Skin Cancer Classification For Seven Types Of Skin Lesions

by

Md. Tawsifur Rahman
20141027
Md. Siam Sadman Azad
20141002
Ali Muhtasim
17301163

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
May 2023

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:

Md. Tawsifur Rahman
ID: 20141027

Md. Siam Sadman Azad
ID: 20141002

Ali Muhtasim
ID: 17301163

Approval

The thesis/project titled "Skin Cancer Classification For Seven Types Of Skin Lesions" submitted by

1. Md. Tawsifur Rahman (20141027)
2. Md. Siam Sadman Azad (20141002)
3. Ali Muhtasim (17301163)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on Jan 22, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty, PhD
Professor
Dept. of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Professor
Dept. of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Dept. of Computer Science and Engineering
Brac University

Abstract

Machine learning (ML) for skin lesion identification employs algorithms, notably convolutional neural networks (CNNs), to categorize and detect skin lesions, aiming to enhance early detection and treatment of skin cancer. CNNs, trained on diverse lesion images, excel in learning features for classification, often rivaling dermatologists' accuracy. Recent studies demonstrate CNNs' effectiveness, achieving accuracy comparable to or surpassing dermatologists. Ongoing research focuses on addressing challenges like dataset diversity and robust evaluation metrics. Despite obstacles, ML's potential to enhance early melanoma detection remains significant, promising to save lives through improved diagnosis and treatment. Notably, our research explored a hybrid approach, combining ResNet50v2 and InceptionV3 models trained on GAN-generated data. This innovative strategy achieved a notable 77% accuracy, showcasing promising results in advancing muticlass skin lesion identification accuracy.

Acknowledgement

All the praise and thanks to Allah (SWT) for His Mercy and Ease in completing this task.
We greatly appreciate the help of our Superviser Dr. Amitabha Chakrabarty.
We also thank Fahim Sir for his help in our later works.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Skin cancer is the most common type of cancer, with over 5 million instances discovered each year in the USA alone. Melanoma is the most lethal type of skin cancer, accounting for the majority of fatalities. For melanoma treatment to be successful, early detection is essential, as it is highly curable when caught in the early stages. However, the early detection of melanoma can be challenging, as it often appears as a small, dark lesion on the skin that may be easily overlooked.

ML algorithms can help with the early detection of melanoma by analyzing images of skin lesions and classifying them as benign or malignant. To understand the characteristics of many kinds of skin lesions, including color, shape, and texture, these algorithms are able to be trained on enormous datasets of tagged photos of skin lesions. Once trained, the algorithms can be used to classify new images of skin lesions and predict whether they are within the 7 classes of lesions. The seven classes are: Melanocytic nevi, also known as "moles," are benign (non-cancerous) growths on the skin that are usually brown or black in color. They can form anywhere on the outermost layer of skin and are brought on by an overabundance of melanocytes, the cells that produce color.

Melanocytes, or the cells that produce pigment, are the source of melanoma, a type of skin cancer. A new mole may develop or an existing mole may change as a result. Melanoma is more serious than other forms of skin cancer because, if it is not detected early, It has the potential for transmission to other bodily parts.

Benign keratosis-like lesions are noncancerous growths on the skin that may have a waxy or scaly appearance. They are usually caused by sun exposure and can appear on areas of skin like the cheeks, ears, or backs of the hands.

Skin cancer of the most prevalent kind is basal cell carcinoma (BCC). It typically manifests as a little, fleshy nodule or lump that is frequently pearly or translucent. It can also take the form of a pink or red, flat, scaly region. BCCs rarely invade other bodily regions and have a moderate rate of growth. Vascular lesions are growths or marks on the skin that involve blood vessels. They can be benign or cancerous and can include angiomas, hemangiomas (benign tumors of blood vessels), and Kaposi sarcoma (a cancer of the blood vessels). A dermatofibroma is a tiny, hard, elevated bump on the skin that is benign (non-cancerous). They can appear anywhere on the body and are often brown or red in color, although the legs are where they are most frequently encountered. An overgrowth of fibrous tissue causes them and are not cancerous.Now the goal would be to recognize which one falls under which class.To achieve that a sizeable dataset is required. HAM10000 dataset has a sizeable amount of images for us to work with and It may contribute to raising the precision and dependability of the Machine Learning algorithms used in training and assessment.

1.1 Research Problem

One of the main research problems in skin lesion detection through machine learning (ML) is the limited availability of high-quality, labeled datasets of skin lesion images. In order to train and evaluate ML algorithms for skin lesion detection, large datasets of images of skin lesions, along with corresponding labels indicating whether the lesion is benign or malignant, are needed. However, obtaining such datasets can be challenging, as the process of collecting, annotating, and verifying images of skin lesions is time-consuming and costly.

Another problem is the variability of the lesion in the images. The lesion may appear in different lighting conditions, different angles and different skin types which can make it difficult to generalize the model.

Additionally, there is a lack of robust and reliable evaluation metrics for skin lesion detection through ML. As there can be a high variability in lesion appearance, evaluating the performance of ML algorithms can be challenging. Developing accurate evaluation metrics that take into account the different factors that can affect the appearance of skin lesions, such as lighting and skin color, is crucial for assessing the performance of ML algorithms for skin lesion detection.

The inter and intra-observer variability of dermatologist's diagnosis of lesions is another important factor that makes the problem even harder to solve.

Another problem is the risk of overdiagnosis and the consequent unnecessary biopsy when using the model, as well as the risk of missing certain types of skin lesions that the model have not been trained on.

Finally, the lack of awareness among the general population and lack of accessibility to medical services in some countries can make it difficult to implement the solution and reach the target population.

1.2 Research Objectives

The main research objective of skin lesion detection through machine learning (ML) is to develop accurate and reliable systems that can aid in the early detection and diagnosis.

Develop artificial intelligence algorithms that can accurately catagorize skin lesion photos into the seven groups. Now the goal would be to recognize which one falls under which class. That requires a huge dataset to be accomplished. We can work with a sizable number of photos from the HAM10000 dataset, which can assist ML algorithms for training and evaluation to become more accurate and reliable.

Another objective is to address the problem of inter and intra-observer variability of dermatologist's diagnosis by creating models that can be used to supplement the human expert, also to address the problem of overdiagnosis and the risk of missing certain types of skin lesions by developing models that can be fine-tuned for specific populations and cases which can be achieved by developing models that can be easily and cheaply implemented in the field, and increase awareness among the general population and medical staff about the benefits and limitations of the ML models.

Our model can also Assist in the reduction of unnecessary biopsies and surgeries by providing an accurate and efficient diagnosis of skin lesions. This research is done to aid Doctors in making diagnosis and to farther improve the treatment protocol for patients because the patient feels their ailment is not being taken seriously when doctors dismiss their skin lesions as benign whereas a quick machine test could easily assure the patient that the lesion is indeed benign and not anything to be afraid of. This takes the fear out of the patient and also prevents farther consultations. General Practitioners tend to make the wrong call when it comes to skin lesions because GP are not well versed in matters of skin lesion compared to other field and its better to have consultation of a dermatologist. The machine could proxy a dermatologist in making judgement.

The model is for aiding dermatologists further confirm his or her prior diagnosis. The model would be there side by side and an image of the patient's skin legion would be fed to the model and the model would be used to predict what class of skin lesion it is.

The objective is solely to make sure our dermatologists have an easier time making diagnosis and if the diagnosis does not match then they can consult to make sure their diagnosis is right or wrong. This protocol could save lives and prevent malpractise.

In summary, the research objective of skin lesion detection through ML is to create systems that can correctly categorize pictures of skin lesions, improve the accuracy, make it accessible, reliable and easy to implement.

Chapter 2

Literature Review

Recently, in the field of medicine, we have seen the successful usage of machine learning. One such instance is the usage of CNN when it comes to feature extraction from images of various kinds of lesions and the determination of their classes. Many studies have demonstrated the effectiveness of using ML algorithms. The work of Han et al. (2018)[2], on the automated skin lesion classification tool, "ModelDerm" is quite notable.

A paper on deep learning for skin lesion classification" by S. Benyahia, B. Meftah, and O. Lézoray (2022)[11] showcased a different approach where features of skin lesions were extracted from pre-trained CNN models and then introduced those characteristics into several classifiers to be used again for extraction. To assess the classification, seven pre-trained CNN architectures, and twenty-four machine learning classifiers were employed. ISIC 2019 and PH2 are two distinct datasets that are used.

A paper by Dhivyaa, C. R., et al.(2020)[8] decided to use decision trees and random forest algorithms to improve the skin image classification's performance and perform a side-by-side comparison with additional datasets. High-resolution feature maps created using the suggested technique can aid in maintaining the image's spatial information. The researchers compared it to two distinct data sets and discovered that the technique was more accurate than earlier methods.

Recently a paper by [14] deals with how to eliminate hair features and show details in dermoscopy images, this model devised a feasible pre-processing method that comprises dilatation and pooling layers. The feature extractor for the processed images was then a deep residual neural network. A deep learning methodology [5] for the identification and categorization of melanoma within dermoscopic images. To specifically extract complex information from the photos, a deep residual neural network was used, while fisher vectors were employed to encode the images for analysis and classification purposes. This approach was chosen due to its efficacy in capturing nuanced characteristics of melanoma in dermatological images.

A paper by Shetty, B., Fernandes, R., Rodrigues, A. P., Chengoden, R., Bhattacharya, S., and Lakshmanna, K. (2022)[13] used k-fold cross-validation to compare the efficacy of CNN with other machine learning methods. Convolutional neural networks, which are used in the suggested work, deliver better accuracy, according to the researchers, than other machine learning techniques. With the CNN model, accuracy of 95.18% was attained in the suggested system. The suggested research aids in the early detection of seven different types of skin diseases, which may then be verified and treated effectively by medical professionals.

A research proposes a completely automated CAD system built on the deep learning architecture by Muhammad Attique Khan, Tallha Akram, Yu-Dong Zhang, and Muhammad Sharif (2020)[10]. The decorrelation formulation technique is used in the ISBI2016,2017 datasets. Later, the DenseNet deep model is given the segmented images as input for feature extraction. The average pool and fully connected layers' features are extracted using the suggested entropy-controlled least square SVM and merged. The feature selection block down-samples the generated vectors. Three datasets—ISBI2016, ISBI2017, and HAM10000—are used for validation, with accuracy levels of 96.3%, 94.8%, and 88.5%, respectively.

In the realm of assessing human organ disorders, a diverse array of imaging modalities is routinely employed. Magnetic resonance imaging (MRI) [4], positron emission tomography (PET) [6], and X-rays [7] stand as prominent techniques. Additionally, computed tomography (CT) [12][9] has been extensively utilized in diagnosing organ-related ailments. In the context of dermatological assessments, dermatoscopy image analysis, clinical screening, and various other methodologies have been historically pivotal in visually diagnosing and evaluating skin lesions.

One of the most commonly used approaches in the literature is the usage of CNN on a dataset of skin lesions and then the trained CNN to classify new images of skin lesions. Several studies have used this approach and have reported high accuracy rates, with some achieving accuracy rates comparable to or even better than those of dermatologists.

Another widely used method is to use image processing and computer vision techniques to extract features from skin lesion images and then use these features to train a classifier such as SVM. Many studies have proposed such methods and have achieved promising results. These algorithms have also been shown to be effective for skin lesion detection.

One important factor that has been identified in the literature is the need for large and diverse datasets of skin lesion images for training and evaluating ML algorithms. Many studies have reported the use of small datasets, which may not be representative of the true population of skin lesions and therefore may not generalize well to new cases.

Another important factor identified in the literature is the need for robust and reliable evaluation metrics for assessing the performance of ML algorithms for skin lesion detection. Different evaluation metrics have been proposed in the literature, but there is still a lack of consensus on which metric is the most appropriate for this task.

In recent years, there have been a lot of advances and breakthroughs in the field of computer vision, deep learning, and generative models, which were also utilized in the field of skin lesion detection. Some studies have shown that these methods have improved the performance of the models, and can also be used to generate synthetic images of skin lesions, which can help to overcome the problem of limited availability of labeled datasets.

Overall, the literature suggests that ML algorithms, particularly CNNs, can be effective for skin lesion detection, with some studies achieving accuracy rates comparable to or even better than those of dermatologists. However, there are still challenges to be overcome, such as the need for larger and more diverse datasets for the development of robust and reliable evaluation metrics.

Chapter 3

Methodology and Dataset

Our research embarked upon the burgeoning field of skin lesion identification through machine learning (ML). Initially, we delineated the challenges, encompassing limited datasets, variability in lesion appearance, and the absence of robust evaluation metrics. Our primary aim was to craft systems capable of accurate and reliable categorization of skin lesions, thereby enhancing accessibility and implementation ease.

Delving into existing literature, we surveyed recent studies, evaluating the categorized lesions using CNN. This examination substantiated the pivotal role of CNNs and prompted the acquisition of a substantial dataset, notably the HAM10000, which underwent meticulous preprocessing to rectify variability in image features and ensure label accuracy.

Building upon this foundation, we developed a CNN-based model, attuned to the intricacies of skin lesion characteristics. Training the model involved an emphasis on differentiating among the seven classes of skin lesions. This stage represents the culmination of extensive work in data acquisition, preprocessing, and model development, exhibiting promising strides in inaccurate classification.

In furthering our efforts to bolster dataset robustness, we employed Generative Adversarial Networks (GANs) to augment images of skin lesions within classes that exhibited limited samples. This augmentation strategy aimed to enhance the diversity and representation of underrepresented classes within the dataset, ensuring a more balanced and comprehensive training regimen for our models.

Moreover, in our pursuit of refining model performance, we engaged in an extensive parameter fine-tuning phase. This involved meticulous adjustments and optimizations aimed at enhancing the learning capabilities and classification accuracy of our models.

Our exploration extended beyond singular model architectures, encompassing a diverse array of models such as ResNet50v2, Vgg19, and ResNet101v2. By employing multiple models, we sought to discern the nuances in performance and classification efficacy across varied architectures, thus gaining deeper insights into the most suitable models for skin lesion classification tasks. Our research trajectory pivoted toward addressing diagnostic challenges prevalent in the field. We endeavored to mitigate inter and intra-observer variability by exploring models complementing human expertise. Simultaneously, our focus extended to fine-tuning models for specific demographics, thus ensuring cost-effective implementation in diverse scenarios.

An integral facet encompassed the evolution of evaluation metrics, intricately de-

signed to encompass varying factors affecting skin lesion appearance. This phase continues to witness ongoing refinement, striving to develop metrics that comprehensively assess model performance vis-à-vis dermatologists' diagnoses.
Moreover, our efforts transcended theoretical frameworks into real-world impact assessment. Strategies were formulated to bolster awareness and accessibility among medical practitioners and the populace, with an aim to reduce unnecessary biopsies and augment early diagnosis and treatment protocols.
In tandem, we initiated clinical integration trials, positioning the ML model as an adjunctive diagnostic tool for dermatologists. Early feedback suggests promising results in reducing diagnostic errors, enhancing efficiency, and potentially averting instances of malpractice.
Our ongoing investigation extends to analyzing user experiences and patient perceptions, deciphering the societal impact of ML-based diagnosis on patient care and mental well-being. As our journey progresses, the aim remains resolute – to streamline dermatologists' diagnoses, empower patients, and significantly influence the landscape of skin lesion identification and treatment protocols.

