Multi-classification Network for Detecting Skin Diseases using Deep Learning and XAI

by

Fahima Hasan Athina
18101234
Sadaf Ahmed Sara
18101284
Nishat Tabassum
18101281
Quazi Sabrina Sarwar
19101666
Mun Tarin Jannat Era
18101245

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 and Engineering

Department of Computer Science and Engineering
Brac University
May 2022

Declaration

It is hereby declared that

1. The thesis submitted is 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:

Sadaf Ahmed Sara
18101284

Fahima Hasan Athina
18101234

Quazi Sabrina Sarwar
19101666

Nishat Tabassum
18101281

Mun Tarin Jannat Era
18101245

Approval

The thesis titled "Multi-classification network for detecting skin diseases using deep learning and XAI" submitted by

1. Fahima Hasan Athina (18101234)
2. Sadaf Ahmed Sara (18101284)
3. Nishat Tabassum (18101281)
4. Quazi Sabrina Sarwar (19101666)
5. Mun Tarin Jannat Era (18101245)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on May 26, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Faisal Bin Ashraf
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

Diagnosis of a disease purely based on the prediction made by a deep learning model and without any further diagnosis by a medical professional can pique an ethical predicament in people. Thus, in our research work, we have ensured the transparency of our model predictions and we also encourage people to seek confirmation of their diagnosis from a medical expert.

Abstract

This research work aims to show a comparative analysis among four different deep learning approaches to classify three rare but deadly skin diseases namely Stevens-Johnson Syndrome, Erythema Multiforme and Bullous Pemphigoid. As the features of these diseases often overlap with each other, it becomes challenging for physicians to distinguish them with their naked eye. Thus, this research work is initiated to find a model that provides an efficient way to identify them for preventing misdiagnosis. This work also attempts to interpret the prediction of these models using LIME based Explainable Artificial Intelligence (XAI). Here, the four pre-trained models namely ResNet50V2, VGG16, Inceptionv3 and InceptionResNetV2 have been used for feature extraction. The top layer of these models have been replaced with a customized 10-layer architecture consisting of Convolution, BatchNormalization, Dropout and Dense Layers. These models have been trained on a hybrid dataset comprising of colored images of the diseases collected from different sources. Moreover, different machine learning classification algorithms (i.e. Random Forest, Logistic Regression, and Support Vector Machine) have been used to classify the images to see how well they perform compared to a neural network approach. Lastly, the accuracy of the attempted models have been compared with each other to identify which algorithm shows the best performance. The analysis shows that the InceptionResnetV2 model provides the highest accuracy of 99.06% while InceptionV3, VGG16 and Resnet50V2 provide 90.27%, 95.92% and 98.26% respectively.

Keywords: Skin Disease; Deep Learning; ResNet50V2; Inceptionv3; InceptionResNetV2; XAI

Dedication

We dedicate our work for the proper detection and diagnosis of Stevens-Johnson Syndrome, Erythema Multiforme and Bullous Pemphigoid in order to aid the unfortunate people who suffer from these diseases. We hope that our small contribution can make a significant difference in the research that will be done in the future by other researchers on these three deadly diseases using deep learning methods.

Acknowledgement

Firstly, all praise and glory be to the Almighty Allah without whom our thesis would not have been completed without any major hindrance.

Secondly, we would like to thank our Supervisor, Dr. Muhammad Iqbal Hossain, for his valuable advice and patient support. We would also like to thank our Co-Supervisor, Mr. Faisal Bin Ashraf. Their expertise and guidance were invaluable in shaping our research methodology. Their insightful feedback guided us to think from a different direction and brought our research to a higher level.

Lastly, we would like to extend our gratitude towards our respected parents for their selfless support and encouragement throughout this research work.

Table of Contents

List of Figures

List of Tables

Nomenclature

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

α Alpha

$\sum$ Summation

ANN Artificial Neural Network

AUC Area Under the Curve

BP Bullous Pemphigoid

CNN Convolutional Neural Network

DNN Deep neural network

e Euler's number

EM Erythema Multiforme

$Grad - CAM$ Gradient-weighted Class Activation Mapping

HSV Hue Saturation Value

KNN K-Nearest Neighbour

$LIME$ Local Interpretable Model-agnostic Explanations

$ReLU$ Rectified Linear Unit

$ResNet$ Residual Network

RGB Red, Green, and Blue

ROC Receiver Operating Characteristic

SJS Stevens-Johnson Syndrome

SVM Support Vector Machine

ten Toxic epidermal necrolysis

VGG Visual Geometry Group

XAI Explainable Artificial Intelligence

$YCbCr$ Green (Y), Blue (Cb), Red (Cr)

Chapter 1

Introduction

1.1 Introduction

Skin is one of the most versatile organs of our body which shields us from harmful external entities and balances our body temperature. Moreover, it provides us with probably the most indispensable senses for our survival which is the sense of touch. If such a key part of our body gets attacked severely, then it may become life-threatening.

To elaborate, skin diseases are ranked to be the fourth most common cause of human illness and from 1990 to 2017, subcutaneous and skin diseases have been found to have grown by 46.8% [15]. However, the recognition and detection of these diseases is a major challenge for medical science because skins vary from person to person on the basis of skin tone, hair on the skin, jaggedness and other features. Some of the most severe skin diseases include Stevens-Johnson Syndrome (SJS), Erythema Multiforme (EM), Bullous Pemphigoid (BP), etc. which can result in death if not treated earlier. In developing countries like Bangladesh, these diseases are very rare and difficult to detect. Also, the detection process can be very expensive. Therefore, an inexpensive, automated skin disease detection system which can distinguish between these rare diseases successfully has become very demandable for both the patients and the medical personnel.

The biggest concern in the detection of SJS, EM and BP is that their symptoms often overlap with each other. As a result, most of the time these diseases are misjudged and therefore medical malpractice may occur. Thus, a patient's condition may get worse and in some cases it may also lead to death. On the contrary, neural network algorithms may prove to easily classify these diseases if trained properly. Therefore, in this experiment, several deep neural network algorithms have been applied to analyze which algorithm works best to classify these diseases with similar clinical presentation. For this research work, 657 images have been collected from different sources to create a hybrid dataset.

Firstly, data augmentation has been applied on the input images to create a fairly large image dataset. Following that, the images have been resized via image processing to make all the images have the same size and then other necessary pre-processing techniques have been applied to fully prepare our image dataset. After that, four pre-trained deep convolutional neural network models namely VGG16, Inceptionv3,

ResNet50V2, and InceptionResNetV2 have been used to extract features from the pre-processed input images. The reason for using these pre-trained models in this experiment is to apply transfer learning techniques. Transfer learning is utilized here to conserve resources and time by not training our model from scratch. Next, the top layer of the pre-trained models have been replaced with a 10-layer CNN architecture and the extracted features have been fed to it for classifying the images. Finally, Random Forest, SVM and Logistic Regression algorithms have been applied on the dataset to compare the performance of these three machine learning algorithms with the 10-layer CNN classifier model. From the comparative analysis, it has been found that InceptionResnetV2 along with its modified top layer is able to classify the diseases with the highest accuracy of 99.06%. However, the decision making techniques of these deep neural network models are quite unclear. As a result, the medical personnel may hesitate to rely on these neural network based systems for diagnosing diseases. Therefore, in this research work, an Explainable AI technique called LIME has been applied to interpret the prediction method of the model and create a transparency among the medical personnel and the deep neural network models. Lastly, we have also built a web application by using the InceptionResnetV2 model (as it shows the best performance) so that the user can get a primary idea about which disease they are suffering from by uploading a picture of the skin lesion into this web application.

1.2 Motivation

The motivation behind this experiment is to build an effective tool to identify the severe forms of skin diseases and provide the results in a short period of time. Since the people in our country are reluctant to visit a doctor at the preliminary stages of their skin disease, this model would act as a blessing for them as it would provide promising results by using only a camera for taking the images and a computer for running the algorithm which will make their skin disease diagnosis very inexpensive and user-friendly.

1.3 Research Problem

Skin diseases can have an enormous impact on our overall health and can impede our life. Despite that, most of the people in Bangladesh do not pay attention to skin diseases as they think it is a normal part of their life. For this reason, they do not generally find it necessary to visit doctors until it becomes severe. Detection of skin diseases is sometimes very difficult for the doctors as many skin diseases have similar symptoms which makes it difficult to detect the actual disease. But if it is detected at an early stage then it would not advance further and thus patients can recover quickly and get back to their normal healthy selves. Thus, detecting skin diseases is very crucial. In this paper, we are mainly focusing on three severe skin diseases namely Stevens-Johnson Syndrome (SJS), Erythema multiforme (EM) and Bullous pemphigoid (BP) which are discussed below :

Stevens-Johnson syndrome (SJS) is an extremely severe and rare skin disease that appears in the form of a drug reaction which results in skin tissues, generally in the

mouth, genitals and eyes, to peel off as the skin tissues die. The medications that often trigger this disease include anti-epileptics, allopurinol, pain relievers, antibiotics, or cancer therapies. The initial signs of this disease frequently include high body temperatures, body pain, cough etc. A red or purplish rash develops within about 1 to 3 days, and then blisters start to appear and peel off. There are many cases where the cause of SJS cannot be determined at all and no universal diagnostic standard has been set for this skin disease yet. At present, the primary way to detect SJS is by seeing the percentage of body surface area that is affected. A skin biopsy is usually performed to confirm the diagnosis [23]. The overall mortality rate is about 10% for SJS [27].

