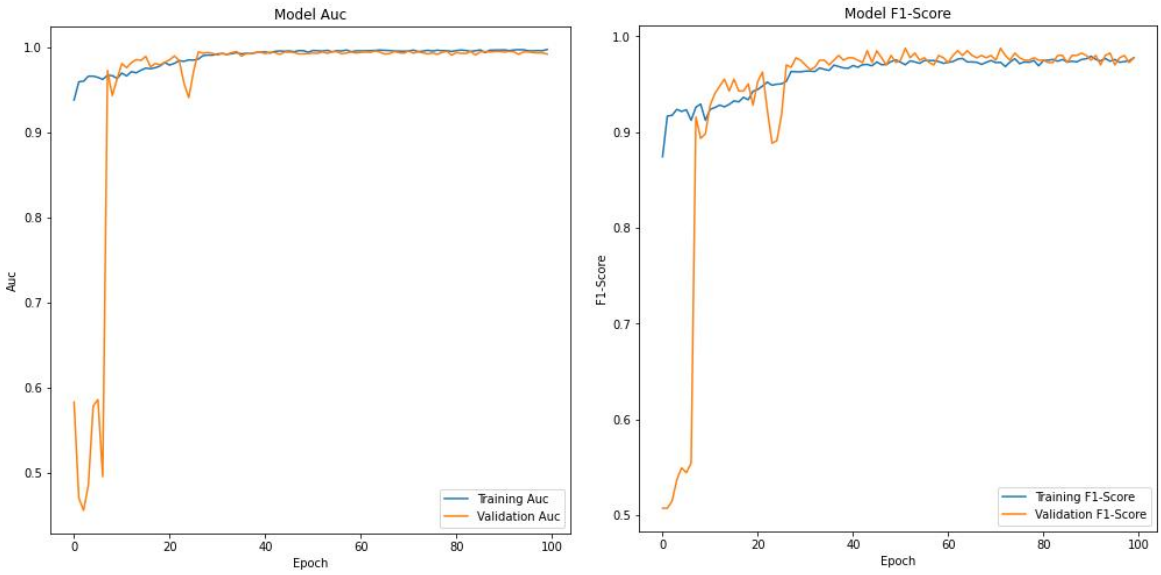

Figure 73: Proposed AUC and F1-Score Trend, Customized Loss and Accuracy

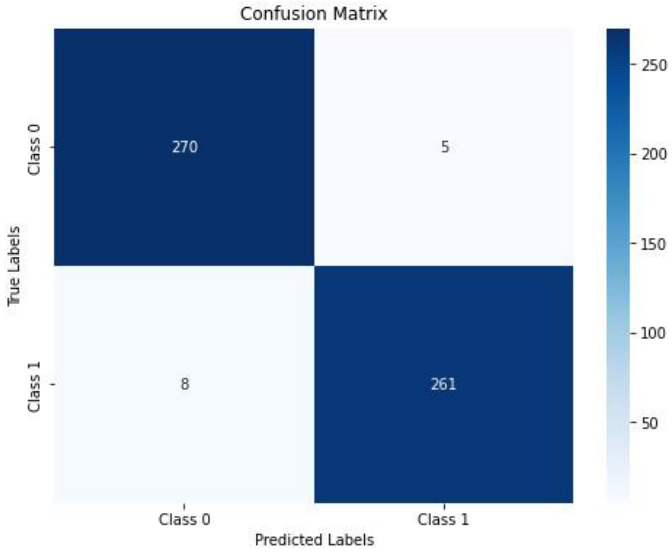

Figure 74: Final Confusion Matrix

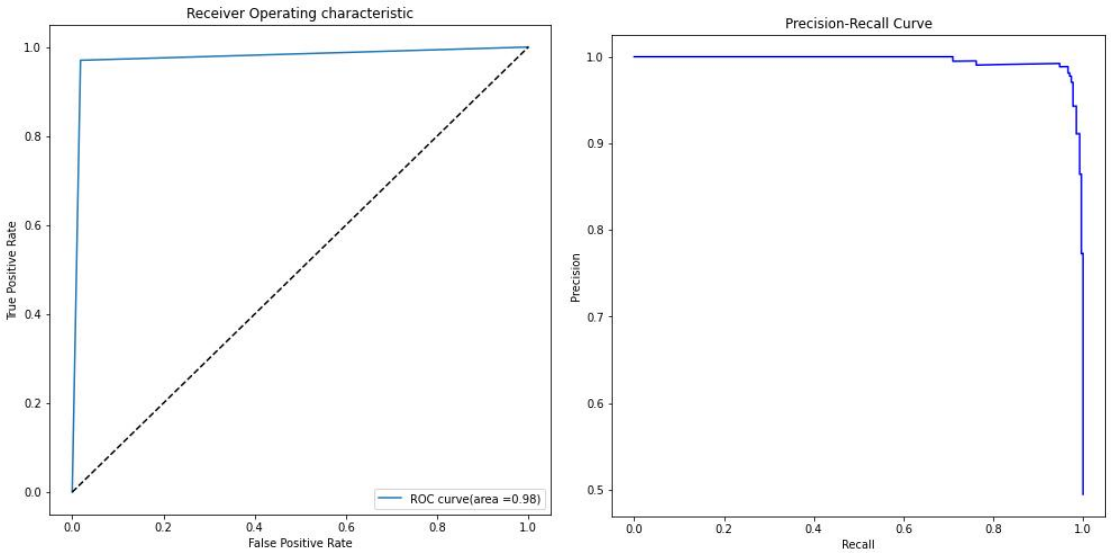

Figure 75: Final ROC and Precision-Recall

```
sensitivity: 0.9702602230483272
specificity: 0.9818181818181818
```

