Implementation of Explainable AI for detection of Skin Cancer using Image Processing leveraging Grad-CAM

by

Fardeen Bin Ibrahim
20201066
Mashnoon Mayad
20201033
Samia Tasnim Orpa
20201115
Oishee Fatima
24341090

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Fardeen Bin Ibrahim	Samia Tasnim Orpa
20201066	20201115
Mashnoon Mayad	Oishee Fatima
20201033	24341090

Approval

The thesis/project titled "Implementation of Explainable AI for detection of Skin Cancer using Image Processing leveraging Grad-CAM" submitted by

1. Mashnoon Mayad (20201033)
2. Fardeen Bin Ibrahim (20201066)
3. Samia Tasnim Orpa (20201115)
4. Oishee Fatima (24341090)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 7 February, 2025.

Examining Committee:

Supervisor:
(Member)

Mr. Rafeed Rahman

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Mr. Tanzim Reza

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

In the ever-evolving landscape of medical diagnostics and treatment, the early detection and precise classification of skin cancer is crucial for effective and successful treatment outcomes. Using deep learning models in cancer diagnostics have been proven to be remarkably successful and transformative, and in some cases, computer-aided diagnosis (CAD) has even been observed to outperform conventional diagnosis techniques and human expertise. However, the complexity of artificial intelligence (AI) and machine learning models used for such diagnosis often leads to a lack of transparency in decision-making, and hence, adversely affects the confidence of medical professionals as well as patients in computer-aided diagnosis. Acknowledging the vital need for transparency in the process of CAD, our study explores the application of Explainable AI techniques in the detection of skin cancer. Using transformer models like Vision Transformer and Swin Transformer and deep Convolutional Neural Networks (CNN) based learning architectures such as VGG-16 and ResNet-50, and by integrating Grad-CAM with image processing, we aim to enhance the techniques of skin lesion classification. Grad-CAM provides a visual explanation of the model prediction allowing for medical professionals to judge the decision-making and prediction process behind the AI's assessments and predictions. Consequently, our research aims to improve the trustworthiness of the black-box AI models in the field of medical diagnostics. Using the knowledge of the aforementioned models, we have developed our own hybrid model, AResNN (Attention-based Residual Convolutional Neural Network), that incorporates attention mechanisms and skip connections to improve accuracy and computational efficiency. We leveraged the power of transfer learning to extract maximum performance from our model. Furthermore, Grad-CAM has been applied to provide visual explainability of our model. Ultimately, our study intends to close the rift between complex AI models and their medical application in skin cancer diagnosis by making the models more transparent.

Keywords: Image Processing, Explainable AI, Deep Learning, Neural Networks, AResNN, CNN, Skin Cancer, Grad-CAM, Residual, Attention, Transfer Learning, Transformer

Acknowledgement

Firstly, all praise to The Almighty Allah for whom our thesis has been completed without any major interruptions.
Secondly, to our supervisor Mr. Rafeed Rahman Sir and co-supervisors Mr. Tanzim Reza Sir and Dr. Md. Golam Rabiul Alam Sir for their kind support and advice in our work. Finally, to our parents as without their support and sincere prayers it may not have been possible.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ADASYN$ Adaptive Synthetic

API Application Programming Interface

$AResNN$ Attention-Based Residual Convolutional Neural Networks

CAD Computer Aided Diagnosis

CAM Class Activation Mapping

CCE Categorical Cross Entropy

$CLAHE$ Contrast Limited Adaptive Histogram Equalization

CNN Convolutional Neural Network

$DCNN$ Deep Convolutional Neural Network

DL Deep Learning

ENN Edited Nearest Neighbor

GAN Generative Adversarial Networks

$Grad-CAM$ Gradient-Weighted Class Activation Mapping

MHA Multi Headed Attention

ML Machine Learning

$ResNet$ Residual Networks

$SMOTE$ Synthetic Minority Oversampling Technique

SVM Support Vector Machine

TL Transfer Learning

$VGG16$ Visual Geometry Group 16

Chapter 1

Introduction

In the light of the rising number of new cases diagnosed every year, cancer has become one of the more prominent ailments afflicting humanity and among the various types, skin cancer is the most common. Therefore, an early detection of it could mean higher chances for effective treatment and improved survival rates. In recent years, medical image analysis using computer vision systems, image processing techniques, and deep learning models has seen significant progress with substantial improvements in accuracy in identifying and classifying skin lesions. Although effective medical devices do exist to cater such assessments and diagnosis, they are integrally dependent on human skills and expertise which, thereby, means that they are prone to human error and require expensive resources. It can also be highly time consuming, as it heavily relies on the expertise of the healthcare professionals. The use of artificial intelligence and deep learning models in medical diagnosis have been proven to be remarkably successful and transformative, and in many cases, computer-aided diagnosis has been observed to outperform traditional diagnosis techniques. Advanced deep learning algorithms such as CNNs, VGG-16 and ResNet-50 are able to efficiently predict and classify whether a skin abnormality is malignant or benign from just an image. Similarly, transformer models such as Vision Transformer (ViT) and Swin Transformer are another efficient approach used in skin lesion detection and classification. We have also produced our own custom model, AResNN (Attention-based Residual Convolutional Neural Network), that utilized groundbreaking features of the aforementioned models to produce a hybrid model that could compete with the well established models in terms of performance and results for skin cancer classification. These models could be incredibly useful in the medical field serving as a quick and primary precautionary method for diagnosing skin cancer, which can be further examined and confirmed by a medical professional.

1.1 Ethical Concerns

Although the use of CAD in skin cancer detection and classification offers significant potential to enhance accuracy of diagnosis and early intervention, the reliability of such technology is often brought into question because there is always a chance of encountering incorrect localizations, inaccurate predictions, and false positives or negatives. If medical professionals are extensively reliant on CAD systems, there is a chance that they might overlook subtle signs missed by the system, which in

turn could result in false negatives or a missed diagnosis. Meanwhile, if not finely tuned, CAD systems could identify abnormalities which are not medically significant and lead to false positives. While a false positive would lead to unnecessary procedures and invasive treatments along with psychological distress for patients, a false negative could delay diagnosis and treatment and result in worsened and often irreversible health outcomes. Furthermore, deep learning models used in CAD are often relatively complex and function as black-boxes, meaning that their decision-making processes are not easily interpretable. Although most AI models are designed to highlight key features in medical images and provide confidence scores, the underlying reasoning and logic behind their predictions are often too complex and difficult to understand. Hence, medical professionals may face challenges due to lack of explainability, low interpretability, and variability in performance. Moreover, patients should be fully informed about the role of CAD in their diagnosis and understand how predictions and diagnosis of CAD systems require expert interpretation. With human wellness at stake, transparency of CAD systems and explanation of deep learning models' chain-of-thought is crucial to avoid any mistakes which may lead to misdiagnosis of a disease. Recognizing this concern, our research delves into how explainable AI techniques help us understand the layout of a model's chain-of-thought in order to assess the reliability and accuracy of the model's assessments and predictions.

1.2 Research Purpose

Recognizing the ethical concerns related to CAD systems in the medical field, especially skin cancer diagnosis, our research delves into how explainable AI techniques help us understand the layout of a model's chain-of-thought in order to assess the reliability and accuracy of the model's assessments and predictions. This would in result bridge the gap of using AI for skin cancer detection in the medical field by building trust with medical professionals. To demonstrate the strength of deep learning classification models in identifying and classifying various types of skin cancer only based on images of skin lesions or skin abnormalities, our research explores multiple types of Convolutional Neural Networks with data augmentation: VGG-16, ResNet-50, and our proposed model AResNN. Our modified CNN algorithm allows for a robust image processing solution. AResNN is a hybrid model that incorporates attention mechanisms, which we can find in transformer models, and skip connections over convolutional layers, which has been first explicitly introduced to us through the ResNet models. By applying the transfer learning methodology to a large skin cancer dataset, which we then transferred the weights onto our smaller dataset, our model was allowed to learn intricate features of different skin lesions. We have also used other CNN-based architectures. VGG-16, considered to be an all-rounder model, allows for effective image classification and easy object detection while being computationally light and efficient. Furthermore, ResNet-50, a powerful and deep version of CNN, uses the concept of residual units which reduces the difficulty of network convergence and increases computational efficiency in the process of image classification. Transformer models such as ViT and Swin Transformer provide an alternative to the CNN-based learning approach by focusing on the global context of features inside the images instead of only localized features

like in CNN models. Transformers are also a branch of neural network architecture with self-attention layers that transform the provided input sequence into an output sequence by learning the context and the meaning via tracing all the relationships in a sequential data. Following classification and detection, comes the most important step, introducing explainability of the models' results through the integration of Gradient-weighted Class Activation Mapping (Grad-CAM) with the image processing models. Explainability can be described as the ability of a model to explain the chain-of-thought it uses to reach its decisions. Grad-CAM is effective in providing a clear visualization of where and on what the models are focusing on during the process of their decision-making and reaching their predictions. This visualization helps us understand a model's behavior and enhances a model's interpretability, which is highly useful when dissecting a model's decision. The visual explanation of the model's prediction allows for medical professionals to judge the decision-making and prediction of CAD systems and AI assessments.

1.3 Contributions

In this work, we design and analyze a novel hybrid deep learning model, AResNN, that offers learning of intricate patterns based on attention mechanisms while also making sure that it is computationally efficient and lightweight compared to state-of-the-art models like ResNet50, VGG16 and transformers and that it competes with them in terms of performance and accuracy. In particular, we propose a more robust and flexible model training mechanism which meets the requirements of a CAD system in this field. Therefore, the key contributions of the paper are as follows:

- We made use of class weights in our loss function instead of opting for traditional oversampling techniques for handling data imbalance on the HAM10000 dataset. Doing so allows our model to focus on minority classes based on the weights allocated to them and not just only on the majority class.
- We devised our own architecture incorporating self-attention mechanisms, similar to those found in transformers, to accurately classify different types of skin cancer and diagnose them.
- We also implemented residual blocks to ensure skip connections that meant greater computational efficiency as the number of parameters for model operations decreased significantly.
- To enforce visual explainability that would, in turn, ensure user trust on our model, we applied Grad-CAM and Grad-CAM++ that highlighted areas of the image based on which the model reached its prediction.

1.4 Thesis Organization

In this paper, we delved into the inner workings of the aforementioned models and their effectiveness in image processing for skin cancer classification. In Chapter 2,

we lay out the synthesis of related works in the form of literature review. Chapter 3 contains an elaborate description of the data used to test the models. Furthermore, Chapter 4 dives into the methodology of our research, providing a more detailed understanding of our models. In Chapter 5, we showcase the results obtained from the models and our analysis. And finally, Chapter 6 marks the end of this paper with a discussion regarding the challenges we faced, our future aims and a summarizing conclusion.

Chapter 2

Literature Review

Now, we analyze some of the related works done in our field of study:

The authors in paper [12] have employed the ResNet50 model for their categorization task to identify malignant and benign skin lesions. ResNet50 has been pretrained on the ImageNet dataset. It uses residual networks that tackle the issue of diminishing gradients in deep networks by skipping layers, namely batch normalization and non-linear activation function layers. Through this the model remains stabilized and speeds up the training process. Weights from the upstream layer are not used, instead weights from neighboring layer links are utilized when a layer is learning. Their model draws inspiration from ResNet18 by including identity shortcut connections that allow certain layers to bypass training, forming what are known as residual blocks. Their model starts with a sequence of batch normalization, ReLU activation, and convolutional layers that are repeated N times instead of the typical convolution layer. In some cases they also adjusted the stride of convolution to control the size of the feature map without using pooling layers. After which, a dense layer with softmax activation was used to do the classification task. The model was designed following two objectives: achieving highest possible accuracy, and reaching that accuracy at the least possible number of training epochs. Due to the vanishing gradient problem, error starts increasing, rather than steadily decreasing, during training because of model overfitting. The authors argue that this is resolved efficiently in a Residual Network through the use of skip connections between every 2 layers and also direct links among all layers. Even though ResNet50 is a 50 layer deep network, through stacking residual blocks, it can efficiently learn parameters from the identity function. They achieved an accuracy of 86.66%. One of their biggest limitations was detecting skin cancer in dark skin tones as their dataset lacked such images which led to such a bias.

In paper [4], the authors explore the detection of melanoma and other types of skin cancer through the implementation of Convolutional Neural Networks (CNN) and another CNN-based architecture, ResNet-50. In addition, the paper provides an overview of various deep learning models used in the detection of diverse types of skin diseases and skin abnormalities. Melanoma and non-melanoma are the two broad classes that skin cancer can be classified into, and differences in the skin lesions and skin abnormalities is the primary factor for detection of cancer. The authors used the dataset of SIIM-ISIC Melanoma Classification Competition from

Kaggle, which has 33,126 images in total. In order to increase detection accuracy up to 95%, they have used image data and CSV data. Before running the classification models, the data is first pre-processed and then augmented. Various pre-processing algorithms have been used, such as Sharpening Image, Grey Image, Smooth filter, Median filter, Histogram, YCbCr, Binary Mask etc. For data augmentation, they have used SMOTE, changing image perspective, color gradient generation, region of interest, image enhancement and segmentation, and feature extraction component. These methods are implemented to balance the dataset and also get rid of any unnecessary background features. After pre-processing and augmentation, ResNet and CNN have been implemented for the classification. The model is trained using CNN on both benign and malignant images, 6000 instances for each. CNN is used for the image dataset and XGBoost classifier is used for the CSV dataset. The implemented model, along with several pre-processing algorithms, was observed to achieve significantly higher accuracy when compared to other CNN-based architectures. Although the ResNet model is observed to give the highest accuracy among the mentioned models, it fails to generalize well with the validation set. The authors have further integrated the model to an android application, which allows users to upload images from their cellphones to get a quick identification of any skin abnormality. Overall, this paper emphasizes how appropriate pre-processing methods along with advanced CNNs can improve the detection accuracy of skin cancer.

Similar to paper [4], paper [20] also delves into the application of deep learning with Inception V3 and ResNet-50 algorithms for early and effective detection of melanoma skin cancer. In this paper, the authors used Inception V3 and ResNet-50 algorithms to classify types of skin cancer using a skin cancer dataset from Kaggle consisting of two classes: benign and malignant. The dataset of 3297 images, have been divided into 660 test data and 2637 training data, while maintaining an approximately equal percentage of benign and malignant images in both test and training data. Minor data pre-processing is done by changing the image size to 224 x 224 pixels. In the classification step, ResNet-50 architecture, a 50 layer Residual Network, is used to train the model for the skin image classification. The main differences between the ResNet used in this research and the original architecture are: 1) two classes of outputs are used on the fully connected layer, 2) the activation function at the fully connected layer becomes ReLU. Before advancing to classification, feature extraction of data is done using a combination of convolutional blocks and identify blocks, from which several filter sizes are used such as 1x1, 3x3 and 1x1. After feature extraction, in the classification process, the model enters the data training process using ResNet-50, and here the ReLU function is used. Moving to validation of results, a confusion matrix is used to visualize the model's performance in classifying the data, which contains four main matrices: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Furthermore, the results of the classification are further elaborated using a detailed model summary. The summary shows that the ResNet-50 algorithm follows a workflow which consists of various important image processing steps. Starting with ZeroPadding2D operation used to expand image size, followed by a Conv2D layer with 64 filters to extract initial features. Then, to introduce non-linearity and accelerate model convergence, ReLU activation function and Batch Normalization are applied. To extract important features from the images, ZeroPadding2D and MaxPooling2D operations are

performed. Then through the repetition of residual blocks with convolutional layers in each block along with addition of shortcut connections, flow of information is strengthened and the vanishing gradient problem is addressed. Lastly, the final convolutional layer generates a feature vector for image classification. From the accuracy and loss values, it is observed that model's accuracy tends to increase and loss tends to decrease as the number of epochs are increased. With epoch 20, the model achieves an accuracy of 0.87 and a loss of 0.28. The main novelty of this research is the modifications to the ResNet-50 architecture and used hyperparameter settings.

The authors in paper [10] had experimented with using a pre-trained Visual Geometry Group 16 (VGG16) architecture model on a Skin Cancer Malignant dataset. They utilized Transfer Learning with the pre-trained VGG16 model by taking the learned weights and parameters to apply them to a new model to suit their classification task. Their image preprocessing included removal of hair in images, reducing noise, contrast enhancement and resizing the image. Their VGG16 model consisted of convolutional blocks that extracted features from the input image. The fully connected layers were then used to make predictions using those features. Lastly, the last output layer replaces a typical softmax layer with a layer containing two nodes - one for malignant and other for benign. This new layer was then trained using weights and parameters from previous layers. Their model achieved a high accuracy with few parameters indicating how powerful the VGG16 model truly is. Researchers used backpropagation to determine the weights of their new output layers in their model. Transfer learning proved advantageous to achieve high accuracy with comparatively few parameters as the network already has started to gain knowledge of the features of skin lesions. Transfer learning on VGG16 proved to also tackle the issue of over fitting. By the end of training their model, both training and validation accuracy converges at about 84.2% proving to be an effective skin cancer classification model.

Similarly, in paper [16], the authors also applied Transfer Learning on a pretrained VGG16 model. However, their main focus was to find the importance of image quality instead of quantity, like usual, in enhancing diagnostic accuracy. VGG16 model has shown significant performances in detecting skin cancer in multiple cases due to its ability to detect minuscule differences in skin lesions making it effective for early diagnosis. The authors plan was to train the VGG16 model with the same hyperparameters for 3 different sets of data all deriving from the same preprocessed HAM 10,000 dataset but only depending on the quality of image while the quantity of images slightly deferred. They applied Transfer Learning by removing the dense classification layers that were trained on ImageNet and replaced with custom layers. The custom layers contained another convolutional block and dense layer that they adjusted to their specific classification task. These custom layers were then trained according to their specific task. In their testing phase, they chose the model with highest validation accuracy which was model 2 that had a relatively smaller dataset with higher quality images. The quality of images were improved through multiple unmentioned preprocessing techniques. Their highest validation accuracy was 94% while its testing accuracy came to an impressive 84.5%. Thus, they came to conclude that in the medical field, quality of images is far more important than quantity of images.

Out of the many various types of malignant skin cancers, Melanoma is the deadliest variation of the few, with a very high mortality rate every year. The authors aim with this study [1] was to analyze lesion images, and extract parameters, such as size, shape, border and color for segmentation and classification using Image Processing tools, for detection of Melanoma. 75% of all skin cancer deaths come from melanoma, making it the deadliest of all the skin cancers. The later it is detected, the more chance it has to sink deeper and deeper into the skin, until it becomes impossible to treat or worse, spreads to other parts of the body. One of the most common ways to examine lesions suspected of melanoma is through dermoscopy. Dermoscopy and clinical images make up a good amount of the data regarding skin cancer. However, the diagnosis of malignant vs benign lesions are not always an easy task, and in some cases, an automatic diagnostic tool would be much more helpful. A computer aided solution for such a problem can be computer vision based diagnosis,through image extraction, feature extraction and then classification or detection of the said lesion image. The methodology proposed in this study [1] is a simplified version, similar to the method in the aforementioned study, where an image containing a suspected melanoma lesion is inputted into a mode. The image will first need to be preprocessed and edited to increase readability and aid in feature extraction and segmentation later on. The enhanced image is then passed onto an automatic thresholding process through threshold generations in RGB planes. After thresholding, the image is passed onto a segmentation model where the model aids in the creation of binary masks. This will help in the task of detection. The proposed method is by choosing the biggest lesion in the image as the reference lesion. Using edge detection and the masks, the lesion part of the image is cut out and separated, to be further analyzed by another model. This is where the work of the classification model comes in. Using geometric features extracted from the image and other parameters stated before (size, shape ,etc.) the image is classified and thus we end up with a result of the severity and possibility of melanoma from the initial lesion image. The main extracted features extracted from the image are crucial to the classification accuracy of the system. The area and number of edge pixels makes sure that only the lesion is extracted in the segmented part of the image. The Circularity index gives shape uniformity and irregularity indexes help decipher any irregularities. This pipeline is similar to the more advanced one used in [17]. The model is effective at making a simple computer aided diagnostic system for melanoma.the proposed system is suggested to be user friendly and accurate for both patients and physicians, as an alternative to when an expert may not be present.

This study [7] also focuses on the various image processing techniques that can be used in the detection of skin cancer early as it recognizes the importance of it and that diagnosis might take a longer time and is more costly. The authors mention that for image analysis three things are really important: pre-processing, data augmentation and feature extraction and selection. They have also talked about the use of segmentation and its parameters. They have described their method in 3 phases. In phase 1, any background glare, shading and hairs are removed in order to efficiently identify the texture, size, shape and color of the skin lesion. For phase 2, segmentation and feature extraction are performed that allow you to pick

specific regions of interest in an image. Lastly, phase 3 involves implementing a machine learning model like CNN and SVM in this study to train and test the data for generating accurate predictions. They have used the ISIC dataset for melanoma skin cancer which consists of roughly 23.000 images but they have only used a subset of 1,000-1,500 images for training and testing purposes. Hough Transformation Method was used to identify lines, ellipses and round shapes. MATLAB filters were used to deal with shading and glare issues. For this paper, the authors implemented architectures like CNN and SVMs for classification. Even on smaller datasets, SVM seemed to generate higher accuracy and when compared to other classifiers, it was observed that SVM performed exceptionally well in their case. Their proposed model obtained an accuracy of 96.5% in detecting skin cancer during the test phase. The authors conclude by saying that their method would help health technicians to conduct faster and perfect diagnosis for the patients.

The authors of this paper [19] begin by emphasizing the importance of early detection of skin cancer as the cases of it have been on the rise especially in America, Canada and Australia over the centuries. They also mention how the use of computer vision systems in the medical domain have increased in popularity as it highly aids with the diagnosis and detection of a particular disease. For their research purpose before implementation, they went through multiple different accounts of applying various Deep CNN and SVM based architectures on the very popular HAM10000 dataset to illustrate the classification of skin cancer. They have also talked about the use of GANs (Generative Adversarial Network) where they create fake images of skin lesions to compensate for the lack of data in hand and that ultimately led to a remarkable increase in accuracy when trained in a CNN. The recent use of deep learning based architectures has somewhat achieved critical success among researchers. The authors mention about a study that was conducted on the basis of 3000 images that were primarily collected from people suffering from skin conditions to classify as malignant or benign and the model that they used was Xception Network which obtained the highest accuracy of 85.303%. The authors of this paper [19], however, have also divided their data into two categories of malignant and benign, and have implemented image processing techniques and DCNNs. In this study [19], the authors have proposed a model that efficiently classifies 7 types of skin cancer. According to them, their approach requires less GPU allocation than a typical search algorithm that needs to train a model. The dataset which they used was International Skin Imaging Collaboration (ISIC) that contains 2357 images with 9 classes. The data was then preprocessed to reduce noise and improve the overall quality of the images. The goal was to eliminate any background features and focus primarily on the lesions. The authors have also used segmentation to separate their region of interest for this study. Their data was also augmented to compensate for the low number of image data. Based on the region of interest, further features were extracted for inputting them in their model. The authors have implemented a CNN model and a ResNet50 model for their work. For their other model, they used a typical ResNet50 consisting of 50 layers with a combination of 1x1. 3x3 and 5x5 kernels to extract features from input. After training and testing, they represented their results both graphically and in matrix form using Confusion Matrices. The accuracy they obtained using their proposed models were 89.03% and 91.32% while precision, recall and F1 score were 76.16%, 78.15%, and 76.92% respectively. They

concluded that their research will play an important role in this field and they will continue to research on the challenges associated with the detection of more different cases using deep learning and AI-based systems.