3.1 Deep Learning Workflow

The images of the dataset are firstly resized into 224 by 224 pixels before normalizing the values between 1 and 0. Normalizing pixel values to the range [0, 1] ensures consistency in scale across different images. This is important because it allows the model to learn patterns and features in a uniform manner, regardless of the original intensity range of the images. Normalizing pixel values to a smaller range reduces memory requirements, which is crucial for efficient computation. This is particularly important in scenarios with limited computational resources, such as when training models on GPUs or edge devices. Many pre-trained models and architectures, especially those trained on large datasets like ImageNet, expect input images to have pixel values in the [0, 1] range. Normalizing input in this way allows for seamless integration with such pre-trained models. After normalizing the data is split into train, test, and validation. Train being 80% and test being 20%. This process of splitting is done after careful testing and observation.
During the training phase, each mini-batch of images undergoes random transformations based on the specified augmentation settings. This process ensures that the model encounters a diverse set of input variations in each epoch, promoting better convergence and learning. Such random transformations are Width Shift and Height Shift, Rotation, Shear, Zoom, horizontal and vertical flip, and fill mode.

Rotation Range (degrees): Randomly rotates the image within the specified range, introducing variations in orientation, e.g., rotations up to 10 degrees.

Width Shift Range and Height Shift Range: Randomly shifts the image horizontally and vertically within the specified ranges, simulating variations in object placement.

Shear Range: Introduces shear transformations to the image, deforming it along one axis.

Zoom Range: Randomly zooms into or out of the image, providing scale variations.

Horizontal and Vertical Flips: Randomly flips the image horizontally and vertically, introducing reflections and augmenting the dataset with mirrored versions.

Fill Mode: Specifies the strategy for filling in newly created pixels during transformations, ensuring a seamless transition in the augmented images.

In pursuit of better accuracy and precision different neural networks are employed for training. The neural networks are ResNet50V2, VGG16, ResNet101V2, InceptionV3, and hybrids of such models. These models are pre-trained on the imagenet dataset. The models are then fine-tuned to achieve better results on this particular dataset. The figure 3.1 is an overview of the whole process.

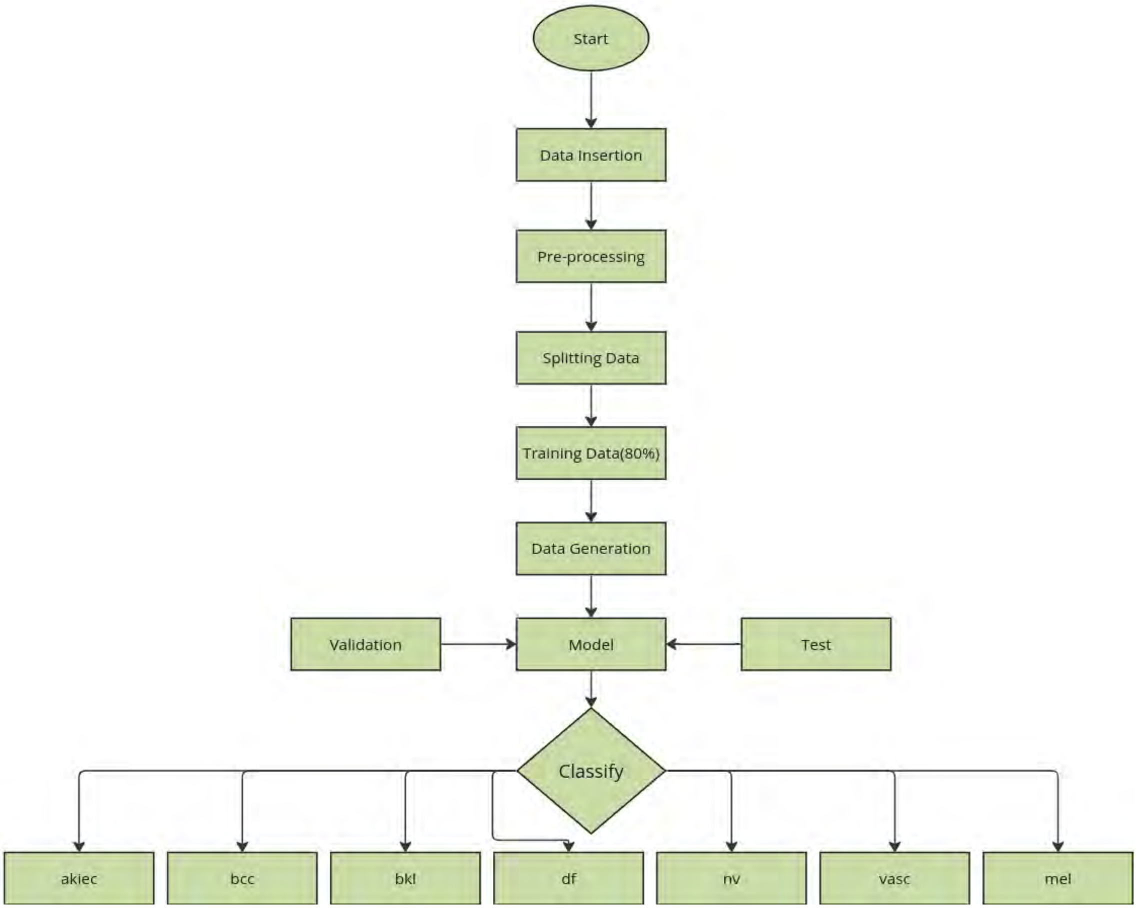


Figure 3.1: Deep learning workflow

3.2 GAN Workflow

The classes with fewer images are selected for GAN(generative adversarial Network). We put all the images belonging to a particular class into one folder and fit them into the model for training. The training is done by the two classes Generator and Discriminator. There is a high-level overview of the generator model in figure 3.2. Figure 3.3 is the model architecture for Discriminator. This process is done separately for each class that requires more samples. After training the generator model is saved and later used to generate the images for each class individually.

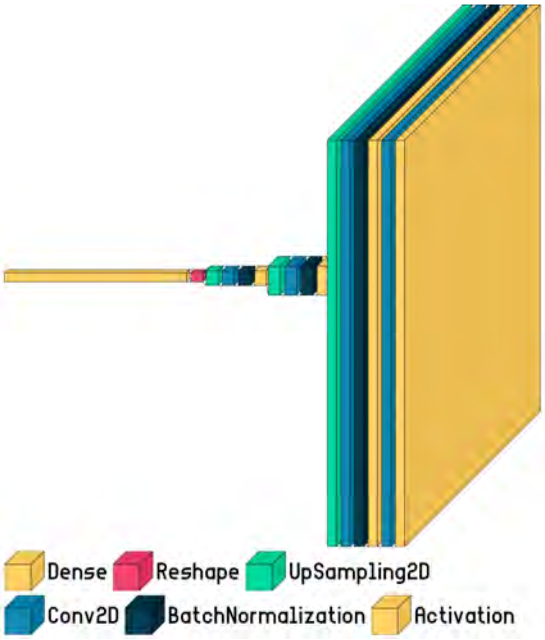


Figure 3.2: Generator Architecture

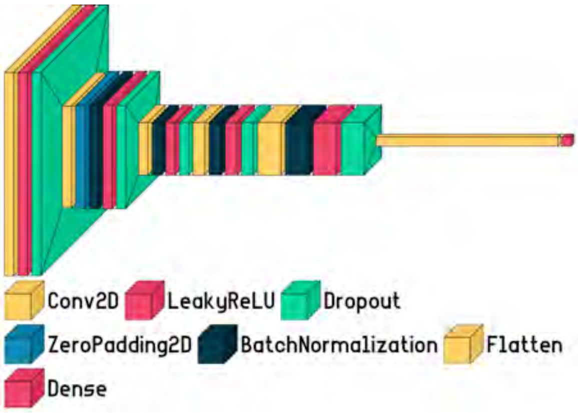


Figure 3.3: Disriminator Architecture

3.3 Application Workflow

Users begin by uploading dermatoscopic images of skin lesions using the file uploader widget. The application supports common image formats such as JPG, JPEG, and

PNG. The uploaded image is then displayed within the application, allowing users to visually confirm and review the selected skin lesion. A pre-trained skin lesion classification model is loaded into the application. This model has been trained on a dataset to recognize different types of skin lesions based on their visual characteristics. The uploaded image undergoes preprocessing steps, including resizing to a standard size (e.g., 224x224 pixels), to ensure compatibility with the model's input requirements. The preprocessed image is passed through the loaded model for inference. The model predicts the likelihood of different skin lesion types, providing a probability distribution over the classes. The application displays the predicted class and the corresponding probability, offering users valuable information about the potential type of skin lesion present in the uploaded image. The entire workflow is encapsulated in an intuitive and user-friendly interface created using Streamlit, making it accessible to healthcare professionals, dermatologists, or anyone interested in obtaining quick and preliminary assessments of skin lesions. The automated nature of the tool facilitates early detection and aids in decision-making related to patient care.

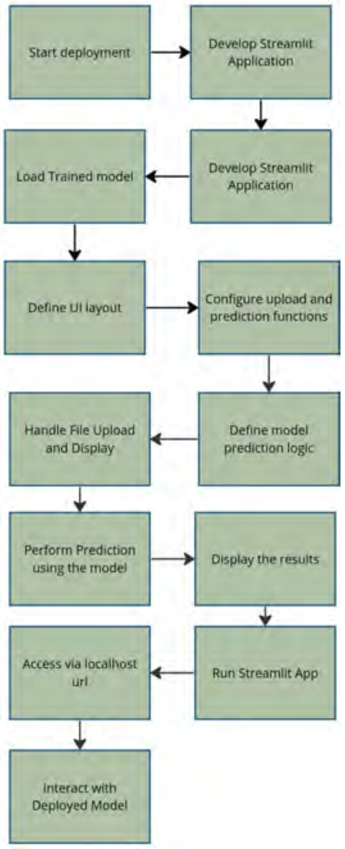

Start deployment
Develop Streamlit Application
Develop Streamlit Application
Load Trained model
Define UI layout
Configure upload and prediction functions
Define model prediction logic
Handle File Upload and Display
Perform Prediction using the model
Display the results
Run Streamlit App
Access via localhost url
Interact with Deployed Model

Figure 3.4: App workflow

3.4 Data description

There is a deficiency of high-quality images and diversity in the dataset of dermatoscopic images that is available. Fortunately, dermatoscopic photos from many populations obtained through various modalities were used to build HAM10000, which refers to Humans against Machines. Histopathology verified more than half of the lesions. There are 10015 pictures in all in the dataset. The courses have been recognized and assigned labels[3]. The following are the classes: 1) Nevi melanocytocytic 2) Melanoma (mel) 3)Lesions that resemble benign keratosis (bkl) 4) BCC, or basal cell carcinoma 5) Akiec, or actinic keratoses 6) Lesiones vascular (vas) 7) Diverticulitis (df).

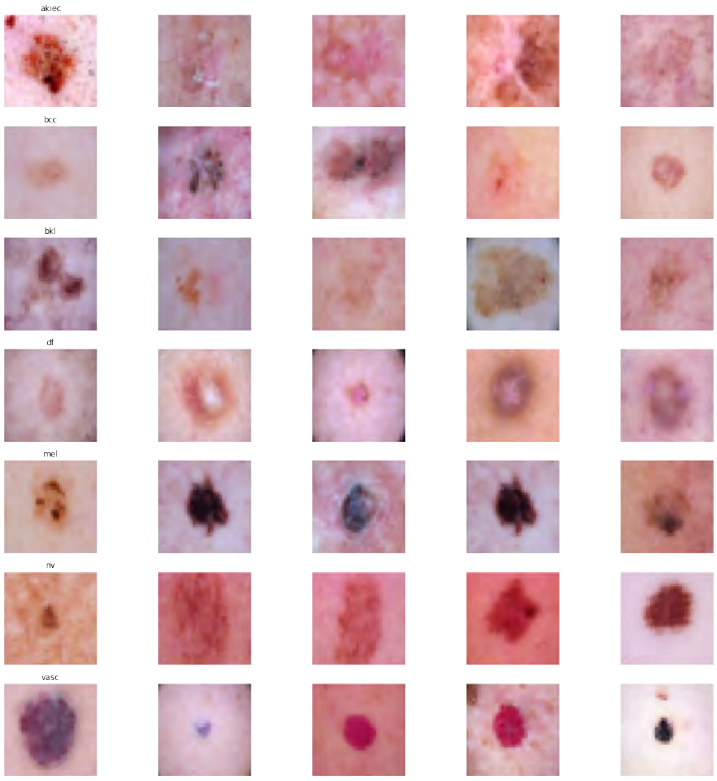


Figure 3.5: classes of skin lesions

The dataset goes further by identifying each individual by their sex, age, and also the position of the lesion on their skin. In fact, the data showcases all the traits necessary to draw correlations among the variables. The graph 3.6 shows how many males and how many females are there in the dataset.
There are a total of 6705 samples of Melanocytic nevi, 1113 smaples of mel or Melanoma, 1099 Benign keratosis-like lesions, 514 bcc or Basal cell carcinoma, 327 samples of Actinic keratoses,142 samples of vascular lesions and 115 samples of Dermatofibroma. The highest number of instances of class within the dataset is 6705 and the lowest being only 142. This imbalance in the dataset is addressed through the use of GAN, by generating images for classes with lesser instances. The imbalance in the dataset can lead to biased model performance. The model may become overly influenced by the majority class, resulting in poor prediction performance for minority classes. Models trained on imbalanced datasets may struggle to generalize well to new, unseen data. The model may learn patterns that are specific to the ma-

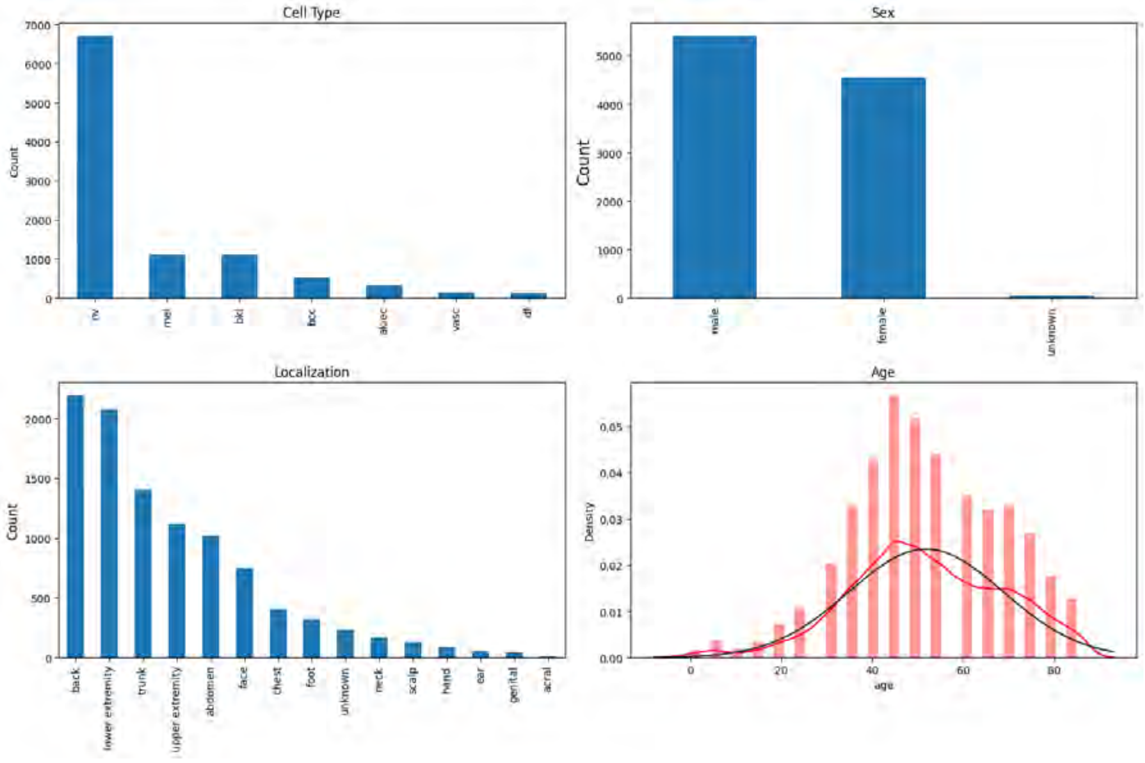


Figure 3.6: Statistics on Dataset

jority class and fail to recognize important features of minority classes. Accuracy, a common evaluation metric, can be misleading in the presence of imbalanced classes. A model that predicts the majority class for all instances may still achieve high accuracy, even though it provides little to no value in real-world applications. The dataset also highlights where exactly the lesion is located in figure 3.6. There are 15 total mentioned places of interest, out of which the back has the highest count, followed by the lower extremity, trunk, upper extremity, abdomen, face, chest, foot, unknown, neck, scalp, hand, ear, genital, and acral.