Erythema multiforme (EM) is a kind of hypersensitive disorder that mostly occurs among unhealthy individuals, children, and young adults. Although the cause is said to be unknown, it seems to be mostly triggered by infections like the herpes simplex virus and medications such as sulfonamides, anticonvulsants, etc. Most often, the symptoms appear to be red spots, ridges, or blisters that mainly form on the forearms and hands. The skin lesions may also appear on the mouth and lips. The dimension and shade of the target lesions are usually used to determine the diagnosis of this disease. Sometimes, to rule out other conditions, a skin biopsy may be necessary [26]. One of the major treatments of EM is to eliminate the problematic medicines and sometimes there is no need for any treatment in minor cases [25]. Though minor erythema multiforme is not very serious, severe erythema multiforme can be fatal and it has a mortality rate of less than 5% [17].

Bullous pemphigoid (BP) is an autoimmune skin disease that is not very common but it can be fatal, particularly without treatment. The primary sign of this disease is redness of the skin and itching [4]. Blisters appear on the limbs, armpits, abdomen, and groin, within a period of weeks to months. Dermatologists diagnose patients with BP by examining these skin lesions. If BP is suspected, a skin biopsy is done for histology and direct immunofluorescence testing which is considered the standard for its diagnosis [22].

From the above description, we can see that SJS and EM have overlapping symptoms such as blisters and redness of the skin. Furthermore, BP which is an autoimmune reaction can also cause lesions which are similar to SJS and EM, that are mainly triggered by drug reactions [4]. Thus, it is vital to identify whether the lesions on the patient's skin are harmful reactions to medications, or whether these blisters are a cause of an autoimmune reaction. As discussed earlier, these diseases are difficult to diagnose so the problem here is that most of the time doctors conclude on a patient's symptoms based on their subjective judgment and medical experiences. If their judgment is incorrect or delayed, it can become lethal for human health. For example, in Bangladesh, a Barisal university student died because his symptoms were a lot like pox and for this reason, doctors failed to identify that he was affected by SJS [24]. Moreover, if the doctors want to distinguish among them through biopsy then it may take two to three days for the results to return which can prove to be life-threatening to the patients. That is why, we aim to develop a CNN based system where if we input images of the affected area of any of these three diseases then the system would output the name of that particular disease in a short time. Thus, the possibility of misdiagnosis can be reduced. Furthermore, a skin biopsy is

a very painful procedure but here we are proposing a skin disease detection model which can detect the particular skin disease quickly and painlessly.

1.4 Research Objectives

This research mainly aims to show a comparative analysis among four different deep learning approaches to classify three rare and fatal skin diseases namely Stevens-Johnson Syndrome, Erythema Multiforme and Bullous Pemphigoid and develop a convolutional neural network based detection system for detecting these severe skin diseases.

The research objectives are listed below :

- To have a deep understanding of CNN
- To develop a model to classify the following three diseases :
  - Stevens-Johnson Syndrome (SJS)
  - Erythema Multiforme (EM)
  - Bullous Pemphigoid (BP)
- To analyze the performance of ResNet50V2, VGG16, Inceptionv3 and InceptionResNetV2 on our hybrid dataset
- To find the best performing model for a newly created dataset
- To see how well the 10-layer CNN architecture can classify the diseases in comparison to other ML algorithms like Random Forest, Logistic Regression and SVM
- To interpret the prediction of the models using LIME
- To develop a system that would reduce the chances of misdiagnosis of these diseases
- To develop a system that would aid doctors in their diagnosis
- To offer a painless, fast and inexpensive skin disease detection process

Chapter 2

Literature Review

2.1 Related Works

In this segment, some of the research works which are related to the field of skin disease detection systems using neural network or machine learning algorithms have been reviewed critically. Moreover, some other techniques and algorithms which were previously used for skin disease detection have also been reviewed and their accuracies have been analyzed .

In this research work [6], 3000 samples of images have been used as a dataset in order to recognise diseases such as lichen planus, acne and sjs ten. Here, the convolution layer has been used for pattern recognition. A Relu function has been used to increase non linearity in the images. Moreover, a pooling function has been used here to reduce the input size. As a result of this, the complexity of computation decreases and also it reduces the chance for overfitting. Then, the training precision of each run has been calculated. Following that, the parameters which were found after running the algorithm have been compared with the five classification models namely kernel SVM, naive bayes, logistic regression, CNN and random forest. Furthermore, they have used a graphical representation as well to show which algorithm works best for detecting skin diseases. For visualizing the errors made by the classifier, confusion matrix has been used. The accuracy results for logistic regression, random forest, kernel SVM, CNN and naive bayes were 68%, 67%, 50%, 96% and 47% respectively. Thus, it has been concluded from the above analysis that CNN gave the best accuracy among the other algorithms and also had the lowest error rate which was equal to 0.04.

In this paper [21], the authors have aimed to develop a method for classifying malignant and benign melanoma images. They have used 4 different datasets containing images of benign and malignant melanoma and these 4 datasets have been collected from the ISIC-Archive repository, the ISBI 2016 dataset, the MED-Node dataset, and lastly, the PH^2 Database. Then they have applied data augmentation techniques to enhance their model's generalization capability by incorporating rotation up to 30 degrees, zoom, flip and shift. Their proposed method consists of a base model and a head model. For the base model, they have used MobileNetV2 to implement transfer learning. Then they have added a global pooling layer followed by 2 fully-connected layers on top of the MobileNetV2 base model as their head model. The output of the base model has been fed into the global pooling layer in the form of a $7\times7\times1280$

feature map. This global pooling layer, which uses an average pooling operation with filter size 7×7, has been used to create a 1D feature vector of size 1×1×1280 and to reduce the data dimension. The 2 fully-connected layers have 128 and 64 neurons respectively along with ReLU activation functions and the output layer has only 2 neurons with a Softmax function. In this study, the authors have used the pre-trained weights of the base model and have only trained the head model with an Adam optimizer. As the Adam optimizer has a fast convergence, they have trained the head model for only 20 epochs and found that this model did not face overfitting. They have then changed the base model with ResNet50V2, InceptionV3, and InceptionResNetV2 to analyze their performance on the ISIC-Archive dataset. They have found that the MobileNetV2 model has the highest accuracy and running time of 85% and 2394s respectively with a specificity, sensitivity and precision of 85%, 85% and 83% respectively. The accuracy of ResNet50V2, InceptionV3, and InceptionResNetV2 were 84%, 81% and 78% respectively. The authors have stated that since ResNet50V2 had the highest sensitivity (87%) and lowest specificity (81%), it implies that it can accurately detect the malignant class but it is comparatively not as good as the other three architectures in detecting the malignant class. The InceptionResNetV2, however, has been found to have the highest specificity (91%) and lowest sensitivity value (63%). Furthermore, they have found that the MobileNetV2 model provides low sensitivity to the PH^2 and ISBI 2016 datasets. This shows that it failed to correctly classify the malignant images because these datasets had a high data imbalance. Thus, they concluded that further processing such as usage of weighted cost function or over-sampling may help this problem of highly imbalanced data.

In this paper [13], the authors have attempted to compare between deep learning and machine learning algorithms. For this purpose, they have used Bagged Tree Ensemble, KNN, and SVM for machine learning and for deep learning, they have used three pre-trained deep neural network models ResNet50, VGG16, and GoogleNet. Bagged Tree Ensemble combines several decision trees to perform the classification or regression operation and uses several weak learners to create a strong learner. VGG16 is a 16 layers deep convolution neural model and it helps to classify images of 1000 objects. GoogleNet is a 22 layers deep convolutional neural model that is not as large as VGG16 and its error detection capability is far lower than other network models. On the other hand, ResNet is a 50 layers deep, pre-trained convolutional neural network which has 4 stages and 1000 neurons that help to classify images of 1000 object categories. All the layers of ResNet help in the deeper classification of an image. Therefore, the classification accuracy of ResNet is much higher than any other deep learning model. After that, they have shown that with VGG16 the accuracy was 58.21%, with GoogleNet the accuracy was 64.67% and with ResNet50 the accuracy was 69.23%. On the other hand, with Bagged Tree Ensemble, the accuracy was 92.99%, with KNN the accuracy was 92.52% and with SVM the accuracy was 83.53%. Therefore, it has been found that the accuracies of the applied deep learning algorithms were comparatively lower than those of the machine learning algorithms that were used.

Artificial Neural Network has played a vital role for differentiating nine varieties of skin diseases namely acne, leprosy, eczema, psoriasis, scabies, foot ulcer, pityriasis rosea, vitiligo and tinea corporis in this paper [2]. For this research work, first of

all, the authors have utilized colored images of skin for pre-processing. After that, they have used eight different image processing algorithms which are sharpening filter, YCbCr, grey image, median filter, Sobel operator, binary mask, histogram and smooth filter to distinguish the features of the color code of the skin's infected place. For pre-possessing, the colored image of the affected area has been transformed into grayscale image and sharpening filter has also been used to fix the shape of the affected area. Furthermore, median filter has been used for matrix configuration so that noise can be removed from the model. They have also used a smoothing filter to change the pixel value of the images with its nearby mean value. Thus, binary images were created to identify the edge of the affected area. Then, their system took user inputs for liquid color, elevation, feeling, gender, age, duration and liquid type. Along with that, ten different features have been extracted from the pre-processed images. These two processes have been done to further train and test their feed forward ANN algorithm. The accuracy of this system for detecting these nine varying diseases has been found to be 90%. However, their system performed poorly for acne and tinea corporis but gave very high accuracy for foot ulcer and vitiligo.