The use of CNN has come to be quite popular in the field of cancer detection using computer vision, due to its robustness and efficacy in dealing with predictions, classifications and even segmentation problems. Skin cancer is one of the most common yet deadly forms of cancer diagnosed in humans in recent years. Studies using Probability Neural Networks, Artificial Neural Networks and Convolutional Neural Networks have proved to be useful in the field of skin cancer diagnosis and detection. Normally, Feature shapes and properties are needed to be fed into the training classifier for it to detect the cancers, to that end, the CNN extracts these information from the image. In the proposed study [6], the authors propose an automatic diagnostic and early detection model. This new model proposed is composed of machine learning followed by deep learning, and finally, its implementation. The dataset used in this study [6] is curated from the International Skin Imaging Collaboration or ISIC, a very popular database for skin cancer images.The images were then preprocessed slightly by noise reduction and increase the detection of Regions of Interest. To that end, some of the preprocessing techniques were Fastest Gaussian Blur, for noise reduction. Following that, the colour space is manipulated to change to a YCbCr color space. Furthermore, a simple Fast Fourier Transform is done to complete the preprocessing stage. In the Classification stage, the study discusses a few method of classification, specifically, Naive-Bayes and Random Forest classifiers. These are our Machine Learning classifiers. As for Deep Learning, the study mainly aims to use CNNs for their efficiency and power for image classification. This leads to the main proposed model in the study, which works as follows:

- After the initial preprocessing stage, the images are shuffled and split
- The images are passed through two separate pipelines. One consists of Machine Learning and another for Deep Learning.
- In the Machine Learning pipeline, the image is first passed through a Feature Extractor, to obtain all hidden features. Next, the input is passed to the two aforementioned ML classifiers: Naive Bayes and Random Forest. Finally, they are passed onto an Evaluation stage and their performance is gauged.
- The Deep Learning pipeline is somewhat similar in practice. A Deep CNN is used with multiple hidden layers for optimal performance and accuracy. The Dataset is then input into the classifier for training and the data then goes through a test/validation process. Finally, The model is evaluated and judged according to its performance metrics.

Overall, using ML models netted a very high performance accuracy of 96% and 97% accuracy, for 660 images. The algorithm used was Principal Component Analysis (PCA). For the Deep CNN, the study went through 200 epochs with an accuracy of 99.5%. Overall, Deep CNN outperformed the Machine Learning model due to it's greater accuracy and ability for feature extraction.

The authors also begin their paper [21] by introducing the various types of skin diseases including the malignant ones and greatly emphasize on creating awareness about the types of skin cancer and the importance of an early detection of such diseases to preserve human life as much as possible. The emergence of deep learning and image processing techniques has opened new doors in dermatology and that it will aid the dermatologists in conducting diagnosis. This study basically presents the implementation of a CNN model with increased classification accuracy of the skin lesions on the HAM10000 dataset. One of their main goals is to implement an innovation data augmentation method to compensate for the imbalance of classes in the dataset. The authors also want to integrate Model Checkpoint Callbacks to maintain high performance in practical implementations. They, furthermore, mention how the combination of AI with Deep Learning has significantly sped up the development of this sector and has enhanced the classification accuracy and precision of skin cancer diagnosis. Moving on, let us take a look at their methodology. For preprocessing, they have resized their image data to a reasonable size for uniformity for the CNN to work properly. Then they have normalized the images so that their pixel value is scaled down to a range of 0 to 1 from 0 to 255 in order to speed up the convergence process. Lastly, they implemented data augmentation by deploying various transformations to diversify their input data. This allows the CNN to generalize among various input images more efficiently and reduces the chance of memorization or overfitting. Therefore, the model's robustness, adaptability and performance. The architecture that they used was a CNN model with ReLU activation function to tackle the lack of non-linearity, Adam optimizer as their optimization function and SoftMax Function in its output layer. Max pooling layers were also added along with each convolutional layer and a CCE function was used as the loss function. All of these layers and functions when arranged in an orderly manner allows the model to efficiently learn and recognize the patterns in visual data. A flatten layer was also added to transform the 2D output from the pooling layer into a 1D vector. After necessary downsampling and feature extractions, a fully connected layer was also added which is connected fully to all other activities of the model. Batch Normalization was also implemented to standardize the input for each layer, thus, enhancing the speed and stability of the training phase. Dropouts were also added to prevent overfitting. Lastly, the output layer makes use of the SoftMax function for multiclass classification which converts the output of the dense later into probability values for each target class, for this case, the seven skin lesion types. Callbacks were implemented as well where validation loss is monitored where training is stopped when there are no significant performance improvements, thereby avoiding overfitting. Hence, Model Checkpoint secures the model at its highest performance on the validation set so that it can be reused for future experiments. The dataset was split as follows: 80% for training, 10% for validation and 10% for testing. After training, the model achieved an accuracy of 97.86% and a 97.78% in the testing phase. The model also obtained over 90% in precision, recall and F1 scores. The authors then concluded that their study thoroughly explores what CNNs can do in identifying various classes of skin cancer using the HAM10000 dataset and that their study paves way for AI to be integrated with healthcare to improve diagnosis and patient outcomes which will take medical technology to new heights.

The research in paper [2] uses Convolutional Neural Networks (CNN), along with Keras Sequential API, to develop a new model to achieve an accuracy of around 80%. Furthermore, to increase accuracy and for comparison, the authors have used pre-trained data in transfer learning models such as VGG-11, ResNet-50, and DenseNet121. Among the mentioned models, ResNet-50 is observed to achieve the highest accuracy, around 90%. Similar to our research, the authors of paper[2] have also used the HAM10000 dataset from the ISIC archive. Initially, they ran CNN on the dataset, and then used VGG-11, ResNet-50, and DenseNet121 to achieve more precise and accurate results. For preprocessing, firstly, the images are resized to 100 x 75 x 3 to ensure that TensorFlow can handle the data. The dataset is then split into a ratio of 20:80 for training and testing sets respectively. Then both the train and test are normalized by subtracting from the average values and dividing by their standard deviation, in order to normalize all values to 0-1 from 0-255. Following normalization, the seven different classes of skin cancer are labeled from 0-6. For data augmentation, the images are rotated by 10 degrees and zoomed in by 10%. Then, in order to further increase the size of the training set, images were shifted by 10% both vertically and horizontally. In the model implementation step, starting with convolutional layers, the authors have used the Keras Sequential API. These layers use filters to learn features from images. In the first two layers, 128 filters are applied, followed by four Conv2D layers with 64 filters each. These filters are known as kernel filters, which transform the images by applying 2D matrices to them. Then a pooling layer (MaxPool2D) is used, which acts as a downsampling filter. This helps lower computational costs and also reduces overfitting. Then both global and local features are learned by using Conv2D and Pooling layers. Using the Dropout method, overfitting is reduced and normalization is improved by forcing the network to learn in a more distributed way. ReLU is used to introduce non-linearity. For optimization, ADAM optimizer is used. This method refines the model's parameters by repeatedly adjusting the weights and biases of neurons, and thus reducing the error. Next, score function, loss function, and tuning algorithm are set up. The loss function measures how well the model performs by calculating the difference between predicted and actual labels. Here, the authors have used categorical cross entropy for classification, and the optimizer is observed to improve the parameters and minimize the loss. To speed up the optimizer's convergence, a technique called annealing the learning rate (LR) is used, and the ReduceLROnPlateau function from Keras is used to halve the LR after 3 epochs without any accuracy improvement. Furthermore, architectures like ResNet-50, DenseNet-121, and VGG-11 are used with Batch Normalization to boost accuracy. ResNet-50 contains 50 layers and classifies images into 1000 categories. This model uses skip-connections by bypassing some layers, which helps to address the vanishing gradient problem. The paper further elaborates how DenseNet simplifies the connections between layers in comparison to ResNet. In ResNets, not all layers are always necessary and each layer has its own weights. Whereas, DenseNets use narrower layers and only add small sets of new feature maps. This design gives each layer access to gradients from the loss function and the original input, improving training by helping with information flow. Paper [3] also provides an overview of the architecture of all VGG models. VGG models use pre-trained data to improve accuracy. The authors have used VGG-11 since desired results could not be achieved after 10 epochs while using VGG-16 and VGG-19. As per validation accuracy, ResNet-50 has consistently

achieved the highest accuracy in classification task among the algorithms used, and it is selected to further evaluate their system. For evaluation, they have formed a confusion matrix and calculated precision, recall, F1 score, and support. According to the authors, their system's detection accuracy is around 90%, making it remarkably efficient in comparison to human detection.

This paper [3] specifically talks about the importance of Data Augmentation. The usage of Data Augmentation allows the model to actually use the data and learn from it rather than simply memorize the dataset. It is also useful for real life image prediction and processing, as most clinical images are usually somewhat imperfect and grainy, and/or unclear. To that end, data augmentation prepares the model for real life applications and predictions. The authors in this the study delves deep into the importance and use of data augmentation in the field of Melanoma Skin Cancer Prediction. The data augmentation is used for the purpose of being fed into Convolutional Neural Networks. Over the years AI integrated diagnoses have been very popular in the field of melanoma detection. Specifically, dermoscopy images have been used extensively and optimized for better performances in CNN based architectures. CNN itself has been crucial in developing image classification models and image feature extraction. The issue with this method is the large amount of data that is required to train these models to reach a satisfactory level of accuracy, something which is not always easy to access nor easy to find, sometimes requiring permission from multiple channels for it to even be used. This causes a problem when trying to create applications that perform with the utmost accuracy, which is essential in such a crucial field as oncology. To answer these concerns, once can use data augmentation to manage these issues and create a way to avoid overfitting and minimize the validation loss. A small dataset can become quite effective, if augmentation is involved. A referenced paper even found that rotation, zooming, cropping and shearing mimicked real life images and gave the model the chance to provide accurate prediction from more 'true-to-life' images. As such, this study aims to investigate the many ways that augmentation can help when using a CNN. In this study [3], the authors first manage a dataset. They ultimately choose the ISIC archive, which is popular for skin cancer datasets, of which they use 3300 RGB images of size 224x224x244. The images are not needed to be preprocessed and are instead only evenly distributed between two classes to avoid uneven distribution. The study then uses a proprietary CNN model employing 15 layers. The metrics used are accuracy and standard deviation using a confusion matrix. Before being fed into the model, images are preprocessed using CLAHE. Furthermore, hair is removed from the images to provide a clear image of the lesion and so the image of the lesion is unobstructed. Finally, the image is smoothed to remove some noise. Different combinations of each are used and the results are compared. Following the preprocessing, the images are now augmented. Some of these methods include simple changes such as geometric transformation, color transform and noise addition , which are used to add variety, small color changes and adding random values to change parts/whole of the image. A very unique method of augmentation is image mix, whereupon slices of images are combined together to make a collage of images, the usefulness of which is arguable. Combining these augmentation models together, it is very easy to create multiple images and expand the dataset.The effectiveness of these augmentations are then studied and the results are logged. Overall, it seems

that image mix methods yielded the best results, whereas color transform yielded the worst ones. Geometric transformations and noise additions make the CNN more robust and learn more effectively, paying attention to fine details to differentiate. In general, a single augmentation layer showed greater accuracy than more layers, which is to be expected. The reason for the greater accuracy is due to the lower amount of augmentation that the images have gone through, thus many of the original features are retained more visibly and clearly. Moreover, if the discussion is moved towards the data expansion section, mixed augmentation provides much better performance in accuracy and standard deviation scores, as it is more preferred than simpler methods of data duplication, as it does not bring any new information that the model can learn. That being said, the study aims to look at more optimized approaches to data expansion. There was an improvement across the board when the dataset was expanded, meaning no matter the method, an expanded dataset unequivocally performed better. Test accuracy however, seemed to hit a cap at an expansion of 300%, which means an optimal expansion is that of 300%, after which there were severe diminishing returns or accuracy became lower. In conclusion, Augmentation is very important when it comes to dataset handling and is an ingenious method to solve the issue of having a small or unbalanced dataset. While we cannot be sure whether there may be further advancements in the future in the field of augmentation, augmentation remains to be a powerful tool for any computer vision task, and an ingenious way to improve pre-existing datasets and models.

This paper [15] primarily focuses on the combination of Vision Transformers and CNN architectures to produce more accurate and efficient predictions. However, understanding these models still remains a challenge in the medical field or among normal people in general. The integration of XAI with deep learning has the potential to develop trustworthiness and transparency in automated skin cancer classification. The authors begin their paper by emphasizing on how far computerized medical image processing has come and could be a potential answer to the automated diagnosis of skin cancer. They also raise awareness about the early detection of skin cancer as the recovery and survival rate decreases drastically with each stage. Furthermore, they also talk about the image segmentation process that helped increase the overall accuracy for melanoma classification. However, segmentation on medical images is onerous due to risk of noise, lack of contrast, varying brightness and irregular skin lesion borders. Additionally, they say the advent of CNNs, DL models and transformers has remarkably enhanced the skin cancer classification while also decreasing the time and resource costs. Among all, CNNs seemed to be the best performing in medical image recognition. The authors have also mentioned XAI methods that have been implemented in this field in the past like LIME and CAM. Even then, skin cancer classification still is a challenging task due to the variety of lesions and their appearances, quality of database, model biasing, imbalance of classes and the lack of explainability in those models. Therefore, the authors proposed a model using ViT and XAI-integrated CNN models to improve the classification accuracy and explainability. The authors implemented the following models: 2 pretrained vision transformers, 3 Swin transformers and 5 pre-trained CNN models for skin lesion classification while also integrating 3 CAM based XAI models that are Grad-CAM, Grad-CAM++ and Score-CAM. They have also implemented SMOTE to tackle the class imbalance problem. They also have used the ISIC dataset which contains 3297

images. 2637 images of the total set were used for training and 660 images for testing. After applying SMOTE, they resized all the images to 224x224 pixels during the preprocessing phase. The authors have implemented ViT-B (base) and ViT-L (large) and in these cases, since the images are 2D the input images are divided into patches which, in turn, are fed as tokens. A patch is a 16x16 small region of an image in the shape of a rectangular. Patches are then represented as vectors which contain the features collected by a CNN. It is then fed to a transformer encoder layer which itself is a self-attention layer which produces a series of vectors that showcase an image's features and then this is used to classify the input image. 3 Swin transformers were applied too: Swin-T, Swin-S and Swin-B. The authors, further, implemented pretrained CNN models like ResNet5, DenseNet201, VGG19 and MobileNetV2. Following this, to enforce trustworthiness in these models, XAI models were integrated: Grad-CAM which is a technique involving gradient calculation of an output and Score-CAM which is a post hoc visualization explanation which also generates higher visual performance. The 5 performance metrics used were accuracy, precision, recall, specificity and F1 score. For simplicity, for review purposes, we will take a look at the paper's accuracy values only. For the transformer models, both ViT-B and ViT-L generated 88.6% accuracy while Swin-S and Swin-T generated 87.7% and Swin-B generated 87.8% accuracy. And, for the CNN models, VGG19 generated 85.6%, ResNet19 generated 82.4%, DenseNet201 generated 83.2%, MobileNetV2 generated 85.3% and ResNet50 outperformed the others and generated 88.8% accuracy. The authors acknowledge that while their models are robust, there is still room for improvements and fine tuning to boost performance. They concluded that they will keep working on this and run the models with more data, test different XAI models and optimize to improve performance of the models.

Using CNN and Vision Transformers approaches, this study [17] aims to build and compare classification models for Melanoma. On top of that, the study also confronts the issue of explainability of these models, by employing Grad-CAM and its "++" version. These Grad-CAMs provide heat maps to explain the models and the way they reached their outcome. Over the years, the use of Deep Learning models and Machine Learning has been steadily growing in the field of medicine,in both use and performance. Multiple studies have been conducted in this exact field tackling the classification problems. However, none so far have been quite successful at providing a practical solution to this problem that can be applied to real-world situations. The study proposes that using a computational model using Deep Learning and transfer learning would benefit in skin cancer detection when applied to digitized dermoscopy images. The dataset used was the HAM10000, and the models used were ResNet50, Xception, VGG16, Inception and MobileNetv2. These models are capable of performing accurate classification, thanks to their powerful architecture made from multiple convolutional layers. Hence, these models handled the task of classification. To further add novelty and make it more human-readable, Vision Transformers were applied to these images, such as U2-Net and Saliency mask-guided vision transformer (SM-ViT). These would segment the images and improve distinction between tokens. Finally, the Grad-CAM would introduce an explainability aspect to the models and allow it to be trustworthy. All of this was then developed to be a web application as a powerful support tool for overall skin cancer detection. The techniques used in this study [17] can be broken down into

two parts: model building and dataset preprocessing. The preprocessing consisted of augmentation of the images through rotation, normalization, flipping and rescaling. After that, the model building part can be broken down into supervised and weakly supervised. In the supervised part, they have worked with U-Net variations. As for the weakly supervised, they worked with classification and detection models and XAI for GradCAMs. For the U-Net variations, U2-Net segmentation models were used for salient object detection, using the ISIC 2017 dataset. This model is especially useful in detecting more very fine details. Furthermore, due to it being lightweight, the model does not rely on pre-trained models for it to be effective. The output masks are then used in the classification pipeline. After the segmentation stage, the results are used and followed up in the CNN based classification stage. By using different CNN models, a comparative study was performed between all of them using the HAM10000 dataset. Firstly, the dataset was slightly modified to remove any and all duplicated images for a balanced image count of 1:1 for benign and malignant patches (specifically melanoma). Next, from this, the study branched out into two parts. In one part, a segmentation model was used conjoined with the CNN. The U-Net model would segment only the benign/malignant skin patch out of the image, and that would then be passed onto the classification model. For the second pipeline, the whole image was used. Afterwards, the images were augmented and trained and the accuracies were compared and tested. The model then loaded the model with the highest accuracy and a GradCAM was then performed on the images using the best model. This concluded the classification based model. For the Vision Transformer based model, the HAM10000 dataset was used once more. The pipeline is very similar to the classification model, in which the dataset is preprocessed and duplicates are removed. From there, two pipelines are taken, one with a saliency mask and the other without. The models were trained with Stochastic Gradient Descent (SGD). Other than that, a basic Vision Transformer was used without saliency masks. All of these features are then combined with the GradCAM, which consists of convolutional layers, rectified convolutional layers and fully connected layers. After the final convolution layer, GAP is applied and then the output is backward propagated for further accuracy onto ReLU. The final results for the study can be divided into three parts:

- For the segmentation models, U2-Net showed a IoU score of 0.93, a loss of only 11.65% and a recall of 92.77%., whereas U0Net had an IoU of 0.84m, a loss of 15.40% and a recall of 85.21%.

- For CNN, the classification accuracy for almost all the models was above 80%, with the exception of InceptionNet which had an an outlier low accuracy of 61.78%.. The highest accuracy and specificity for the mask guided experiments were for Xception at 98.37% and 99.07% respectively, with the highest sensitivity was achieved by ResNet50 at 97.95%. As for the non-mask guided models, VGG16 gave the highest accuracy and specificity at 93.49% and 94.45% respectively, with ResNet50 getting a very high sensitivity of 99.90%.

- For the Vision Transformer models, the Base ViTs had a accuracy of 84.68%, a sensitivity of 83.87% and a specificity of 85.48%. As for the accuracy, sensitivity and specificity of the Saliency Mask ViT, they are 92.79%, 91.09% and 93.54% respectively.

Overall, with the use of the Grad-CAM, and the addition of segmentation models in conjunction with classification models provide a very powerful method for making explainable models that are useful in the real world, for early cancer detection, classification and segmentation. The authors provide a very deep dive into how they have achieved this using very rigorous methods of testing, with descriptive explanations of how to achieve the same process.

In another article [8], the authors tackled the task of skin cancer classification with DenseNet201 and MobileNetV2 deep learning models. Both models were pre-trained on the ImageNet dataset and then modified with additional layers to effectively do the task. Initially, contrast stretching was done on the images to make faded skin lesion marks more significant. Augmentation was then done on the dataset to increase the number of training images effectively. They applied three augmentation techniques: rotation, flipping and noise addition for both the models. Lastly they formed the modified model by applying a few layers at the end of each model right before the classification head block and using Transfer Learning (TL) to reduce computational cost. The additional layer consisted of 2 convolution layers each having 128 filters of size 3x1 that are stacked below 1 convolution layer containing 64 filters of kernel size 3x3. The MobileNetV2 model is lightweight. One of the main blocks is the inverted residual block (IRB). It requires less memory compared to a convolutional block. This block does both pointwise and depthwise convolution. The depthwise convolution is responsible for removing redundant features helping it to maintain low computational cost. There are 17 such IRBs. Conversely, the original DenseNet201 model consists of 201 layers. The model has 4 dense blocks where convolutional layers are directly connected with each other. Between the dense blocks, there are transitional layers that act as downsampling layers by using average pooling. Modified MobileNetV2 achieved 91.86% accuracy compared to the normal MobileNetV2 model that had 90.54% accuracy. Similarly, DenseNet201 had a 94.09% accuracy while modified DenseNet201 got 95.5% accuracy. Lastly, the authors applied Gradient-weighted Class Activation Mapping (Grad-CAM) to make the deep learning models that are considered as black boxes to be more explanatory. It provides a visualization to explain the model's prediction. Through this we will see that it correctly learns features and is able to localize the lesion during training. It will weakly localize the lesion spot that can aid the dermatologist to detect and diagnose the skin lesion.

In this paper [18], the authors proposed their own deep convolutional neural network (DCNN) model specifically for the task of skin cancer lesion classification. They dealt with two datasets, HAM10000 and ISIC-2019 to determine model performance comparative to other deep learning models like VGG16, DenseNet121, MobileNetV2, etc. using the same datasets. Normalization and image reshaping was done for effective model training. Normalization makes the model less sensitive to variations in image intensity making it more consistent. Image resizing increases computational efficiency while decreasing memory utilization. The main concern of their project was dealing with the unbalanced datasets. To handle the major class imbalance that causes the model to become biased, they used the oversampling method. They used SMOTE, SMOTEENN, ADASYN sampler, and random oversampling methods to compare results. The random oversampling method pro-

vided them with the highest accuracy, recall, precision and F1 values. Without handling the class imbalance, their proposed model performance was comparatively the least accurate. The architecture of the proposed DCNN model consisted of 8 convolutional layers. In between each pair of convolutional layers, they included a batch normalization layer with ReLU activation function. Batch normalization is used to stabilize and speed up the training process. Kernel size of convolutional layer remained 3x3. The number of filters increased to 32, 64, 128, 256 the model could start learning more intricate features. After all convolutional layers, they used a MaxPooling layer to reduce spatial dimensions of feature maps. This followed with a flattening layer and then a dropout of layer of 0.5 rate before it was passed down to three dense layers. In between the dense layers they included a batch normalization layer. Lastly, the output layer was a softmax layer that gave class probabilities as output. The model was compiled with a categorical cross-entropy loss function, learning rate of 0.001 and the Adamax optimizer. Also, they applied a learning rate reduction callback to adjust the learning rate during the training phase. Before handling the class imbalance, the proposed model had an accuracy of only 0.665% for the HAM10000 dataset. After random oversampling, the accuracy reached an impressive 98.5% on the same dataset. It was similar for the ISIC-2019 dataset too. They also applied Grad-CAM to generate a rough localization map, focusing on crucial regions within an image for predicting output. This helps users to observe whether the model has learnt the correct features. They also printed 32 feature maps of a single input image at different layers to see what patterns are being learnt at each layer. The proposed model works specifically well for classifying skin cancer lesions.