3.5 Model Description

A set of models are employed for training over the custom dataset. Each model has its own set of strengths and weaknesses. The structures and parameters of such models are highlighted below.

3.6 ResNet50v2

A version of the ResNet architecture called ResNet50v2 was released to help with the difficulties associated with developing very deep networks of neural networks. ResNet50v2's ”50” stands for the number of layers, or more precisely, the depth of the network. The ”v2” denotes a revised version that enhances the initial ResNet.

3.6.1 Model Structure

Depth and Layer Configuration: ResNet50v2 is a variant of the ResNet architecture, specifically ResNet50, which includes 50 layers. It consists of a series of residual blocks that contain convolutional layers, batch normalization, ReLU activations, and shortcut connections (or skip connections). The network architecture is designed to mitigate the vanishing gradient problem and facilitate the training of deeper networks.

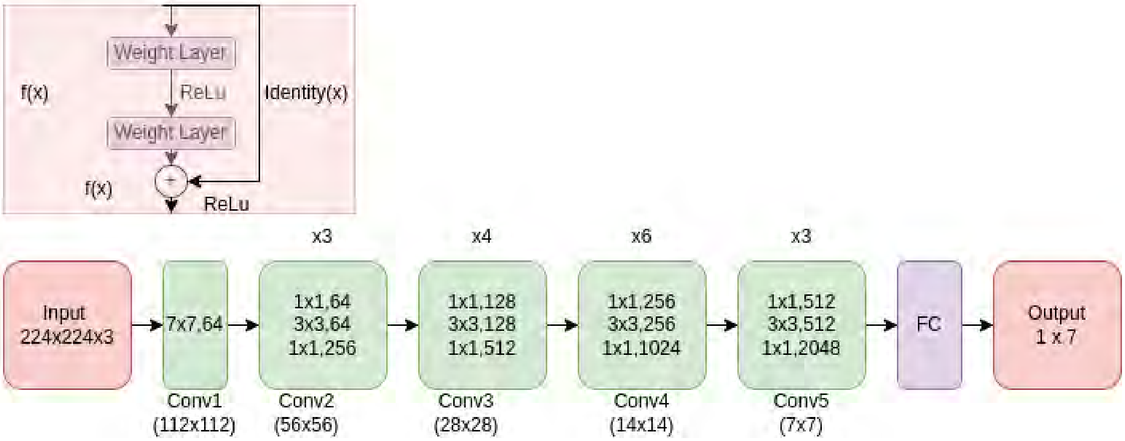


Figure 3.7: ResNet50v2 model Architecture

Bottleneck Design: ResNet50v2 employs a bottleneck architecture that utilizes 1x1, 3x3, and occasionally 1x1 convolutional layers. This design optimizes computational efficiency by reducing dimensionality before applying more computationally expensive operations, allowing for deeper networks with fewer parameters.

3.6.2 Model Parameters

ResNet50v2 has approximately 24.6 million parameters. The bottleneck design, which includes fewer convolutional filters in the intermediate layers, contributes to the reduction in parameters compared to deeper ResNet variants like ResNet101 or ResNet152.

Category	**Parameters**
ResNet50v2	23,564,800
overall model	24,617,479
Trainable parameters	1,052,679
non-trainable	23,564,800

Table 3.1: Resnet50v2 Parameters

3.6.3 Memory usage

Compared to deeper networks like VGG16 or even the original ResNet, ResNet50v2 generally demands less memory due to its relatively shallower depth. While it still requires memory for storing intermediate feature maps during training and inference, the reduced number of layers and parameters compared to deeper architectures can lead to lower memory consumption.

Type	**Usage**
total memory usage	93.91 MB
Trainable parameters	4.02 MB
non-trainable	89.89 MB

Table 3.2: ResNet50v2 Memory Usage

3.6.4 Model Summary

In summary, this model is a Sequential model with a ResNet50v2 base, followed by custom layers, including a Dropout layer. It is designed for a classification task with 7 output classes. The majority of parameters come from the ResNet50v2 layer, indicating the utilization of a pre-trained model for feature extraction.

3.7 VGG16

The deep convolutional neural network architecture known as VGG16, or Visual Geometry Group 16, is intended for image classification. The total amount of weight layers in the network is indicated by the ”16” in VGG16.

3.7.1 Model Structure

Convolutional Layers: The network primarily consists of 3x3 convolutional filters applied with a stride of 1 and padding of 1 to maintain the spatial dimensions of the input. These convolutional layers are stacked on top of each other, enabling the network to learn hierarchical features of increasing complexity.

Pooling Layers: VGG16 incorporates max-pooling layers with a 2x2 window and stride of 2, reducing spatial dimensions and providing translation invariance to certain features.

Fully Connected Layers: Following the convolutional layers are three fully connected layers with the last layer producing class probabilities using softmax activation for classification tasks. There is a brief overview of the said model in figure 3.8

3.7.2 Model Parameters

VGG16 contains approximately 138 million parameters. The convolutional layers account for the majority of these parameters, especially because of the multiple layers and the large number of filters in each layer.

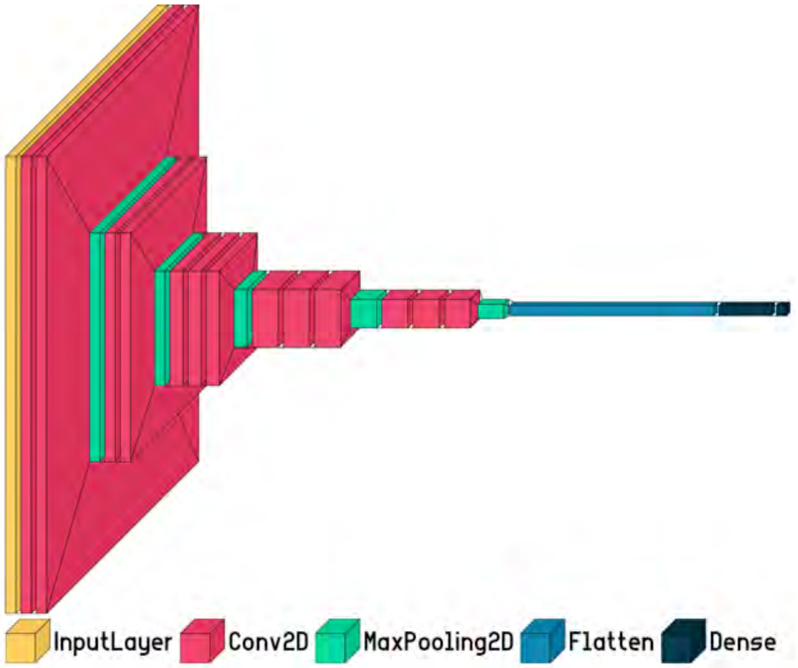


Figure 3.8: VGG16 model Architecture

Category	**Parameters**
overall model	14,980,935
Trainable parameters	266,247
non-trainable	14,714,688

Table 3.3: VGG16 Parameters

3.7.3 Memory usage

While VGG16 is known for its simplicity in architecture, it demands a considerable amount of memory due to its depth and a large number of parameters. The repetitive stacking of convolutional layers increases memory usage, particularly when storing intermediate feature maps during both training and inference.

Type	**Usage**
total memory usage	57.15 MB
Trainable parameters	1.02 MB
non-trainable	56.13 MB

Table 3.4: VGG16 Memory Usage

3.7.4 Model Summary

In summary, this model is an extended version of VGG16 with additional layers, potentially fine-tuned for a specific task with 7 output classes. The majority of the parameters come from the pre-trained VGG16 model, and the model has been configured with dropout for regularization.

3.8 ResNet101v2

ResNet101v2 (Residual Network 101 version 2) is a deep convolutional neural network (CNN) architecture that addresses the degradation problem in deep networks. ResNet101v2 is an extension of the ResNet (Residual Network) architecture, specifically ResNet101, which denotes the depth of the network with 101 layers. The "v2" indicates improvements and modifications made over the original ResNet to enhance performance and training efficiency.

3.8.1 Model Structure

The architecture follows a building block structure featuring residual blocks with skip connections. These blocks contain a series of convolutional layers, batch normalization, ReLU activations, and shortcut connections, ensuring that the network can learn more complex features while mitigating the vanishing gradient problem associated with deeper networks. Additionally, ResNet101v2 employs bottleneck blocks, where 1x1 convolutions are utilized to reduce dimensionality before applying 3x3 convolutions, optimizing computational efficiency while maintaining representational capacity as shown in figure 3.9.

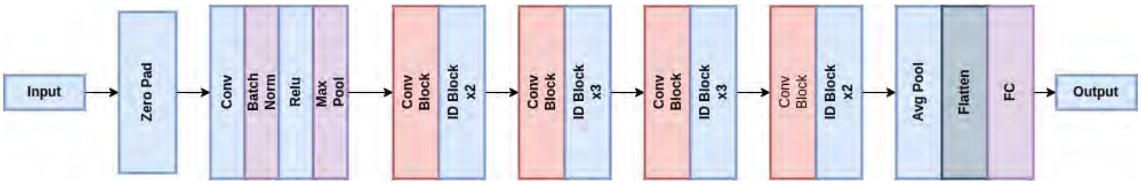


Figure 3.9: ResNet101V2 model Architecture

3.8.2 Model Parameters

Deeper networks have more parameters (weights and biases) to store during training and inference, resulting in increased memory consumption.

Category	Parameters
ResNet101v2	20,024,384
overall model	20,290,631
Trainable parameters	266,247
non-trainable	20.024,384

Table 3.5: ResNet101v2 Parameters

3.8.3 Memory usage

ResNet101v2, being a deeper and more complex convolutional neural network, generally requires more memory compared to shallower networks or earlier versions of the ResNet architecture, like ResNet50 or ResNet18.

Type	Usage
total memory usage	77.40 MB
Trainable parameters	1.02 MB
non-trainable	76.39 MB

Table 3.6: ResNet101v2 Memory Usage

3.8.4 Model Summary

Overall, ResNet101v2 is a powerful and versatile CNN architecture for image recognition tasks. Its improved accuracy, faster inference speed, and pre-trained weights make it a valuable tool for researchers and developers alike.

3.9 InceptionV3

Google's research team created the potent convolutional neural network (CNN) architecture known as InceptionV3. This deep learning model is well-known for how effective it is at classifying images.

3.9.1 Model Structure

Inception Modules: The usage of "Inception modules," that are blocks of several convolutional layers of various sizes (1x1, 3x3, 5x5) and pooling procedures concatenated together, is what distinguishes InceptionV3. As a result, the network can learn and record information at different scales as shown in figure 3.10.

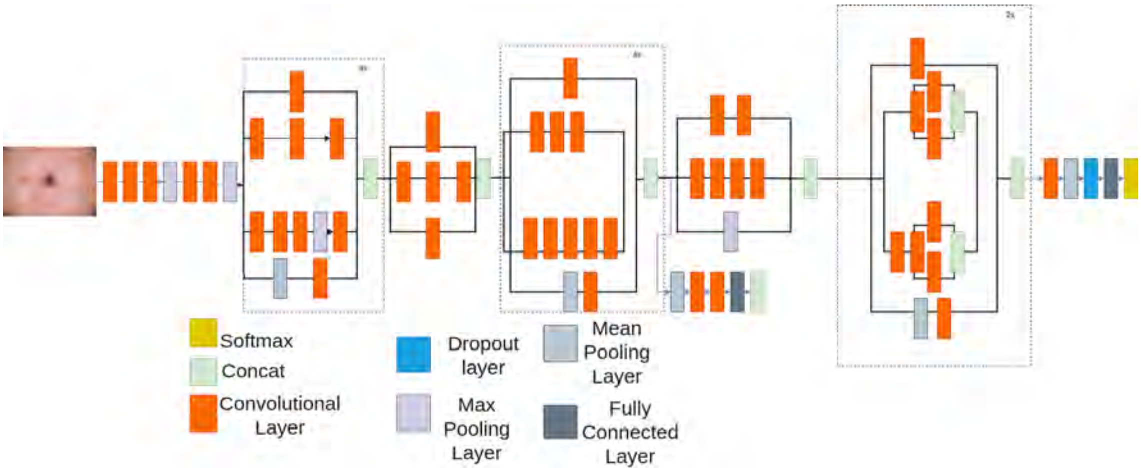


Figure 3.10: InceptionV3 model Architecture

Factorization: InceptionV3 uses factorization to reduce computational complexity. It replaces large convolutions (e.g., 5x5) with a combination of smaller convolutions (e.g., two consecutive 3x3 convolutions) to decrease the number of parameters and computational cost while retaining expressive power.

Auxiliary Classifiers: This architecture introduces auxiliary classifiers at intermediate layers during training, aiding in combating the vanishing gradient problem and providing additional regularization, thus improving the model's ability to learn useful representations.

Pre-Trained on Large Datasets: InceptionV3 like many other modern CNN architectures, was initially trained on large-scale image datasets like ImageNet, enabling it to learn a wide range of features useful for various visual recognition tasks.

3.10 ResNet50v2VGG16

We propose an innovative hybrid model that amalgamates the robust features of two widely acclaimed convolutional neural network architectures, ResNet50V2 and VGG16. The ResNet50V2, distinguished by its utilization of residual connections, excels in capturing intricate details while mitigating the vanishing gradient issue. Concurrently, VGG16, characterized by its deep and homogeneous architecture, has demonstrated excellence in feature representation. By fusing ResNet50V2 and VGG16 within a unified framework, our hybrid model aims to harness the distinctive advantages of each architecture. Through a concatenation of feature extraction layers, the model captures a diverse range of features, offering a more comprehensive representation of input data. Extensive experimentation and evaluation are conducted to gauge the hybrid model's performance across diverse tasks. Our findings shed light on the synergistic benefits of combining ResNet50V2 and VGG16, highlighting the potential for improved classification accuracy and model robustness. This research contributes to the exploration of hybrid neural network architectures, offering insights into effective strategies for leveraging the strengths of diverse convolutional neural network paradigms.