In the research paper [8], the authors have tried to identify six quite common facial skin diseases using five well known CNN algorithms (i.e. InceptionV3, InceptionResNetV2, Xception, ResNet50 and DenseNet121). At first, they have build a hybrid dataset consisting of 4394 clinical images of the six facial skin diseases - actinic keratosis, lupus erythematosus, rosacea, squamous cell carcinoma, seborrheic keratosis and basal cell carcinoma. These images have been assembled from China's largest skin disease dataset Xiangya–Derm and among them 60.45% images were facial images and 39.55% images were of other body parts. For each of the 5 CNN networks, images of 300×300 dimensions have been used as input. Then, to minimize the no. of parameters and to manage the spatial data, the authors have replaced the 1st fully-connected layer (which was after the last conv layer) with global average pooling and a 1×1 convolution. After that, they have finally applied a 1024-d fully-connected layer in every network and then, Softmax was used to get six outputs for the six different skin diseases. In this research work, the five CNN models have been trained using only the facial clinical images at first and among all the models, InceptionResNetV2 had the highest recall of 67.2% and highest precision of 63.7%. Next, the model has been pre-trained with the images of different body parts and for the new model, the parameters that the authors have got from this pre-trained model was applied as the initial parameters. InceptionResNetV2 has performed better than the other algorithms for this new pre-trained model too. It has given the highest recall of 77% and highest precision of 70.8%. Moreover, by comparing the two models, it has been found that, the models which have been pre-trained on the different body part images performed better than the models that have been trained only based on the facial skin images. Furthermore, guided propagation and grad-CAM was also used in this paper for visualizing the results of InceptionResNetV2.

Here [1], color image processing methods, color gradient methods and k-means clustering have been used in order to recognize the part of the skin that was affected by the disease. At first, both visual and non-visual data from the patient's history have been collected by the authors for developing the system. Following that, the images have been pre-possessed to separate the affected skin from healthy ones and then,

the images have manually been cropped in a way that it will only contain healthy and diseased areas. A color-based customized Sobel operator has been used for generating color gradients and after doing so, k-means clustering has been performed on those by applying a threshold. For obtaining a binary mask, morphological closing has been performed and by applying the mask, diseased skin has been separated from the healthy ones. After that, automated visual and external feature extraction have been applied. Then, for training, they have used a feed forward back propagation neural network which has then been validated using a tenfold cross validation process. Lastly, the system has been tested on 2055 diseased areas in 704 images of six diseases all together. The accuracy of detection has been found to be 95.99% and the accuracy for identification was found to be 94.016%.

In this research study [18], the authors have proposed an ensemble architecture for the diagnosis and classification of malignant skin lesions. This architecture consists of three transfer learned deep vision networks namely VGG16, InceptionResNetV2 and DenseNet. These deep learning algorithms have been implemented on two different datasets named HAM1000 and ISIC and on these datasets, data augmentation had been applied beforehand. In their proposed model, the authors have obtained the average of the three models (i.e. VGG16, Inception ResNetV2 and DenseNet) to get the optimal accuracy with maximum authenticity. For getting the average for this combined architecture, the equation that have been used is -

$$X(\mathrm{y} : \alpha) = \sum_{j=1}^{n} \alpha_j x_j(\mathrm{y}) \tag{2.1}$$

Here, X is represented as the output of the three different models, input image is denoted with y, model weights are represented as α and n is represented as the summation of the numbers of models used and so, in this case, n is three. Here, for the final predicted result, the weighted average predictions of all the models have been used because α_j provides the same weights to each of the three algorithms. In this research work, the authors have implemented other deep learning algorithms and from their experiment, they have found that VGG16, Dilated VGG16, MobileNet, Mod MobileNet, RestNet50, InceptionResnetV2, VGGNet and DenseNet201 have shown 85%, 87.42%, 83.1%, 88.7%, 87.1%, 87%, 81.3% and 86% accuracies respectively. But the best accuracy was shown when InceptionResNetV2 and ResNeXt101 were applied together and it was 92.83%. Here, the proposed network has given the accuracy of 90% which is not the highest but it has shown the highest recall and precision which were 89% and 89.5% respectively. This high recall and precision proves the correlation and robustness of the ensemble model.

In another research work [7], a deep CNN model has been introduced to classify various skin diseases. In this experiment, the authors have used MED-NODE, Derm (IS and Quest) and ISIC datasets to train and verify their model. The problem that they had to face during the experiment was that the datasets were very small and it was not sufficient to train the model properly. The ISIC dataset has a total of 2000 images consisting of Seborrheic Keratosis, Melanoma and Nevus. The MED-NODE dataset has a total of 170 images consisting of melanoma and nevus. The third dataset Derm (IS and Quest) has a total of 206 images consisting of melanoma and nevus. Since these are very small datasets, these datasets may not be able to

learn the model properly. To solve this problem, in the paper, the authors have introduced data augmentation. Using this method, the authors have increased the number of images and created large datasets. For data augmentation, firstly, they have applied random rotation on each of the images where each image was rotated 72 times using the angle range of 0-335 degree. Secondly, they have applied 5 degree fixed step angle rotation. After completing data augmentation, a pre-trained deep CNN model namely AlexNet has been used to classify the images into their respective diseases. Although the authors have already applied data augmentation to increase the number of images, it is not sufficient to train a CNN model from scratch. Therefore, to save time and resources, they have used a pre-trained AlexNet model. In their proposed model, the authors have excluded the fully-connected layers and replaced it with Softmax layer. To train the new weights, they have introduced a learning rate and they have also applied back propagation and after that, the weights have been fine-tuned. For Derm (IS, Quest) they have got an average accuracy rate of 96.86% and for MED-NODE and ISIC it has been 97.70% and 95.91% respectively.

CNN has been used for extraction of features and SVM has been used for classifying skin diseases in the following paper [3]. A straightforward morphological closing operation has been used for removing background noise like air bubbles and hair to prevent any inaccuracy which might arise due to them. This closing operation has been used along with a circular structuring component. Here, a media filter has also been used for smoothing the images by eliminating extra noise. In this research paper, three major steps- (a) Convert Color Space, (b) Feature Extraction and (c) Clustering, have been taken for segmenting the images based on pixels. RGB color space does not work well for detection and analysis which are done on the basis of color. Thus, for describing colors in terms of their shade and brightness, HSV has been used here. Texture segmentation has been used here as lesion classification is hard to carry out when it only depends on color features. K-means has been applied for identifying collections of pixels which are similar and this has been done on the basis of the texture features. Because of its flexibility in extracting discriminating and intrinsic features, here, the authors have used CNN for feature extraction. On the extracted feature, a linear SVM has been used for classification. SVM has been used here because it is robust, simple and prevents overfitting of train data. Their system accuracy for identifying non melanoma as well as melanoma patients was 94.12% and the comprehensive precision for correctly diagnosed patients was 94.12%.

In this paper [10], the authors have established their experiment on the ISIC public dataset containing beyond 10,000 images of diseases which are either malignant or benign. As the original dataset had data imbalance, the authors had made a subset of the dataset by randomly taking twenty percent images of both the malignant and benign classes. Then they proceeded to use the VGG16 convolutional neural network model to extract 1000 features from each of the input images. This VGG16 model consists of convolutional and pooling layers. Each of these convolutional layers has 3×3 dimensional filters. These filters have been used to extract the low level features in the primary stage and then they have been used to extract the intermediate and high level features at a later stage. To elaborate, the authors have conducted their experiment in 2 levels. In the initial level, they have extracted the low, intermediate

and high level features from the images in the dataset. Then, in the next level, they have inserted labels into the extracted features. Lastly, these features have been fed into SVM, KNN, Decision Tree and Linear Regression for classification of the images. These classifiers have been used here as they work well for linear and binary classification problems. Then, the authors have evaluated the accuracy of these classifiers using ROC curve and confusion matrix. After implementing their experiment, the authors have found that Linear Discriminate, SVM, Decision Tree, K-Nearest Neighbor, and Ensemble Tree have shown 99.9%, 98%, 99.9%, 99.9% and 48.2% accuracies respectively. Therefore, these accuracies show that the dataset that has been used here suits linear binary models and does not suit course multi-class algorithms.

From the above discussion, it can be observed that various different approaches have been taken in the past in order to detect different skin diseases. Some researchers had used machine learning and neural network algorithms like SVM, KNN, random forest, ANN, etc. while others had used deep learning algorithms such as InceptionV3, MobileNet, Resnet, etc. Several of these previously proposed models had high accuracy. For instance, the model in [13] had an accuracy of 90% using ANN. However, no such research has been done in the past for the classification of SJS, BP and EM together. Moreover, in our experiment, we have created a completely new hybrid dataset consisting of skin lesion images of SJS, BP and EM collected from various different sources. Thus, no other experimentation has been performed on this dataset except ours. Since these three diseases have skin lesions that are very similar to each other, it acts as a challenge for our classifier to detect these similar diseases and have high accuracy. If our proposed model can succeed in distinguishing among them accurately, it will prove to be a great aid to the medical personnel in diagnosing these diseases quickly and with a reduced chance of misdiagnosis.

Chapter 3

Background Study

There are various types of pre-trained models available for feature extraction such as Xception, VGG, ResNet, Inception, DenseNet etc. However, in this research work, VGG16, ResNet50V2, InceptionV3 and InceptionResNetV2 have been examined. These are popular pre-trained deep learning models for extracting features from input images. Then, to classify the images into three categories - SJS, EM and BP, a customized 10-layer CNN model has been set up. Furthermore, some machine learning algorithms i.e. Random Forest, Logistic Regression and SVM have been used here for classifying these diseases and for comparing their accuracy with that of the 10-layer CNN model.

3.1 CNN

Convolutional Neural Network or CNN is most commonly used for analysis of images in deep learning models. However, it can also be used in other types of data analysis and classification problems. It is a class of ANN that can detect patterns and classify images according to those patterns. The main architecture of CNN has an input layer, multiple hidden layers which can be made up of convolutional layers and non convolutional layers and an output layer. The hidden layers mainly help to detect patterns and analyze the images. CNN works based on the shared weight concept. The neurons of each layer have a weight and a threshold value and they connect with other neurons that have the same weight. A neuron is activated when the output of that neuron is greater than the threshold and thus that output is fed into the subsequent layer of the network. The layers of a general CNN are:

- Convolutional layer
- Pooling layer
- Fully-Connected layer

The convolutional layer is the most significant layer of CNN where most of the computation occurs. This layer receives the input and transforms the inputs using some filters (kernels) and sends this transformed input to the next layer. This process of transforming the input is known as convolution operation. In convolution, we need to define the number and size of filters to be used and these filters help to detect the patterns of the images. These filters go over the input image using the

specified stride and filter the input. After completing the filtering process, a feature map is created. Usually, the activation function utilized in this layer is a ReLU function. This ReLU activation function removes all the negative values from the feature map and replaces them with zero. It only activates when the input of the neuron is above the threshold value. Therefore, when the input of the neuron is less than zero, it gives zero as output. On the other hand, when the input crosses the threshold value, the function shows a linear relationship with the dependent variable as shown in figure below.