Chapter 3

Dataset Description

We already know in order to train a deep learning model or a neural network for automated diagnosis of skin cancer, we need a large variety of dermatoscopic images. However, in most scenarios, the data acquired is usually small in size or there is a lack of diverse datasets in hand. To avoid such issues, we have chosen the HAM10000 (Human Against Machine with 10000 training images) dataset [22]. It is a wide collection of dermatoscopic images collected from various populations which contains roughly 10,015 images divided into 7 classes of skin cancer. The authenticity of the skin lesion images were also verified by health experts and pathologists before it was made public for research purposes. The 7 classes in question are:

- **AKIEC**: Actinic Keratoses and Intraepithelial Carcinoma
- **BCC**: Basal Cell Carcinoma, malignant
- **BKL**: Benign Keratosis-like Lesions
- **DF:** Dermatofibroma, benign
- **MEL**: Melanoma, malignant.
- **NV:** Melanocytic Nevus
- **VASC:** Vascular lesions, malignant/benign (We have considered it malignant, in this case.)

An example of each class can be found in the following page:

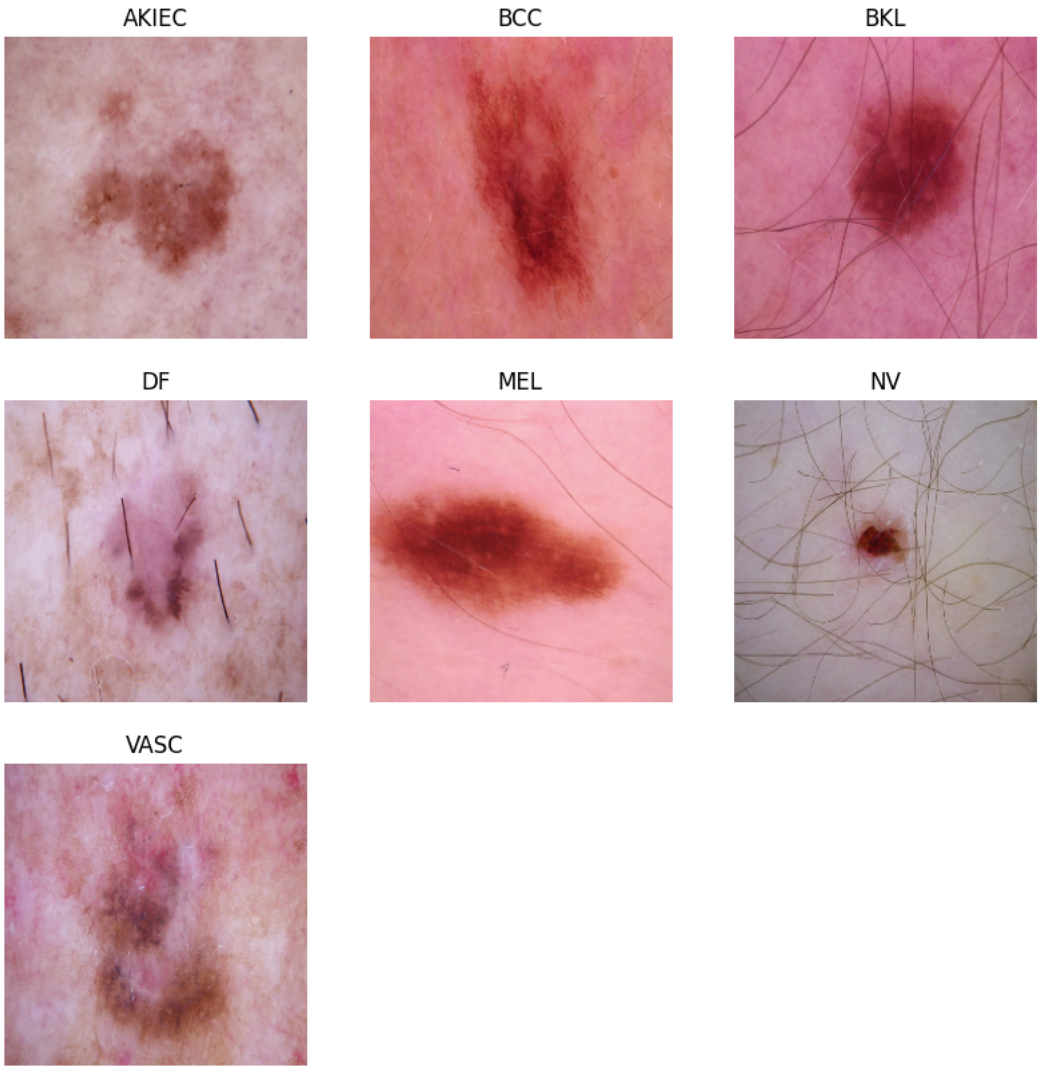

Figure 3.1: Class Examples

Besides the HAM10000 dataset, we also utilized the ISIC 2019 (The International Skin Imaging Collaboration 2019) dataset [23] for transfer learning purposes. It is an extensive dataset containing exactly 25,331 images ranging over 9 separate classes of skin cancer. The skin lesion images were also validated by medical professionals ranging from well renowned companies before the dataset was publicised for research purposes. The 9 separate diagnosis classes are namely: Melanoma (MEL), Melanocytic nevus (NV), Basal cell carcinoma (BCC), Actinic keratosis (AKIEC), Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis) (BKL), Dermatofibroma (DF), Vascular lesion (VASC), Squamous cell carcinoma (SCC) and Unknown (UNK). 7 out of the 9 classes are similar to the HAM10000 dataset which made this dataset suitable for the transfer learning purpose. The class images in the ISIC dataset are similar to those shown in the examples in Figure 3.1.

3.1 Dataset Imbalance

Even though the HAM10000 dataset is large and diverse with its many classes, it is also heavily imbalanced. An imbalanced dataset is where one class tends to be the majority class i.e. if one class is more common than another. The less common classes then become the minority class and the implemented models often tend to skew towards the majority class. Total number of images per class are as follows - AKIEC: 327 images, BCC: 514 images, BKL: 1099 images, DF: 115 images, MEL: 1113 images, NV: 6705 images and VASC: 142 images. It is evident that the NV class overwhelmingly dominates the dataset. The imbalance in the HAM10000 dataset can be visualized in the bar chart attached in the Figure 3.2.

To address this serious issue, we opted to make use of **class weights** in our loss functions. Class weights assign greater importance to minority classes and less to majority classes in the loss calculation. By doing this, the loss function penalizes misclassifications of minority class samples heavily and thus encourages the model to learn more from the minority classes. As a result, the classes impact the loss function in a balanced manner. In our task, we put the estimated class weights in our loss function, namely cross entropy loss function, as ('AKIEC', 'BCC', 'BKL', 'DF', 'MEL', 'NV', 'VASC') = (0.17, 0.17, 0.21, 0.07, 0.24, 0.07, 0.07). It can be observed that the majority class was given the least weight and minority classes were given the most weight. By doing this we can ensure that the model's accuracy does not increase only through getting the majority class correct, but rather to maximize recall to avoid false negative outputs that may be dangerous.

Instead of using traditional generative algorithms like SMOTE (Synthetic Minority Over-sampling Technique) to tackle the imbalance in our dataset, we have decided to use class weights as SMOTE poses some issues that do not go well with our purpose. Firstly, since the dataset is heavily imbalanced, SMOTE might not properly represent the minority class. Secondly, it can over-sample the minor classes which might result in overfitting towards the minor classes. Lastly, SMOTE increases the computation cost of our models and from our experience, it has caused our models to crash numerous times due to memory limitations.

The imbalance in dataset is represented by the bar charts attached below:

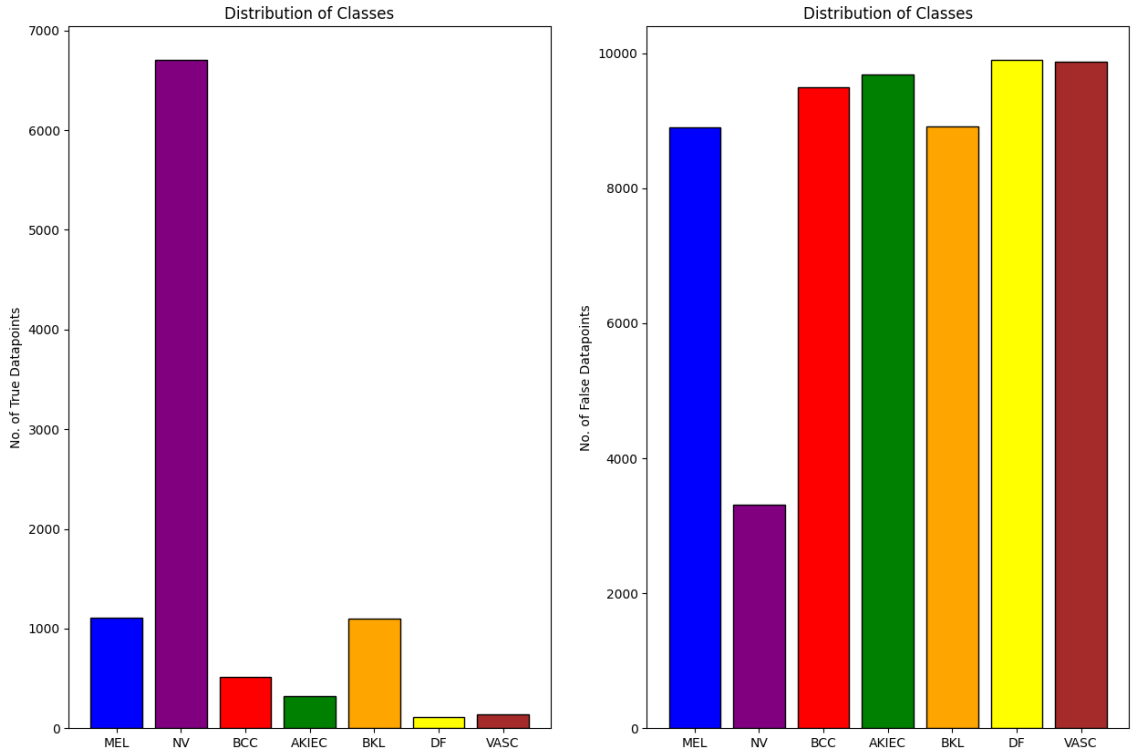

Figure 3.2: Bar Chart representing number of True and False values for each class

Chapter 4

Methodology

4.1 Background Study

4.1.1 Transformers

In order to understand some of the models we have used for our research purposes, let us go through what Transformer models are and how they function. Transformer is a neural network architecture that has been used for numerous ML tasks in research ever since it was first introduced in 2017. Transformer architecture makes use of a set of mathematical approaches, called self-attention, to detect how data elements in a series depend on one another. In other words, a transformer model is a neural network that learns the context of sequential data and proceeds to generate fresh data from it. Transformers turned out to be a huge leap from the orthodox RNNs and LSTMs which suffered from either vanishing gradient problems or failed to capture the relevant context from its data. On the other hand, transformers, via its self-attention, tackled these issues and proved to be efficacious at understanding the context. Therefore, nowadays, transformers are widely used for NLP tasks, computer vision, speech recognition, recommendation systems etc.

Transformers primarily work best in converting input sequences to output sequences. It is the first transduction model which completely relies on self-attention without taking the help of any RNNs or convolution layers. Just like some other neural network models, transformer models also consist of large encoder and decoder blocks that process data. To be more specific, the native architecture had only 6 encoders and 6 decoders but we can use "n" number of layers of it for our purpose. The main function of the encoder is to transform input tokens into such representations such that the encoder can capture context for each individual token with respect to the full sequence. The bottom most encoder consists of an input embedding that captures the semantic context of the tokens and converts them to mathematical vectors. Unlike RNNs, transformers do not have a sound understanding of sequential order so to solve this issue, Positional Encodings are added to give information regarding the positions of each and every token in a sequence. Now let us take a look at the encoder and decoder layers used in this architecture.

The encoder layer consists of 2 modules: a self-attention mechanism and a feed forward network. Moreover, each sublayer has residual connections which is then

followed by layer normalization. The attention mechanism used in this architecture uses scaled dot-product by calculation weighted sums of query, vectors and keys. The multi-head attention applies this mechanism multiple times parallely to increase model performance. The outputs are then concatenated and transformed to generate the final output. This allows the model to extract varying relationships and create richer, more refined representations of the input sequence. MHA improves the transformer model's capability to process complex data. The output of this layer is then fed into a feed-forward neural network that consists of two linear transformations with the ReLU activation function. Now comes the decoder which is also quite similar to the encoder layer in terms of its workflow: starts with output embeddings followed by positional encoding and then the decoder layer. The decoder layer consists of 3 modules: self-attention, encoder-decoder attention and a feed-forward network. There is an additional attention layer between the self-attention mechanism and the feed-forward network layer, unlike the encoder, to help the decoder concentrate on the more important parts of the input sequence.

4.1.2 Visual Transformers (ViT)

Since we are using image data for our research purpose, we will be using Vision Transformers (ViT). For the last decade, any research field with images involved was heavily based on the use of CNNs as their primary models but with the advent of Vision Transformers, this field is changing its approach. Vision Transformers make the implementation of the transformer architecture possible on image data thus opening new ways to process visual information. The main idea, therefore, would be to treat images as a sequence of data. To understand how ViT works we must understand a few basic steps for this architecture:

Image Patching: Patching is the first step in implementing a ViT. This step involves dissecting the images into smaller patches of a specific size. Each patch then acts as a part of the sequence and is processed accordingly. Therefore, the model then can analyze the full image via its components. For instance, a 224x224 pixel image can be divided into 16x16 pixel patches which gives us a total of 196 patches. Then the patches are flattened into vectors, allowing the model to process these smaller, feasible slices of the image.

Positional Encoding: It is the same concept as we have seen previously for regular Transformers except this time we are adding positional encoding to the patch embeddings to maintain the positional details of the patches. It ensures that the model comprehends where each patch is situated in the original image, enabling it to extract the spatial relationships effectively.

Multi-Layer Transformer Encoder: This is one of the core structures of ViT which consists of Self-Attention layers: these layers allow the model to evaluate the relationships between various patches and aids it to comprehend how they behave with one another. The evaluation is done by calculating attention scores that determine how much focus should each patch have on other patches and thus enabling ViTs to deduce complex connections and dependencies throughout the entire image; and,

Feed Forward Network Layers: these layers add non-linearity to the output generated from the self-attention layer, improving the model's potential to learn complex, hierarchical or abstract patterns in the data.

Classification Head: It is the most important component of ViT that is used to make predictions for image recognition and classification tasks. Tokens called Classification Tokens (CLS) integrate information from all patches and produce the final predictions. This collection of data ensures the model leverages all the details from the entire image rather than secluded patches.

All in all, ViTs can produce more complete and distinct feature representations, recognising complicated patterns that might have been missed by conventional neural network models.

4.1.3 Swin Transformers

Swin Transformer was first introduced in 2021 and since then there has been a huge shift in Transformer architectures. Swin Transformer familiarizes us with a hierarchical architecture with shifted windows. It not only enhances computational performance but also enhances scalability, making it a great choice for a variety of image processing and computer vision tasks. Since we have also used a Swin Transformer for our research purposes, let us take a deeper look into how it functions. An interesting point to consider though is that Swin Transformer is basically a ViT with new additional features. To understand how a Swin Transformer works, we must first try to grasp the core concepts:

Swin Embedding: This essentially converts the input RGB images into non-overlapping patches which is similar to Image Patching in ViT. Each patch here is considered as a token and is then projected into a higher-dimensional space with the help of a linear embedding layer. What it does is it converts each pixel's raw RGB pixel value into a feature representation.

Patch Merging: It acts as a merging block that primarily reduces the patch numbers at each deeper layer of the model. For example, an input patch tensor of shape (batch size, number of patches, color channels) is fed to the merging block which then outputs a merged patch tensor with reduced number of patches and twice the channel dimensionality. To implement this, the input patches are resized into a grid format and thus reducing the grid size for both the height and width dimensions by half. Following this, a linear transformation is applied to stretch the channel dimension by two times. Lastly, layer normalization is added so that the output is stable.

Relative Position Embeddings: In order to work out the self-attention, we need to include a relative position bias to each head's calculation of similarity. This allows the model to understand the positional relationships and dependencies between tokens. Smaller sized bias matrix that extracts relative positions in a window range. Moreover, positional embeddings are learnt during the training phase rather than using

sinusoidal frequencies. These embeddings are added to the attention formula while keeping the matrix shape constant. As a token's position can only vary by a small margin in each attention window, we just need to keep a record of the parameters for the smaller-sized matrix. This simplifies the whole process and makes it efficient.

Shifted Window Attention: This is what makes the Swin Transformer unique. To understand this we should first, briefly, learn about 'Windowed Attention'. In windowed attention, the attention score is computed in parallel for the non-overlapping patches of the image. [diagram] However, on the contrary, Shifted Window Attention introduces a shift mechanism that permits overlapping and reusing of tokens between adjoining windows. Therefore, it enhances the model's potential to cater to contextual details across various scales without raising the computational complexity as it enables the model to extract dependencies efficiently while also keeping spatial coherence. The whole concept is to establish a cross window communication for the model.

Swin Encoder: When the encoder receives the input tensor, layer normalization is applied to it in order to regularize feature distributions. Then, the tensor goes into the Shifted Window Multi-Headed Self-Attention (WMSA). After that, a skip connection is implemented between the input tensor and the output of WMSA to store information. Then it undergoes another layer normalization before the tensor is fed to the Multi-Layer Perceptron (MLP). The output from this MLP module is then added to the output from the skip connection that leads to another skip connection. Lastly, dropout regularization is added as a measure to prevent overfitting and then the final output tensor is yielded from the Swin Encoder.

Finally, we combine all these 5 blocks together to build a Swin Transformer which offers hierarchical partitioning, patch feature dimensionality, linear transformation etc unlike ViTs.

4.1.4 Neural Network Models

Even though we are well acquainted with Convolutional Neural Networks (CNN) by now, let us briefly review what it is. CNNs are Deep Learning Neural Networks used in various image classification and recognition based tasks for its ability to recognize patterns in images. CNN models consist of an Input Layer, a number of Hidden Layers and an Output Layer where the Input and the Output Layer have neurons equal to the number of features in the data. The Hidden Layers, consisting of a convolutional layer, a pooling layer and a dense layer, are what actually analyzes the input data. The deeper the network is, that is the more hidden layers a CNN has, and it can be assumed, the more efficiently and accurately it can generate results. The convolutional layer applies kernels or filters to the input image for feature extraction, the pooling layer performs downsampling on the images to reduce computation and finally, the dense layer makes the final prediction.

For our research purposes we have used two variations of CNN models for comparison with our proposed model, i.e., the Attention Based Residual Convolutional

Neural Network (AResNN) which we will see in detail in later sections of this paper. The models used for said comparison are: VGG16 and ResNet50.

(a) VGG16:
VGG16 is indeed a CNN model which has 16 layers: 13 convolutional layers and 3 fully connected layers. The main charm of this model is its effectiveness and simplicity of its architecture, and its ability to obtain quality performance on different computer vision tasks. The model's architecture is basically a stack of convolutional layers followed by Max-Pooling layers with increasing depth which enables the model to learn intricate hierarchical patterns of visual features leading to robust results. Even though many different architectures have made its way in this field, VGG16 still remains relevant for numerous deep learning applications for its performance and versatility.

(b) ResNet50:
ResNet50 is a powerful CNN model consisting of 50 layers often used for image classification tasks that has been pre-trained on a large dataset called the ImageNet dataset [over 14 million images and 1000 classes] and one that can obtain excellent results. One of the core but unique features of this model is the use of residual connections, also known as skip connections. These residual connections allow the model to learn much deeper architectures and negate the vanishing gradients problem. The architecture of ResNet50 is divided as such:

Convolutional Layers: These layers, followed by batch normalization and ReLU as its activation function, extract features from the input images which are then passed into the Max-Pooling layers that, in turn, reduce the spatial dimensions of the feature maps while keeping the most important features.

Identity Block: This block is where the network learns the residual functions that correspond the input to a certain output. This is done by passing the input through a series of convolutional layers and then adding the input back with its output.

Convolutional Block: It adds a 1x1 convolutional layer to reduce the number of kernels before the 3x3 convolutional layers.

Fully Connected Layers: These layers are the ones that generate the final classification. The output of the last fully connected layer is passed through a Softmax function to produce the final class probabilities.

4.1.5 Skip Connections

Previously, we have discussed ResNet or Residual Networks that gave us a brief overview of what skip or residual connections could be. Skip connections are a vital component of our proposed model, the AResNN which we will see further into this paper. Skip connections are a component from the neural network architecture introduced to make training of very deep networks possible. They also minimize gradient vanishing and loss of information caused during passing through the mul-

tiple layers of a very deep network. Skip connections allow layers to skip layers and connect to layers further up the network, enabling easier flow of information and gradient through the network. To implement such connections we need to build a residual block that will include a few convolutional layers followed by batch normalization and a ReLU Activation function. The way it works is the input is added to the output of the convolutional layers and this creates the skip. Since they reduce gradient vanishing and so allow deeper networks, they improve the model's accuracy and generalizing ability. They allow the network to learn more intricate patterns from the data and also significantly decrease the number of parameters and operations required for the network to operate. Moreover, they can also make training times faster for deeper networks. However, like most things, it does have some disadvantages like usage of skip connections may increase complexity and memory requirements or may add noise to the data. This is why they need to be carefully designed and implemented to match the network architecture where the location of the skip connections in the network plays an important role when it comes down to impacting the network performance.

4.2 Data Preprocessing

During the transfer learning process, we discarded the classes SCC and UNK from our ISIC 2019 dataset. We did this to remain consistent with the classes of our main HAM10000 dataset. Our final classes were: 'AKEIC', 'BCC', 'BKL', 'DF', 'MEL', 'NV' and 'VASC' for both the datasets. This resulted in reducing the total dataset size to 24,703 images.