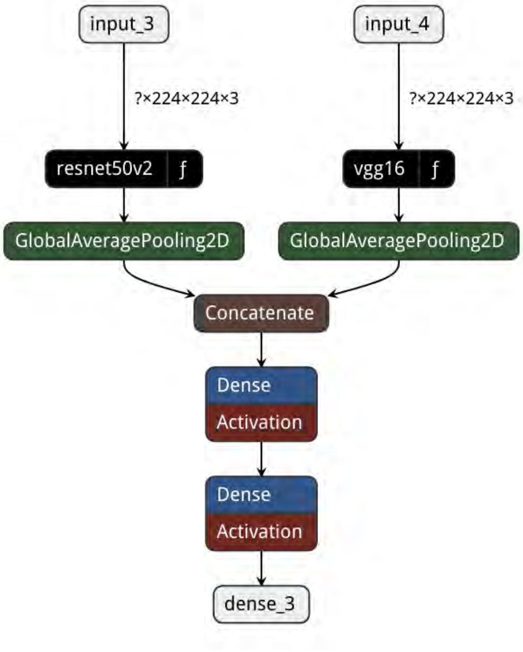


Figure 3.11: ResNet50v2VGG model Architecture

ResNet50V2 incorporates residual connections and consists of residual blocks with skip connections.Introduce VGG16-like blocks (stacks of convolutional layers followed by occasional pooling) within ResNet50V2's structure. Creating a hybrid model by combining ResNet50V2 with VGG16-like components involves merging the residual connections of ResNet50V2 with stacks of convolutional layers and occasional pooling layers reminiscent of VGG16's architecture as shown in figure 3.11. This integration aims to leverage the depth capabilities of ResNet with certain structural elements of VGG16 to potentially enhance the model's feature learning and representation capabilities.

3.11 ResNet50v2InceptionV3 hybrid

In this study, we propose a novel hybrid model that leverages the distinct strengths of two state-of-the-art convolutional neural network architectures, ResNet50V2 and InceptionV3. By integrating these models into a unified framework, we aim to exploit the complementary features and representations learned by each architecture. The ResNet50V2, known for its residual connections, excels in capturing intricate details and overcoming vanishing gradient problems, while the InceptionV3, with its inception modules, excels in efficiently capturing multi-scale features. The hybrid model amalgamates the unique characteristics of ResNet50V2 and InceptionV3 through a concatenated feature extraction process, allowing for a more comprehensive and diversified representation of the input data. Through extensive experimentation and evaluation, we assess the performance of this hybrid model across various tasks, demonstrating its potential to enhance classification accuracy and robustness compared to individual architectures. This research contributes to the exploration of model fusion strategies, shedding light on the benefits of combining distinct neural network architectures for improved deep learning outcomes.

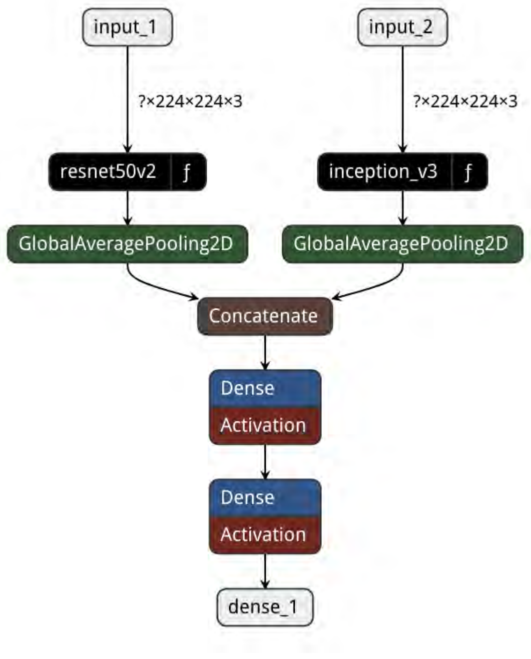


Figure 3.12: ResNet50v2Inceptionv3 model Architecture

Embed specific InceptionV3 modules within ResNet50V2's residual blocks, possibly replacing certain convolutional blocks with Inception-like structures.Utilizing the ResNet50V2 architecture as the foundational backbone while integrating selected InceptionV3 modules within the ResNet50V2 architecture. The figure 3.12 showcases a high level overview of the said model.

Chapter 4

GAN for Data Augmentation

Two competing neural networks, a generator, and a discriminator, make up a Generative Adversarial Network (GAN), a kind of machine learning model. Think of it as a game of cat and mouse, where the generator tries to create realistic data (like images, music, or even text) that can fool the discriminator, while the discriminator tries to distinguish real data from the generator's fakes. In Goodfellow's tutorial [1] on Generative Adversarial Networks (GANs) presented at the Neural Information Processing Systems (NIPS) conference in 2016 provides valuable insights into the significance of generative models. Goodfellow elaborates on the workings of GANs, highlighting their innovative approach compared to other generative models. The tutorial delves into frontier research areas within the realm of GANs, offering a comprehensive overview of their potential applications and advancements in the field of generative modeling. We used Generative Adversarial Networks (GANs) in the context of GAN augmentation in our study to address class imbalance in our dataset which comprised seven distinct skin lesion classes. Recognizing the inherent challenge posed by disparate sample sizes across these classes, our approach involved employing GANs to generate synthetic skin lesion images for underrepresented classes. By doing so, we aimed to equalize the number of samples per class, enhancing the robustness and balance of our dataset. This innovative use of GAN augmentation not only mitigated the class imbalance issue but also contributed to the overall effectiveness of our image classification model in accurately discerning diverse skin lesion categories.

4.1 Architectural Overview of Generative Adversarial Networks

Generative Adversarial Networks (GANs) represent captivating machine learning architectures comprising a tandem of neural networks: the 'generator' and the 'discriminator'. Engaged in an adversarial learning scheme, these networks undergo a strategic contest. This contest stimulates the generator to continually enhance its proficiency in generating increasingly realistic outputs. Consequently, GANs have emerged as formidable instruments applicable in diverse realms, including but not limited to image generation, text-to-image synthesis, and style transfer.

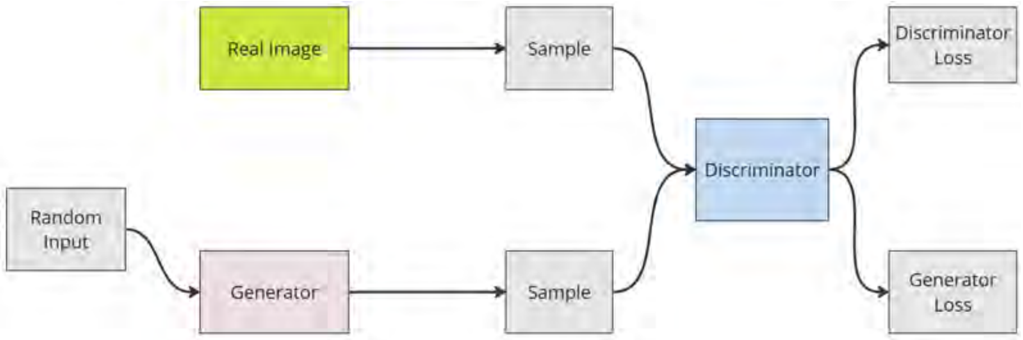


Figure 4.1: GAN Architecture

4.1.1 Fundamental Components

GAN is composed of mainly two opposing components Generatot and Discriminator. Both of these classes are made by stacking layers as shown in the figure 4.2.

Generator (G) Architecture

Input Layer (Seed Size) represents the initial random seed input for generating synthetic data. Dense Layer (ReLU) Applies a dense layer with Rectified Linear Unit (ReLU) activation to capture complex patterns. Reshape Layer (4x4x256) reshapes the output to a 4x4x256 tensor to serve as the foundation for further upsampling. Multiple upsampling layers progressively increase the spatial dimensions of the tensor. Each upsampling layer consists of Convolutional (Conv2D) operations, Batch Normalization, and ReLU activation. The last upsampling layer generates the final synthetic data with a specified number of channels using the tanh activation.

Discriminator (D) Architecture

Input Layer (Image Shape) accepts input data representing real or generated images.A series of convolutional layers with Leaky Rectified Linear Unit (Leaky ReLU) activation, Batch Normalization, and Dropout. These layers extract features at different spatial resolutions from the input images.Flatten Layer flattens the tensor to prepare for the final dense layer. A dense layer with a sigmoid activation function produces a binary output indicating whether the input is real or generated.

Training Connection:

The generator and discriminator are connected during training to facilitate the adversarial training process. Synthetic data generated by the generator is fed into the discriminator to distinguish between real and generated samples. The training connection is a crucial part of the GAN architecture, ensuring that both components learn and improve iteratively.

4.1.2 Adversarial Training Process

A Generator and Discriminator are used in the GAN training process to constantly improve through learning, feedback, and trial and error. If D correctly identifies a fake, G receives negative feedback and adjusts its parameters to make future forgeries more convincing. If D mislabel real data as fake or vice versa, it penalizes itself and updates its parameters to refine its discrimination ability.

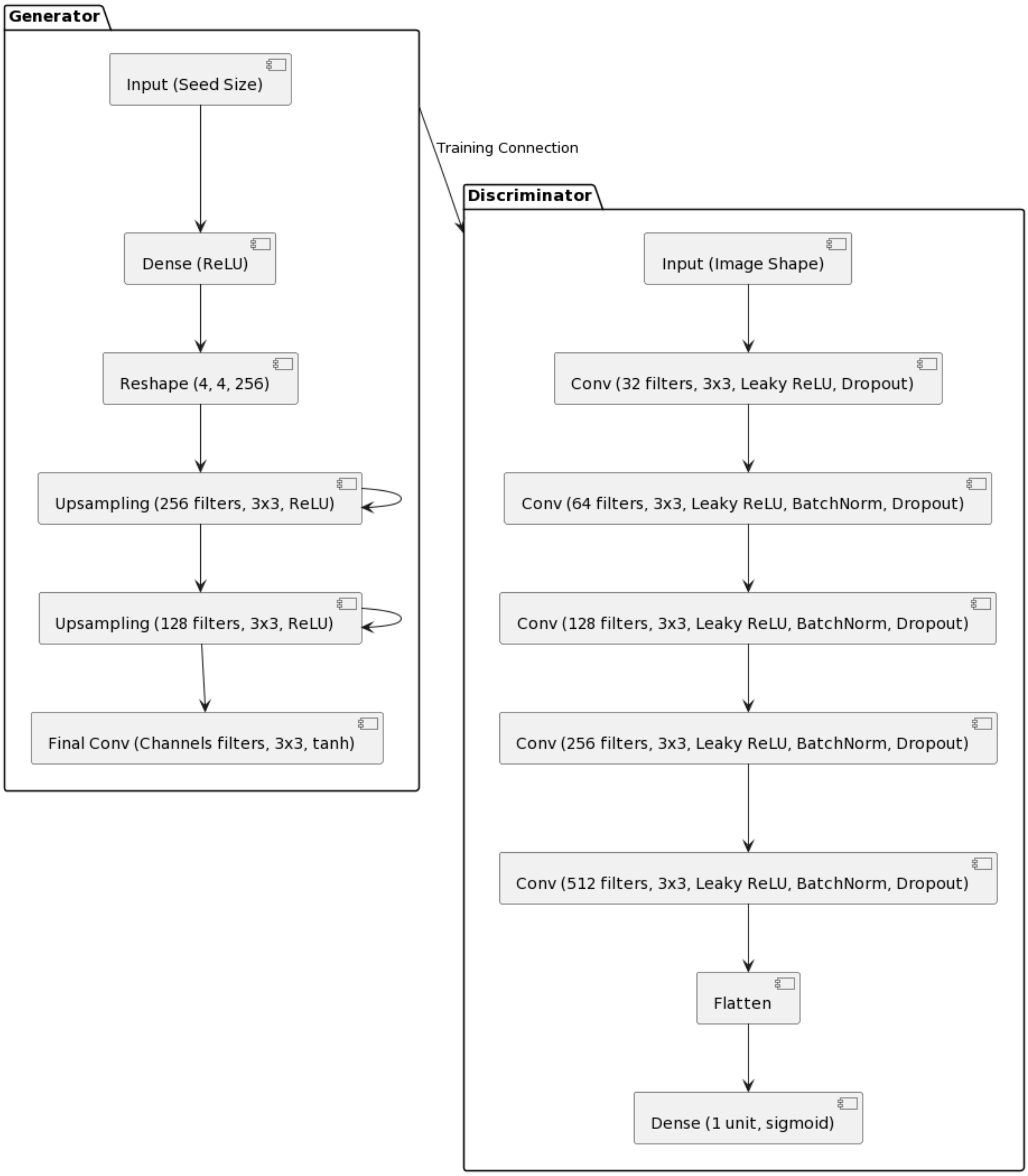

Generator
Input (Seed Size)
Dense (ReLU)
Reshape (4, 4, 256)
Upsampling (256 filters, 3x3, ReLU)
Upsampling (128 filters, 3x3, ReLU)
Final Conv (Channels filters, 3x3, tanh)
Training Connection
Discriminator
Input (Image Shape)
Conv (32 filters, 3x3, Leaky ReLU, Dropout)
Conv (64 filters, 3x3, Leaky ReLU, BatchNorm, Dropout)
Conv (128 filters, 3x3, Leaky ReLU, BatchNorm, Dropout)
Conv (256 filters, 3x3, Leaky ReLU, BatchNorm, Dropout)
Conv (512 filters, 3x3, Leaky ReLU, BatchNorm, Dropout)
Flatten
Dense (1 unit, sigmoid)

Figure 4.2: GAN Generator and Discriminator Structure

Generate Adversarial Examples: Adversarial examples are created by introducing small, imperceptible perturbations to the input data in order to mislead the model. These perturbations are calculated using optimization techniques like gradient ascent to maximize the model's prediction error.

Augment Training Data: Adversarial examples are added to the training dataset alongside the original, clean examples. This expanded dataset is used to retrain the model.

Update Model Parameters: The model is retrained on the augmented dataset, including both original and adversarial examples. The objective is to minimize the overall loss function, which now considers the model's performance on both clean and adversarial examples.

Repeat Iteratively: Steps 1-3 are repeated for multiple iterations to further enhance the model's robustness. In each iteration, new adversarial examples are generated, and the model is updated accordingly. The process of adversarial training helps the model learn to generalize better by recognizing and adapting to potential adversarial perturbations in the input space. It is commonly used in deep learning, especially in computer vision tasks, to improve the reliability and security of models. It's important to note that adversarial training is not a silver bullet, and the robustness of a model may still be limited to the specific perturbations used during training. Continuous research is being conducted to explore more effective ways of defending against adversarial attacks and enhancing the generalization capabilities of machine learning models.

4.1.3 Equilibrium and Generation

After extensive training, an equilibrium emerges where G's fakes are so convincing that even D struggles to tell them apart from real data. This signifies that G has captured the essence of real data and can effectively generate realistic samples. At this point, the focus shifts to using G for its intended purpose, such as generating novel images, music, or text.