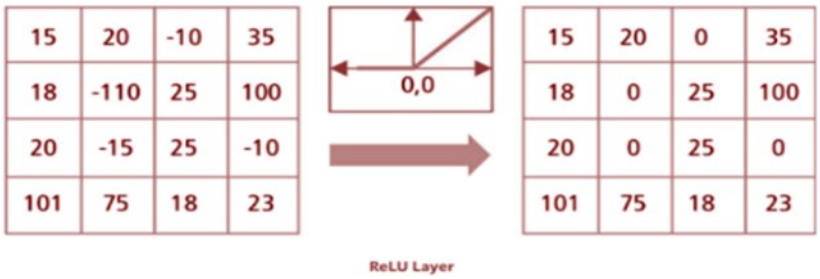

Figure 3.1: A visual representation of how ReLU works

After applying convolution, we can apply the following equation to determine the dimension of the output image:

$$W_0 = \frac{W_i - F + 2P}{S} + 1 \tag{3.1}$$

$$H_0 = \frac{H_i - F + 2P}{S} + 1 \tag{3.2}$$

Where, Wo = width of the output image
Ho = height of the output image
Wi = width of the input image
Hi = height of the input image
F = size of the filter
S = the value of stride
P = padding value

Then comes another important layer of CNN which is the pooling layer. Pooling layers can be of 2 types which are max pooling and average pooling. This layer is mainly used to lessen the number of parameters as the lesser the number of parameters, the easier it is to learn the model. The final layer of CNN is the fully connected layer which is basically used to classify the images based on the outputs of the previous layers. Usually Softmax function is used as activation function in this layer.

3.2 Transfer Learning

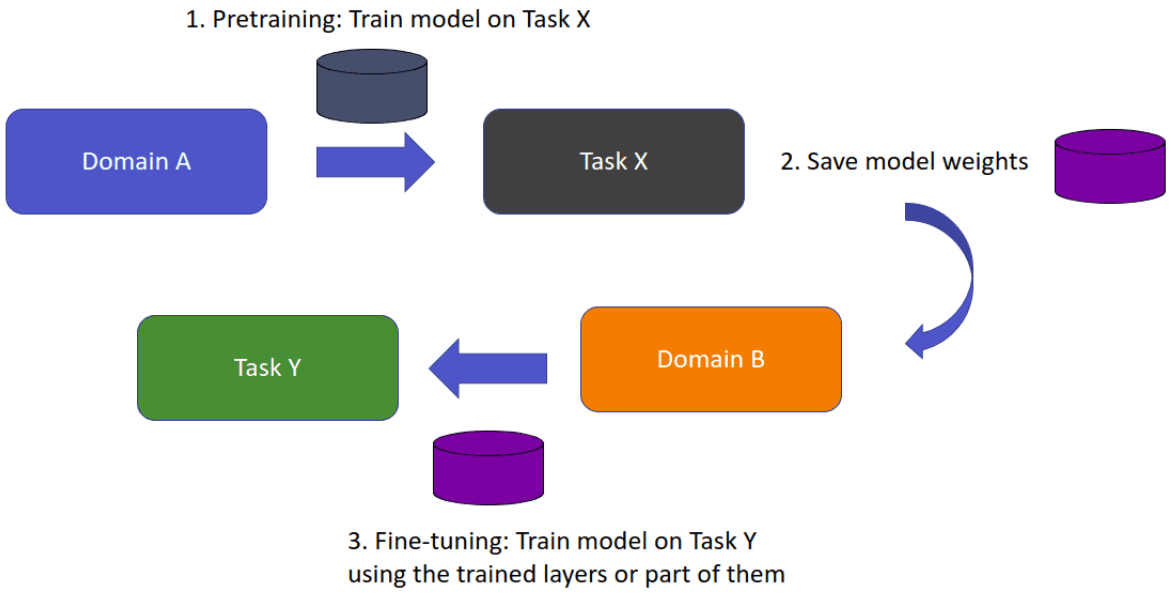

Figure 3.2: Visual representation of transfer learning

Transfer learning is a method of reusing a pre-trained model as the base for a new model to execute a new and different task. From Figure 3.2, it can be visualized that a pre-trained model is a model that has been previously trained on a particular dataset acquired from Domain A to execute a particular task, Task X. This model contains weights and biases that represent features of Domain A. In neural networks, the initial layers usually try to detect and extract the low level features such as the edges in the images whereas the intermediate layers try to detect different shapes in the images. On the other hand, the more task-specific features are detected in the final layers of the network. Thus, in transfer learning, the initial and intermediate layers of the pre-trained model are used and only the latter layers are retrained on a different dataset belonging to Domain B to perform a different task, Task Y. Transfer learning usually only works if the model features learned from Task X, that is the first task, are general.

In deep learning, a significantly large amount of data is required to train a neural network from scratch. However, it is not always possible to gain access to such a large quantity of data. Thus, using transfer learning, a firm model can be built with relatively less training data because the network is already pre-trained. This also reduces the training time significantly because training a deep learning model on a complex task from scratch can take several days to weeks. It also results in better performance of neural networks in most cases.

3.3 Pre-trained Transfer Learning Models

3.3.1 VGG16

VGG16 is a convolutional neural network architecture which is sixteen layers deep and it has thirteen convolutional layers and three fully-connected layers. It was proposed by Karen Simonyan and Andrew Zisserman in 2004. The VGG16 model reached 92.7% test accuracy on ImageNet, which is mainly a dataset of 14 million images of one thousand classes, in 2004. This model brings improvement over AlexNet architecture. The figure below illustrates the architecture of the VGG16 model.

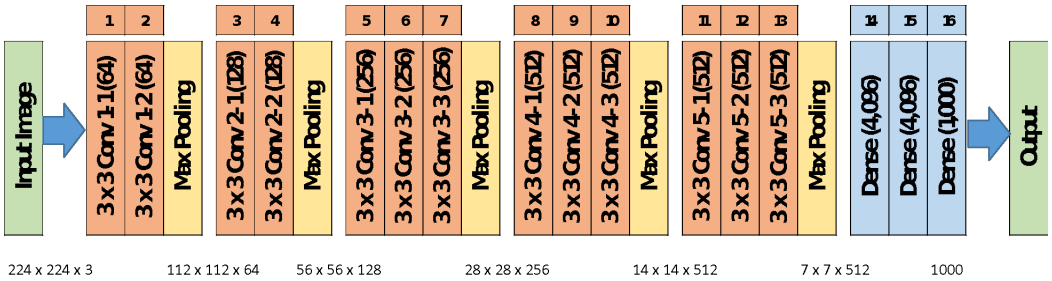

Figure 3.3: VGG16 Network Architecture

In this architecture, a fixed size of 224×224 images are given as input in the first convolution layer and these inputs are further passed to the subsequent convolution layers. In most of the configurations, these conv layers consist of filters with 3×3 dimension. However, in one of the configurations, these layers also use 1×1 convolutional filters which is basically a linear transformation. The stride of these convolutional layers is always the same and it is equal to 1. The padding of the convolutional layers with 3×3 dimension is also of 1. Spatial pooling is performed by five max pooling layers. These max pooling layers are carried out by a pixel window of 2×2 dimensions and the stride of these layers are equal to 2. Finally, the architecture is completed with fully-connected layers along with a Softmax for computing the output probabilities. Also, all the hidden layers of VGG16 use an activation function called ReLU which helps to reduce vanishing gradient problem in the model [5].

3.3.2 ResNet50V2

ResNet or residual neural network is a deep neural network architecture. It has many variants that follow the same notion but have different layers and among them ResNet50V2 is a variant which consists of fifty layers. It is a modified version of ResNet50 and shows a better performance compared to ResNet50. The modification in ResNet50V2 from the ResNet50 architecture is in the propagation formulation of the connections between the convolutional blocks. The key concept of ResNet is to introduce skip or shortcut connections in the architecture that skip one or more layers and perform identity mappings, as shown in the figure below:

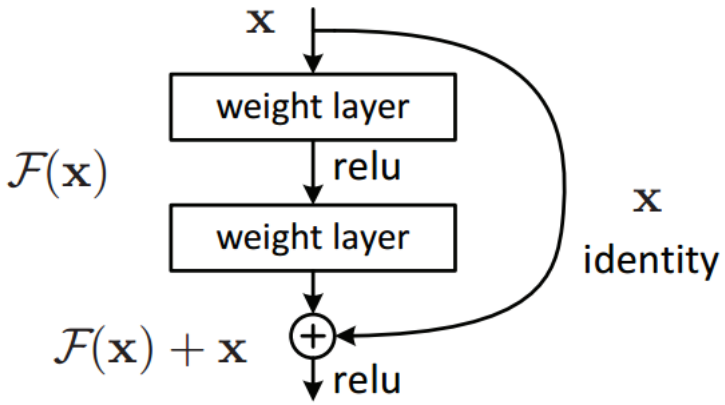

Figure 3.4: A residual block

In a skip connection, the convolution layers are not only stacked on top of one another but the original input is also added to the output of the convolution block. As a result of skip connections, the problem of vanishing gradients that occur as neural networks get deeper is alleviated because the gradient has another shortcut path to flow through. This helps in making the architecture deeper while reducing the rate of error. Moreover, skip connections make it simpler for the model to learn identity functions and they make sure that both the higher and lower layers perform similarly well. Since the ResNet50 model has been trained on the ImageNet dataset, it has a good capability to represent features of a broad spectrum of images.