Figure 76: Sensitivity and Specificity

```
Test loss: 0.0828901156783104
Test accuracy: 0.9774305820465088
Test precision: 0.9763636589050293
Test recall: 0.9763636589050293
Test AUC: 0.9941454529762268
Test F1 Score: 0.9763636589050293
```

Figure 77: Final Results on Test set

	precision	recall	f1-score	support
0	0.97	0.98	0.98	275
1	0.98	0.97	0.98	269
accuracy			0.98	544
macro avg	0.98	0.98	0.98	544
weighted avg	0.98	0.98	0.98	544

Figure 78: Proposed Classification Report

Additionally, figure 79 provides the graph when kernel initializer is not included in proposed model.

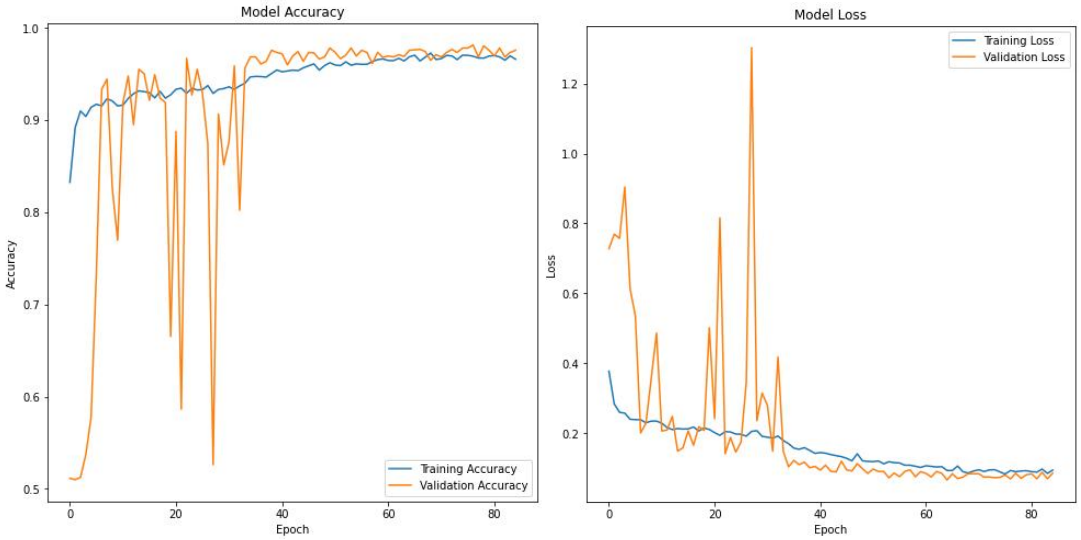

Figure 79: Model without “he_normal” initializer

Based on the hyperparameter adjustments, table 3 compares the different exact value of fine tuning the final model.

Loss&Accuracy	Dropout	Batch size	Val-Loss	Val_Acc	Sensitivity	Specificity	Time/Epoch
Inbuilt	0.4	16	0.0622	0.9802	0.9740	0.9855	54s
Inbuilt	0.5	8	0.0988	0.9752	0.9764	0.9832	59s
Customized	0.3	16	0.1193	0.9615	0.9787	0.9867	36s-42s
Customized	0.3	64	0.0981	0.9800	0.9800	0.9810	43-46s
Final_Proposed	0.3	32	0.0953	0.9800	0.9702	0.9819	35s-40s

Table 3: Modification Comparison

Lastly, Grad-CAM was incorporated to enhance interpretability, with the areas of focus illustrated as follows in figure 80. This technique functions similarly to thermal imaging: the higher the weight value, the more attention is paid to that area, resulting in a redder appearance.