4.1.4 Visualizing the Training Process

$$E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))] \tag{4.1}$$

The objective function for GANs is formulated to optimize the interplay between the discriminator (D) and the generator (G). Here's a detailed explanation of each element:

- D(x): This term represents the discriminator's assessment of the probability that a given real data instance x is genuine.
- E_x: This denotes the expected value operator overall real data instances. In other words, it represents the average probability estimation of the discriminator for real data.
- G(z): Here, G is the generator, and z is a random noise input.
- G(z) is the output of the generator when provided with this noise, creating a synthetic or fake data instance.
- D(G(z)): This term signifies the discriminator's estimation of the probability that a generated (fake) instance G(z) is real.

- E_z: This represents the expected value operator over all random inputs to the generator, essentially averaging the discriminator's assessment of fake instances generated by G.

The formula is rooted in the concept of cross-entropy between the probability distributions of real and generated instances. The generator aims to minimize the loss function, which is equivalent to minimizing log(1-D(G(z))) since the generator has no direct influence on the log(D(x)) term. In TensorFlow-GAN (TF-GAN), the functions minimax-discriminator-loss and minimax-generator-loss implement this specific loss function. These functions encapsulate the adversarial nature of GAN training, where the discriminator tries to maximize its ability to distinguish real from fake, while the generator aims to minimize the discriminator's confidence in distinguishing between the two.

4.2 Comparison with alternatives

GANs are often favored for their remarkable ability to achieve realistic and diverse outputs but there are others capable of more or less the same feat.

4.2.1 Variational Autoencoders (VAEs)

VAEs are a kind of generative model that can generate fresh, similar samples and reconstruct data by learning a latent representation of the input. Imagine a skilled architect meticulously compressing a building's blueprint (encoding) and then flawlessly reconstructing it (decoding). That's essentially how VAEs work. They excel at generating data similar to the training set but might struggle with novelty or capturing intricate details.

4.2.2 PixelRNNs and Autoregressive Models

Think of these models as meticulous storytellers, crafting data pixel by pixel, word by word. They excel at high-fidelity outputs but can be computationally expensive and slow to generate, especially for complex data.

4.2.3 Diffusion Models

Imagine adding noise to a real image until it becomes pure static, then gradually reversing the process to recover the original image. That's the core idea behind diffusion models. They offer impressive controllability and can generate high-quality outputs, but often require careful hyperparameter tuning and can be computationally intensive.

4.2.4 Why Choose GANs?

GANs often produce sharper, more photorealistic data compared to VAEs, as they directly compete with the discriminator to mimic reality. GANs can generate a wider range of outputs, sometimes venturing beyond the training data, while VAEs tend to stay closer to known examples. GANs typically train faster than PixelRNNs,

especially for high-resolution data. GANs bypass the sequential generation process, making them less prone to error accumulation and potentially able to capture long-range dependencies effectively. GANs have a more straightforward training process compared to diffusion models, which can be sensitive to hyperparameter settings. While GANs and Diffusion models can generate realistic data, GANs might offer more inherent potential for diverse and creative outputs due to the adversarial training dynamic.
There are several reasons why GANs (Generative Adversarial Networks) might be chosen over models like PixelRNNs and PixelCNNs, some are discussed briefly:

1. **Sample Variety and Quality:**
   - GANs often generate more diverse and visually appealing images than autoregressive models like PixelRNNs and PixelCNNs. This is because GANs are not strictly tied to the sequential order of the data, allowing them to explore a wider range of possibilities.
   - GANs excel at capturing high-frequency details and complex textures compared to the smooth, sometimes blurry outputs of autoregressive models.
2. **Training Speed and Stability:**
   - While PixelRNNs are slower to train due to their sequential nature, GANs can often achieve similar results in less time. This is because they utilize parallel processing during both training and generation.
   - GAN training is generally more stable than PixelRNNs. This is because GANs do not explicitly model the data distribution, which can be tricky, especially for high-dimensional data.
3. **Flexibility and Control:**
   - GANs offer more flexibility and control over the generated data compared to autoregressive models. This is because the generator and discriminator can be separately manipulated to achieve specific effects or incorporate additional information.
   - Conditional GANs can incorporate additional data like class labels or text descriptions to control the generated outputs, which is not straightforward with autoregressive models.
4. **Other Applications:**
   - While autoregressive models primarily focus on image generation, GANs have a wider range of applications beyond just image synthesis. They can be used for tasks like video generation, text generation, music generation, and even game development.
   - If the highest possible quality and fidelity are paramount, and training time is not a major concern, PixelRNNs and PixelCNNs might be preferred.
   - If efficiency and flexibility are priorities, GANs is the better choice.

4.2.5 Generate Skin Lesion Image using GAN

In our comprehensive analysis of the dermatological dataset, we meticulously examined the distribution of instances across different classes. There were 327 instances of class Akiec, 514 instances of class bcc, 1000 instances of bkl, 115 images of class df, 1000 images of mel,1000 images of nv, 142 images of vasc. We generated 673 more images for Akiec, 486 for bcc, 885 for df, and 858 images for class vasc.

In order to augment the dataset and address potential imbalances, particularly in the Akiec class, we employed Generative Adversarial Networks (GANs) to generate an additional 673 images for the Akiec class,486 for bcc, 885 for df,858 images for class Vasc. GANs, renowned for their ability to produce synthetic data with realistic features, were instrumental in enhancing the diversity and robustness of our dataset. This augmentation process not only contributed to a more equitable representation of classes but also fostered a more nuanced understanding of the 7 classes by introducing variations in their visual characteristics.

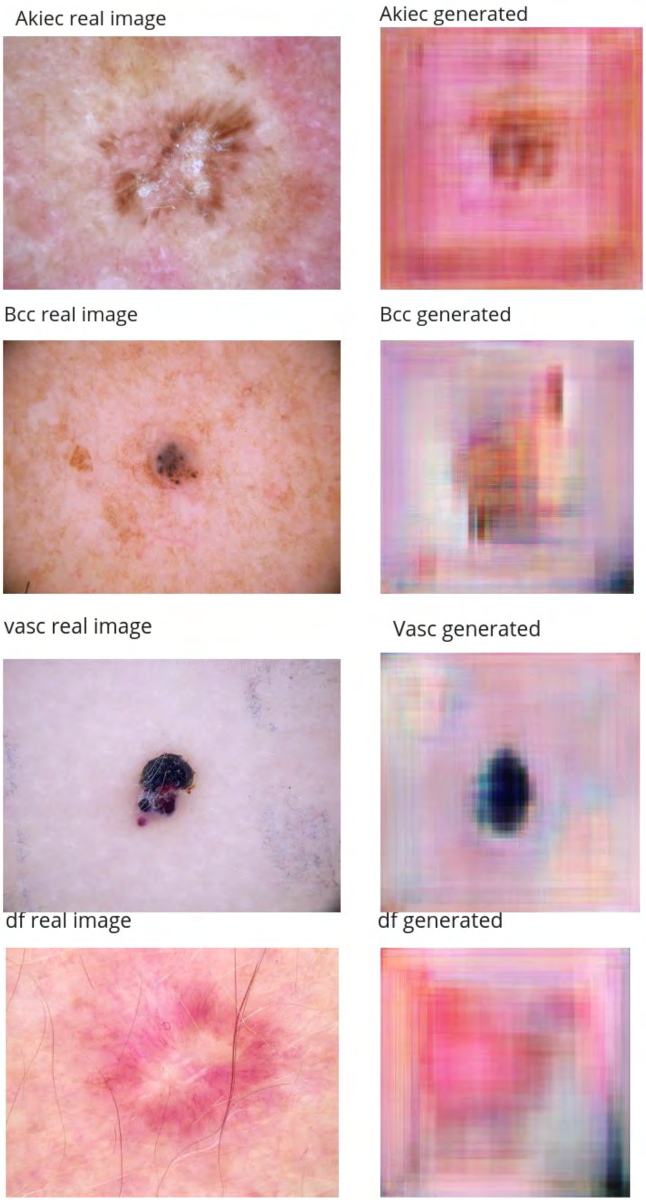


Figure 4.3: image comparison between real and generated

4.2.6 Limitations of GAN

Generative Adversarial Networks (GANs) have proven to be powerful tools for generating realistic data, but they also come with certain limitations and challenges.
GAN training can be unstable, and finding the right balance between the generator and discriminator can be challenging. GANs are sensitive to hyperparameters, and small changes can lead to drastic differences in performance. GANs are sensitive to hyperparameter choices, including learning rates, network architectures, and initialization methods. Finding optimal hyperparameters for different datasets and tasks can be time-consuming. Quantitatively evaluating the performance of GANs is challenging. Metrics like Inception Score and Frechet Inception Distance are commonly used, but they may not always correlate well with human perception of image quality.

Training GANs can be computationally expensive and time-consuming, especially for high-resolution images. It often requires powerful hardware, and convergence may take a long time. GANs do not provide explicit control over the characteristics of generated samples. While conditional GANs offer some control, generating specific, user-defined samples can be challenging. GAN training can be affected by noisy labels, and the discriminator may become saturated or reach a state where it provides little useful feedback to the generator. GANs can be used to create deepfake content, raising ethical concerns related to the generation of fake and potentially misleading information. This includes generating realistic but fake images of people, which can be misused. Understanding the internal representations learned by GANs can be difficult. The generated features may not have clear semantic meanings, making it challenging to interpret the learned representations.

While GANs have shown success in generating images, their application across different domains, such as text or structured data, is still an area of ongoing research, and results may vary. Despite these limitations, ongoing research is addressing many of these challenges, and GANs continue to be a vibrant area of study in the field of machine learning and artificial intelligence.

Chapter 5

Deep Learning Implementation and Results

5.1 Data Pre-processing

After reading the images from folder, the images are opened using PIL and then converted into array. The labels which were named according to their class is transformed into numerical values and then configured into categorical values since this is a multi class classification. The images from prior HAM10000 dataset along with the newly generated images using GAN are then read and saved into a npy formated file named image_data_with_labels.npy.
The data from the file is read and the pixel rgb information is then flattened for labeling and put in dataframe for ease of use. After converting to an array, we normalize pixel values to a range [0, 1]. Normalization ensures that the model can learn more effectively, and it often helps with model convergence during training.

5.2 Spitting Data

The Data is split 75% for training and 25% for testing. Another split was made separately for validation with 25% of the dataset.

5.3 Training Parameters

Custom ResNet50v2,Custom ResNt101V2,Custom VGG16,Custom InceptionV3, hybrid ResNetInceptionv2 and hybrid ResNetVGG16 were built and fine tuned for training with the given parameters as shown in 5.1. The training process consisted of 100 epochs. Each epoch represents a complete pass through the entire training dataset. The choice of 100 epochs was determined based on an empirical analysis of convergence and model stability.Data augmentation techniques were employed to enhance the model's ability to generalize. SGD (Stochastic Gradient Descent) was used instead of ADAM optimizer.

Model Compilation	Model Optimizer = ‘SGD’ Loss Method =’binary-Cross Entropy’
Iteration	EPOCHS = 100

Data Enhancement	rotation_range = 10,
	width_shift_range = 0.1,
	height_shift_range = 0.1,
	shear_range=0.2,
	zoom_range=0.2,
	horizontal_flip=True,
	vertical_flip=True,
	fill_mode=’nearest’

Table 5.1: Model Training Parameters

5.4 ResNet50v2 Results

An accuracy of 73% indicates the overall correctness of the model’s predictions across all classes.. However, to gain a deeper understanding of its performance, it’s crucial to analyze the confusion matrix. The fact that class 3 (dermatofibroma or df) has the highest number of correct predictions suggests that the model excels in identifying instances of this class. On the other hand, class 4 (melanocytic nevi) exhibiting the lowest correct predictions implies a potential area for improvement. class ”mel” (melanoma) has a high precision, recall, and F1-score, indicating good performance. On the other hand, class ”bkl” (benign keratosis-like lesions) has relatively lower precision, recall, and F1-score, suggesting that the model might struggle with this class.

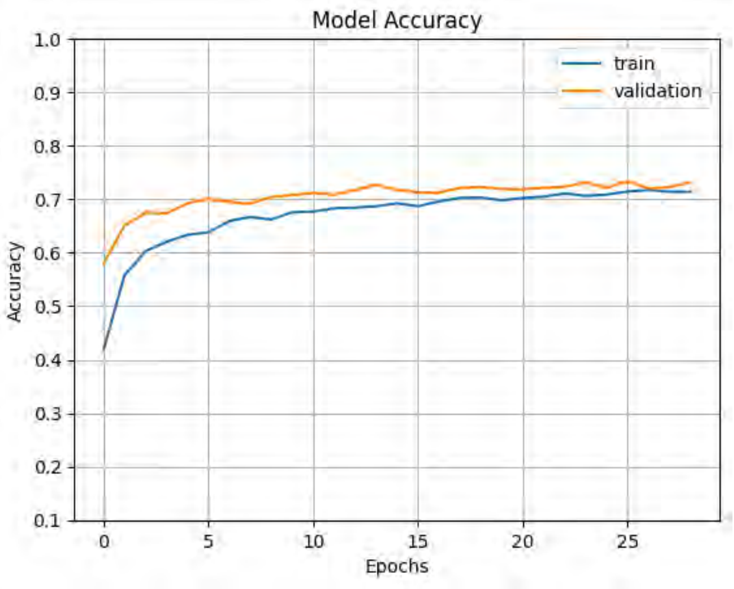


Figure 5.1: Resnet50v2 accuracy graph

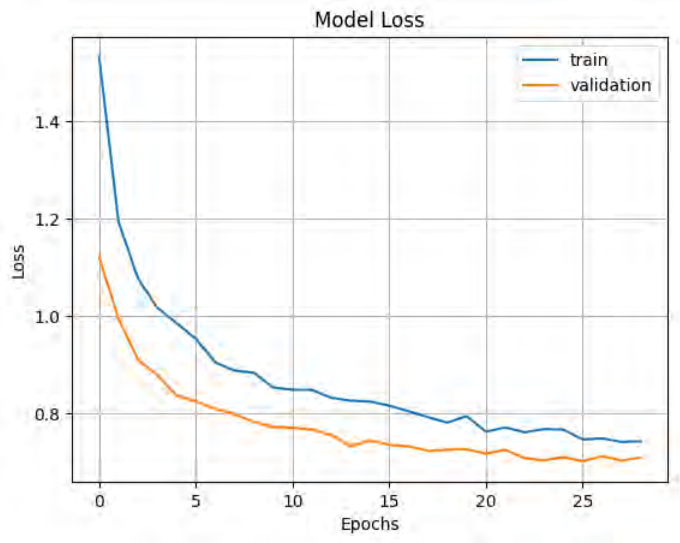


Figure 5.2: Resnet50v2 loss graph

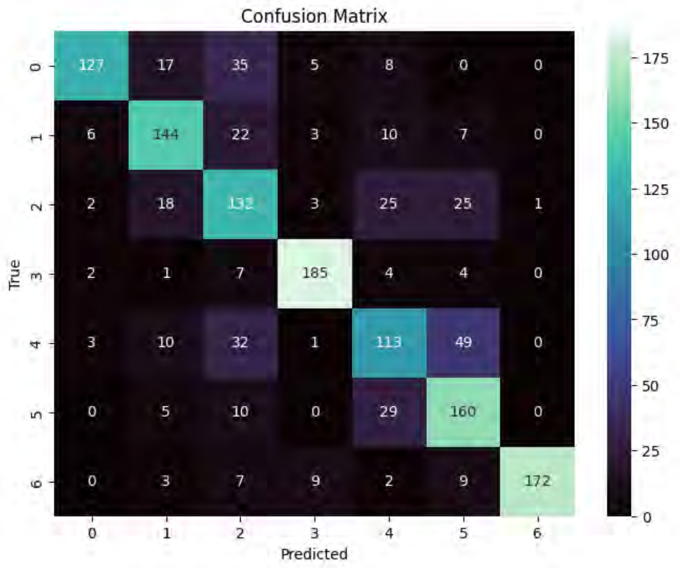


Figure 5.3: Resnet50v2 Confusion matrix

	precision	recall	f1-score	support
akiec	0.91	0.66	0.77	192
bcc	0.73	0.75	0.74	192
bkl	0.54	0.64	0.59	206
df	0.90	0.91	0.90	203
nv	0.59	0.54	0.57	208
vasc	0.63	0.78	0.70	204
mel	0.99	0.85	0.92	202
accuracy			0.73	1407
macro avg	0.76	0.73	0.74	1407
weighted avg	0.75	0.73	0.74	1407

Table 5.2: ResNet50v2 Classification Report

5.5 ResNet101v2 Results

ResNet101V2 has acquired an accuracy of 70%. In confusion matrix, class 4 has the lowest numbers of true predictions whereas class 6 or Melanoma, 'df' for short has the highest number of true predictions.