The ResNet50v2 architecture starts with a convolution layer with 64 different kernels, each having a size of 7×7 and a stride of 2. It is then proceeded by a max pooling layer with kernel size of 3×3 and stride of 2. Next, there is a block (Block 2) that contains a convolution layer with 64 kernels of size 1×1 followed by another convolutional layer with 64 kernels of size 3×3 and then followed by a third convolutional layer with 256 kernels of size 1×1. These three convolutional layers are repeated two more times giving a total of nine layers in Block 2. The next block (Block 3) consists of a convolution layer with 128 kernels of size 1×1 followed by another convolutional layer with 128 kernels of size 3×3 and then lastly followed by a third convolutional layer with 512 kernels of size 1×1. These three convolutional layers are repeated three more times giving a total of twelve layers in Block 3. In the next block (Block 4), there is a convolution layer with 256 kernels of size 1×1 followed by another convolutional layer with 256 kernels of size 3×3 and then followed by a third convolutional layer with 1024 kernels of size 1×1. In a similar manner, these three layers are repeated five more times giving a total of eighteen layers in Block 4. After that, the next block (Block 5) consists of a convolution layer with 512 kernels of size 1×1 followed by another convolutional layer with 512 kernels of size 3×3 and then lastly followed by a third convolutional layer with 2048 kernels of size 1×1. These three layers are repeated two more times giving a total of nine layers in Block 5. All these blocks are then stacked by a classifier consisting of an average pooling 2D layer, a Flatten layer, a Dense Fully-Connected layer containing 1000 nodes, a dropout layer and lastly the output layer with a Softmax function. This classifier is considered a single layer so in total the ResNet50V2 architecture has $1 + 9 + 12 + 18 + 9 + 1 = 50$ layers [11]. The figure below demonstrates the ResNet50V2 architecture.

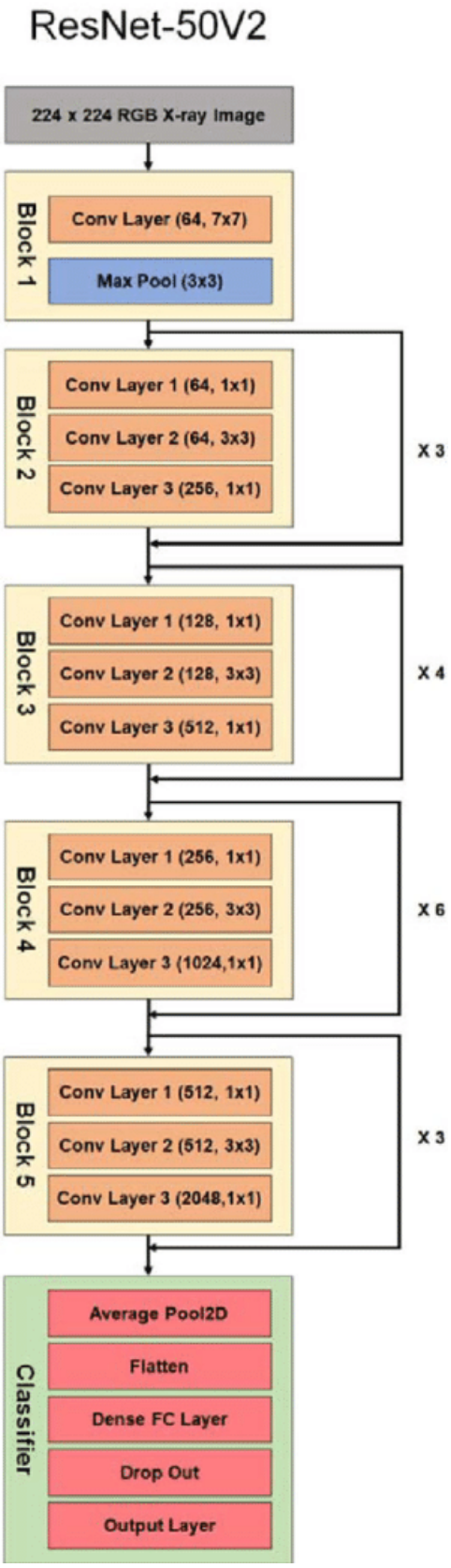

Figure 3.5: ResNet50V2 architecture

3.3.3 InceptionV3

InceptionV3 is a convolutional neural network architecture which is used for tasks such as image classification, image analysis and object detection. It is the third version of the inception model introduced by a team of researchers at Google and it has been trained on the ImageNet dataset. This inception version is superior to InceptionV1 in terms of accuracy and speed due to the many modifications made to it.

The concept of Inception V1 emerged because the main features of an image can drastically vary in size. As a result, the location of those salient parts can vary largely. This makes it difficult to choose the correct kernel size for the convolutional layers because globally distributed features require larger sized kernels whereas locally distributed features require smaller sized kernels. Moreover, as networks become deeper, the model tends to overfit and the computational cost tends to increase. Thus, in the inception architecture, different sizes of kernels are used in the same level. The outputs of these kernels are then concatenated and fed to the next inception module. Therefore, due to having parallel layers, inception architecture gets wider than deeper which helps the problem of overfitting. Additionally, to reduce the dimensions and increase the computational speed of deep networks, a 1×1 convolutional layer is added before each convolutional layer. In the InceptionV3 model, the larger convolutions are factorized into smaller ones to reduce the dimensions even further. For instance, the 5×5 convolutional layers are replaced by two 3×3 convolutional layers which reduces the number of parameters. Furthermore, the 3×3 convolutions are replaced with a 1×3 convolutional layer followed by a 3×1 convolutional layer i.e. spatial convolutions are turned into asymmetric convolutions. If the input and output filter numbers are the same, this 2 layer solution proves to be thirty-three percent cheaper for the same number of output filters. Auxiliary classifiers are also used here along with Batch Norm as a regularizer. Label smoothing is used as well in InceptionV3 as a regularizer to reduce overfitting. Lastly, the activation dimensions of the network filters are extended to reduce the grid size of the feature maps by utilizing 2 parallel blocks of pooling and convolutional layers which are later concatenated together. For instance, a m×m grid of k filters are reduced to m/2×m/2 grids with 2k filters [12]. The figure below shows the InceptionV3 architecture:

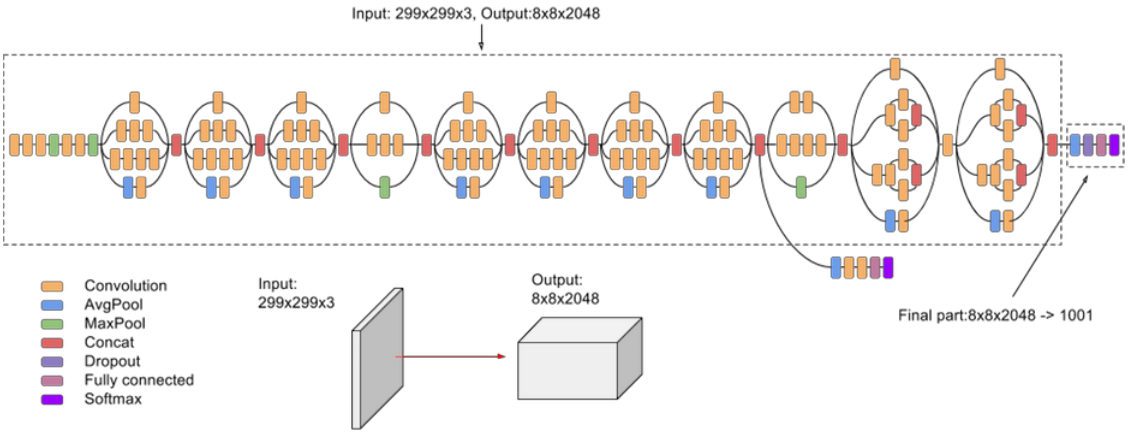

Figure 3.6: InceptionV3 Network Architecture

3.3.4 InceptionResNetV2

InceptionResnetV2 is a 164 layer architecture which has been built based on the Inception family. It was trained on over a million images which belong to the Imagenet dataset and so, it is possible for this network to classify images into thousands of categories. Thus, for an extensive variety of images, this architecture has learned rich feature representations.

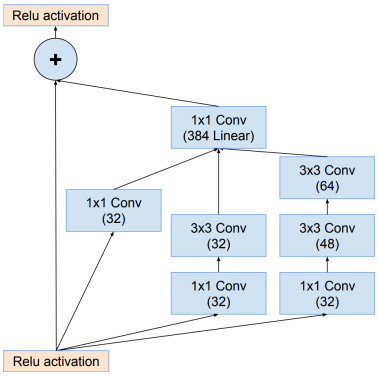

Figure 3.7: A Residual Inception Block

In InceptionResNetV2, residual connections are combined with Inception architecture. In the Residual Inception block, convolution filters of various sizes are added with the residual connections. There are filter expansion layers inside the Residual Inception block which are used to scale up the filter bank dimensionally before matching the depth of the input. Here, batch-normalization is generally used on top of the traditional layer and not on the top of other layers.