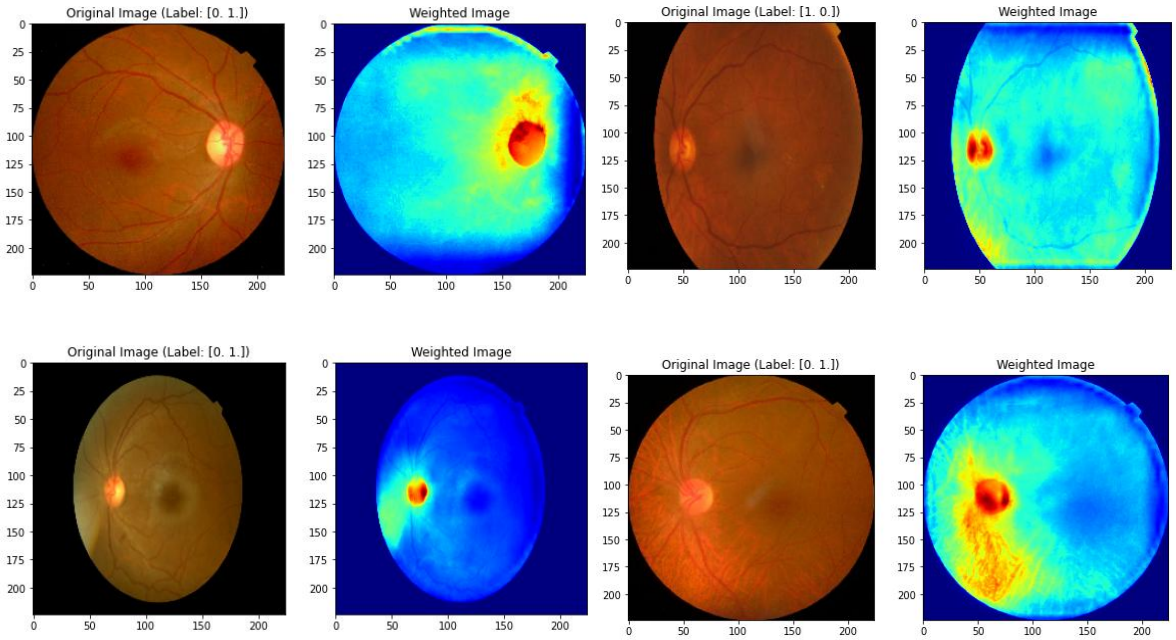

Figure 80: Grad-CAM Images for Visualization

Figure 81 presents the final internal and external structure of the model which concludes the experiments taken during the construction