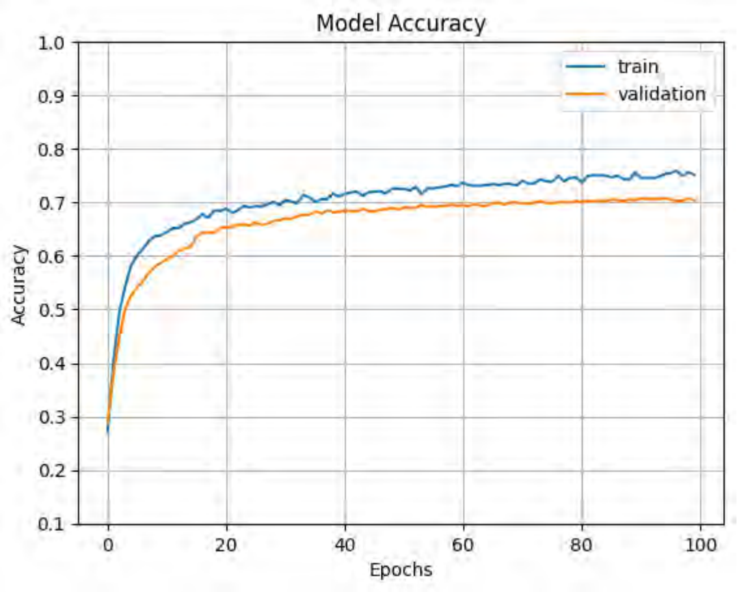


Figure 5.4: Resnet101v2 accuracy graph

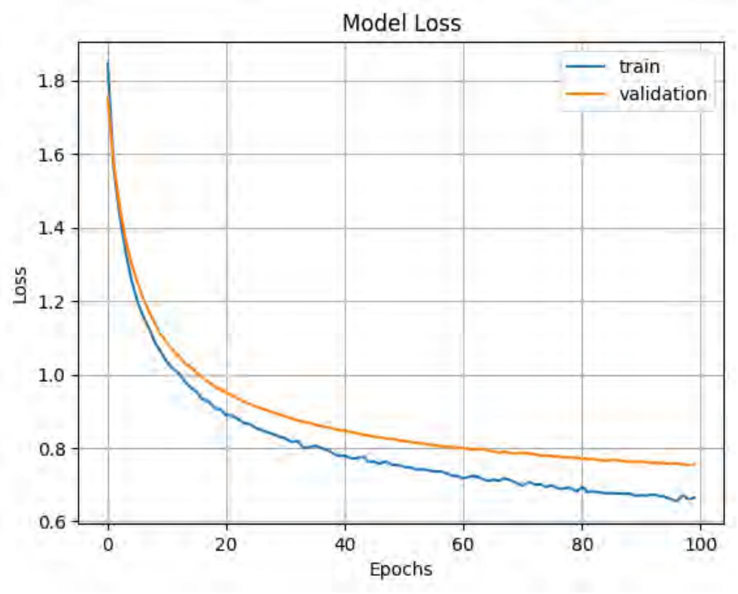
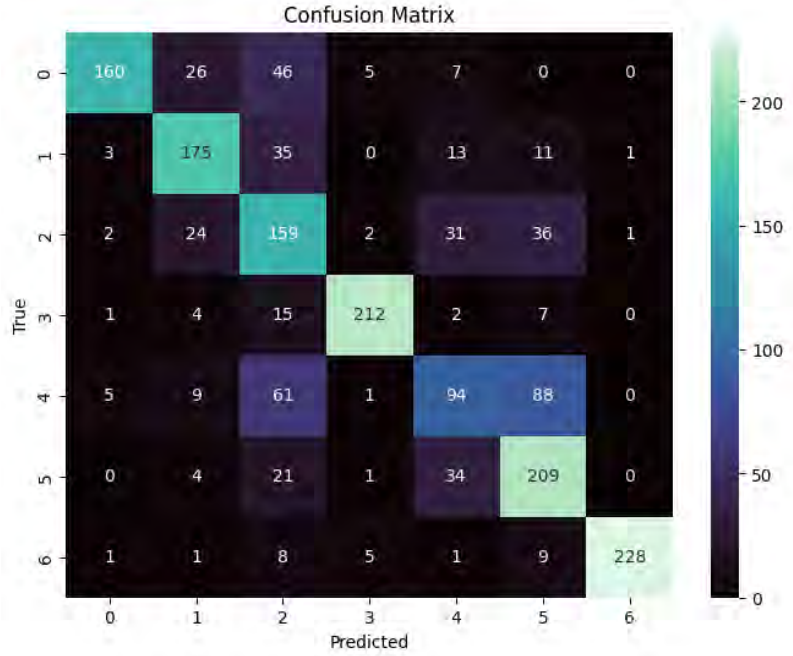


Figure 5.5: Resnet101v2 Loss graph



True \ Predicted	0	1	2	3	4	5	6
0	160	26	46	5	7	0	0
1	3	175	35	0	13	11	1
2	2	24	159	2	31	36	1
3	1	4	15	212	2	7	0
4	5	9	61	1	94	88	0
5	0	4	21	1	34	209	0
6	1	1	8	5	1	9	228

Figure 5.6: Resnet101v2 Confusion matrix

	precision	recall	f1-score	support
akiec	0.93	0.66	0.77	244
bcc	0.72	0.74	0.73	238
bkl	0.46	0.62	0.53	255
df	0.94	0.88	0.91	241
nv	0.52	0.36	0.43	258
vasc	0.58	0.78	0.66	269
mel	0.99	0.90	0.94	253
accuracy			0.70	1758
macro avg	0.73	0.71	0.71	1758
weighted avg	0.73	0.70	0.71	1758

Table 5.3: ResNet101v2 Classification Report

5.6 VGG16 Results

After evaluation, the model had an accuracy of 70%. For class melanoma the number of true predictions were the highest and for class 4 '(nv', ' melanocytic nevi')they were the lowest.

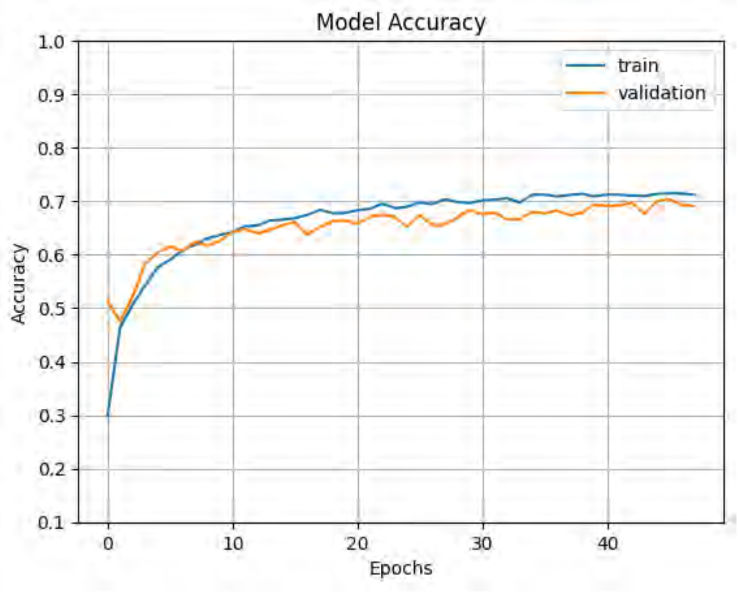


Figure 5.7: VGG16 Accuracy graph

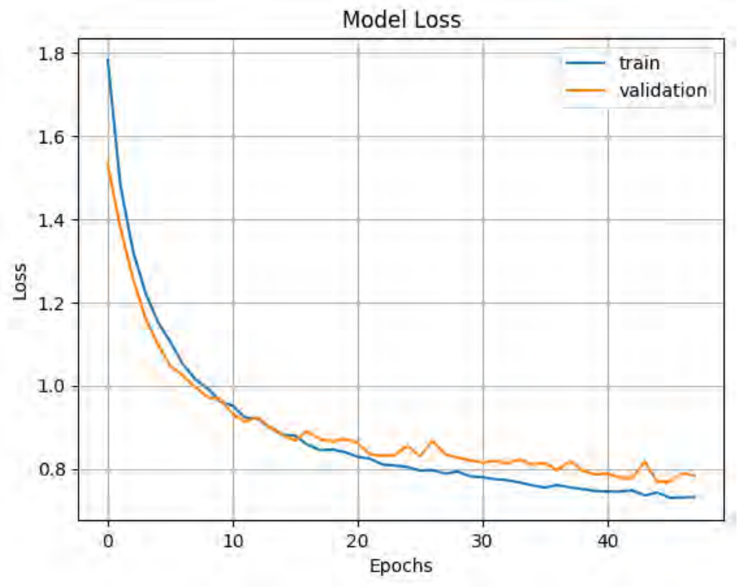
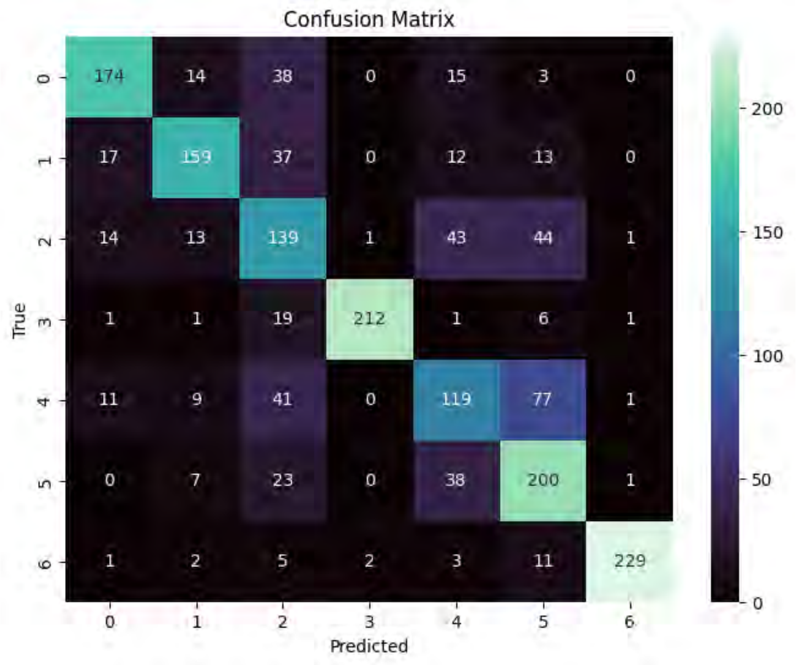


Figure 5.8: VGG16 Loss graph



True \ Predicted	0	1	2	3	4	5	6
0	174	14	38	0	15	3	0
1	17	159	37	0	12	13	0
2	14	13	139	1	43	44	1
3	1	1	19	212	1	6	1
4	11	9	41	0	119	77	1
5	0	7	23	0	38	200	1
6	1	2	5	2	3	11	229

Figure 5.9: VGG16 Confusion

	precision	recall	f1-score	support
akiec	0.80	0.71	0.75	244
bcc	0.78	0.67	0.72	238
bkl	0.46	0.55	0.50	255
df	0.99	0.88	0.93	241
nv	0.52	0.46	0.49	258
vasc	0.56	0.74	0.64	269
mel	0.98	0.91	0.94	253
accuracy			0.70	1758
macro avg	0.73	0.70	0.71	1758
weighted avg	0.72	0.70	0.71	1758

Table 5.4: VGG16 Classification Report

5.7 InceptionV3 results

InceptionV3 achieved an accuracy of 71.8%. The confusion matrix appeared well balanced and class 6 has the highest number of true predictions.