The incorporation of residual connections with the Inception structure has not shown a significant improvement in InceptionResNetV2's performance as it performs quite similarly as the InceptionV3 network. However, the actual benefit of combining the residual connections in InceptionResNetV2 is that it minimizes the training time and also, it puts a stop to the degradation problem which is caused due to deep structures. The training time is reduced because the training of Inception architectures speeds up notably when it is trained with residual connections [9]. The figure below shows the compressed view of InceptionResNetV2 architecture :

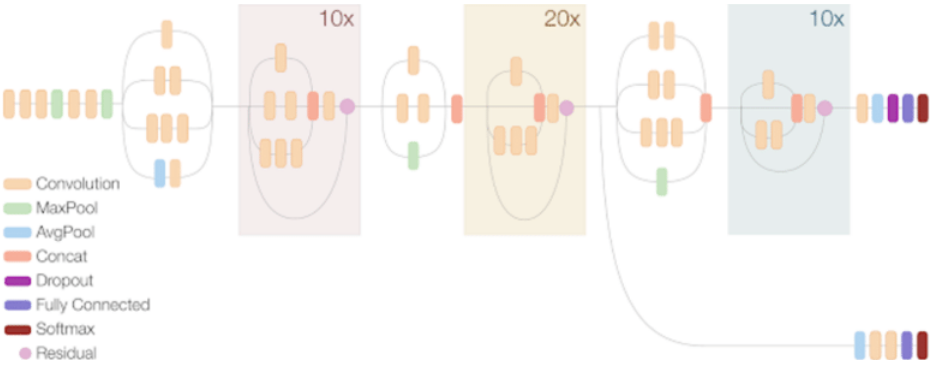

Figure 3.8: InceptionResNetV2 architecture

3.4 Explainable AI (XAI)

The calculation process of a Neural Network or a Machine Learning model is impossible to interpret. That is why these models are referred to as 'black-box'. It can not be determined what is exactly happening inside those models or how these models are getting the specific results. Thus, to understand the behaviors of deep Neural Network models, Explainable AI which is also known as XAI is required. It is a tool that can interpret the prediction mechanism of the deep learning models and helps to create transparency among the classification models and the human beings. It aims to produce more explainable models that human beings can trust and effectively use in the field of artificial inteligence.

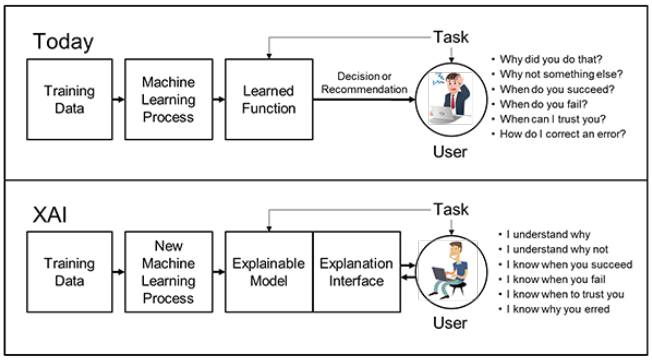

Figure 3.9: Explainable Artificial Intelligence

Various tools and techniques have been developed by the researchers for the explainability and the interpretability of the deep learning models and a few of them have shown very promising results over the recent years. Some of these techniques are Gradients, Integrated Gradients, DeepLIFT, Guided Backpropagation, Deconvolution, Class Activation Maps (CAMs), Grad-CAM, Grad-CAM++, Layer-wise Relevance Propagation (LRP), SmoothGrad, RISE algorithm, Concept Activation Vectors (CAVs), DeepTaylor, PatternAttribution, Local interpretable-model agnostic (LIME), DLIME, SHAP, Ancors, Contrastive Explanations Method (CEM), Permutation Importance (PIMP), L2X, PDPs, Accumulated Local Effect (ALE), ICE, LIVE, the BreakDown method, ProfWeight etc.

3.4.1 LIME

LIME is a technique that has the ability to interpret any black-box model. It is one of the simplest and most popular XAI techniques. The full form of LIME is Local interpretable-model agnostic. Here, the model agnosticism refers to the explainability of LIME for any supervised deep learning models in the world. When a prediction model and a test sample is given as input into the LIME, it first does the sampling and obtains a surrogate dataset. By default, it applies normal distribution to the feature vector and generates a surrogate dataset of 5000 samples. Then using the prediction model, the target variable of these 5000 samples is acquired. Each row of the surrogate dataset is weighted based on the closeness of the newly created samples with the original sample. Next, to select the top and important features, a feature selection technique called Lasso is performed and then the prediction result is outputted [16].

3.5 Machine Learning Classifiers

3.5.1 Random Forest

Random Forest is an ensemble tree that is a collection of many random decision trees. An ordinary decision tree algorithm is highly sensitive to training data and this could result in it having high variance. Therefore, the decision tree might end up failing to generalize. Random Forest algorithm, on the contrary, is much less sensitive to the training data. In this algorithm, sub-samples are formed from the original training data by using the process of bootstrapping and then various decision trees are made on these bootstrapped datasets. A random subset of the features are then selected for each of the decision trees during their training process. The average of all the results of these decision trees are then computed to get a better accuracy. This process of combining the results from multiple models is called aggregation. Here, bootstrapping ensures that the same data is not being used for each decision tree and this helps the model to be less sensitive to the original data. The random feature selection, on the other hand, helps to reduce the correlation between the trees. If it used every feature, then most of the trees would have the same decision nodes and will act very similarly which will lead to increased variance. Random Forest has also been proven to solve the issue of overfitting. However, despite providing better results than the decision tree algorithm, it shows some lacking like taking more time compared to the decision tree. The figure below demonstrates a Random Forest Tree.

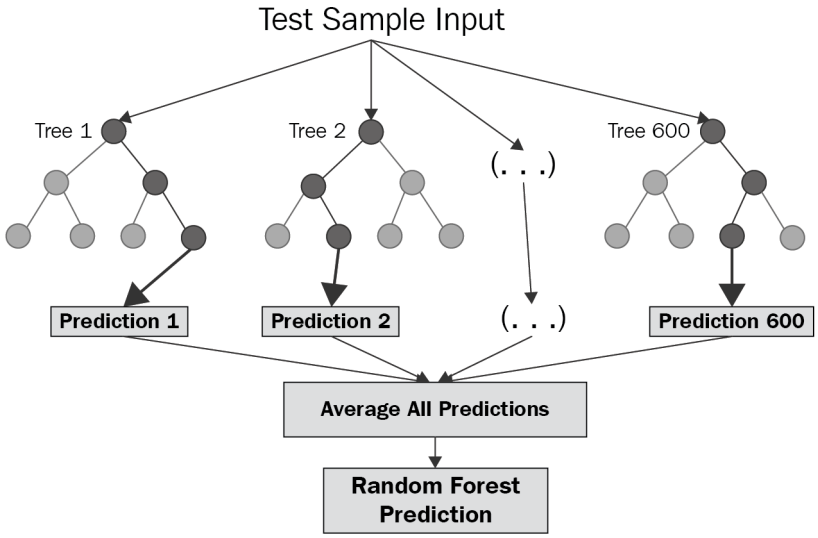

Figure 3.10: Random Forest Tree

3.5.2 Logistic Regression

Logistic Regression is a machine learning algorithm which is used for classification problems. It is based on the concept of probability and thus, it is a predictive analysis model. We can classify it into three classes which are binomial, multinomial and ordinal.

Logistic regression is used whenever we want to use a categorical variable as our dependent variable. For example, if we want to classify whether cancer is malignant or not, we will use logistic regression rather than linear regression as logistic regression gives the appropriate result for these kinds of categorical variables.

The aim for using logistic regression is to find out the occurrence of a variable using probability and these prediction values are confined between 0 and 1 which is predicted using a sigmoid function. The formula of sigmoid function is given below:

$$Y = \frac{1}{1 + e^{-x}} \tag{3.3}$$

x = Independent variable
Y = Dependent variable

Using this sigmoid function the independent variable has been transformed into an expression of probability that spans between zero to one with respect to the dependent variable. The following figure demonstrates how a sigmoid function classifies data points between 0 and 1 in logistic regression.

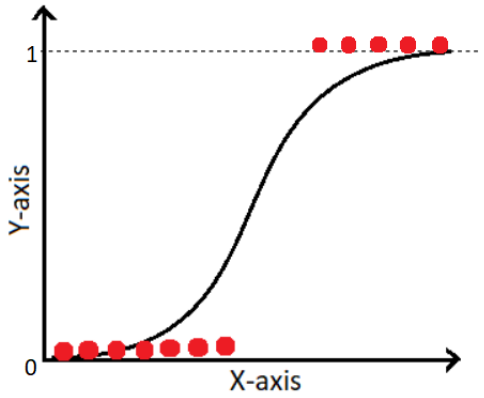

Figure 3.11: Logistic Regression

3.5.3 SVM

Support Vector Machines or SVM is a swift and reliable supervised machine learning classification algorithm which is utilized for both regression and classification problems. It performs significantly well when used to analyze small datasets.

The main purpose of SVM is to create a hyperplane that separates different classes of data points. After creating the hyperplane, it also creates two planes which are parallel to that hyperplane. The distance between the original hyperplane and the parallel planes is called the margin. In SVM, we can create multiple hyperplanes but our main aim is to create a hyperplane that has maximum marginal distance as the model will be more generalized when we have a higher margin. The data points which are nearest to the hyperplane that passes through the marginal parallel planes are the support vectors and they play a big role in classifying the data points correctly. All these techniques in SVM are used for linearly separable points. Figure 3.10 demonstrates the separation of linear data points using SVM.