Both our datasets were preprocessed for easier handling in the similar manner. The initial dataset for both ISIC 2019 and HAM10000 consisted of a single folder containing all images and a single .csv file as metadata. While simple, it is not effective nor efficient for our task at hand. Thus, for ease-of-use later, we segmented the dataset into multiple folders, according to the seven different classes. This meant that each folder contained images of that class only, thus leaving us with seven individual folders respective to the seven classes. This was done by using the .csv file for class data, and then placing images according to their image ID. After that, the dataset was ready to be used. The train-test split was kept at 90-10 for the HAM10000 dataset, so we used around 9013 images for training and 1002 for testing purposes. For the ISIC 2019 dataset, the train-test split ratio was 80-20, meaning we had 19,762 images for training and 4,941 images for testing.

4.3 Data Augmentation

Data augmentation plays a crucial role in improving the generalization of the models for skin cancer detection. Data augmentation ensures that the models do not overfit to the limited small training data and can generalize better to unseen cases. These transformations are applied to simulate diverse imaging conditions and improve the model's ability to detect patterns regardless of orientation or lighting. This also helps in overcoming the limited diversity in the original dataset. Therefore, our

purpose of data augmentation is to bring variety to the image, like having the skin lesion in different positions of the image, making the image brighter or darker, making the mark larger or smaller, etc. This helps the model to learn of various sized and positioned skin lesions. The augmentations described below were applied to both the datasets.

First, we used RandomResizedCrop(64) to resize each image to a fixed size of 64*64 while randomly cropping different parts of the image. This ensures uniform input dimensions while retaining essential lesion features, helping the model focus on key patterns regardless of where they occur in the image. Next, we applied RandomHorizontalFlip(p=0.5), which flips images horizontally with a probability of 50%. This augmentation introduces variations in lesion orientation, simulating the different perspectives that might occur in real-world scenarios. To account for variations in lighting and skin tone, we incorporated ColorJitter where the brightness, contrast and saturation was set to 0.2 and hue to 0.1. These adjustments mimic the natural variability in image conditions caused by different lighting environments and camera settings, making the model more robust to such inconsistencies. We also included RandomRotation(15) to randomly rotate images by up to ±15 degrees. This ensures the model is not overly sensitive to minor changes in orientation, which can occur in real-world image capture conditions. Finally, we normalized the pixel values using Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]). These mean and standard deviation values correspond to the ImageNet dataset and ensure that pixel values are standardized, enabling the model to converge more effectively during training. By applying these augmentations, we significantly improved the variability of the dataset, enabling the model to learn robust and generalized features. This is critical for accurately classifying skin cancer images across varying conditions and orientations.

Attached below is an example of all augmented images in a batch of size 16:

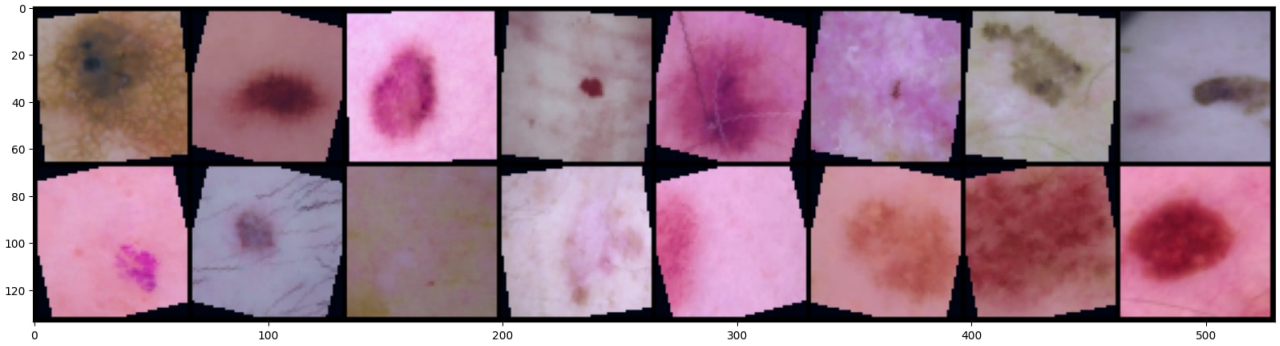

Figure 4.1: An Augmented Batch of Images

4.4 Flowchart

Following is the visual representation of our workflow:

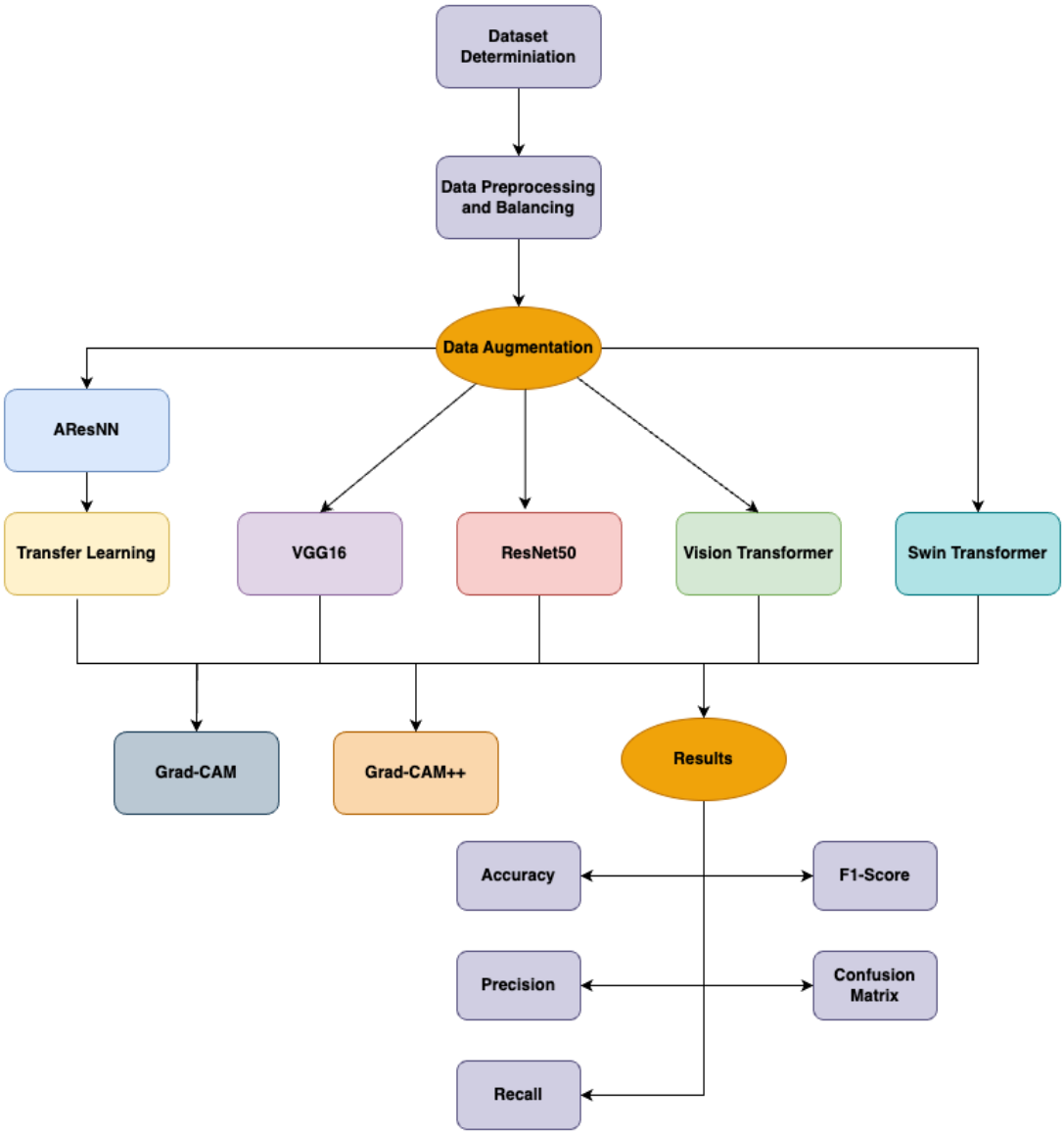

Figure 4.2: Workflow

4.5 Model Implementation and Architecture

4.5.1 ResNet50

We utilized the ResNet-50 model, a widely recognized and powerful CNN architecture known for its 50-layer depth and strong performance in image classification tasks. Pretrained on the ImageNet dataset, the ResNet-50 model was adapted to meet our purpose. All input images were resized to 64*64 pixels, and the model's parameters were fine-tuned to suit our task at hand. We allocated 90% of the data for training and 10% for validation to monitor model performance. A consistent batch size of 16 was maintained across all our models. The Adam optimizer, configured with a learning rate of 0.001, was selected to ensure efficient convergence. Furthermore, the categorical cross-entropy loss function was employed to handle the multi-class nature of our skin cancer dataset. The model's final pooling layer was set to its standard global average configuration, effectively reducing the feature maps before classification. Two dense layers were added at the end, featuring ReLU and softmax activation functions, respectively. Lastly, to align with the number of classes in our dataset, the final dense layer was configured with seven nodes. This setup capitalized the power of transfer learning while customizing the model to address our classification needs effectively.

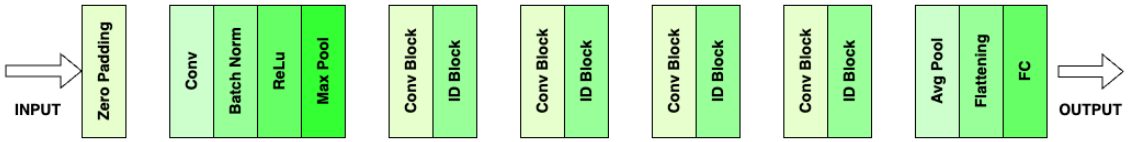

Figure 4.3: ResNet50 Architecture

4.5.2 VGG16

Opposing the previous model, we utilized the VGG16 model, a well-established CNN architecture known for its simplicity and effectiveness in image classification tasks with its 13 convolutional layers followed by 3 dense layers. This was also pretrained on the ImageNet dataset to leverage the model's feature extraction capabilities. The model was adapted to our skin cancer classification problem by resizing all input images to 64*64 pixels. The dataset was similarly split into 90% training data and 10% validation data to monitor performance during training. Batch size of 16 was set throughout the process. The kernel size and stride was set to the default 3*3 and 1 respectively for all the convolution layers. Max Pooling layers had also a kernel size of 3*3 but with a stride of 2. Dense output layer with seven nodes and a softmax activation function was added to match the number of classes in our dataset. This configuration enabled us to use the VGG16 model effectively through task-specific customizations to achieve accurate skin cancer classification.

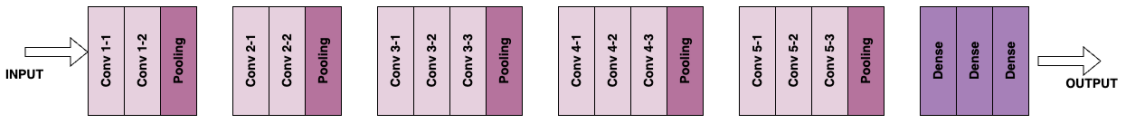

Figure 4.4: VGG16 Architecture

4.5.3 Vision Transformer (ViT)

Moving to transformer models, the ViT model switches up the CNN-based approach of learning by focusing on a global context of the entire image rather than just localized regions. For our task, again, the dataset was split into 90% training and 10% validation, and a batch size of 16 was maintained throughout training. We adapted the ViT model to handle 64*64 images, which were divided into 16*16 sized patches, resulting in a total of 16 patches. This reduces the computational complexity without losing significant performance. Similar to other models, we maintained the learning rate of 0.0001 and configured with the Adam optimizer to make effective convergence. Additionally, a cosine annealing learning rate scheduler was added to slowly decrease the learning rate over epochs, enhancing performance further. The cross entropy loss function was also utilized to handle our multi-class dataset classification.

In the ViT model, the transformer architecture's core components were configured as follows:

1. Hidden size = 128: This determines the size of the embeddings and the internal representations in the model. It controls the capacity of the model to learn intricate patterns in the data. A hidden size of 128 ensures a balance between computational efficiency and the model's ability to capture meaningful features.

2. Number of layers = 8: This represents the number of transformer encoder blocks. Each layer contributes to learning increasingly abstract features from the data. A depth of 8 layers ensures sufficient complexity to model our task without overfitting to our relatively small dataset.

3. Attention heads = 8: The multi-head attention mechanism uses 8 attention heads, allowing the model to focus on multiple parts of the image simultaneously. This means the model learns diverse and complementary features, enhancing its overall representation power.

This configuration allowed the ViT model to process global image features effectively, leveraging its unique transformer-based architecture to classify skin cancer accurately.

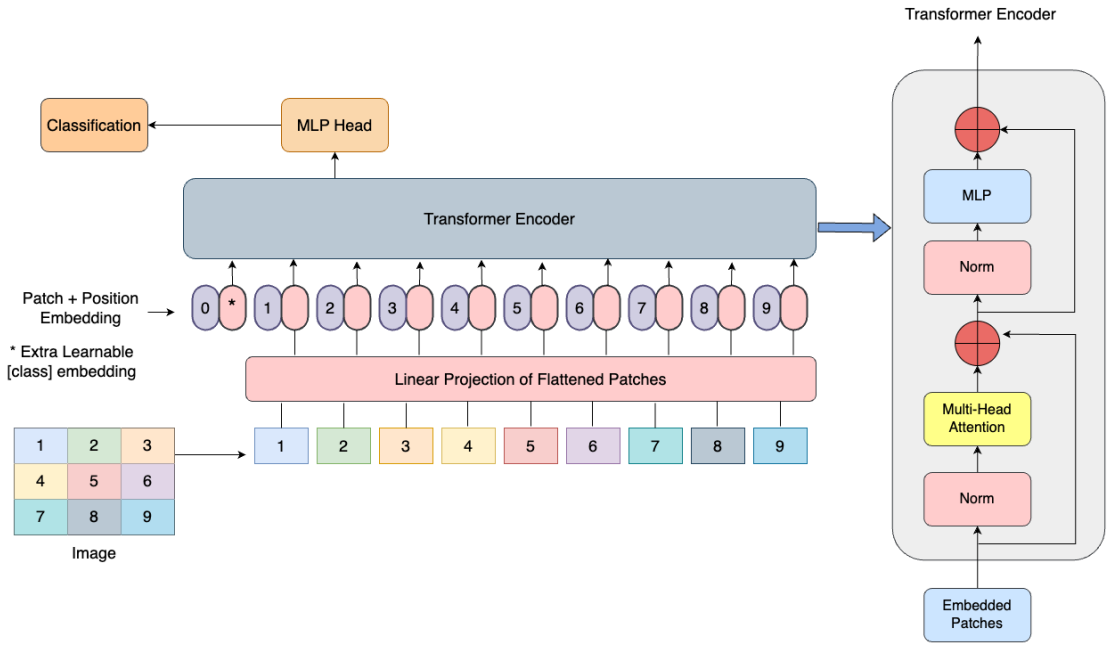

Figure 4.5: ViT Architecture

4.5.4 Swin Transformer (Swin-T)

The Swin Transformer (Swin-T) introduces a hierarchical vision transformer architecture with a focus on local attention mechanisms. For our implementation, the model was trained using input images resized to 224*224, ensuring compatibility with the model architecture. A batch size of 16 was selected to balance computational efficiency and training stability, while the learning rate was set to 0.001 for optimal convergence. The Adam optimizer, combined with a weight decay of 0.0001, was used to regularize the model and prevent overfitting. Consistently, to monitor performance, just like other models, 10% of the dataset was reserved for validation. Additionally, label smoothing with a factor of 0.1 was introduced to improve generalization by preventing the model from becoming overly confident in its predictions. Together, these configurations ensured a robust and efficient training process for the Swim-T model.

In the Swin-T model, the transformer architecture's core components were configured as follows:

1. Patch size = (16*16): Input images were divided into 16*16 patches, creating a sequence of patches for the model to process. Since the image size is 224*224, the resulting number of patches was 196, which in return also determined the size of the MLP layers.
2. Embed dimension = 32: Each patch was embedded into a feature vector of size 32, serving as the input to the transformer blocks.
3. Dropout rate = 0.2: A dropout rate of 0.2 was applied to the embeddings and intermediate layers to reduce overfitting and improve generalization.

4. Attention heads = 8: Identical to the ViT model, the multi-head attention mechanism used 8 heads, enabling the model to extract various complex patterns in the image by attending to multiple regions simultaneously.

5. Window size = 1: The Swin-T model operates on local windows rather than the entire image, with a window size of 1 in this configuration. This allowed the model to focus on localized regions.

6. Shift size = 1: The shifting mechanism, which offsets the windows between layers, helped the model capture cross-window relationships, ensuring comprehensive feature extraction.

7. Query, Key, and Value bias = True: Learnable biases were added to the query, key, and value projections, enhancing the model's adaptability to specific tasks.

Lastly, the final output dense layer with softmax activation, had 7 nodes to match the number of classes of our dataset. This hierarchical attention mechanism and efficient architectural design allowed the Swin-T model to process both local and global features on the skin cancer classification task.

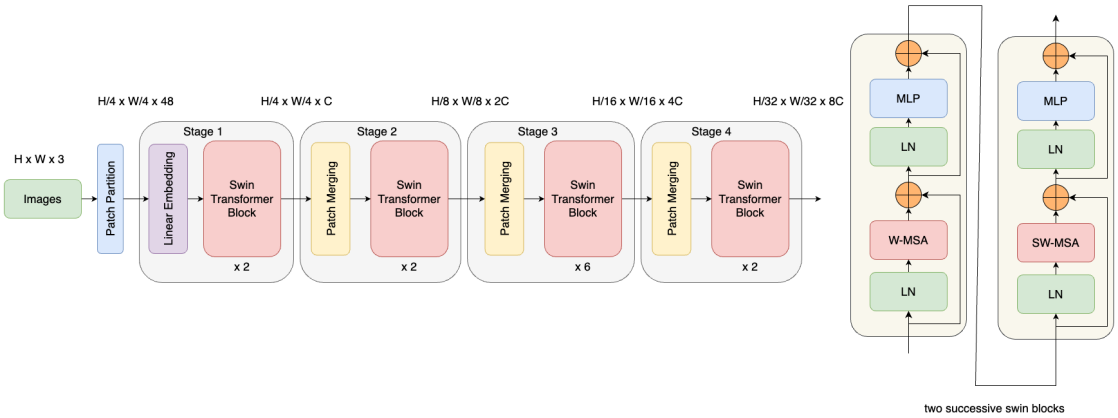

Figure 4.6: Swin Architecture

4.5.5 Proposed Model: AResNN

Our proposed CNN model combines state-of-the-art design principles from multiple influential architectures, offering a blend of accuracy and performance. We opted to produce a lightweight model that could compete in performance with other well renowned models for skin cancer classification. Our model has only approximately 5 million trainable parameters which is very low compared to other CNN models like ResNet50 (24 million) and VGG16 (14 million). We integrated the best features from Vision Transformer (ViT) and ResNet models that had a breakthrough in the image processing world. Our model's main 2 attributes are the use of the multi-head attention mechanism and the use of skip connections. One of its standout features is the incorporation of a multi-head attention (MHA) mechanism, inspired by the ViT model. MHA is able to generate global contexts or relationships in images as it

allows the model to access data from different subspaces at separate positions improving its ability to focus on prominent regions of input images. Additionally, the model leverages skip/residual connections introduced in the ResNet architecture. These residual connections jump over multiple sequential layers by directly adding input of a layer to the output of a later layer in the architecture. It has been proven that models with residual connections converge much easily as it enables easier gradient flow even in deep networks overcoming the vanishing gradient problem. The use of layer normalization before attention layers enhances training stability, while batch normalization in convolutional blocks improves convergence by standardizing activations. Together with dropout, the model balances complexity and regularization, making it robust to overfitting and adaptable to challenging tasks. By utilizing well known functionalities in the image processing world, we produced our proposed model "AResNN".

Model Architecture: The model architecture is a hybrid of CNNs, MHA mechanisms, and residual connections, designed to effectively capture spatial and contextual features while mitigating common issues like vanishing gradients. Here's a detailed breakdown:

1. Attention Mechanism:
   (a) Input data first goes through Conv2D (conv1) where input_channels = 3, output_channels = 64, kernel size = (3, 3), stride = (1, 1) and padding = 1 and padding_mode = reflect.
   (b) This 64-channel output is then processed by the attention mechanism:
      i. Data is reshaped for attention (batch_size, channels, height*width) where batch_size = 16, channels = 64, height = 64 and width = 64.
      ii. Layer normalization is applied where normalized_shape = 64.
      iii. Self multi-headed attention is computed where the number of heads is set to 4.
      iv. Attention output is reshaped back to the original dimensions.
   (c) The attention output is scaled by the learnable scale parameter which has been previously initialized to 0.
   (d) Scaled attention is added back to the original conv1 output to form a residual connection.
   (e) ReLU activation is applied to this combined output.

2. First Residual Connection Block:
   (a) Main Path:
      i. Output of ReLU activation is input in a Conv2D layer (conv2) where in_channels = 64, out_channels = 64, kernel_size = (3, 3), stride=2, padding=1 which forms H/2 and W/2.
      ii. Batch normalization is done on this output keeping num_features = 64.

iii. ReLU activation is applied to this output.

iv. Output put in another Conv2D layer (conv2b) where in_channels = 64, out_channels = 64, kernel_size = (3, 3), stride=1, padding=1.

v. Batch normalization is done on output keeping num_features = 64.

(b) Skip Connection Path:

i. Input of conv2 is downsampled (downsample1) to match dimensions with output from main path using a Conv2D layer where in_channels = 64, out_channels = 64, kernel_size = (1, 1), stride=2, padding=0 which also forms H/2 and W/2.

ii. Batch normalization is done on this output keeping num_features = 64.

(c) Main path and skip connection path are added together to form the residual connection.

(d) ReLU is applied to this sum.

3. Second to Fifth Residual Connection Block:

(a) Same structure as First Residual Connection Block in forming 2 different paths to form a residual connection in each block.

(b) Differences:

i. Second Residual Block: input_channels = 64 and output_channel = 128, H/4 and W/4 is formed as output by setting stride = 2 (in only 1 Conv2D layer) and padding = 1.

ii. Third Residual Block: input_channels = 128 and output_channel = 128, H/8 and W/8 is formed as output by setting stride = 2 (in only 1 Conv2D layer) and padding = 1.

iii. Fourth Residual Block: input_channels = 128 and output_channel = 256, H/16 and W/16 is formed as output by setting stride = 2 (in only 1 Conv2D layer) and padding = 1.

iv. Fifth Residual Block: input_channels = 256 and output_channel = 512, H/16 and W/16 is unchanged as output by setting stride = 1 kernel = (3, 3) and padding = 1.

4. The 512-channel feature map is flattened.

5. Passed through a Dropout layer which is set to 0.2.

6. First dense layer (fc1) reduces to 256 dimensions and ReLU is applied.

7. Final dense layer (fc_out) produces 7 class output.

We trained this model with a batch size of 16, just like the rest of the models, where all images were resized to 64*64 pixels due to our limited computational resources. Our loss function was the cross entropy loss function that was configured with the

weights = (0.17 0.17 0.21 0.07 0.24 0.07 0.07) to handle class imbalance. An attempt was made to set the learning rate to 0.00001, however it ended up crashing the training process. So we next attempted to set it to 0.0001 which brought about a good performance result to our model. Thus, we set the Adam optimizer with a learning rate of 0.0001.