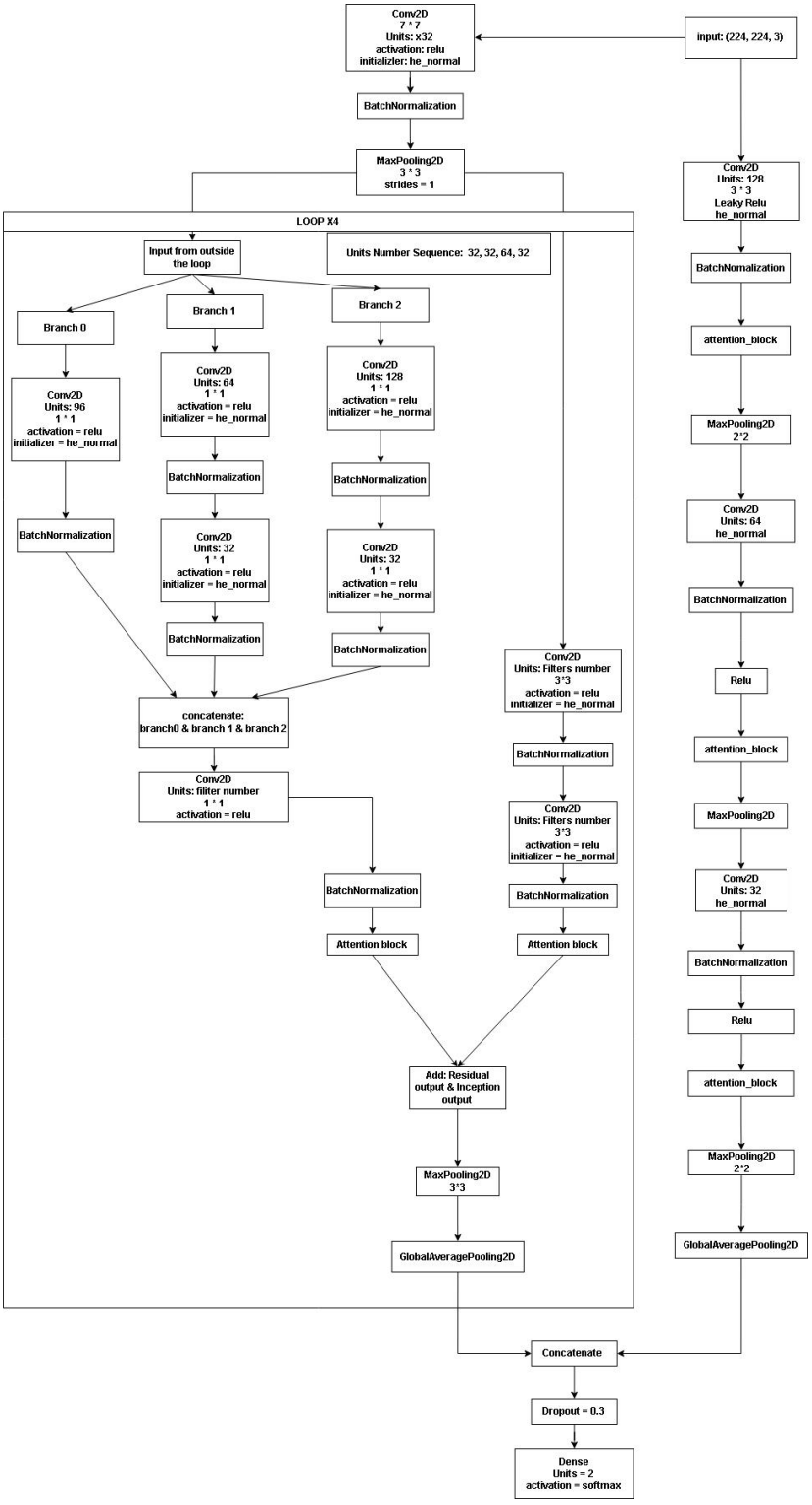

Figure 81: Proposed Internal and External Structure

4.2. Discussion

First of all, based on the equation of the (5) - (9), the results of F1-Score, Precision and Recall are identical when calculating using the micro-average method in binary classification whether it is treated as multi-class or not. Micro-averaging involves summing the individual true positives, false positives and false negatives of the system for different sets and then calculating the metrics. Particularly in this situation, two classes are quite balanced, the sum of true positive for on classes is the sum of true negatives for the other. Furthermore, the accuracy is also align with these values, which is because with Mirco-averaging, the emphasis is on each instance, leading to a unified calculation across all classes that effectively treats both classes as one aggregated singular set, and it displayed the overall evaluation of the model.

Secondly, models involved with using combination of ReLU or Leaky ReLU with "kernel_initializer = 'he_normal'". According to the general understanding, kernel initializer plays a paramount role in setting the initial weights of a neural network in a way that prevents the vanishing or exploding gradient problems, which are common challenges in training deep networks such as the proposed one. Based on the idea of [16], the "he_normal" initialization is tailored for rectifiers like ReLU and Leaky ReLU. The advantage of ReLU, as opposed to traditional function similar to Sigmoid, it is asymmetry. As it is shown in figure 82.

This means that the mean response of ReLU is always non-negative, which significantly influences both the theoretical convergence and empirical performance of the model [16]. Moreover, the "he_normal" initializer provides a theoretically sound method for initializing deep models. This is particularly beneficial for training very deep networks.

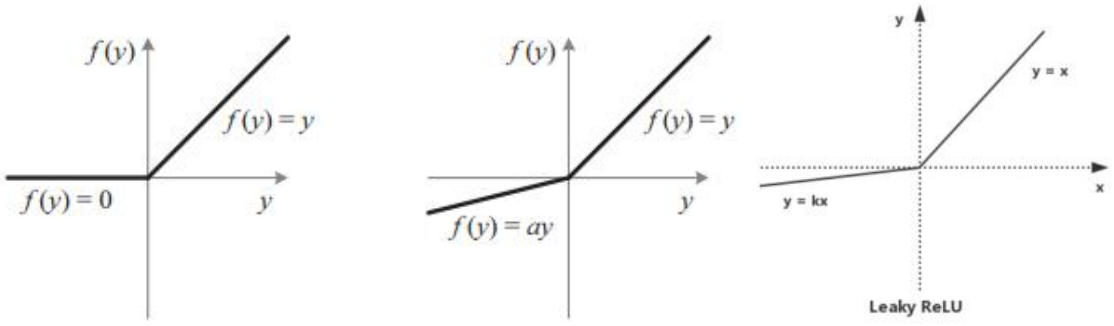

Figure 82: ReLU, PReLU and Leaky ReLU Function [16]-[17]

According to figure 78, it is extreme obvious that without the “he_normal” as the initializer to set the initial weight, the model showed evident fluctuation throughout the first half rounds, and even revealing signs of underfitting which is far worse than the proposed one with kernel_initializer incorporate with rectifiers. And this is the reason kernel_initializer was implemented from the start.

From the beginning of the model construction to the proposed model, a numbers of the steps and versatile of methods were applied. Following will be analyzing all the statistics of the model and the reason why each steps were taken.

For the individual convolutional network, the overall accuracy was increasing and nearly reached 97% with validation a bit higher than training accuracy. Confusion matrix were displaying a high rate of correctness with only 32 samples predicted incorrectly. According to figure 12, specificity was lower than sensitivity, which aligned with the confusion matrix and thus showing minus in recognizing true negative, in this case, true diabetic images. Additionally, the fluctuation stop at around 50^{th} epoch. These all indicates that the model has a level of underfitting, whether the smallness of batch size or the simplicity of the model. However, the trend shows that the model will be performing with a acceptable instability after optimization. Moreover, models are required to be simple while operating effectively to reach a high performance within the hardware situation, and therefore, the model is applied later.

For the second and third models, as shown in figure 17 and figure 25, which represent the Inception and Residual networks respectively, there are instances where certain model structures do not perform well on specific datasets. This suggests a limitation in their compatibility or effectiveness with those particular types of data. And this was the reason to implement these two networks so as to exam the compatibility. With the provided results in figure 18 and figure 26, Inception Network had the accuracy around 95%, though with validation higher than training which implies underfitting, but it is likely the reason the number of filters in each branch are quite small considering the fact that parameter reduction was used to simplify the model, confusion matrix and specificity also illustrated the lack of ability to distinguish the diabetic class. ResNet on the other hand provides a excellent results with stable validation accuracy and low in loss. Furthermore, it is able to classify both classes accurately based on figure 27 and 28. These factors demonstrated that residual network could address the problems once integrated.

Building on these observations, combinations of individual models were tested with and enhancement through the integration of attention mechanism. Incorporating a relatively simple convolutional block with an attention mechanism enhances the complexity of feature extraction which partly resolves the deficiency caused by the simplicity of model. It also maintains the model's efficiency and a modest computational parameter count.