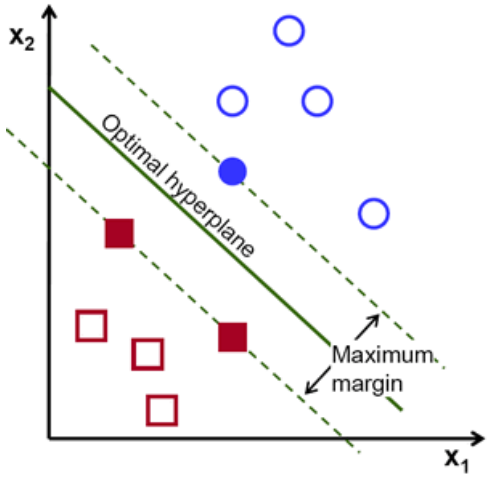

Figure 3.12: Linear data separation using SVM

But in non-linearly separable points, we cannot just draw a line to separate two different kinds of data points. Thus, in the case of non-linearly separable points, SVM uses a technique named SVM kernels. The goal of SVM kernels is that it tries to transform the low dimensional data into high dimension. When the low dimension gets converted into a higher dimension, the data points get arranged differently. As a result, it becomes easy for SVM to create a hyperplane between two different classes of data points for distinguishing them. After classifying the data points, the algorithm will again map back to the lower dimension where the data points originally were. Figure 3.11 demonstrates the separation of non-linear data points using SVM.

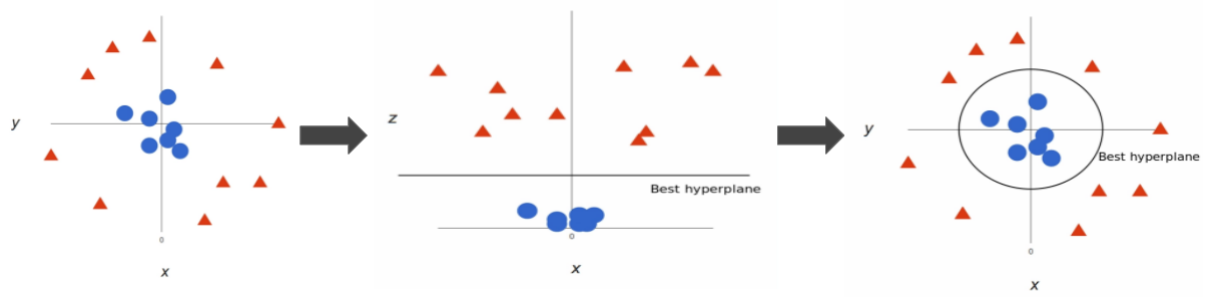

Figure 3.13: Non-linear data separation using SVM

Chapter 4

Methodology

4.1 Research WorkFlow

The methodology of our research work proceeds through the following steps :

1. Data collection
2. Data pre-processing
   (a) Background removal and cropping
   (b) Data Augmentation
   (c) Rescale
   (d) Resize
   (e) Encoding
3. Selection of the pre-trained models to be experimented on for feature extraction
4. Feature extraction using the chosen pre-trained models
5. Train-test split of the dataset
6. Development of a customized CNN architecture for classification
7. Training of the CNN classifier using the extracted features
8. Classification of SJS, EM and BP using the customized CNN classifier and finding the accuracies that the model provides when it is trained on the features extracted using the different pre-trained models
9. Interpretation of the prediction of the model using LIME
10. Comparative analysis of the performance of the CNN classifier using different pre-trained models as feature extractors on our new hybrid dataset
11. Classification of the diseases using SVM, Logistic Regression and Random forest
12. Analysis of how well our custom CNN architecture works in comparison with the machine learning models on our hybrid dataset for classification

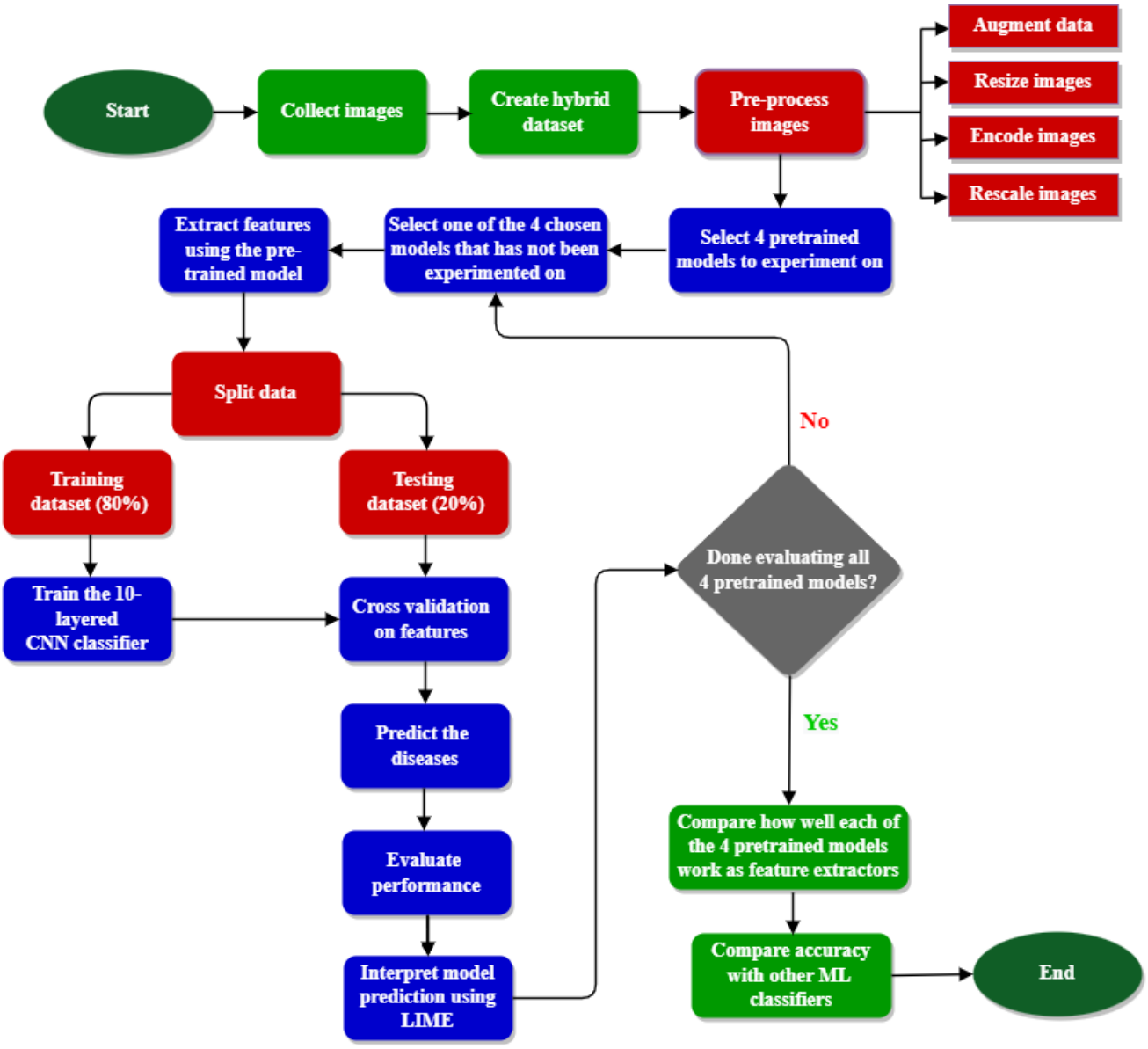

Figure 4.1: The flow chart of the research work

4.2 Dataset

In this experiment, the models have been trained on a combined dataset consisting of colored images which are collected from seven different sources (i.e. Dermnet NZ, DermIS, Dermatology Atlas, Global Skin Atlas, SJS Awareness UK, Dermnet and University of IOWA Health Care). This combined dataset contains a total of 657 images of skin lesions. Among them, 209 images belong to Stevens-Johnson syndrome (SJS), 262 images belong to Erythema Multiforme (EM) and the rest of the 186 images belong to Bullous Pemphigoid (BP).

4.2.1 Dataset Collection

Initially, from the various different sources, 646 images of skin lesions were collected for the dataset. Among those images, 207 images belonged to Stevens-Johnson syndrome (SJS), 250 images belonged to Erythema Multiforme (EM) and the remaining 189 images belonged to Bullous Pemphigoid (BP).

Out of the 207 SJS images, 16 images have been collected from the site named 'SJS Awareness UK', 24 images have been collected from the site called 'Global Skin Atlas', 116 images have been collected from the site 'Dermnet NZ' and 51 images have been collected from the site 'DermIS'.

Out of the 250 EM images, 3 images have been collected from the dermatology department of University of IOWA Health Care, 92 images have been collected from the site named 'Global Skin Atlas', 28 images have been collected from the site 'Dermnet NZ', 20 images have been collected from the site 'DermIS', 52 images have been collected from the site 'Dermatology Atlas' and lastly, 55 images are collected from the site 'Dermnet'.

Out of the 189 images of BP, 5 images have been collected from the dermatology department of University of IOWA Health Care, 24 images have been collected from the site called 'Global Skin Atlas', 47 images have been collected from the site 'Dermnet NZ', 72 images have been collected from the site 'DermIS' and finally, 41 images have been collected from the site 'Dermatology Atlas'.

The table 4.1 demonstates the details of data collection from the seven sources. All of these collected images were of different sizes and all of them were in .jpg format. In the figure 4.2, the sample images of the three diseases are shown.

Figure 4.2: Sample of Dataset (a) SJS, (b) EM, (c) BP

Site Name	SJS	EM	BP	Total
Dermnet NZ	116	28	47	191
Derm IS	51	20	72	143
Global Skin Atlas	24	92	24	140
Dermatology Atlas	0	52	41	93
Dermnet	0	55	0	55
SJS Awareness UK	16	0	0	16
University of IOWA Health Care	0	3	5	8

Table 4.1: Number of images collected from seven sources

4.2.2 Data pre-processing

Background Removal and cropping

Most of the sites from which these images have been collected had watermarks on them. Thus, to prepare them for the experiment, those images had to be manually cropped to remove the watermarks. Next, the images have been cropped further to remove the background so that the images only contain the affected skin and the lesions on that affected skin. This has been done to ensure that the images are more focused on the skin lesions, making it easier for the feature extractors to detect the lesions. Moreover, some of the images have been cropped to create multiple focused images. The website and apps that have been used to crop these images and prepare the dataset are removebg, Adobe Photoshop and Windows Photo Editor.