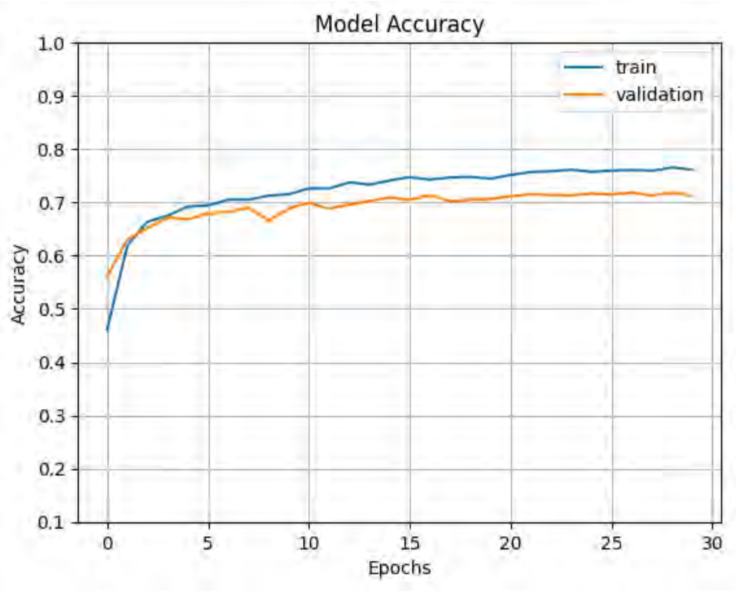


Figure 5.10: Inceptionv3 Accuracy graph

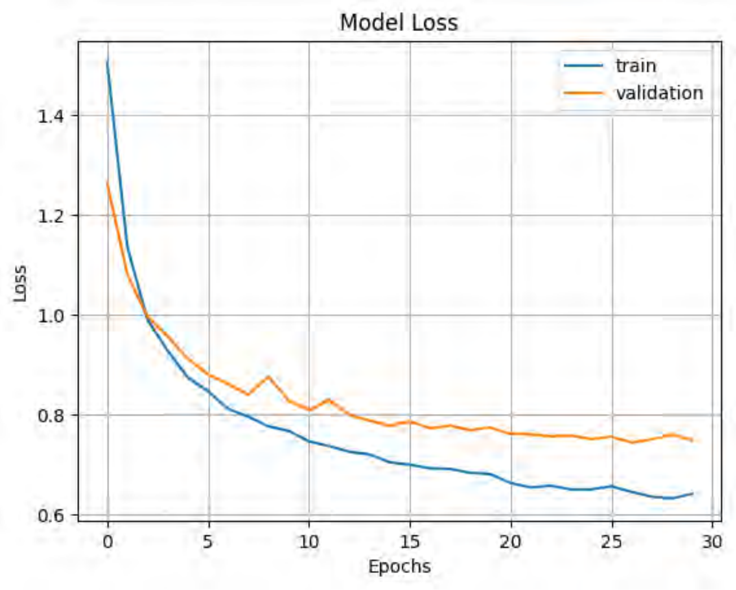
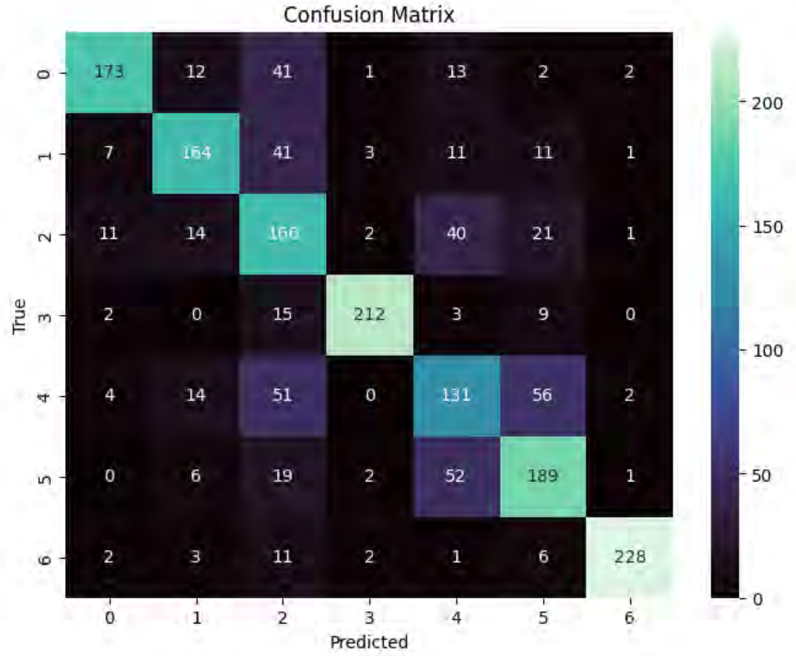


Figure 5.11: Inceptionv3 Loss graph



True \ Predicted	0	1	2	3	4	5	6
0	173	12	41	1	13	2	2
1	7	164	41	3	11	11	1
2	11	14	166	2	40	21	1
3	2	0	15	212	3	9	0
4	4	14	51	0	131	56	2
5	0	6	19	2	52	189	1
6	2	3	11	2	1	6	228

Figure 5.12: Inceptionv3 Confusion matrix

	precision	recall	f1-score	support
akiec	0.87	0.71	0.78	244
bcc	0.77	0.69	0.73	238
bkl	0.48	0.65	0.55	255
df	0.95	0.88	0.92	241
nv	0.52	0.51	0.51	258
vasc	0.64	0.70	0.67	269
mel	0.97	0.90	0.93	253
accuracy			0.72	1758
macro avg	0.74	0.72	0.73	1758
weighted avg	0.74	0.72	0.73	1758

Table 5.5: Inceptionv3 Classification Report

5.8 ResNet50V2VGG16 results

ResNet50v2VGG16 hybrid has an accuracy of 74%. The confusion matrix showcased all the classes with decent correct predictions.

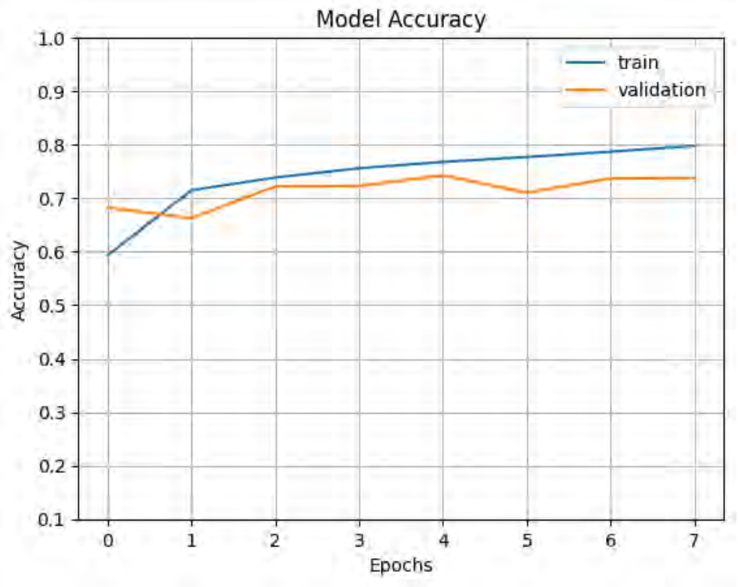


Figure 5.13: ResNet50V2VGG16 Accuracy graph

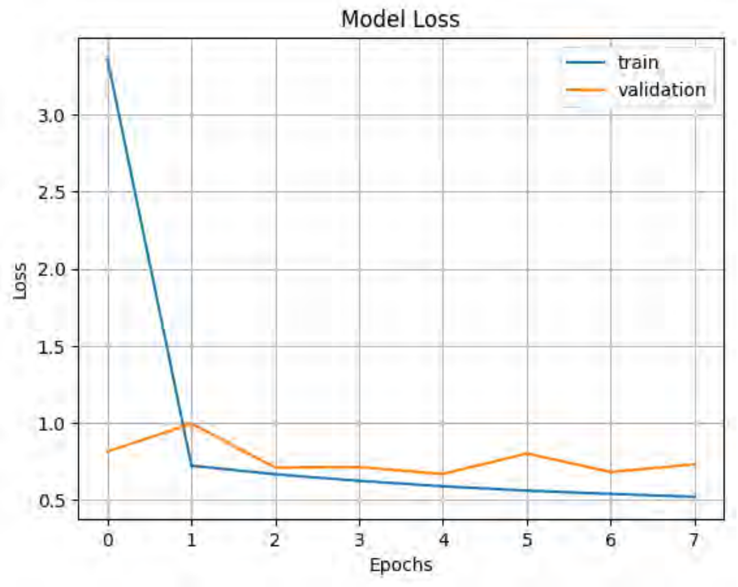


Figure 5.14: ResNet50V2VGG16 Loss graph

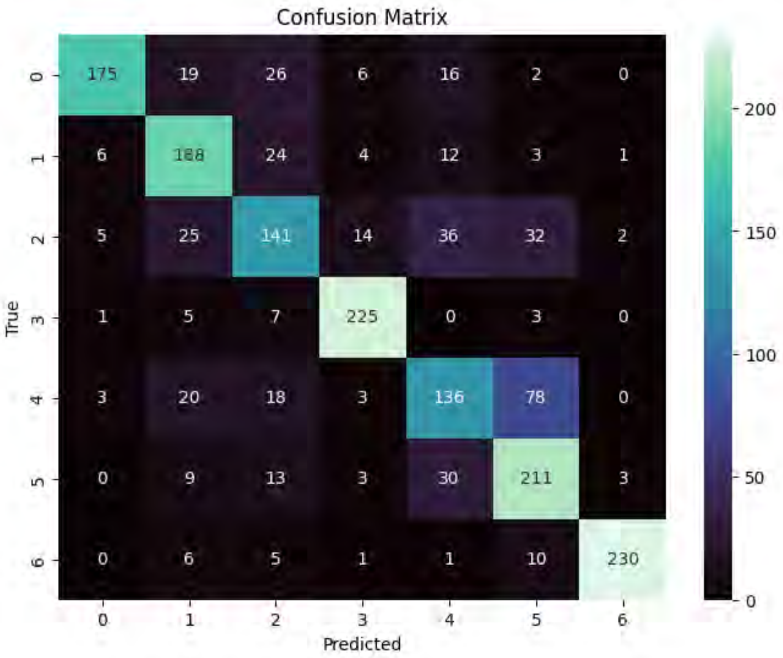


Figure 5.15: ResNet50V2VGG16 Confusion matrix

	precision	recall	f1-score	support
akiec	0.92	0.72	0.81	244
bcc	0.69	0.79	0.74	238
bkl	0.60	0.55	0.58	255
df	0.88	0.93	0.91	241
nv	0.59	0.53	0.56	258
vasc	0.62	0.78	0.69	269
mel	0.97	0.91	0.94	253
accuracy			0.74	1758
macro avg	0.75	0.74	0.75	1758
weighted avg	0.75	0.74	0.74	1758

Table 5.6: Resnet50v2VGG16 Classification Report

5.9 ResNet50V2Inceptionv3 results

Among all the models trained ResNet50V2InceptionV3 hybrid achieved the highest accuracy of 77%. All the classes had optimal correct predictions with classes 3 and 6 with the most amount of correct predictions.

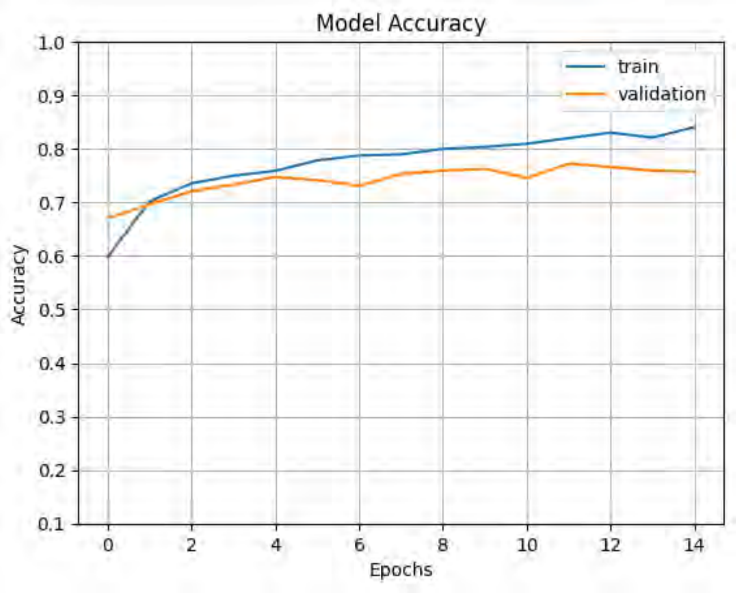


Figure 5.16: ResNet50V2Inceptionv3 Accuracy graph

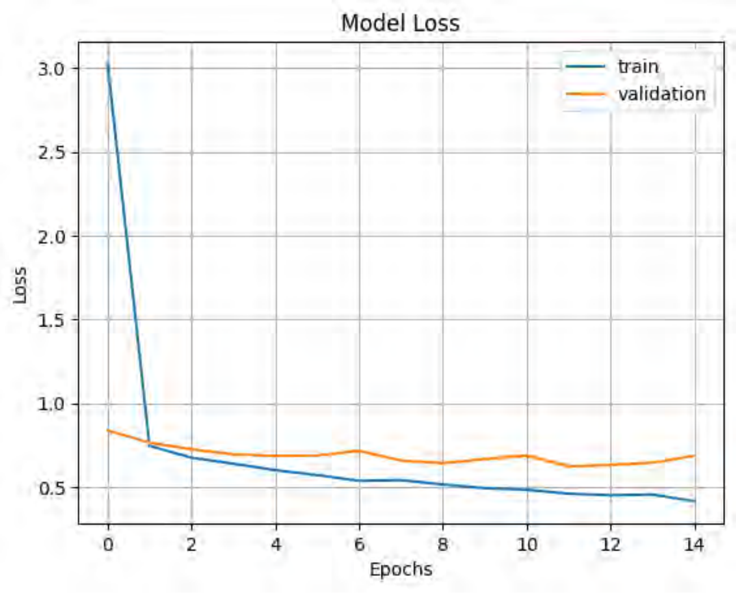


Figure 5.17: ResNet50V2Inceptionv3 Loss graph

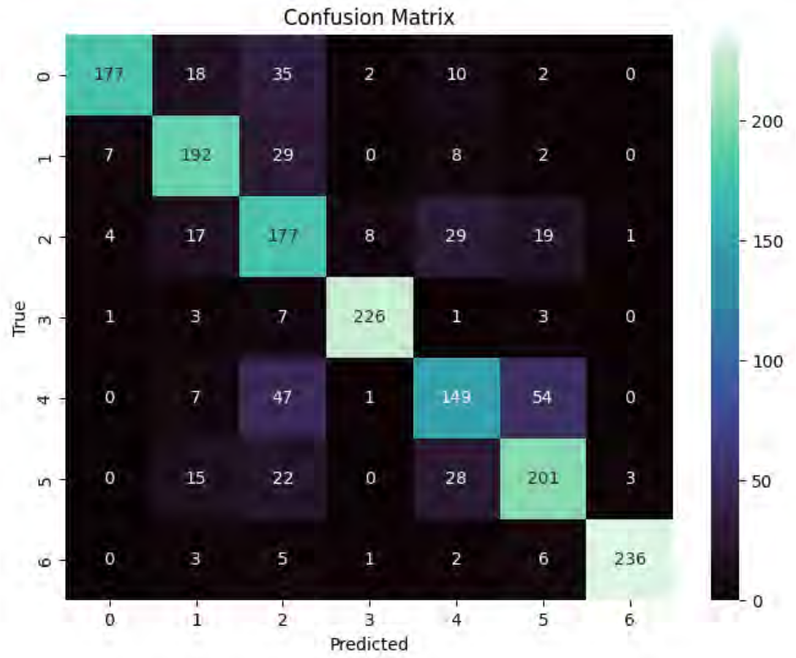


Figure 5.18: ResNet50V2Inceptionv3 Confusion matrix

	precision	recall	f1-score	support
akiec	0.94	0.73	0.82	244
bcc	0.75	0.81	0.78	238
bkl	0.55	0.69	0.61	255
df	0.95	0.94	0.94	241
nv	0.66	0.58	0.61	258
vasc	0.70	0.75	0.72	269
mel	0.98	0.93	0.96	253
accuracy			0.77	1758
macro avg	0.79	0.77	0.78	1758
weighted avg	0.79	0.77	0.78	1758

Table 5.7: Resnet50v2inceptionv3 Classification Report

Chapter 6

Model Comparison

The calibration graph shows the calibration of six different machine learning models against the mean predicted probability. The x-axis shows the mean predicted probability, while the y-axis shows the calibration. A perfectly calibrated model would have a calibration curve that goes straight along the diagonal line. As you can see, all of the models are more or less well-calibrated.