The idea of combining Residual Network with Inception module draws inspiration from the Inception-ResNetV2 model. Firstly, each module contains branched convolutional layers, enhancing breadth to capture a variety of features. These features are then aggregated, enabling the extraction of wider range of feature types. This method differs from continuously abstracting an image after extracting basic features, which can lead to a reduction in the variety of them and potentially dominant low-level convolution in modes precision.

However, implementing branched convolutions also significantly increases the networks depth and complexity. This can lead to issues like network degradation and gradient vanishing. The introduction of residual blocks effectively addresses these challenges by preventing gradient disappearance. The results of the singular ResNet-Inception Model stands with the theoretical estimation where sensitivity and specificity reached around 0.98, implying sufficient capability and compatibility in this certain dataset.

Moving on to the total concatenation of both Attention-Convolutional Model and Attention-ResNet-Inception Model. It observable that after adding attention mechanism and combined the two models together using concatenate fusion methods, the results became better, performance on the dataset increased and stability had also been optimized as well with only few fluctuations occurred. Furthermore, the fitting ability on the dataset had became better. Reason for using concatenate fusion instead of others like voting or average is that concatenate fusion possesses flexibility and extendability, while provide robustness and strengthen the capability of model.

The experimentation with two data preprocessing techniques, CLAHE (Contrast Limited Adaptive Histogram Equalization) and Grayscale image transition, resulted in diminished performance compared to using raw images. This outcome suggests that these methods not only negatively impacted the stability of the model but also adversely affected certain key performance indicators. Specifically, the application of CLAHE and Grayscale processing, which alters the original color distribution and intensity of the images, may have inadvertently obscured vital color-related information. This finding underscores the importance of color as a critical feature in the diagnosis of Diabetic Retinopathy (DR), even though to the naked eye, the

color differences in retinal images may appear subtle or indistinct. It indicates that the model relies significantly on color cues to discern between normal and diabetic conditions, highlighting the inherent complexity and nuanced nature of medical image analysis in DR detection.

After testing and selecting the proper pre-processing method which is raw image with ImageDataGenerator augmentation, model was fine-tuned at the last stage according to table 3.

By testing on inbuilt first then moved to customized loss function, the final hyperparameter was set with dropout rate and batch size equals to 0.3 and 32 respectively. And the training time was around 35s per epoch on this device.

The last part shows the region the model paid most attention to with the use of Grad-CAM. It employs gradient information from the last convolutional layer of the CNN model to get weights of each pixels by fetching backwards from the last layer [13]. In this case, the weight value is the average of the last layer of straightforward convolutional block and Inception-ResNet block.

4.3 Fair comparison With other Deep Learning Models

In this part, the coursework compares the proposed model with three existing pre-trained models on the same dataset so as to provide completeness of the evaluation. The three models are VGG16, ResNet50V2 and InceptionV3.

The condition for the following are all the same, with only difference on input shape in order to match the model's input.