The block diagram of our architecture can be found in the figure below:

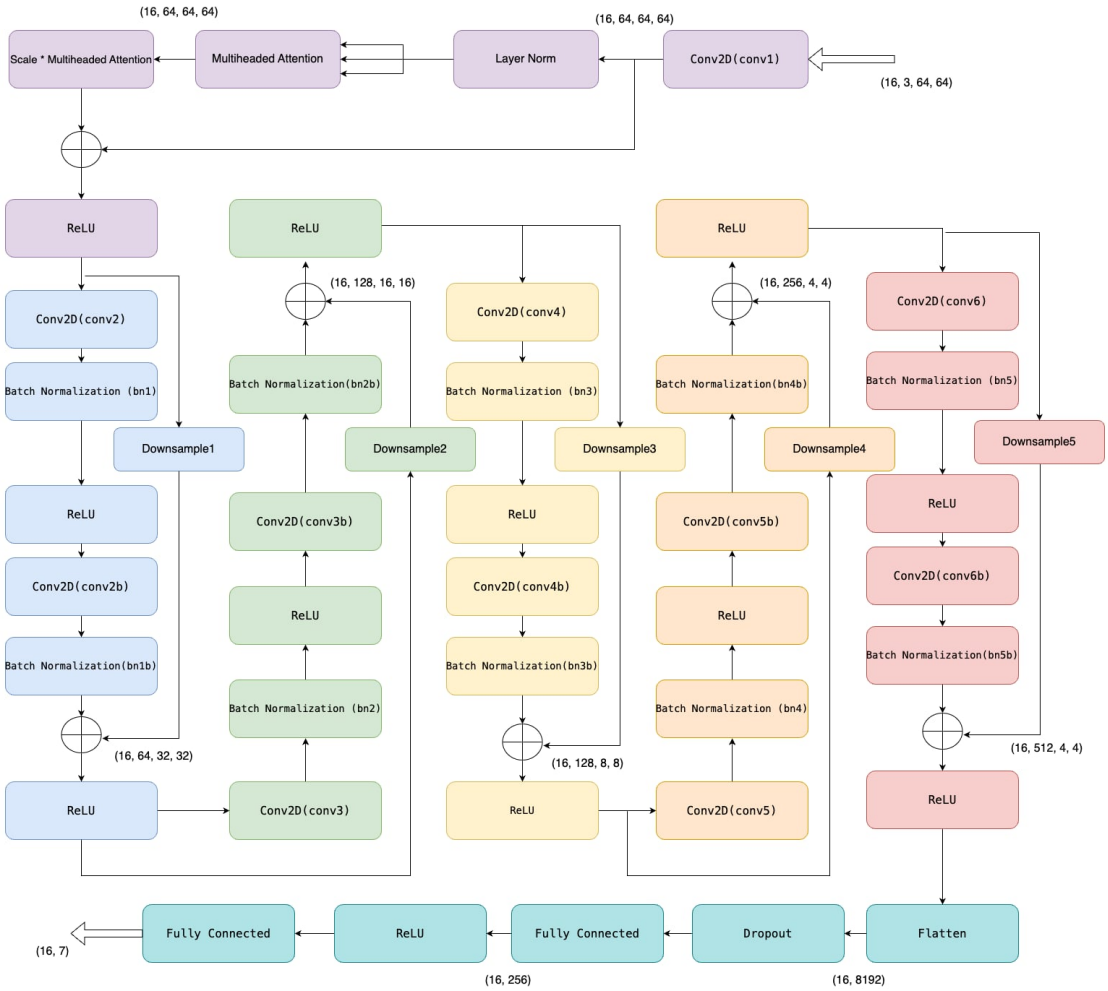

Figure 4.7: AResNN Architecture

4.5.6 Grad-CAM and Grad-CAM++

Gradient-Weighted Class Activation Mapping (Grad-CAM) is a well-known and versatile explainable artificial intelligence (xAI) technique used in the field of deep learning to visualize and comprehend the decisions made by CNNs without altering model architecture or compromising the accuracy. It identifies the areas of importance from the input for making decisions by computing gradients. Basically it breaks down the importance of each feature map for a particular class. The resultant Grad-CAM heatmap highlights the areas on the input that are most important for making the decision by analyzing the gradients, obtained from backpropagation,

passing into the final convolutional layer of the CNN. By calculating the gradient of the predicted class score regarding the feature maps of the final convolution layer, it determines the significance of each feature map for a particular class. The Grad-CAM output is later modified by adding a ReLU function to it to identify specific points of the image that have a positive impact on the image. The Grad-CAM is then resized to our image size and normalized so that it can be placed right on top of our image.

Now, one might wonder why we are even using a Grad-CAM. As previously mentioned, we use it to visualize and understand how the models have reached their conclusion. It addresses the interpretability of a deep learning model. Moreover, it enhances model transparency which allows researchers or users to decipher and analyze the reasoning behind a model's conclusion by producing visual explanations. Such transparency is utmost important in fields where AI systems can influence critical decisions like for instance medical diagnosis. Grad-CAM being class discriminative in nature also localizes model output by producing heatmaps that particularly highlight which parts of the image contributed the most to the model's decision, thus allowing researchers to further analyze the features in those regions. Not only this, Grad-CAM also plays a vital role in developing trustworthiness in AI systems. Surveys and researches involving human evaluations reflect that Grad-CAMs were pivotal in developing user trust in automated systems as they provided transparent cognizance into model predictions.

There is another xAI model that is also widely used called Grad-CAM++ which essentially is the same technique as the regular Grad-CAM but it utilizes "Guided Backpropagation" to backpropagate via only the positive gradients which influences the class prediction. Regular Grad-CAMs suffer in identifying and focusing when there are multiple instances of the same object or when objects have low spatial footprint. So the optimization in Grad-CAM++ resolves that issue. Grad-CAM++ gives more importance to the gradient pixels that are most appropriate for our class predictions by scaling them with a greater factor instead of a constant like in the regular version. High order gradients are used in Grad-CAM++.

For both of our XAI models we have set the class as NV. When the model successfully predicted the NV class, the Grad-CAM models also highlighted how the model made that conclusion by generating a heatmap showing which features of the image received the most attention. The Grad-CAM and Grad-CAM++ outputs have been showcased further in Chapter 5. We followed the architecture shown in the following page to achieve this:

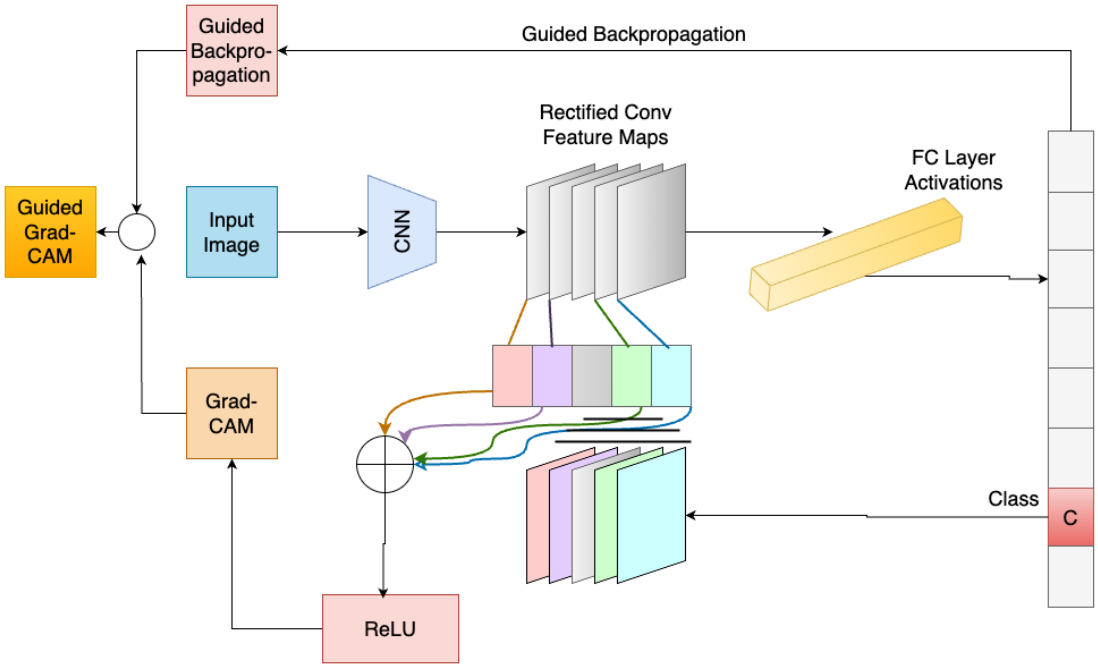

Figure 4.8: Grad-CAM Architecture

Chapter 5

Results and Discussion

5.1 Acquired Results

For our classification tasks, we have implemented five models: our main hybrid model AresNN, ResNet50, VGG16, Vision Transformer (ViT) and Swin Transformer (Swin-T). We have run each and every model for 50 epochs as a base line and more if we had observed that the accuracy function was increasing for any model. Then we thoroughly analysed and compared the results obtained and model performance for comparison with each other. This was done to identify which models were outperformed by our proposed model and which were not. We also checked if our hybrid model was more efficient in terms of number of parameters used for operations than the other models. All of the obtained values have been recorded in the Table 5.1, 5.2 and 5.3.

Our main performance metrics were accuracy, precision, loss, recall and F1-score where, TP: True Positives; TN: True Negatives; FP: False Positives; and FN: False Negatives.

1. Accuracy: It refers to the percentage of correct predictions the model can make. The formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2. Precision: It refers to the percentage of actual "true positive" instances out of all true and false positives. This ensures that the predictions are correct and close to the true value. The formula is as follows:

$$Precision = \frac{TP}{TP + FP}$$

3. Recall: It refers to the amount of times the model correctly identifies "true positives" from all the actual positive samples in the dataset. The formula is as follows:

$$Recall = \frac{TP}{TP + FN}$$

4. Loss: It is a measure by which it is checked if a model's prediction match with the actual true outcomes..

5. F1-score: It refers to the harmonic mean of the precision and recall scores of a model. The formula is as follows:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Model	Epochs	Accuracy in %	Loss	Precision	Recall	F1-Score
ResNet50	50	82.04	0.4635	0.8613	0.7846	0.8212
VGG16	50	77.95	0.6002	0.8456	0.7122	0.7732
ViT	50	74.34	0.5501	0.7110	0.7433	0.7114
Swin-T	50	72.16	1.030	0.8405	0.5932	0.6677
AResNN	50	76.92	0.4051	0.7513	0.7742	0.7434
AResNN (TL)	50	82.10	0.3655	0.8205	0.8311	0.8115

Table 5.1: Results during Training Phase

Model	Epochs	Accuracy in %	Precision	Recall	F1-Score
ResNet50	50	71.73	0.7508	0.6833	0.7155
VGG16	50	75.85	0.8333	0.7064	0.7646
ViT	50	73.35	0.6907	0.7307	0.7007
Swin-T	50	69.73	0.8516	0.5344	0.6682
AResNN	50	73.65	0.7231	0.7399	0.7110
AResNN (TL)	50	78.94	0.7756	0.7923	0.7738

Table 5.2: Results during Validation Phase

Model	Number of Paramaters (in millions)
ResNet50	24
VGG16	14
AResNN	5
ViT	1
Swin-T	0.074

Table 5.3: Number of Parameters for Operations

For our ResNet50 model, which was trained over 64x64 images with a batch size of 16 and a learning rate of 0.001 and CCE as our loss function, we obtained the following results in the training phase: 82.04% accuracy, a loss of 0.4635, a recall of 0.7846, a precision of 0.8613 and a pretty high F1-score of 0.8212. And, as for validation, we obtained: 71.73%, a recall of 0.6833, a precision of 0.7508 and F1-score of 0.7155. We have observed that there is a noticeable dip in the recall value

here during the validation phase which might indicate that the dataset has one or more underwhelming classes due to which the result might skew towards the major class. This model also has a significantly large number of parameters i.e. 24 million parameters which shows how heavy the model is. The graphical representations are as follows:

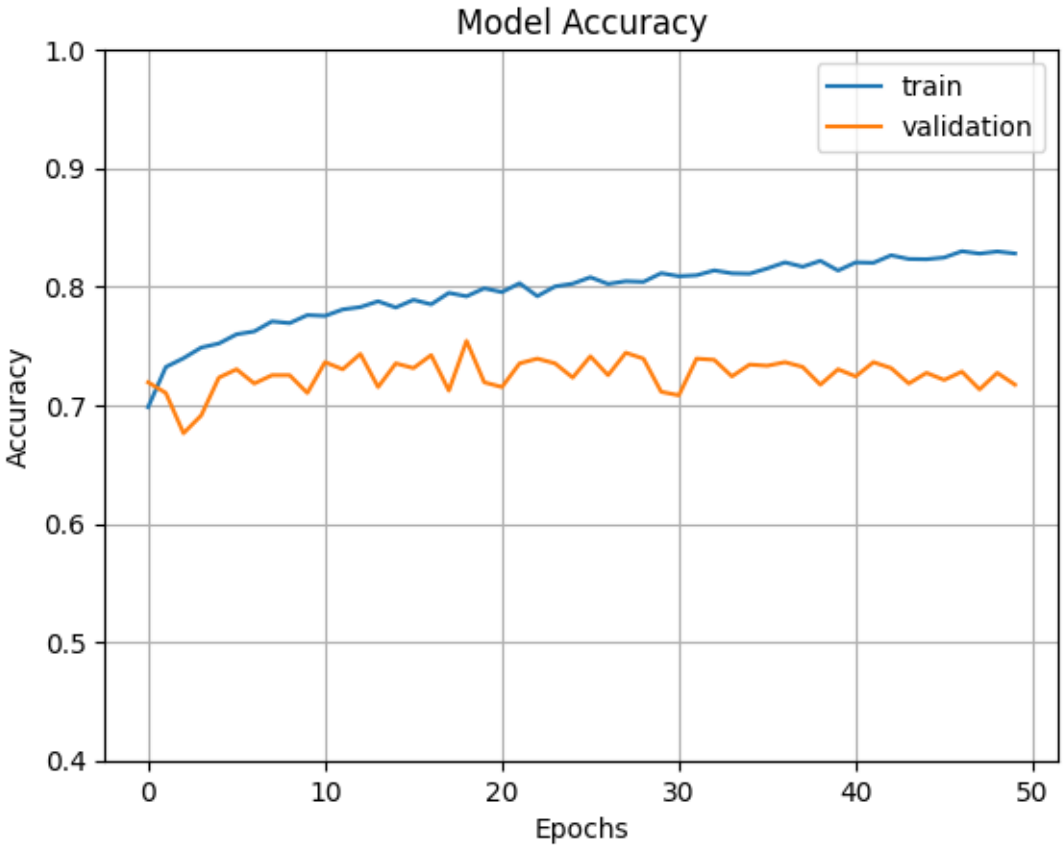

Figure 5.1: ResNet50 Accuracy Graph

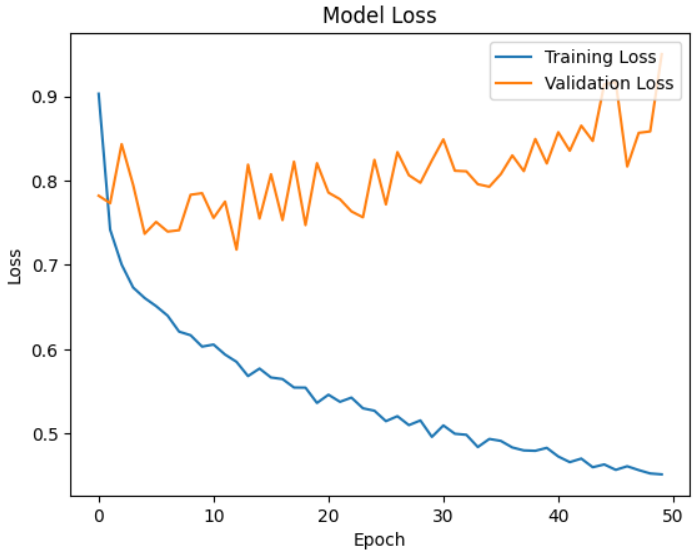

Figure 5.2: ResNet50 Loss Graph

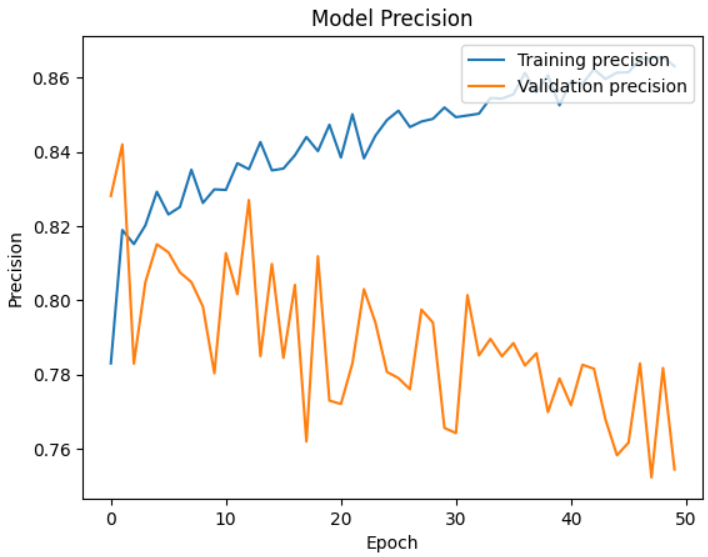

Figure 5.3: ResNet50 Precision Graph

Figure 5.4: ResNet50 Recall Graph

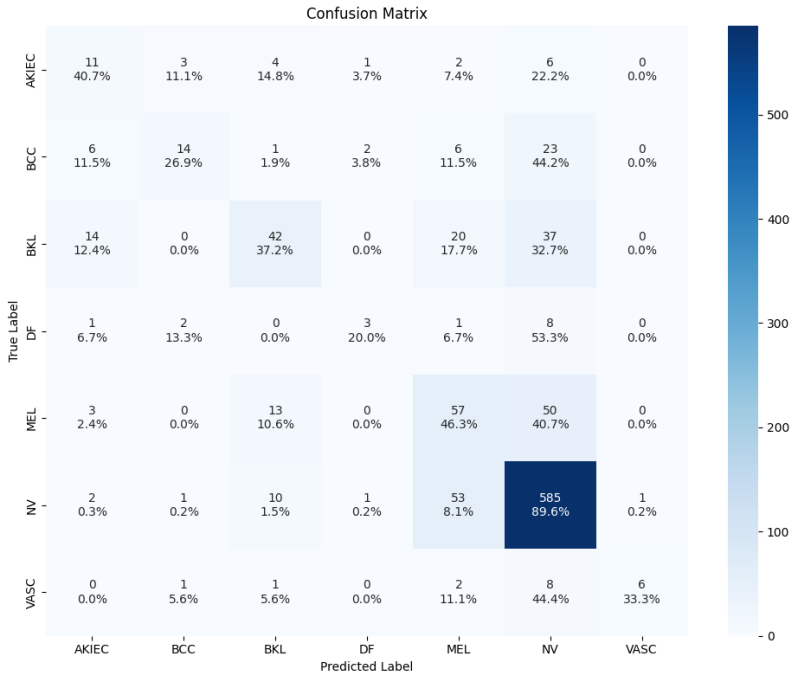

Figure 5.5: ResNet50 Confusion Matrix

For our VGG16 model, which was also trained over 64x64 images with the same batch size and learning rate used for ResNet50 and just like before as well, CCE as our loss function. We obtained the following results in the training phase: 77.95% accuracy, a loss of 0.6002, a recall of 0.7122, a precision of 0.8456 and a pretty high F1-score of 0.7732. And, as for validation, we obtained: 75.85%, a recall of 0.7064, a precision of 0.8333 and F1-score of 0.7646. All in all, the values obtained in both the phases seem to be consistent with each other. This model has 14 million parameters which is significantly lower than the ResNet50 indicating this model is lighter. Let us take a look at the graphs:

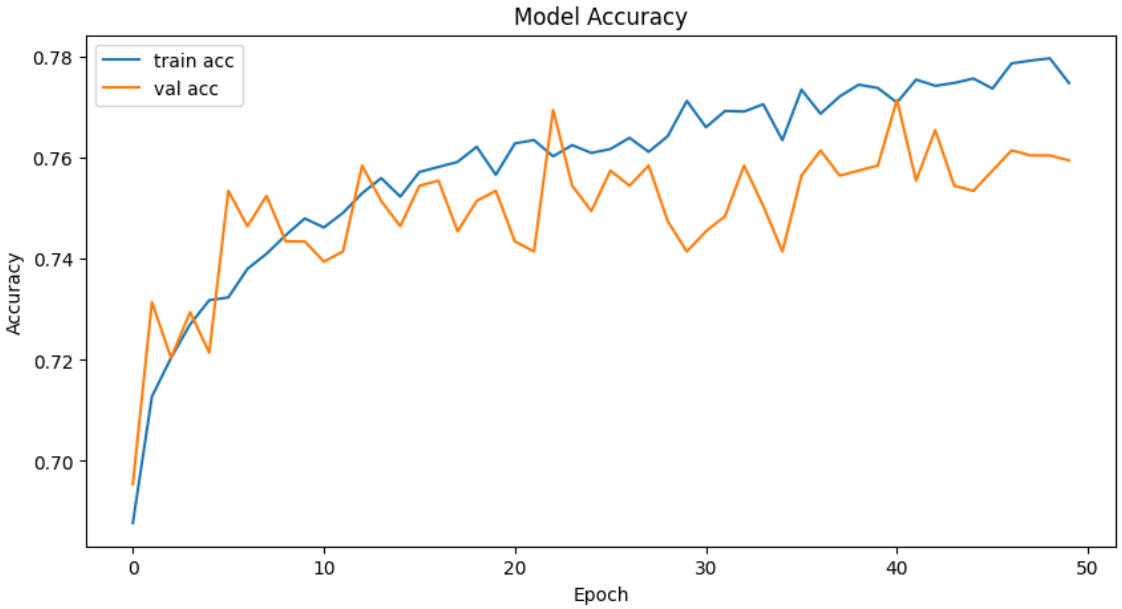

Figure 5.6: VGG16 Accuracy Graph

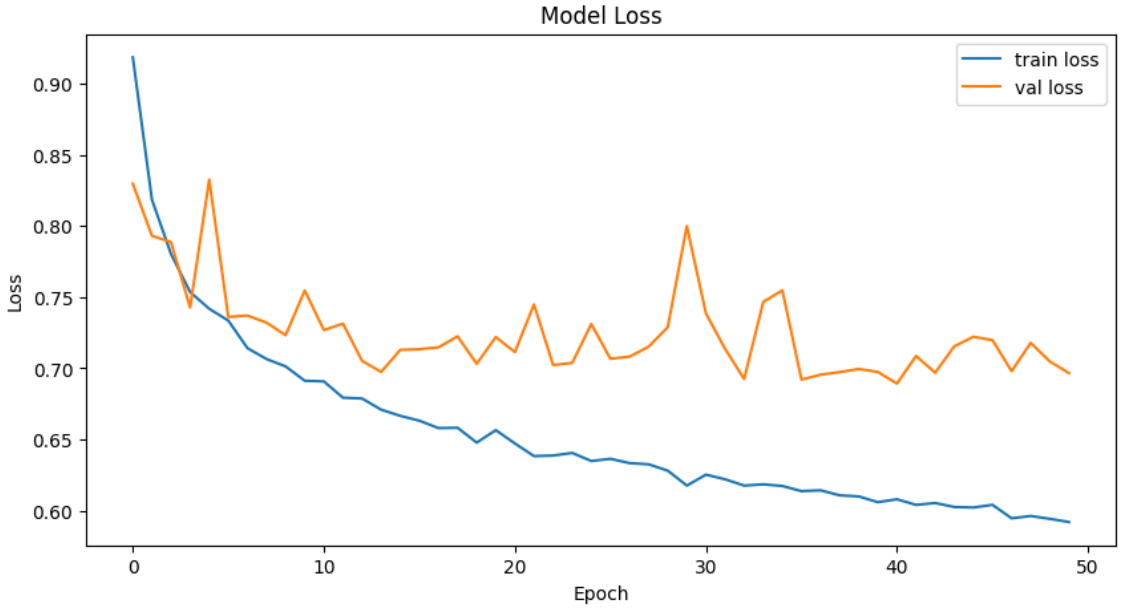

Figure 5.7: VGG16 Loss Graph

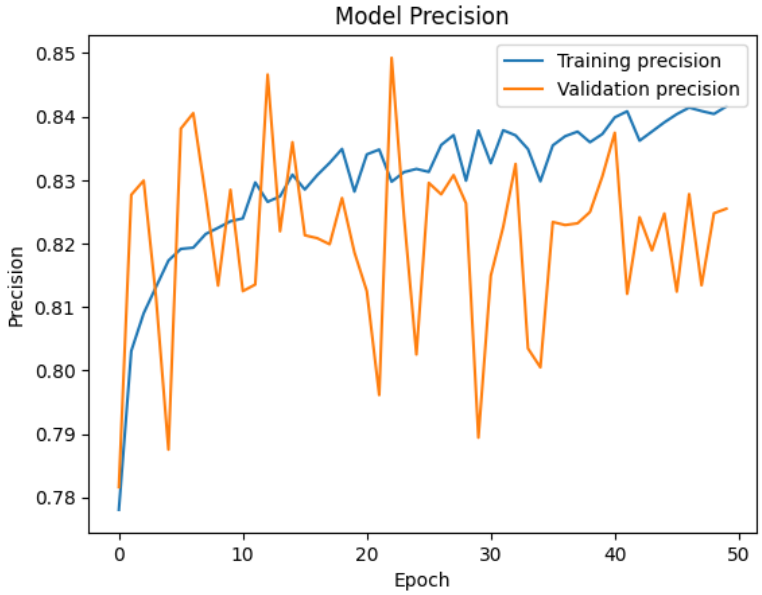

Figure 5.8: VGG16 Precision Graph

Figure 5.9: VGG16 Recall Graph

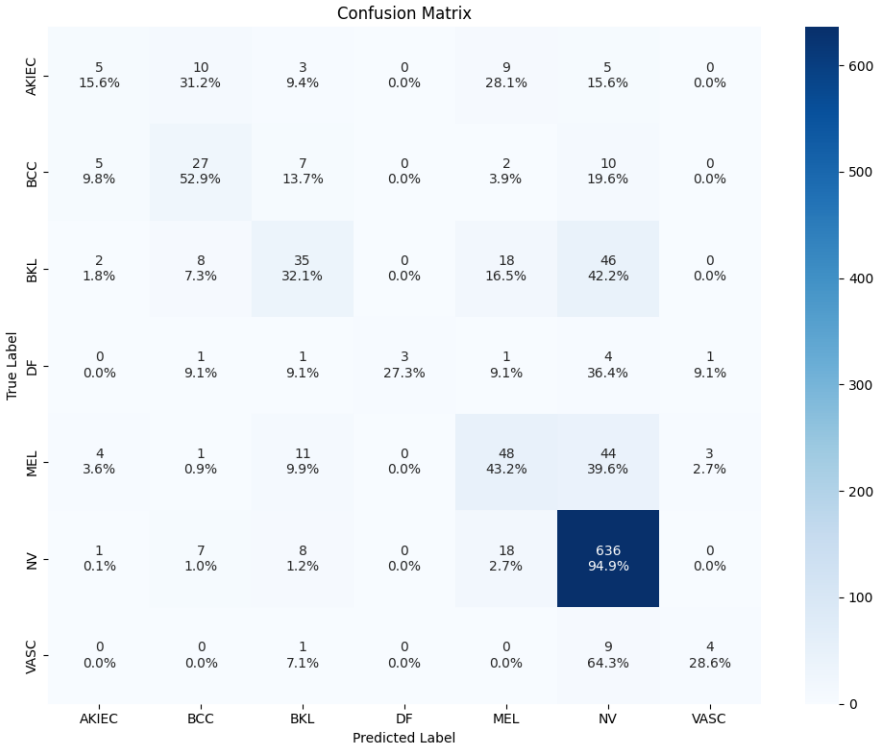

Figure 5.10: VGG16 Confusion Matrix

Now, let us move on to our transformer models. Before we go for the results, an important point to note is that the transformers were not pre-trained and made from scratch so there will be a noticeable drop in training accuracy as it was only trained in our dataset and not on other larger dataset. This is a limitation that we will focus on in a later section. First, let us take a look at our Vision Transformer (ViT). It was trained over 64x64 images which were divided into 16x16 sized patches leading to 16 patches in total. Similar to our CNN models, batch size was 16 and learning rate was 0.0001. During training, we obtained the following results: 74.34% accuracy, a loss of 0.5501, a recall, precision and F1-score of 0.7433, 0.7110 and 0.7114 respectively. For validation, we obtained: 73.35% accuracy, a recall, precision and F1-score of 0.7307, 0.6907 and 0.7007 respectively. Lastly, the ViT model only uses a million parameters for its operations. The graphs are as follows:

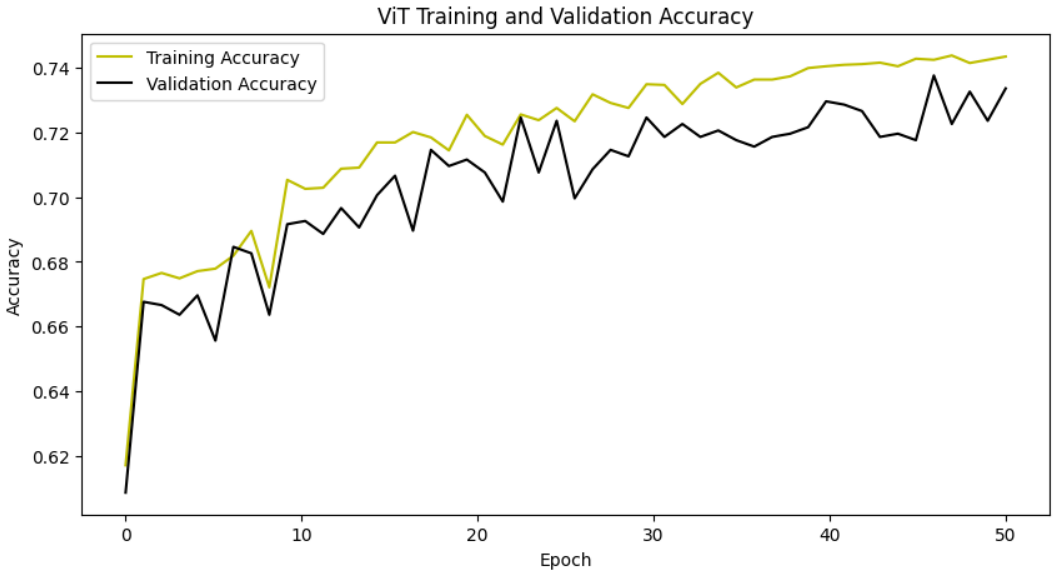

Figure 5.11: ViT Accuracy Graph

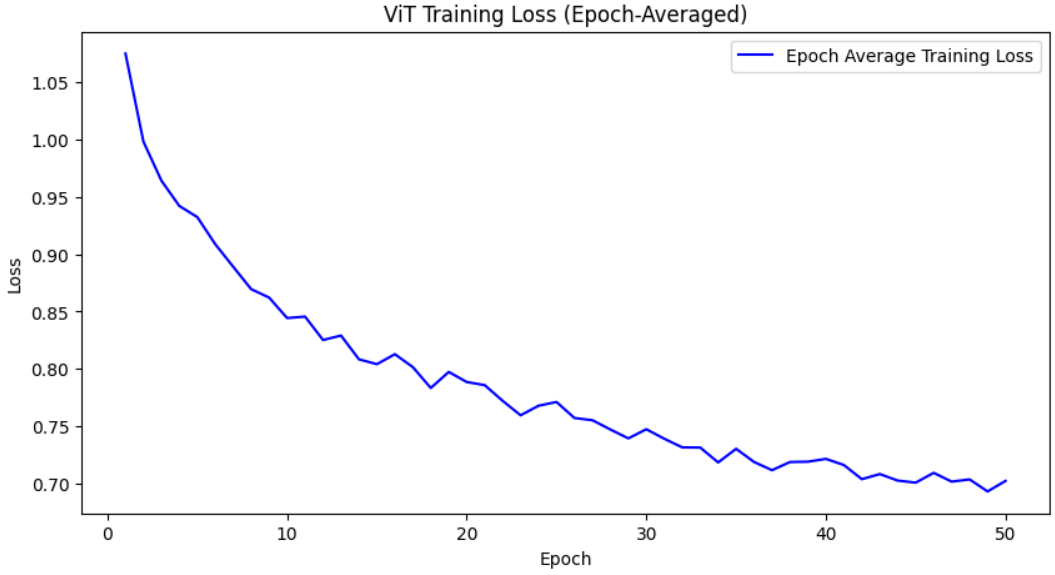

Figure 5.12: ViT Loss Graph

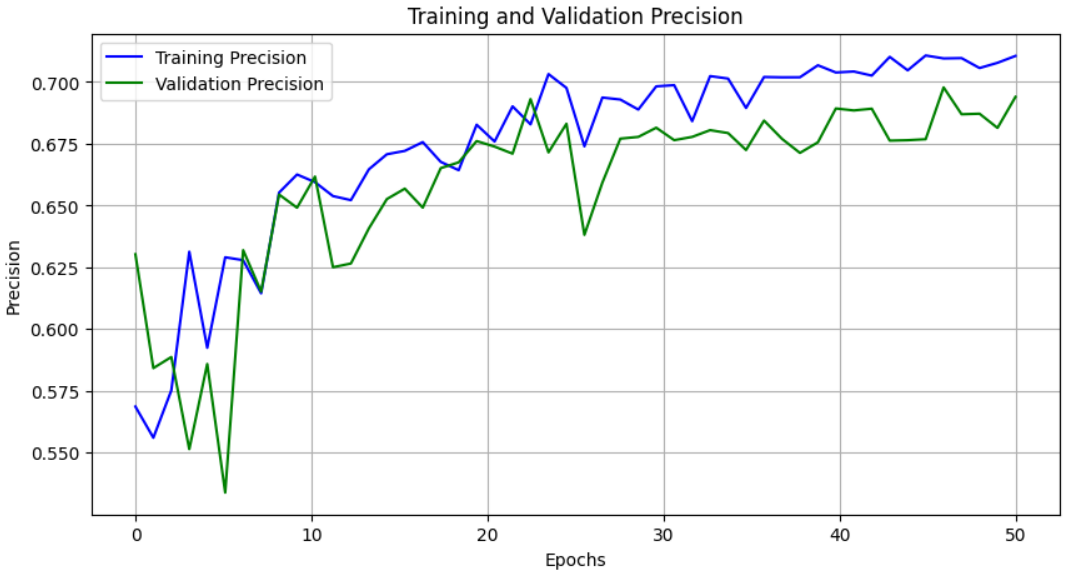

Figure 5.13: ViT Precision Graph

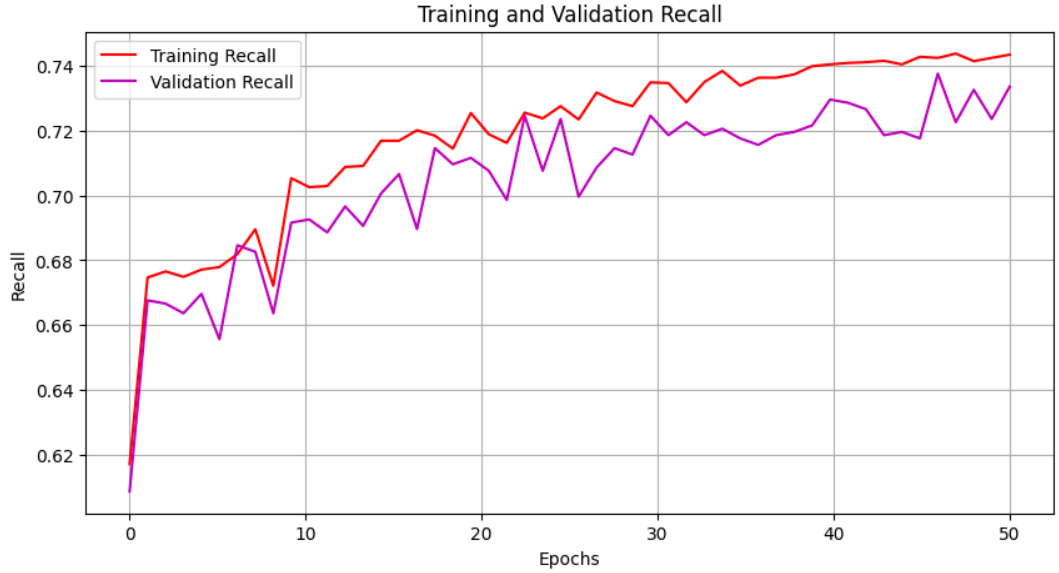

Figure 5.14: ViT Recall Graph

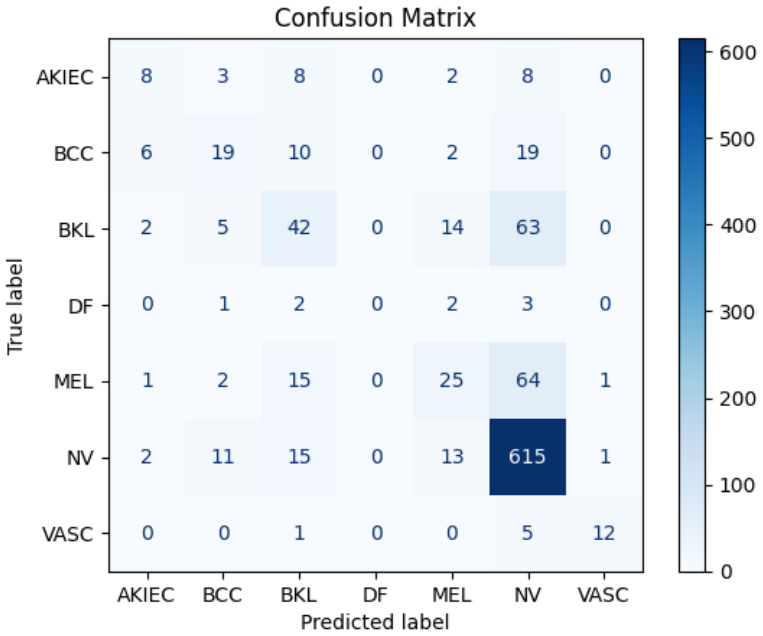

Figure 5.15: ViT Confusion Matrix

For the Swin-T model, however, we had to use 224x224 images in accordance with the architecture for training with a batch size of 16, 196 patches, a learning rate of 0.001 and weight decay of 0.0001. During the training phase, we acquired the following: 72.16% accuracy, a loss of 1.030, a recall, precision and F1-score of 0.5932, 0.8405 and 0.6677 respectively. For validation, we recorded the following: 69.73% accuracy, a recall, precision and F1-score of 0.5344, 0.8516 and 0.6682 respectively. All in all, we observed that this model was not performing as well as the others in terms of generating accurate predictions for our target classes. Finally, in comparison to the ViT, it has a very low number of parameters for its operations i.e. only 74,000 but in terms of performance metrics and accurate results, the ViT is superior. Since the positive samples of 6 of our classes are relatively lower in our dataset, models may tend to favor precision by avoiding false positives, at the cost of missing many true positives leading to the lower recall value. However, it could also mean that there might be some noisy samples which makes the model more cautious about predicting positive samples. Therefore, again, precision increases and recall decreases.

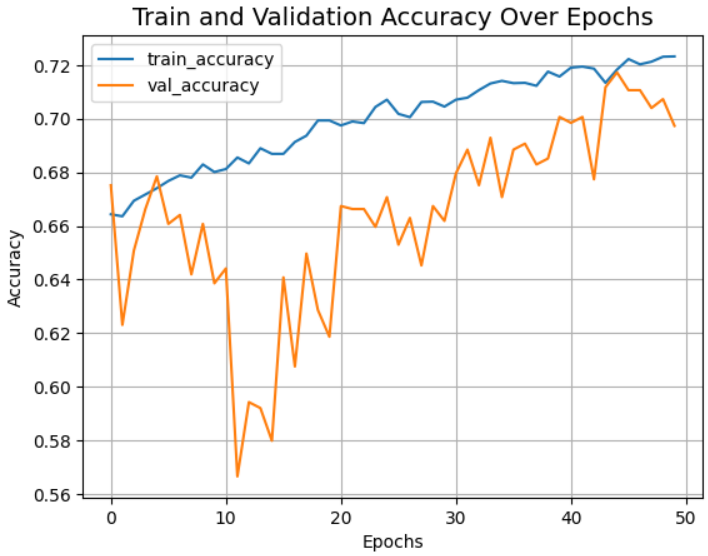

Figure 5.16: Swin Accuracy Graph

Figure 5.17: Swin Loss Graph

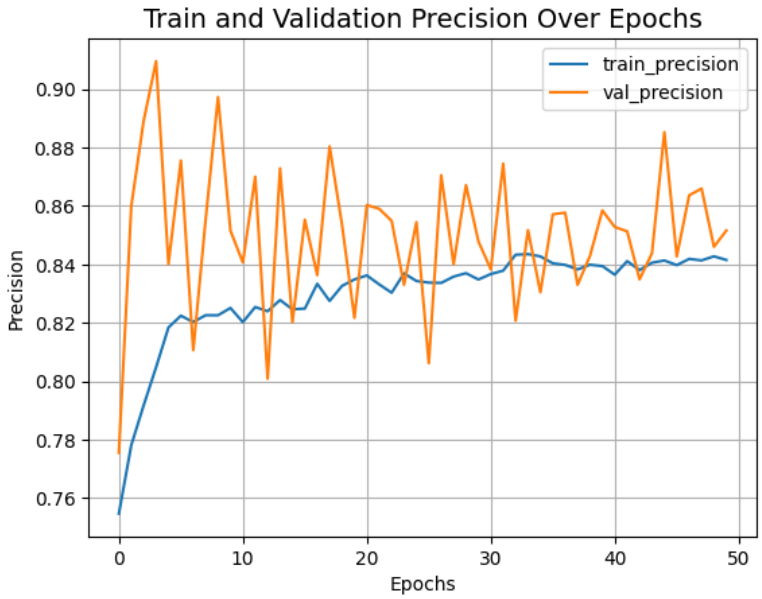

Figure 5.18: Swin Precision Graph

Figure 5.19: Swin Recall Graph

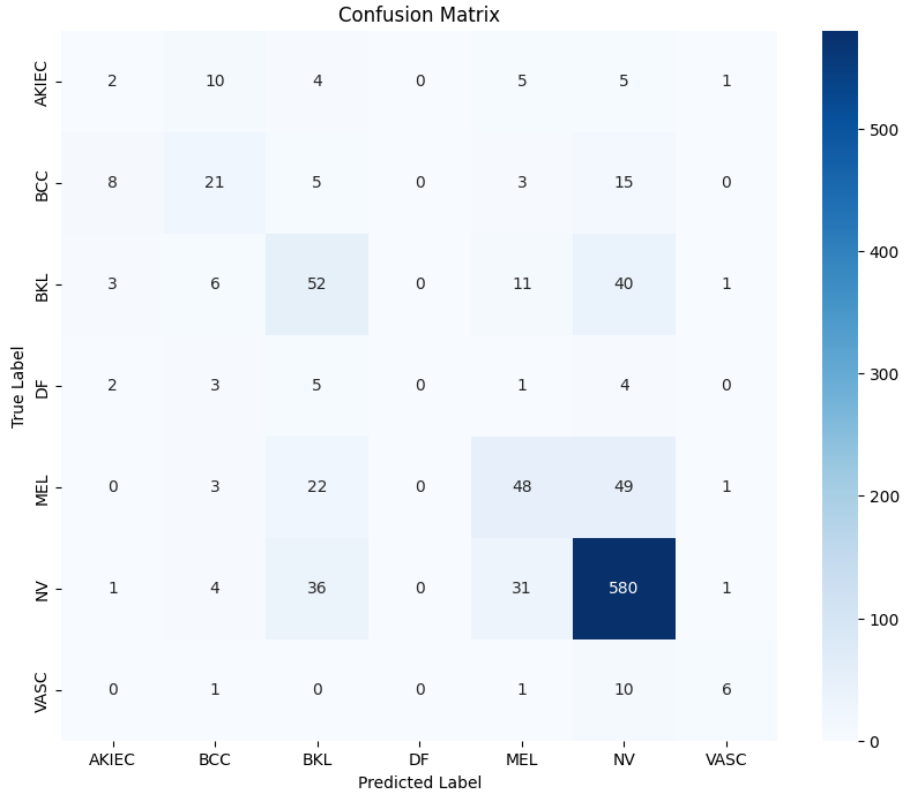

Figure 5.20: Swin-T Confusion Matrix

Now, we move to our proposed model, AResNN, which again is made from scratch and was not trained on any other dataset other than HAM10000 dataset, the AresNN, that was trained over 64x64 images with batch size of 16 and learning rate of 0.0001 like most of our other models. During the training phase, we obtained the following: 76.92% accuracy, a loss of 0.4051, a recall, precision and F1-score of 0.7742, 0.7513 and 0.7434 respectively. For validation, we obtained: 73.65% accuracy, a recall, precision and F1-score of 0.7399, 0.7231 and 0.7110 respectively. To sum up, all of the results seem to be consistent and the performance is pretty good and consistent. The number of parameters in this model was 5 million which is significantly less than the ResNet50 and VGG16 models.