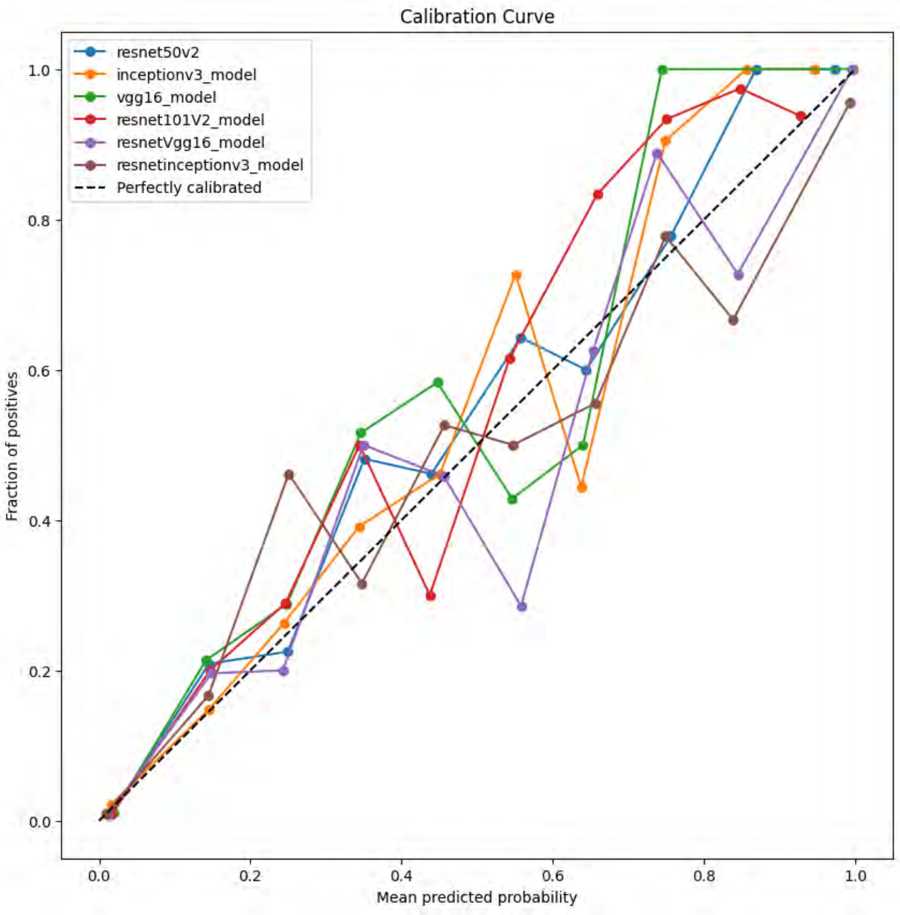


Figure 6.1: Calibraction Curve

6.1 ROC-AUC Score for different models

ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) is a metric used to evaluate the performance of classification models. It measures the area under the curve of the ROC curve, which represents the model's ability to distinguish between classes.
Different models may have different ROC-AUC scores based on their performance. For instance, a high ROC-AUC score (closer to 1) indicates better class discrimination. Among the models, ResNet50v3Inceptionv3 has the highest index.

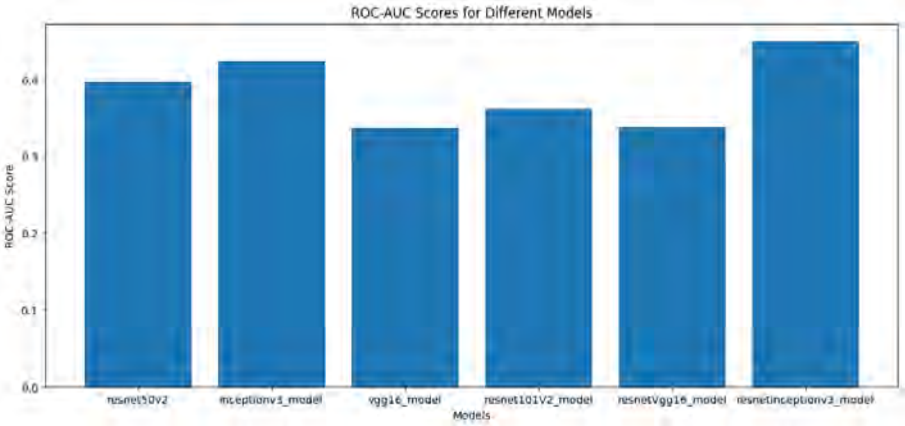


Figure 6.2: ROC-AUC Score

6.2 Categorical Accuracy Comparison of different models

While comparing all the models ResNet50v2InceptionV3 hybrid has the best results overall. The only drawback however is the size of the model. The model is significantly larger than its competitors. In this era of lightweight models, the sheer size of the model puts it at a disadvantage.

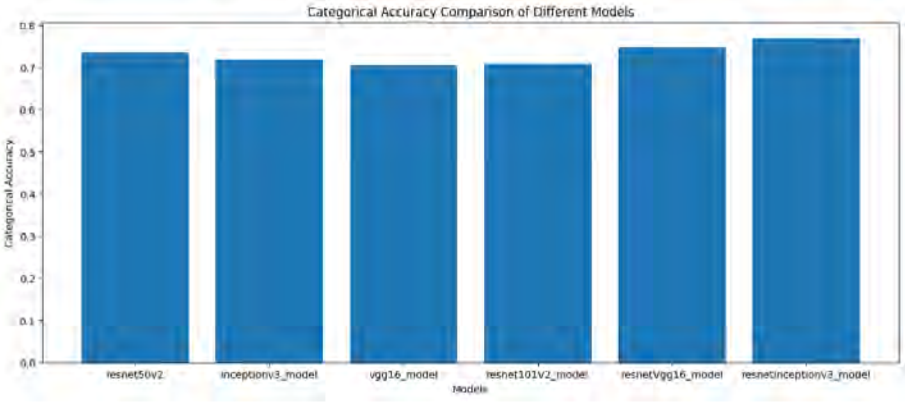


Figure 6.3: Accuracy Comaprison

Chapter 7

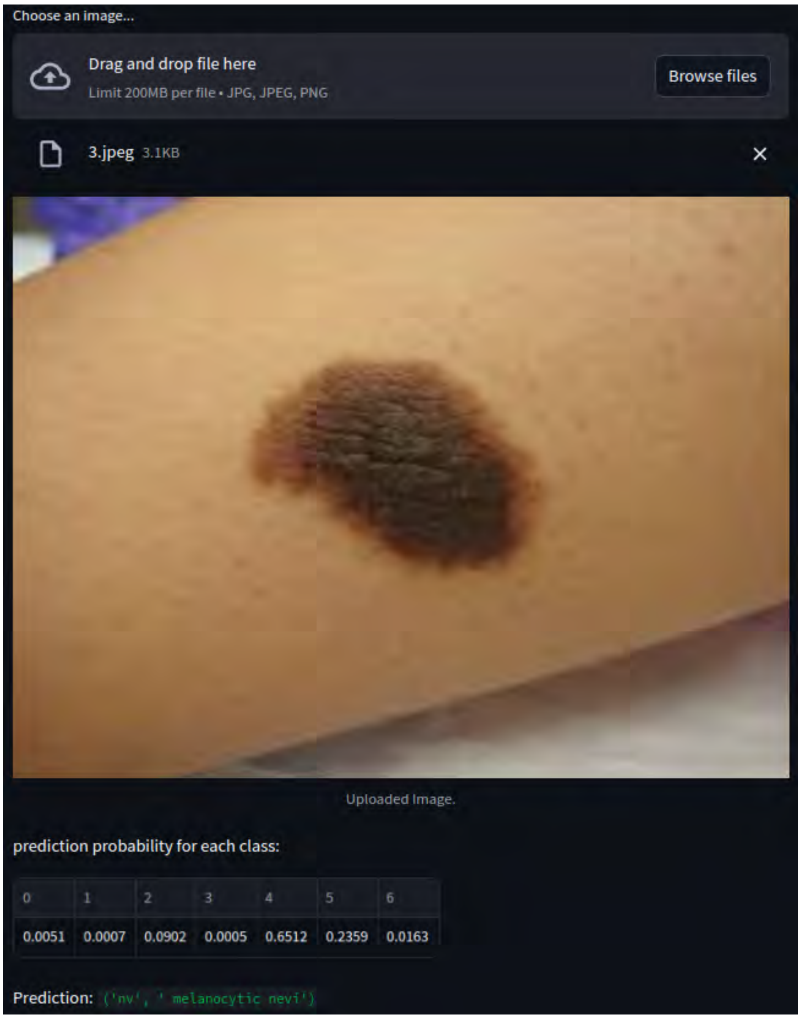
Model deploy

7.1 Streamlit Framework

Streamlit, a Python library designed for creating web applications with minimal effort, was chosen as the deployment framework. Its simplicity, flexibility, and interactivity make it an ideal choice for showcasing the model to end-users. The Streamlit application was developed to allow users to upload skin lesion images for classification. The uploaded image undergoes preprocessing before being fed into the pre-trained model. The model's prediction, including the predicted class and confidence score, is then displayed to the user. The web application features an intuitive interface, enabling users to easily upload images and receive instant predictions. A sidebar provides additional information about the application and its functionalities.

7.2 Prediction Process

Upon user file upload, the application reads the image using the Image.open function from the PIL library. The image is then preprocessed using the preprocess_image function. The preprocessing involves resizing the image to the expected input size (e.g., 224x224 pixels) and normalizing pixel values to the range [0, 1]. The preprocessed image is passed through the loaded model to obtain predictions. The TensorFlow model is capable of handling batched inputs, and in this case, a single image is converted to a batch of size 1 using tf.convert_to_tensor. The resulting prediction contains the model's output probabilities for each class. The final step involves post-processing the model's output to interpret the prediction. The classes dictionary maps the numeric class indices to their corresponding labels and descriptions. The class with the highest probability is determined using np.argmax, and the prediction results are displayed, including the probabilities for each class and the predicted class label. This prediction process provides users with valuable insights into the classification of skin lesions based on the uploaded image, enhancing the application's usability in a healthcare or dermatological setting.



Uploaded Image.

prediction probability for each class:

0	1	2	3	4	5	6
0.0051	0.0007	0.0902	0.0005	0.6512	0.2359	0.0163

Prediction: `('nv', ' melanocytic nevi')`

Figure 7.1: Prediction of type of skin lesion

7.3 Limitations

Streamlit uses GitHub for hosting purposes therefore we are bound by the restrictions put on the case of storage. There is a cap of 100 mb over uploading files to GitHub meaning our model ResNet50V2InceptionV3 hybrid which is 877 mb, can not be hosted through GitHub.

Chapter 8

Future Work and Conclusion

In conclusion, this research emphasizes the critical need for effective and accurate skin lesion detection systems, particularly focusing on the prevalent and deadly form of skin cancer, melanoma. The alarming statistics of over 5 million skin cancer cases annually in the United States alone underscore the urgency of early detection, with melanoma being a significant contributor to skin cancer fatalities.

Machine learning (ML) algorithms emerge as a promising solution for early melanoma detection by analyzing images of skin lesions and classifying them as benign or malignant. The utilization of the HAM10000 dataset presents a valuable resource to train and evaluate these algorithms, contributing to improved accuracy and reliability in skin lesion classification.

However, challenges persist in this domain, including limited availability of high-quality labeled datasets, variability in lesion appearance across different conditions, and the inherent difficulties in evaluating ML algorithm performance. The research objectives thus center on addressing these challenges, aiming to develop ML algorithms that accurately classify skin lesions, mitigate inter and intra-observer variability, and provide practical solutions for overdiagnosis and missed diagnoses.

The proposed model serves as a complementary tool for dermatologists, assisting in confirming diagnoses and reducing unnecessary biopsies. Importantly, this research seeks to bridge the gap in skin lesion diagnosis, where general practitioners may lack expertise, by offering a reliable proxy for dermatologist consultations.

Ultimately, the research objectives align with the overarching goal of making skin lesion detection through ML accessible, reliable, and easy to implement. By doing so, the research aims not only to enhance the accuracy and efficiency of skin lesion diagnoses but also to contribute to saving lives and preventing medical malpractice, ensuring that patients receive timely and appropriate care for their skin lesions. In summary, the research underscores the potential of ML in revolutionizing skin lesion detection, ultimately benefiting both healthcare professionals and patients.

Bibliography

[1] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[2] A. Narla, B. Kuprel, K. Sarin, R. Novoa, and J. Ko, "Automated classification of skin lesions: From pixels to practice," *Journal of Investigative Dermatology*, vol. 138, no. 10, pp. 2108–2110, 2018.

[3] P. Tschandl, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," Tech. Rep., version V4, 2018. DOI: 10.7910/DVN/DBW86T. [Online]. Available: https://doi.org/10.7910/DVN/DBW86T.

[4] M. Wolf, A. de Boer, K. Sharma, *et al.*, "Magnetic resonance imaging t1- and t2-mapping to assess renal structure and function: A systematic review and statement paper," *Nephrology Dialysis Transplantation*, vol. 33, no. Suppl. S2, pp. ii41–ii50, 2018. DOI: 10.1093/ndt/gfy198.

[5] Z. Yu, X. Jiang, F. Zhou, *et al.*, "Melanoma recognition in dermoscopy images via aggregated deep convolutional features," *IEEE Transactions on Biomedical Engineering*, vol. 66, pp. 1006–1016, 2018. DOI: 10.1109/TBME.2018.2866166.

[6] J. Hooker and R. Carson, "Human positron emission tomography neuroimaging," *Annual Review of Biomedical Engineering*, vol. 21, pp. 551–581, 2019. DOI: 10.1146/annurev-bioeng-062117-121056.

[7] A. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. Rodrigues, "Identifying pneumonia in chest x-rays: A deep learning approach," *Measurement*, vol. 145, pp. 511–518, 2019. DOI: 10.1016/j.measurement.2019.05.076.

[8] C. Dhivyaa, K. Sangeetha, M. Balamurugan, S. Amaran, T. Vetriselvi, and P. Johnpaul, "Skin lesion classification using decision trees and random forest algorithms," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2020.

[9] M. Jinzaki, Y. Yamada, T. Nagura, *et al.*, "Development of upright computed tomography with area detector for whole-body scans: Phantom study, efficacy on workflow, effect of gravity on human body, and potential clinical impact," *Investigative Radiology*, vol. 55, p. 73, 2020. DOI: 10.1097/RLI.0000000000000603.

[10] M. A. Khan, T. Akram, Y.-D. Zhang, and M. Sharif, "Attributes based skin lesion detection and recognition: A mask rcnn and transfer learning-based deep learning framework," *Pattern Recognition Letters*, vol. 143, pp. 58–66, 2021.

[11] S. Benyahia, B. Meftah, and O. Lézoray, “Multi-features extraction based on deep learning for skin lesion classification,” *Tissue and Cell*, vol. 74, p. 101 701, 2022.

[12] J. Morawitz, F. Dietzel, T. Ullrich, *et al.*, “Comparison of nodal staging between ct, mri, and [18F]-fdg pet/mri in patients with newly diagnosed breast cancer,” *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 49, pp. 992–1001, 2022. DOI: 10.1007/s00259-021-05502-0.

[13] B. Shetty, R. Fernandes, A. P. Rodrigues, R. Chengoden, S. Bhattacharya, and K. Lakshmanna, “Skin lesion classification of dermoscopic images using machine learning and convolutional neural network,” *Scientific Reports*, vol. 12, no. 1, pp. 1–11, 2022.

[14] F. Alenezi, A. Armghan, and K. Polat, “A multi-stage melanoma recognition framework with deep residual neural network and hyperparameter optimization-based decision support in dermoscopy images,” *Expert Systems with Applications*, vol. 215, p. 119 352, 2023.