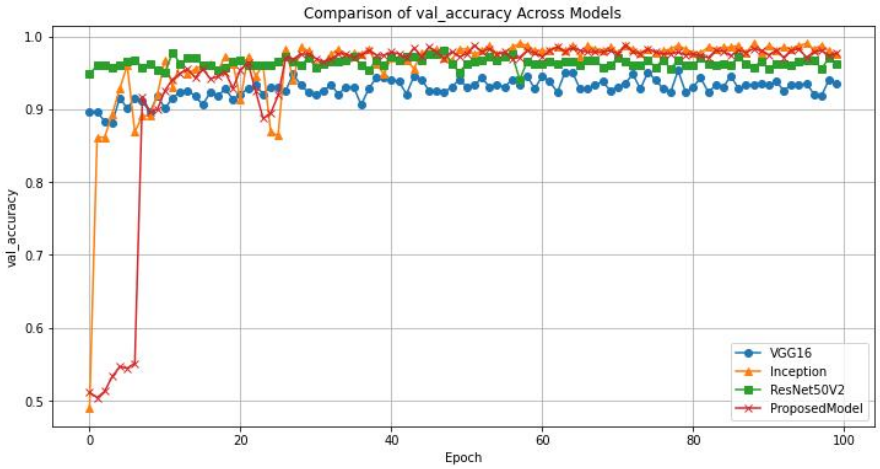

Figure 83: Validation-Accuracy Comparison

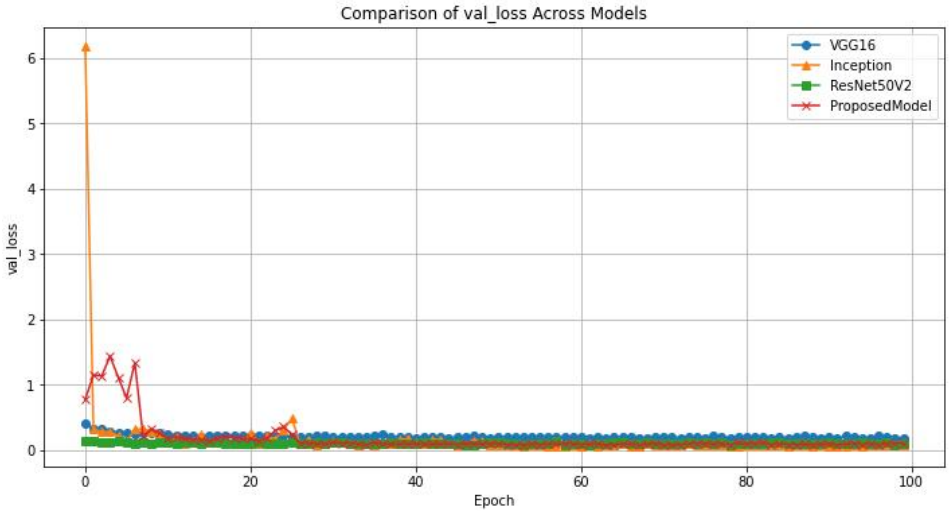

Figure 84: Validation-Loss Comparison

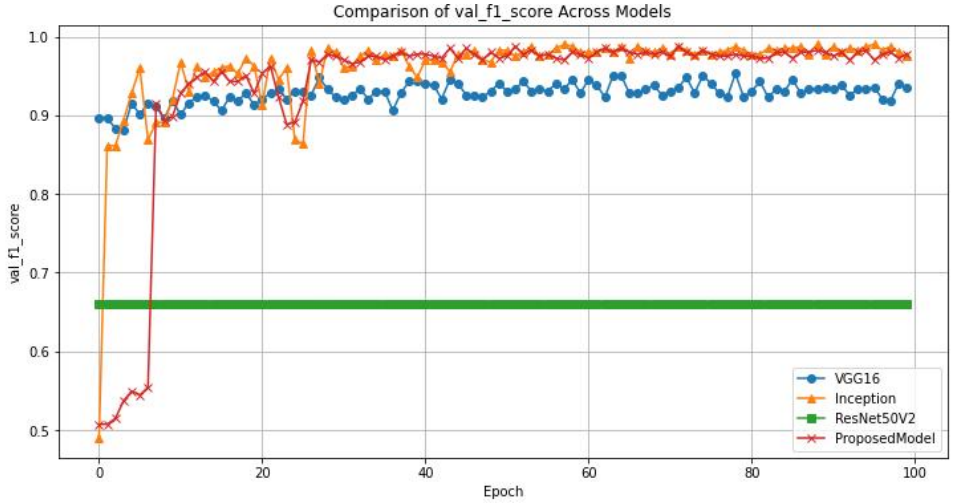

Figure 85: Validation-F1-Score Comparison

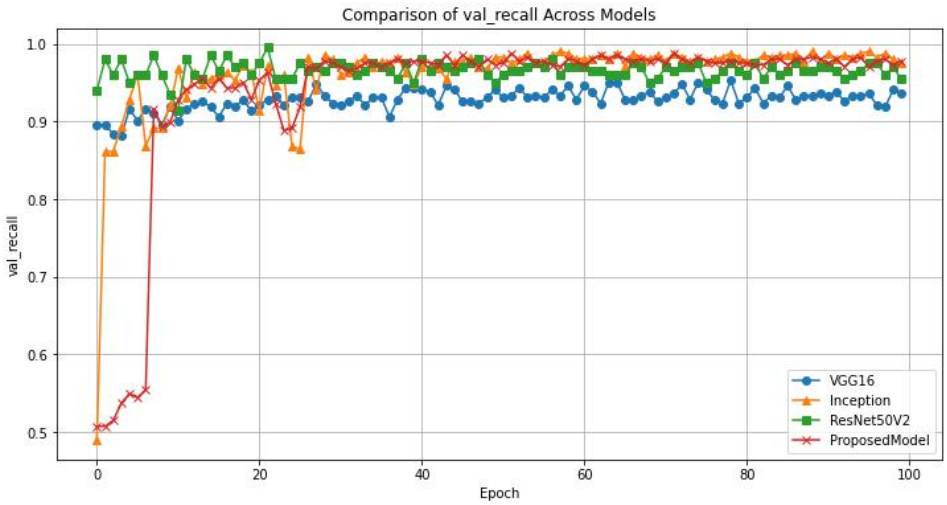

Figure 86: Validation-Recall Comparison

Model	Loss&Acc	Weights	Batch size	Val-Loss	Val_Acc	Sensitivity	Specificity
VGG16	Inbuilt	ImageNet	32	0.1817	0.9406	0.97	0.85
ResNet50V2	Inbuilt	ImageNet	32	0.0910	0.9629	0.98	0.89
InceptionV3	Inbuilt	ImageNet	32	0.0741	0.9752	0.98	0.97
Final_Proposed	Customized	Proposed	32	0.0953	0.9769	0.97	0.98

Table 4: Comparison Among Pre-Trained Models

Based on the graphs plotted from the test results, all four models – including the proposed model, VGG16, InceptionV3, and ResNet50V2 – demonstrated excellent performance on the APTOS Diabetic Retinopathy (DR) dataset, highlighting their high capability and compatibility for binary classification. Each model achieved impressive validation accuracies exceeding 90% and maintained relatively low loss rates around 6%. However, a notable observation was made regarding the Validation F1-Score: ResNet50V2 consistently showed an F1-Score around 66%, which did not exhibit significant variation.