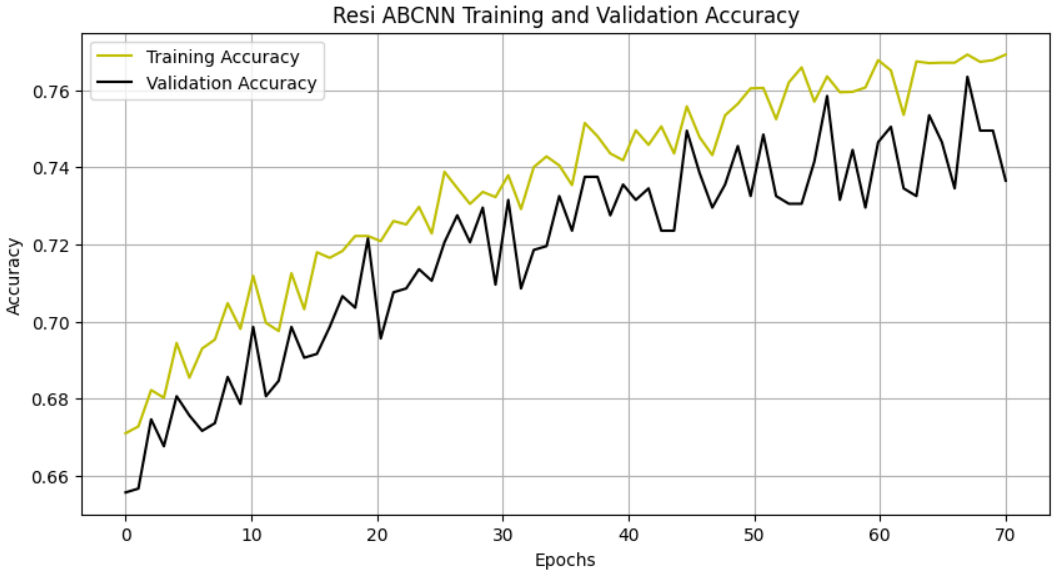

Figure 5.21: AResNN Accuracy Graph

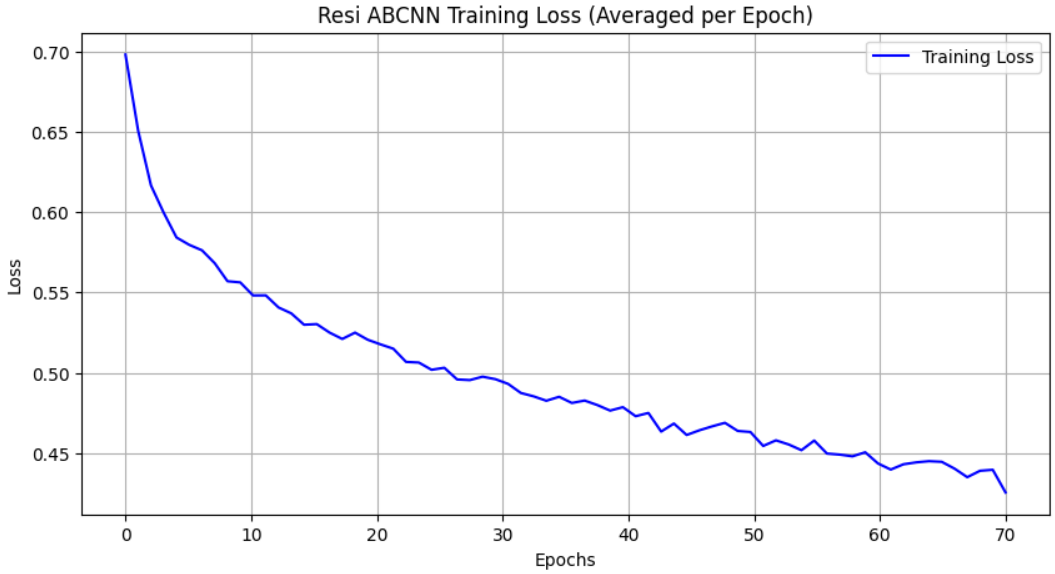

Figure 5.22: AResNN Loss Graph

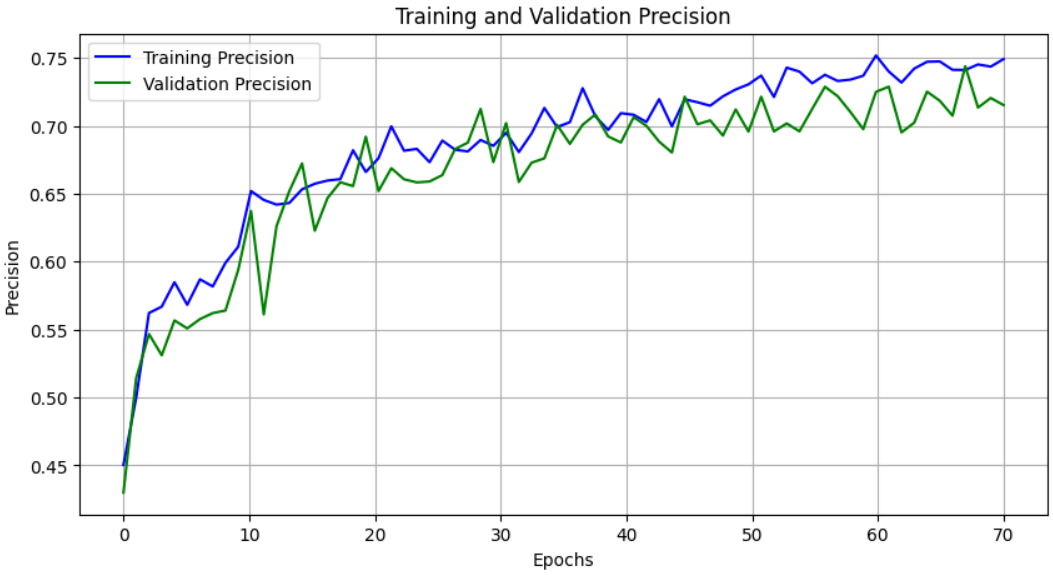

Figure 5.23: AResNN Precision Graph

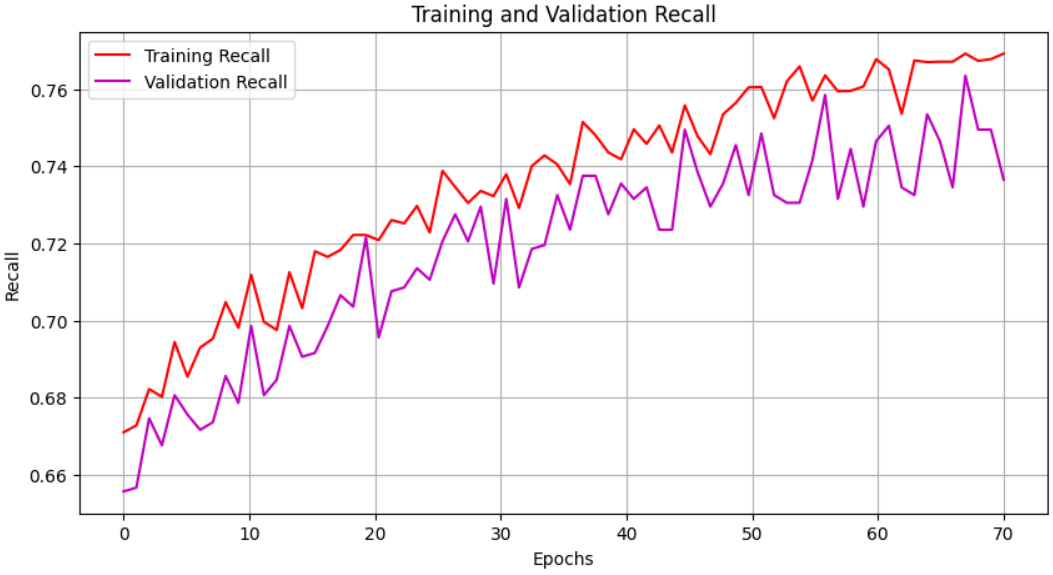

Figure 5.24: AResNN Recall Graph

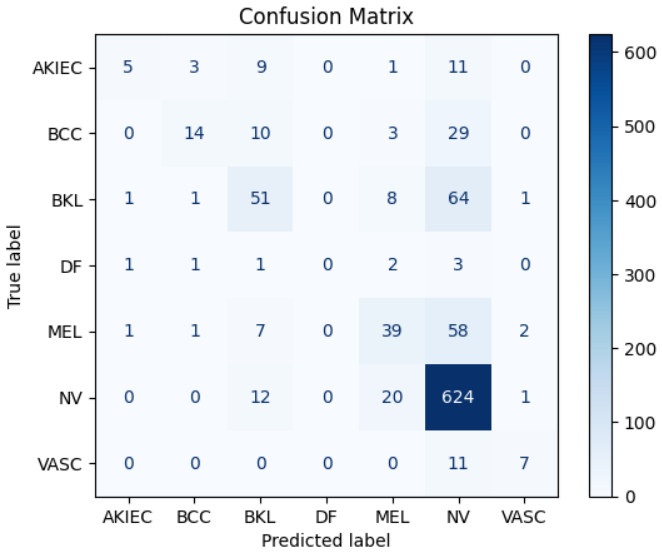

Figure 5.25: AResNN Confusion Matrix

Finally, we see that our proposed model AResNN, which utilized the transfer learning mechanism on the ISIC 2019 dataset before it was trained on the HAM10000 dataset, provided us with the superior result. During the training phase, we obtained 82.10% accuracy, a loss of 0.3655 and for recall, precision and F1-score we obtained 0.8311, 0.8205 and 0.8115 respectively. For validation, we obtained: 78.94% accuracy, a recall, precision and F1-score of 0.7923, 0.7756 and 0.7738 respectively.

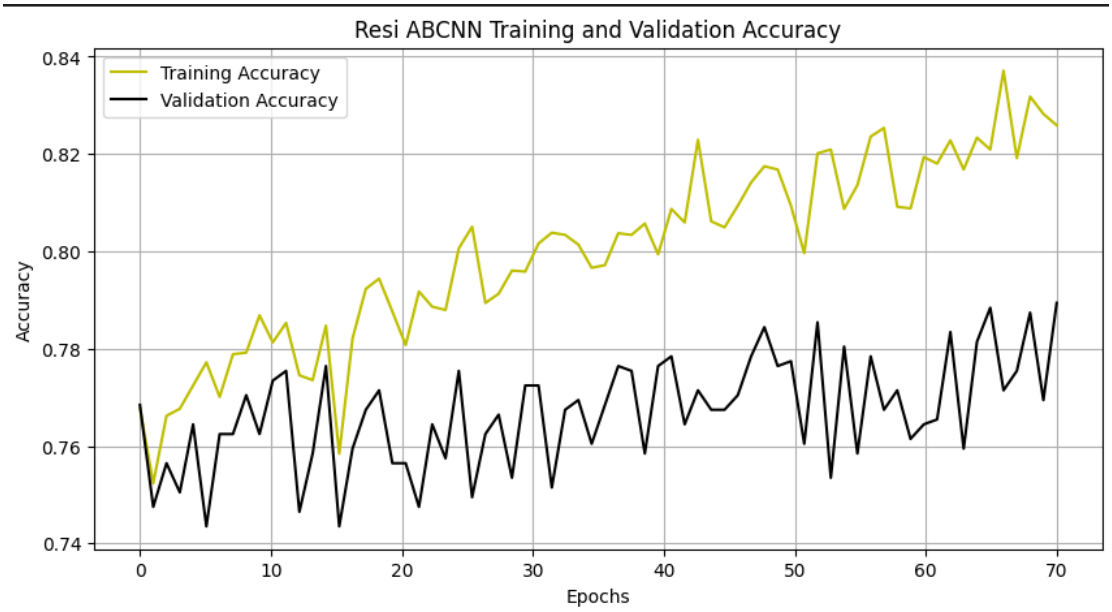

Figure 5.26: AResNN [with TL] Accuracy Graph

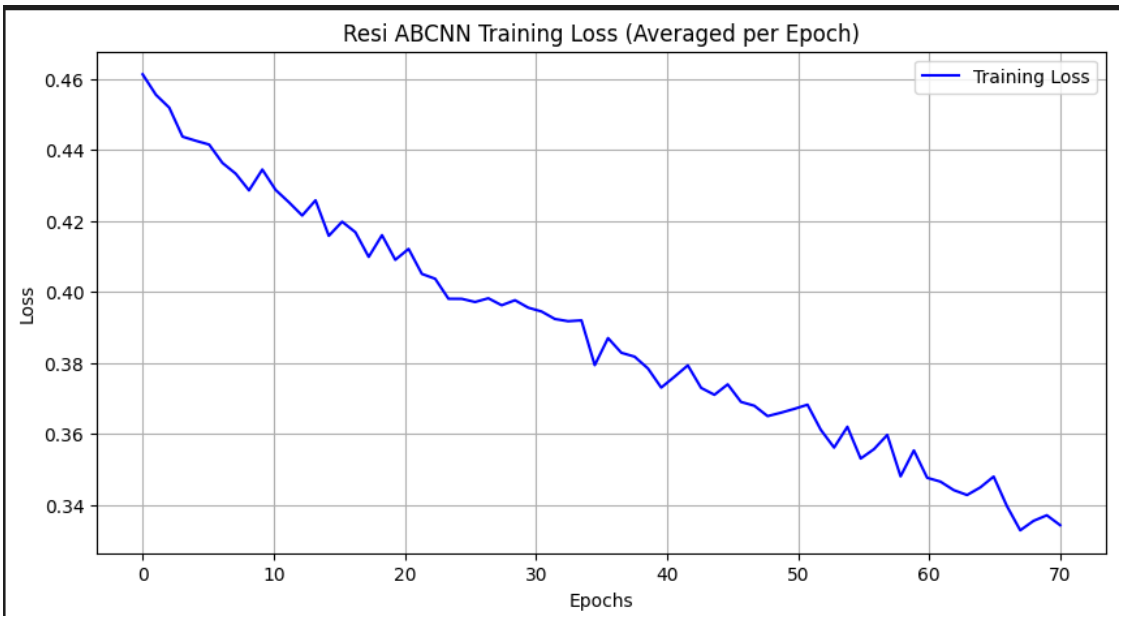

Figure 5.27: AResNN [with TL] Loss Graph

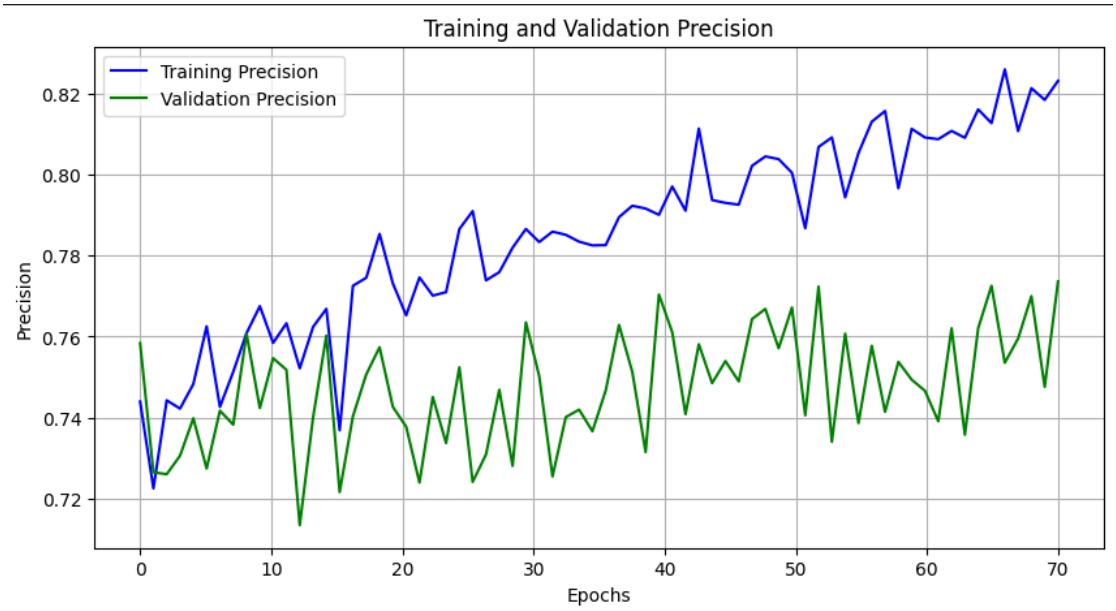

Figure 5.28: AResNN [with TL] Precision Graph

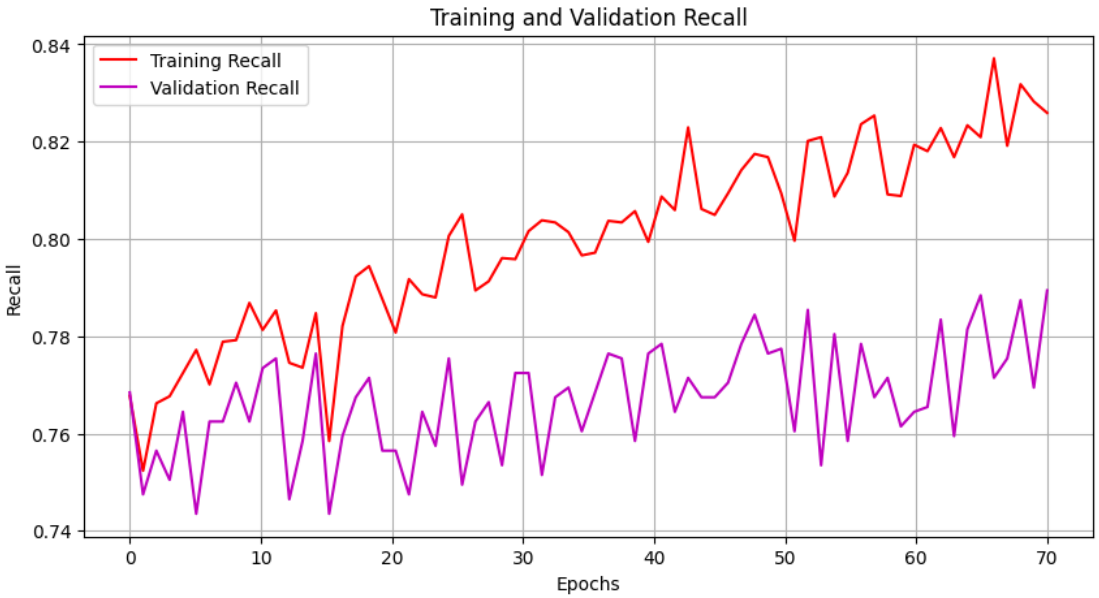

Figure 5.29: AResNN [with TL] Recall Graph

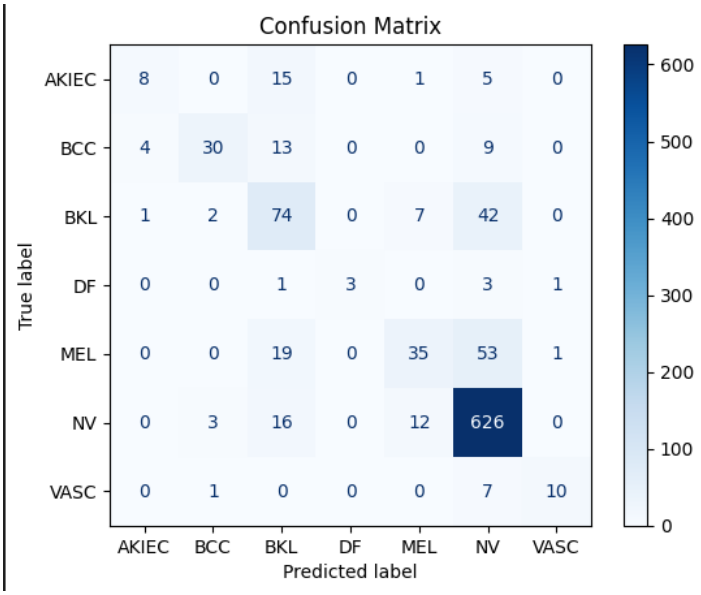

Figure 5.30: AResNN [with TL] Confusion Matrix

The parameter is an innate model configuration that is learned or estimated during training. The number of parameters cannot be initialized and are affected by the hyperparameter values chosen. The number of parameters keeps changing as the model learns from the data and reaches its final value only when the training is complete. Simpler models tend to have very low parameter numbers like regression and Naive Bayes but complex architectures like neural networks contain a large number of parameters. For example, the parameters in neural networks contain main weights and biases. For a typical neural network, the weight parameter number would be n*p where n is the layer input and p is the number of neurons. Each neuron would have its bias, so for each p, there would be a p bias. Therefore, the total parameters would be approximately (n*p) + p. The complexity thereby increases with each addition of layers, where with each layer the total increases by (n*p) + p parameters. The number of parameters not only affects model complexity but also affects the risks of overfitting and computational efficiency. With more parameters, a model can capture more intricate patterns from the data but if the parameters are too much then the risk of overfitting increases and so does the noise accumulation. On the other hand, a model with less parameters is harder to overfit as it has limited learning capacity. Finally, the more the parameters, the less the computational efficiency for the model. Therefore, in practice, a computationally heavy model will be less feasible to deploy and will cost more resources for processing and deployment.

Finally, let us discuss how our proposed model performs against all the other models used for our research purpose. The training accuracy that we achieved for AresNN (without TL) outperformed both the transformer models and was almost the same as the VGG16 model but could not outperform the ResNet50 model in this regard as the ResNet50 model was pretrained on a huge dataset. However, our model does not suffer from a high training loss like the Swin-T and is significantly lower than all the other models. In terms of validation accuracy, our model, AResNN (without TL), outperformed all the other models except VGG16 by a 2% margin but AResNN is significantly more computationally efficient than it as it has 9 million less parameters. Even though the Swin-T uses minimal parameters, its performance was not up to our model's standards and suffered from a high training loss. Our model's F1-score that summarizes the performance of the model as a whole was better than the transformers and on par with the ResNet50 and VGG16 model. Lastly, the AResNN model that used Transfer Learning visibly outperformed in all factors among the rest of the models. It had the privilege of learning from 2 vast datasets allowing it to gain knowledge of intricate details. This means that the ISIC 2019 dataset allowed for the model to gain a general idea of skin cancer images and it had an easier time to converge during the training process of HAM10000 dataset allowing for the large jump in performance. All in all, we can conclude that the AResNN model is the best choice here as it has good accuracy, consistent performance and is computationally efficient and less costly.

5.2 Ablation Study for AResNN

Ablation studies is a process by which a machine learning model is dissected and the influence of each component is evaluated. By eliminating certain features or

layers within the machine learning model and recording the new results or changes in performance, we can evaluate the importance of each component in achieving the required output.

Ablation studies help us to look at each component in a model individually to assess its importance or impact on the complex model's performance. Through this hard, rigorous approach, we can judge which components are influential enough to be considered an important factor in the model and which may be just redundant features. This model analysis can be a guidance for reproducing the model for different tasks by providing a path for model optimization and establishing reliability of research findings.