Upon further examination, it was confirmed that this was not due to a technical error. The dataset used was relatively balanced, appropriate default threshold values were set, and the complexity of the models was aptly suited for the task. Additionally, all models utilized techniques for automatic adjustment of learning rates, ensuring minimal interference from

potential image noise. The other metrics of ResNet50V2 were high, ruling out overfitting as a cause for the stagnant F1-Score.

The most plausible explanation for this anomaly seems to be the compatibility of ResNet50V2 with this specific dataset. The constant F1-Score might indicate that ResNet50V2, while effective in general, may not align as well with the nuances of the APTOS dataset compared to the other models. This suggests a relative disadvantage in certain aspects when using ResNet50V2 for this particular dataset, potentially indicating that it is not the most suitable model for this specific application.

In this comparative test, while all models generally performed well, a few indicators hinted at possible overfitting or mismatch, suggesting varying levels of model compatibility. The proposed model, in this context, showed comparable or superior performance to the other models, underlining its effectiveness and potential suitability for the task at hand.

4.4 Indirect comparison With Existing Literature

This part, course conduct a indirect comparison among a number of publications in DR classification to display the capability of the model while showing possible flaws through comparing. Table 5 shows the details

Author	Model	Acc	Loss	F1-Score	Sensitivity	Specificity	Precision	Recall	AUC
Nahiduzzaman et al [1]	CNN-SVD with ELM Algorithm	Binary:99.73% Five: 98.09%	X	100%	X	X	100%	Binary: 100% Five: 96.26%	100%
Saeed et al [2]	ResNetGB	Binary: 99.72%(Eye PACS) Binary: 99.54% (Messidor)	X	X	Binary: 96.04% Binary: 97.32%	Binary: 99.81% Binary: 99.26%	X	X	0.98
Zang et al [3]	DcardNet	Binary: 94.2%	X	X	96.0%	90.5%	X	X	X
AL-Antary and Afara	MSA-Net	87.5%	X	76.7%	90.6%	78.7%	X	X	X

[4]									
Khan et al [5]	VGG-NiN	85%	X	59.6%	X	91%	67%	55.6%	X
Mustafa et al [6]	CNN-Based with Adaboost and random forest	95.58%	X	77%	X	X	Five-Class Average 74.6%	Five-Class Average 81.2%	X
Ali et al [7]	Inception-ResNet-CNN	96.85%	X	98.65%.	99.28%	98.92%	96.46%	X	X
Raiaan et al [8]	ResNet 10	99.46%	X	98.65%	X	99.67%	98.65%	98.66%	X
Chen et al [9]	Integration of Multi-shallow CNNs	Overall Acc: 88.2%	Overal Below 18% based on graph	X	X	X	X	X	X
Farag et al [10]	CBAM&DenseNet169 Encoder	97%	X	X	97%	98.3%	X	X	X
He et al.[11]	CABNet	93.1%	X	91.5%	X	X	92.9%	90.2%	96.9%
Final_Proposed	AIRSC	97.69%	0.953	97.77%	97.02%	98.18%	97.77%	97.77%	98%

Table 5: Indirect Comparison(AIRSC stands for Attention Inception-ResNet-Straightforward Convolution)

For Nahiduczzam et al [1] who proposed a CNN-SVD with extreme machine learning algorithm, got the results of 99.7% and 98.09% accuracy on both Five classes and binary classification, while possesses 100% on F1-Score, precision, AUC and recall of binary classification.This illustrates the ability for generalization on different mode of dataset.Though the results this coursework had obtained is similar towards his, it is still limited on multi-classes classifications, implying that it is lacking generalization on various modes of datasets. Saeed et al [2] proposed

the model of ResNetGB that got high accuracy on both datasets, sensitivity and specificity are also ideal in both datasets which is similar to this work's results.

For most of the work within the comparison, the results of proposed model are either better or at least equal to theirs. In this case, it is considered that the proposed model shows a decent reliability and capability in dealing with DR classification. Especially the model from Ali et al [7] is quite alike to the proposed one and it also got high performance on DR classification.

5. Conclusion

In this report, a novel Convolutional Neural Network (CNN) model is introduced, which uniquely combines a simple yet effective convolutional framework with an advanced Inception-ResNet block, augmented by an attention mechanism. This ensemble model has shown remarkable proficiency in diagnosing diabetic retinopathy, evidenced by its high accuracy rate of approximately 98%, a loss maintained at or below 10%, and other key metrics surpassing the 95% threshold. The model's architecture is adept at feature extraction through its attention-based convolutional approach, while the Inception-ResNet block, empowered with attention mechanisms, is particularly effective in identifying and focusing on critical and diverse features within the retinal images. By integrating both the compactness of small convolutional blocks and the complexity of the Inception-ResNet structure, the model excels in pinpointing essential characteristics crucial for accurate disease categorization. This innovative design is strategically developed to enhance the diagnostic process for diabetic retinopathy, showcasing its specialized application in this critical field of medical research.

However, despite these promising statistical outcomes, the model faces certain limitations. Presently, it is specifically configured for a singular dataset, which doesn't adequately represent the complexities and imbalances often encountered in real-world clinical data. Such imbalances, prevalent in diverse and non-ideal datasets, present a significant challenge to the model's ability to generalize effectively. Furthermore, the model's current structure, not being originally designed for handling unbalanced data, might not be ideally suited for broader categorical classification tasks. For future development, enhancing the model's adaptability to a variety of complex, imbalanced datasets, such as those seen in related works (like the Messidor 2-class and Messidor 5-class datasets), becomes a priority. Addressing these limitations is not only crucial for increasing the model's practical applicability but also for ensuring its effectiveness and reliability in diverse and realistic clinical settings. Such advancements will be key in

realizing the full potential of this model in the broader context of medical image analysis and disease diagnosis.

References

[1] Md. Nahiduzzaman, Md. R. Islam, S. M. R. Islam, Md. O. F. Goni, Md. S. Anower, and K.-S. Kwak, "Hybrid CNN-SVD Based Prominent Feature Extraction and Selection for Grading Diabetic Retinopathy Using Extreme Learning Machine Algorithm," *IEEE Access*, vol. 9, pp. 152261–152274, 2021, doi:https://doi.org/10.1109/access.2021.3125791.

[2] F. Saeed, M. Hussain, and H. A. Aboalsamh, "Automatic Diabetic Retinopathy Diagnosis Using Adaptive Fine-Tuned Convolutional Neural Network," *IEEE Access*, vol. 9, pp. 41344–41359, 2021, doi: https://doi.org/10.1109/access.2021.3065273.

[3] P. Zang *et al.*, "DcardNet: Diabetic Retinopathy Classification at Multiple Levels Based on Structural and Angiographic Optical Coherence Tomography," vol. 68, no. 6, pp. 1859–1870, Jun. 2021, doi: https://doi.org/10.1109/tbme.2020.3027231.

[4] M. T. Al-Antary and Y. Arafa, "Multi-Scale Attention Network for Diabetic Retinopathy Classification," *IEEE Access*, vol. 9, pp. 54190–54200, 2021, doi: https://doi.org/10.1109/access.2021.3070685.

[5] Z. Khan *et al.*, "Diabetic Retinopathy Detection Using VGG-NIN a Deep Learning Architecture," *IEEE Access*, vol. 9, pp. 61408–61416, 2021, doi: https://doi.org/10.1109/access.2021.3074422.

[6] H. Mustafa, Syed Ahmad Ali, M. Bilal, and Muhammad Shehzad Hanif, "Multi-Stream Deep Neural Network for Diabetic Retinopathy Severity Classification Under a Boosting Framework," *IEEE Access*, vol. 10, pp. 113172–113183, Jan. 2022, doi: https://doi.org/10.1109/access.2022.3217216.

[7] G. Ali, Aqsa Dastgir, Muhammad Waseem Iqbal, M. Anwar, and M. Faheem, "A Hybrid Convolutional Neural Network Model for Automatic Diabetic Retinopathy Classification From Fundus Images," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 11, pp. 341–350, Jan. 2023, doi: https://doi.org/10.1109/jtehm.2023.3282104.

[8] M. Azam et al., "A Lightweight Robust Deep Learning Model Gained High Accuracy in Classifying a Wide Range of Diabetic Retinopathy Images," IEEE Access, vol. 11, pp. 42361–42388, Jan. 2023, doi: https://doi.org/10.1109/access.2023.3272228.

[9] W. Chen, B. Yang, J. Li, and J. Wang, "An Approach to Detecting Diabetic Retinopathy Based on Integrated Shallow Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 178552–178562, 2020, doi: https://doi.org/10.1109/access.2020.3027794.

[10] M. M. Farag, M. Fouad, and A. T. Abdel-Hamid, "Automatic Severity Classification of Diabetic Retinopathy Based on DenseNet and Convolutional Block Attention Module," *IEEE Access*, vol. 10, pp. 38299–38308, 2022, doi: https://doi.org/10.1109/access.2022.3165193.

[11] A. He, T. Li, N. Li, K. Wang, and H. Fu, “CABNet: Category Attention Block for Imbalanced Diabetic Retinopathy Grading,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 143–153, Jan. 2021, doi: https://doi.org/10.1109/tmi.2020.3023463.

[12] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul, “Grayscale Image Colorization Methods: Overview and Evaluation,” *IEEE Access*, vol. 9, pp. 113326–113346, 2021, doi: https://doi.org/10.1109/access.2021.3104515.

[13] Anjar Wanto, Yuhandri Yuhandri, and Okfalisa Okfalisa, “Optimization Accuracy of CNN Model by Utilizing CLAHE Parameters in Image Classification Problems,” Aug. 2023, doi: https://doi.org/10.1109/iconnect56593.2023.10327100.

[14] Tasmim *et al.*, “A Lightweight-CNN Model for Efficient Lung Cancer Detection and Grad-CAM Visualization,” Sep. 2023, doi: https://doi.org/10.1109/icict4sd59951.2023.10303569.

[15] Q. Shen, H. Zhang, and C. Liao, “EwLoss:A Exponential Weighted Loss Function for Knowledge Graph Embedding Models,” *IEEE Access*, vol. 11, pp. 110670–110680, Jan. 2023, doi: https://doi.org/10.1109/access.2023.3322204.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, doi: https://doi.org/10.1109/iccv.2015.123.

[17] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, “Reluplex made more practical: Leaky ReLU,” *2020 IEEE Symposium on Computers and Communications (ISCC)*, Jul. 2020, doi: https://doi.org/10.1109/iscc50000.2020.9219587.