Ablation	Component Removed	Parameters (in millions)	Training Accuracy	Validation Accuracy	Loss	F1-Score
0	None	5	82.10	78.94	0.365	0.81
1	TL	5	76.92	73.65	0.381	0.74
2	Class Weights	5	75.18	73.08	0.99	0.73
3	Multi Headed Attention	5	73.62	72.75	0.99	0.72
4	Residual Block	2	72.55	71.46	0.99	0.72

Table 5.4: Ablation Table

An ablation study was conducted on our proposed model i.e. the AResNN. In each ablation, we removed a certain component and trained the entire model again to record what changes had occurred. First, we removed the Transfer Learning mechanism, which means we only trained the model on the HAM10000 dataset. This caused a significant drop in training and validation accuracy from 82.1% to 76.92% and 78.94% to 73.65% respectively. It also had a major impact on the F1-score where it dropped from 0.81 to 0.74. However, the loss value was only slightly impacted. Then, we removed the class weights component from our model which essentially meant training on an imbalanced dataset. This ended up increasing the loss per epoch reaching a value of 0.99 which essentially meant that the model had started generating erroneous output while reducing the training accuracy and validation accuracy to 75.18% and 73.08% respectively. The other performance metrics were not affected significantly. We noticed a slight 0.01 change for most of the metrics. However, all the values have still been recorded and can be seen in the ablation table for reference. Next, the Multi Headed Attention component was removed which meant the model did not have its multiple heads to properly focus on different features of the image individually which resulted in a significant drop in accuracy to 73.62% and 72.75% for training and validation accuracy respectively. Finally, we removed a residual block and some convolutional layers along with it and made the convolutional block smaller which led to a drastic drop in the number of parameters for the model i.e. it fell from 5 million to 2 million but at the cost of a significant decrease in training and validation accuracy which are 72.55% and 71.46% respectively.

5.3 Grad-CAM and Grad-CAM++ Outputs

The figures below show the heatmap outputs for the given models. The outputs for both of XAI models are consistent with the NV class from our dataset. However, we

observed that Grad-CAM++ concentrates more on the specific features of the actual skin lesion while Grad-CAM sometimes seems to deviate away from the actual lesion.

(a) AResNN

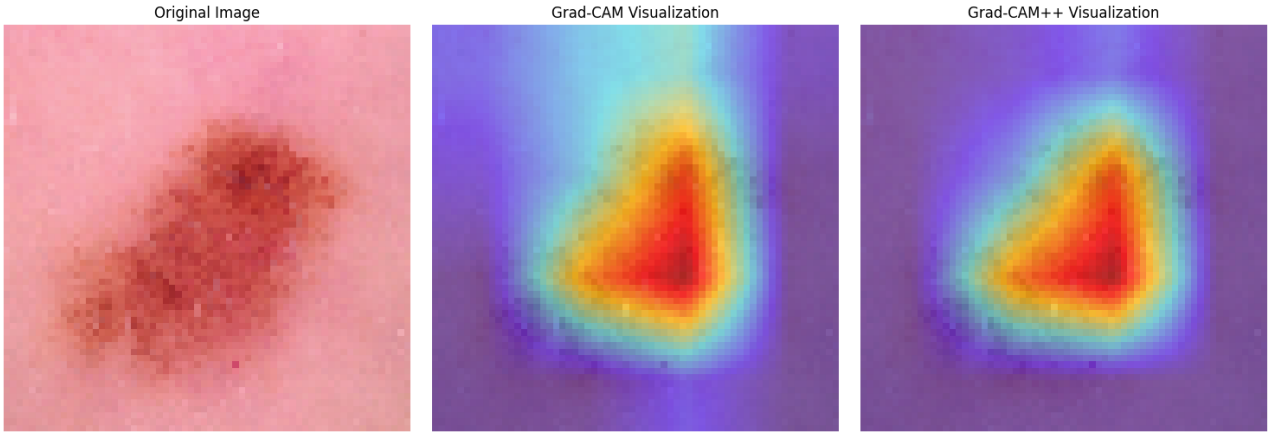

Figure 5.31: Heatmap output for AResNN

(b) ResNet50

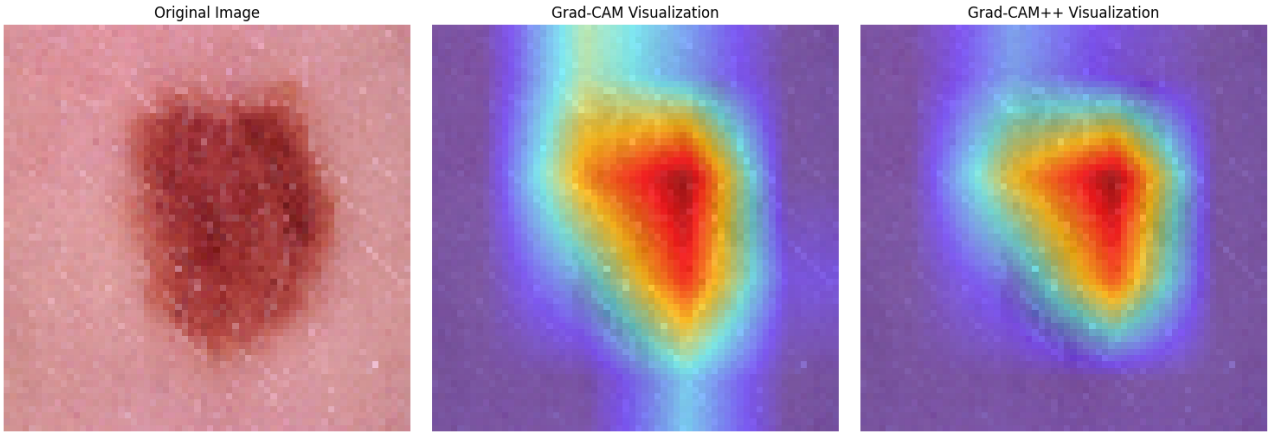

Figure 5.32: Heatmap output for ResNet50

(c) VGG16

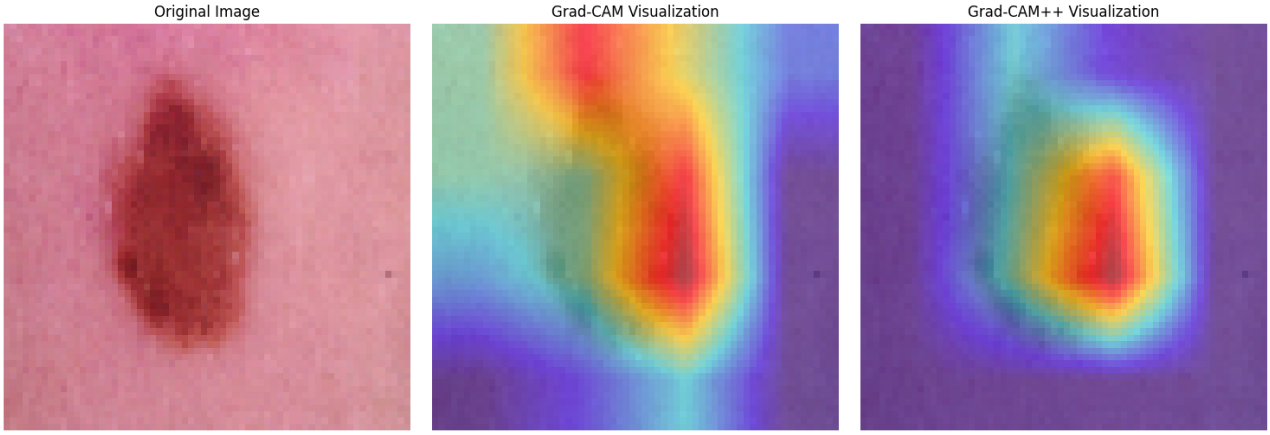

Figure 5.33: Heatmap output for VGG16

(d) Vision Transformer (ViT)

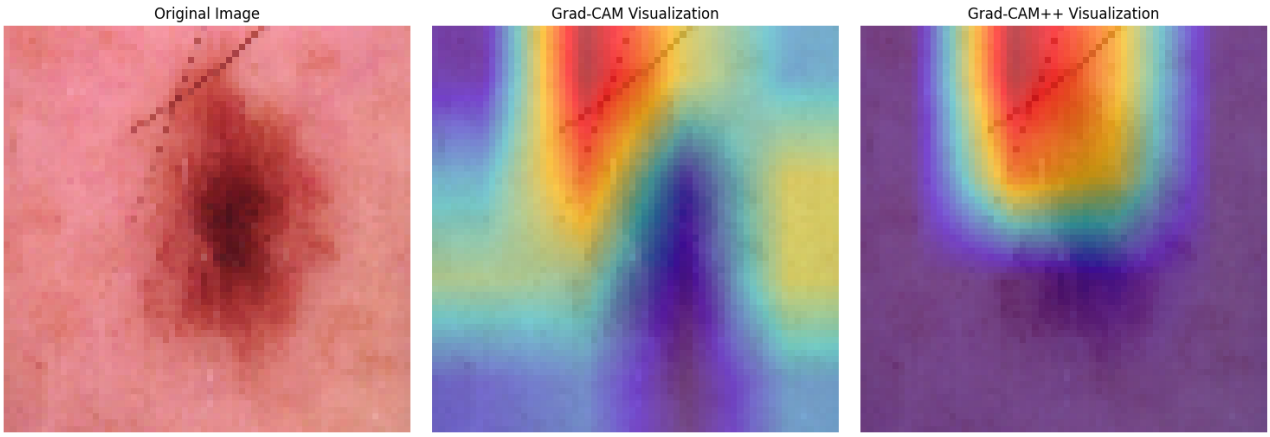

Figure 5.34: Heatmap output for Vision Transformer

(e) Swin Transformer (Swin-T)

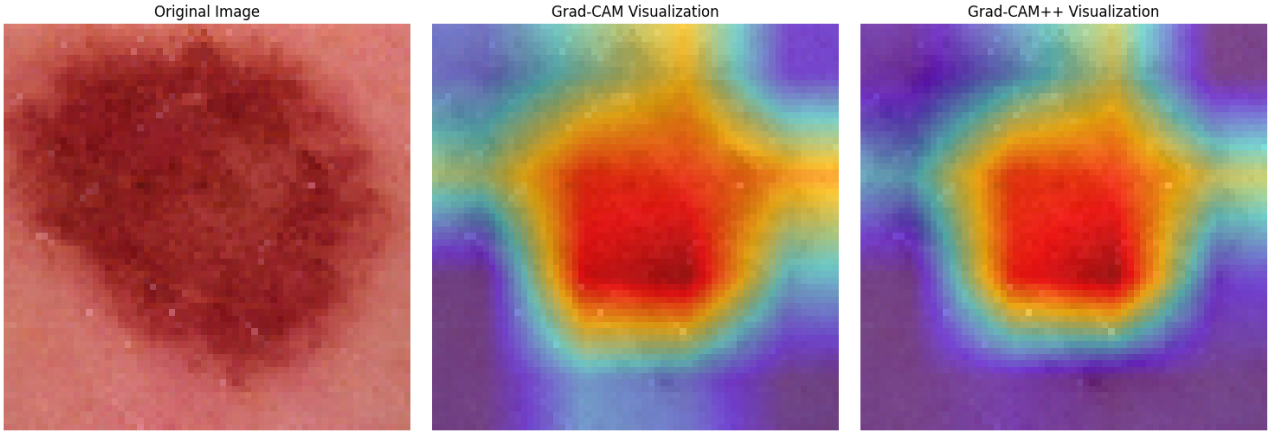

Figure 5.35: Heatmap output for Swin-T

Chapter 6

Conclusion

While AresNN demonstrates strong performance in classifying skin cancer images, it has several limitations that must be acknowledged. One key limitation is its ability to distinguish between visually similar skin cancer types. Some categories, such as melanoma (MEL) and basal cell carcinoma (BCC), or actinic keratosis (AKIEC) and benign keratosis (BKL), may share overlapping features that make precise classification challenging. This issue underscores the importance of professional medical evaluation, as AresNN should not be solely relied upon for diagnosis.

Real-world conditions present another significant challenge. The model was trained on a curated dataset where images were captured under controlled conditions with uniform lighting and consistent angles. However, in practical applications, images may be taken from various perspectives, under different lighting conditions, and with varying degrees of focus, all of which can impact model performance. Furthermore, the dataset used primarily consists of images from individuals with lighter skin tones. This lack of diversity in skin pigmentation raises concerns about the model's generalizability to individuals with darker skin tones, as different skin types may present variations in lesion appearance that the model has not been adequately trained to recognize.

Another challenge faced during model development was class imbalance. Certain skin cancer categories, such as dermatofibroma (DF) and vascular lesions (VASC), had significantly fewer samples in the dataset compared to more common types like nevus (NV) and melanoma (MEL). This imbalance affected the model's ability to generalize well across all classes, leading to poorer performance in underrepresented categories. Addressing this limitation would require access to a high-quality, validated medical dataset with a sufficient number of images for all classes, which is often difficult to obtain due to privacy concerns and limited availability. While AresNN integrates attention-based convolutional neural networks to improve feature extraction and focus on relevant regions, these limitations highlight the need for further research and improvements, including dataset expansion, augmentation strategies, and potential real-world testing to assess its robustness in diverse clinical settings.

An issue of great significance was that of hardware limitations. We did not have access to larger more powerful GPUs until the tail end of our development. As a

result, this led to much longer training times, decreasing the amount of experiments we could run. Memory limitations were also a significant factor, which forced us to use 64x64 images instead of larger ones that may have allowed for more feature extraction. Attempting to run with even 128x128 images led to our kernel or the entire device to crash, limiting our scope to smaller sized images. Ultimately, AresNN should be seen as a supportive tool for dermatologists rather than a standalone diagnostic system.

Our model holds significant potential for further refinement and performance enhancement. One of the primary directions for future work involves exploring architectural modifications to optimize the model's efficiency and accuracy, by experimenting with varying sizes of convolutional blocks and adjusting the number of attention heads to identify the most effective configuration for specific tasks. The skip connections within the model could also be evaluated and redesigned to improve gradient flow and feature integration across layers. We also plan on using other attention mechanisms, such as sparse attention or axial attention, to reduce computational overhead and possibly enhancing-the model's dependency-capturing capabilities. More importantly, access to higher-end hardware will also make these experiments feasible by allowing the training and evaluation of larger, more complex model variants in shorter amounts of time.

We also plan to investigate advanced training strategies to further enhance AresNN's performance by using methods like label smoothing and weight decay. Other strategies involve learning rate scheduling, gradient clipping, and adaptive optimization methods like AdamW that might also be combined in order to stabilize training and accelerate convergence. This will help in making necessary adjustments in model learning dynamics that would make certain the performance level is as intended across a broad set of data and tasks. Simultaneously, it can allow a detailed overview of what point in its capabilities it lacks or should be adjusted.

For our future work on AresNN, we will perform rigorous evaluation in order to further push the state-of-the-art boundaries of image processing. These would lead to a robust, efficient, versatile model for a wide range of real-world challenges and a new standard in neural network-based image processing. Overall, explainable AI plays a significant role in enhancing the trustworthiness and reliability of deep learning models and CAD systems in the field of medicine. Integrating visualization tools, such as Grad-CAM, allows us to understand the decision-making process of AI models, and especially aids in making the AI models more explainable and easily interpretable. This further helps to build confidence of medical and healthcare professionals in CAD systems and AI predictions, which not only reduces the relative time to reach a diagnosis but also enhances the overall treatment outcomes. Ultimately, our research aims to use the strength of our proposed AResNN model along with the visualization power of Grad-CAM++ to further enhance the reliability of skin cancer diagnosis using artificial intelligence and CAD systems. To summarize the results obtained from our curated proposed model, it outperforms all the other traditional models for our research purpose i.e. the classification of skin cancer.

Bibliography

[1] S. Jain, V. Jagtap, and N. Pise, "Computer aided melanoma skin cancer detection using image processing," en, *Procedia computer science*, vol. 48, pp. 735–740, 2015, ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.04.209. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050915007188.

[2] M. M. I. Rahi, F. T. Khan, M. T. Mahtab, A. K. M. Amanat Ullah, M. G. R. Alam, and M. A. Alam, "Detection of skin cancer using deep neural networks," in *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, vol. 20, IEEE, 2019, pp. 1–7, ISBN: 9781728163031.

[3] K. W. Lee and R. K. Y. Chin, "The effectiveness of data augmentation for melanoma skin cancer prediction using convolutional neural networks," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, IEEE, 2020.

[4] N. Abhvankar, H. Pingulkar, K. Chindarkar, and A. P. I. Siddavatam, "Detection of melanoma and non-melanoma type of skin cancer using cnn and resnet," in *2021 Asian Conference on Innovation in Technology (ASIANCON)*, IEEE, 2021, pp. 1–6, ISBN: 9781728184029.

[5] S.-H. Tsang, *Review — Grad-CAM++: Improved visual explanations for deep convolutional networks (weakly supervised object localization)*, en, https://medium.com/swlh/review-grad-cam-improved-visual-explanations-for-deep-convolutional-networks-weakly-760264e66bc6, Accessed: 2025-2-7, Jan. 2021.

[6] Z. Civelek and M. H. S. Kfashi, "An improved deep cnn for an early and accurate skin cancer detection and diagnosis system," *International Journal of Engineering Research and Development*, 2022. DOI: 10.29137/umagd.116295. [Online]. Available: https://dergipark.org.tr/en/download/article-file/2426304.

[7] G. Shinde, *Skin cancer detection using image processing*, 2022. [Online]. Available: https://www.semanticscholar.org/paper/Skin-Cancer-Detection-Using-Image-Processing-Shinde/ec182cf3a5a289cf08e8142fa6a7a3713b6cfc46.

[8] M. Zia Ur Rehman, F. Ahmed, S. A. Alsuhibany, S. S. Jamal, M. Zulfiqar Ali, and J. Ahmad, "Classification of skin cancer lesions using explainable deep learning," *Sensors*, vol. 22, p. 6915, Sep. 2022. DOI: 10.3390/s22186915.

[9] C.-Y. Chang, *Building a customized residual CNN with PyTorch*, en, https://medium.com/@chen-yu/building-a-customized-residual-cnn-with-pytorch-471810e894ed, Accessed: 2025-2-7, Nov. 2023.

[10] A. M. Ibrahim, M. Elbasheir, S. Badawi, A. Mohammed, and A. F. M. Alalmin, "Skin cancer classification using transfer learning by vgg16 architecture (case study on kaggle dataset)," *Journal of Intelligent Learning Systems and Applications*, vol. 15, pp. 67–75, Aug. 2023. DOI: 10.4236/jilsa.2023.153005. [Online]. Available: https://www.scirp.org/journal/paperinformation?paperid=126855 (visited on 01/12/2024).

[11] M. Inuwa, *Swin transformers*, en, https://www.analyticsvidhya.com/blog/2023/08/swin-transformers-modern-computer-vision-tasks/, Accessed: 2025-2-7, Aug. 2023.

[12] K. Sambyal, S. Gupta, and V. Gupta, "Skin cancer detection using resnet," *SSRN Electronic Journal*, 2023. DOI: 10.2139/ssrn.4365250. (visited on 03/31/2023).

[13] T. Sardana, *Visualizing how CNN thinks: CAM, Grad-CAM, Grad-CAM++*, en, https://medium.com/@tanishqsardana/visualizing-how-cnn-thinks-cam-grad-cam-grad-cam-1dabcf886cc5, Accessed: 2025-2-7, Jun. 2023.

[14] N. Vishwakarma, *A guide to Grad-CAM in deep learning*, en, https://www.analyticsvidhya.com/blog/2023/12/grad-cam-in-deep-learning/, Accessed: 2025-2-7, Dec. 2023.

[15] G. Dagnaw, E. Mouhtadi, and M. Mustapha, "Skin cancer classification using vision transformers and explainable artificial intelligence," *J Med Artif Intell*, 2024. DOI: https://doi.org/10.21037/jmai-24-6. [Online]. Available: https://cdn.amegroups.cn/journals/jlpm/files/journals/33/articles/8962/public/8962-PB1-3284-R1.pdf.

[16] K. Djaroudib, P. Lorenz, R. B. Bouzida, and H. Merzougui, "Skin cancer diagnosis using vgg16 and transfer learning: Analyzing the effects of data quality over quantity on model efficiency," *Applied Sciences*, vol. 14, pp. 7447–7447, Aug. 2024. DOI: 10.3390/app14177447. (visited on 10/17/2024).

[17] L. Gamage, U. Isuranga, D. Meedeniya, S. De Silva, and P. Yogarajah, "Melanoma skin cancer identification with explainability utilizing mask guided technique," en, *Electronics*, vol. 13, no. 4, p. 680, 2024, ISSN: 2079-9292. DOI: 10.3390/electronics13040680. [Online]. Available: https://www.mdpi.com/2079-9292/13/4/680.

[18] E. H. Houssein, D. A. Abdelkareem, G. Hu, M. A. Hameed, I. A. Ibrahim, and M. Younan, "An effective multiclass skin cancer classification approach based on deep convolutional neural network," *Cluster computing*, Jun. 2024. DOI: 10.1007/s10586-024-04540-1. (visited on 07/26/2024).

[19] C. Kavitha, S. Priyanka, M. P. Kumar, and V. Kusuma, "Skin cancer detection and classification using deep learning techniques," en, *Procedia computer science*, vol. 235, pp. 2793–2802, 2024, ISSN: 1877-0509. DOI: 10.1016/j.procs.2024.04.264. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050924009438.

[20] N. Khasanah and M. N. Winnarto, "Application of deep learning with resnet50 for early detection of melanoma skin cancer," en, *Journal Medical Informatics Technology*, pp. 16–20, 2024, ISSN: 2988-7003. DOI: 10.37034/medinftech.v2i1.31. [Online]. Available: https://medinftech.org/index.php/medinftech/article/view/31.

[21] M. M. Musthafa, Mahesh, V. Kumar, and S. Guluwadi, "Enhanced skin cancer diagnosis using optimized cnn architecture and checkpoints for automated dermatological lesion classification," en, *BMC medical imaging*, vol. 24, no. 1, 2024, ISSN: 1471-2342. DOI: 10.1186/s12880-024-01356-8. [Online]. Available: http://dx.doi.org/10.1186/s12880-024-01356-8.

[22] https://www.kaggle.com/datasets/surajghuwalewala/ham1000-segmentation-and-classification?select=GroundTruth.csv&fbclid=IwY2xjawIS1ydleHRuA2FlbQIxMAABHf
3Zy6wj1bc1NVc55gDQLN8CGiv151g_aem_uteQM5Lr43XpWNJii9_lpg, Accessed: 2025-2-7.

[23] https://www.kaggle.com/datasets/anwarhawash/skin-lesions?fbclid=
IwY2xjawIS1x9leHRuA2FlbQIxMAABHUAoJIbvD1GGjED9WlD475YNIHYYXGgbRDZKPr
lI11A_aem_yvmRQEpP_9U-T7uqw1CH_A, Accessed: 2025-2-7.