After all these cropping, the final hybrid dataset had a total of 657 images. Out of these 657 images, 209 images belonged to SJS, 262 images belonged to EM and lastly, 186 images belonged to BP.

Data Augmentation

Data augmentation is a method which can be used to increase the amount of data from the existing dataset. With the help of data augmentation, we can improve the accuracy of machine learning and neural network models as large amounts of data are created by using this technique which aids the training process. There are various different kinds of techniques in data augmentation such as :

- Flipping
- Rotating
- Zooming and scaling
- Cropping
- Translating
- Adding Gaussian noise

In this experiment, initially there had been 209 images of SJS in our hybrid dataset before augmentation. However, after augmentation, it increased to 3421 images. Furthermore, there had been 262 images of EM before data augmentation but after augmentation the number of images increased to 4128. Similarly, using data augmentation, the number of images of BP had been increased from 186 to 3102. Therefore, after performing data augmentation, the total number of images of SJS, EM and BP summed up to be 10651. For this experiment, a rotation method has been used to rotate the images of SJS, EM and BP by 40 degrees. Translation methods have also been used in the images of SJS, BP and EM by setting both height_shift_range and width_shift_range to 0.2. The brightness range has also been set to [0.2, 0.1] for brightness augmentation in all the images. Following that, the zooming method has been applied by setting zoom_range to 0.2 in all the images.

Also, the flipping method has been used by setting horizontal_flip to "True". Lastly, the shear_range has been set to 0.2 and fill_mode to "reflect".

After data augmentation, the augmented images have been fed into the feature extractor in each iteration of the training process. This ensures that the model does not face the same image in every iteration. Thus, even though the batch size of the training process remains the same in every iteration, the images used in each iteration varies and the resulting model proves to have a better generalization capability. As a result of this enhanced generalization capability, the neural network will have the ability to deal with unseen testing data.

Resize

All the images that had been collected to create the hybrid dataset used in this experiment were of different sizes. However, pre-trained models have been used in this research work for feature extraction so the input images have been resized according to the recommended input size of these pre-trained models. For the VGG16 and ResNet50V2 models, all the images have been resized to 244×244 before inputting them into the respective models whereas for the InceptionV3 and InceptionResNetV2 models, all the images have been resized to 150×150 and 299×299 respectively before inputting them into the respective models. Another reason for resizing the images before inputting them was that processing high resolution images can be time consuming and complex.

Encoding

As there are 3 classes of images in our hybrid dataset, the image labels have been encoded using one hot encoding. This was done by setting the class mode as 'categorical' during feature extraction which returns a 2D numpy array of one hot encoded label.

Rescale

Before feature extraction, the image pixel values have also been rescaled between 0 to 1 to normalize the data. Usually the pixel values of images lie in between 0 to 255. This greater range of pixels is not suitable for deep learning as it leads to instability in the neural network and makes it difficult for the model to process such large values. Hence, for better results, the pixels must be rescaled before inputting them into the model. Thus, the pixels have been divided by 255 to rescale them.

4.3 Feature Extraction

In this experiment, VGG16, ResNet50V2, InceptionV3 and InceptionResNetV2 have been used to extract the features from the images. For feature extraction, instead of re-training the weights, the pre-trained imagenet weights of these models were used. The top layer of these models were excluded as these models have only been used for extracting deep features, not for classifying images. Each of these models perform a convolution operation where a filter with a specified stride slides over the

input image to recognize certain patterns and create a feature map. In our experiment, VGG16 has been used firstly to extract features from the images and those features have been used afterwards to train and validate the 10-layer customized CNN architecture. Later on, those features have also been used to check the training and testing accuracy of the machine learning algorithms (i.e. Random Forest, SVM and Logistic Regression). Initially, the input shape for the model has been set to (224,224,3) and a total of 14,714,688 parameters have been used by this model. Since the trainable parameters had been set to false, these parameters were non-trainable. For ResNet50V2, InceptionV3 and InceptionResNetV2, a similar process has been carried out. Here, the pre-trained imagenet weights of ResNet50V2, InceptionV3 and InceptionResNetV2 have been used respectively and the top layer has not been included. Initial input size of ResNet50V2, InceptionV3 and InceptionResNetV2 has been set to (224,224,3), (150,150,3) and (299,299,3) respectively. Only the number of non-trainable parameters differ among them. The total parameters used in ResNet50V2 was 23,564,800. The total parameters used in InceptionV3 was 21,802,784 and the total parameters used in InceptionResNetV2 was 54,336,736. Each of the four pre-trained models have extracted a total of 10,651 feature vecotrs from all the images. These four models have been used to extract features in this experiment in order to make a comparison among them to see which feature extractor works best on this newly created hybrid image dataset.

4.4 Train-Test Split

After extracting the features from the pre-trained models, the hybrid dataset has been divided into a 80:20 ratio where 80% of the data has been included in the training set and 20% of the data in the validation set.

Chapter 5

Proposed 10-layered CNN Classifier

To achieve the best possible result for multiclass classification, we have proposed a 10-layered CNN classifier. The classical top layer of the pre-trained models has been replaced with this CNN classifier and then experimented on our hybrid dataset to analyze the performance of each of these four models on our dataset. The layers of this classifier are described below:

5.1 Convolution Layer

The proposed CNN classifier consists of two Convolution2D layers. These Two Convolutional layers are used so that when the extracted features from the pre-trained models are passed onto this classifier, it can learn more about the high level features from those input features. For the first Convolution2D layer, 32 filters of size 2×2 have been used and for the second Convolution2D layer, 64 filters of size 2×2 have been used. For both layers, ReLU has been used as the activation function.

5.2 Batch Normalization

In this proposed classifier, a Batch Normalization layer has been added after each convolution layer. Batch Normalization optimizes the performance of convolutional networks and decreases the generalization error. It is used as a regularization method which helps to block the overfitting problem of the model. It also standardizes the inputs in deep layers which helps to reduce the covariate shift problem.

Batch Normalization has the following steps :

1. It first normalizes the output from the activation function using the formula below:

$$z = \frac{x - m}{s} \tag{5.1}$$

   Here, x = output from activation function
   m = mean of the data
   s = standard deviation

2. It then multiplies the normalized output by an arbitrary parameter, g.

$$z * g \tag{5.2}$$

3. Lastly, it adds an arbitrary parameter, b, to the resulting product

$$(z * g) + b \tag{5.3}$$

Here, the calculation with the arbitrary products g and b sets a new standard deviation and mean to the data and the parameters m, s, g and b are all optimized during the training process.

5.3 Dropout

To reduce the overfitting problem, three dropout layers have been used. Dropout layer decreases the overfitting problem by randomly dropping the redundant and error-prone data. It randomly switches off the incoming and outgoing connections of some neurons and thus those neurons get dropped off so that the network can never learn those unnecessary data. This is how the learning of a sequential model gets enhanced. For this experiment, the value of the dropout rate has been set to 0.2. That means 20% of the neurons in the network will randomly get dropped out.

5.4 Global Average Pooling

Instead of the Flatten layer, we have used the Global Average Pooling layer. Global Average Pooling selects one feature per feature map by taking the mean of all the elements of a feature map and also it removes the spatial information. Thus, it creates a 1D feature vector. This feature vector is easier to interpret and that is why we have used Global Average Pooling in the classification layer. Moreover, as we do not need to optimize any parameter in Global Average Pooling, it prevents overfitting problems.

5.5 Dense Layer

Two dense layers have been used in our CNN architecture where the first dense layer has 128 neurons and the second dense layer has 3 neurons. Each neuron of these dense layers takes input from the previous layer and passes the output to the next layer. These 2 dense layers of the classifier model are mainly used to classify the images into their respective diseases.

5.6 Activation Function

In sequential neural network models, activation functions are required to transform the inputs of the neurons into the output. In our classifier model, two types of activation functions have been used which are Rectified Linear Unit or ReLU and Softmax. The details of these two activation functions are given below:

5.6.1 ReLU

The most common non-linear activation function ReLU is mostly applied to all the hidden layers of a sequential model. It helps to decrease the vanishing gradient problem because its derivative is one for input values greater than zero. As multiplication of one by itself several times still provides one, this basically helps with the vanishing gradient problem. Also, it allows the models to learn faster and helps CNN models to perform better.

5.6.2 Softmax

Softmax function is usually used with the dense layers. This function generates the probabilities for the target classes. By maximizing the logit values of the target classes, it allows the model to assign those logit values to those certain target classes. This function also has normalization properties. For an individual class x_i, the function can be represented as,

$$\sigma\left(x_i\right) = \frac{e^{x_i}}{\sum_j^c e^{x_i}} \tag{5.4}$$

Classifier architecture summary

Based on the pre-trained models, the input shape, output shape and the number of parameters of the CNN classifier changes. The summary of the architecture of this CNN classifier when it is used as the top layer of InceptionResnetV2 is shown below:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 7, 7, 32)          196640

 batch_normalization_2 (Batc  (None, 7, 7, 32)         128
 hNormalization)

 dropout_3 (Dropout)         (None, 7, 7, 32)          0

 conv2d_3 (Conv2D)           (None, 6, 6, 64)          8256

 batch_normalization_3 (Batc  (None, 6, 6, 64)         256
 hNormalization)

 dropout_4 (Dropout)         (None, 6, 6, 64)          0

 global_average_pooling2d_1   (None, 64)               0
 (GlobalAveragePooling2D)

 dense_2 (Dense)             (None, 128)               8320

 dropout_5 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 3)                 387

=================================================================
Total params: 213,987
Trainable params: 213,795
Non-trainable params: 192
_________________________________________________________________
```

Figure 5.1: Summary of the CNN classifier (used as the top layer of InceptionResnetV2)

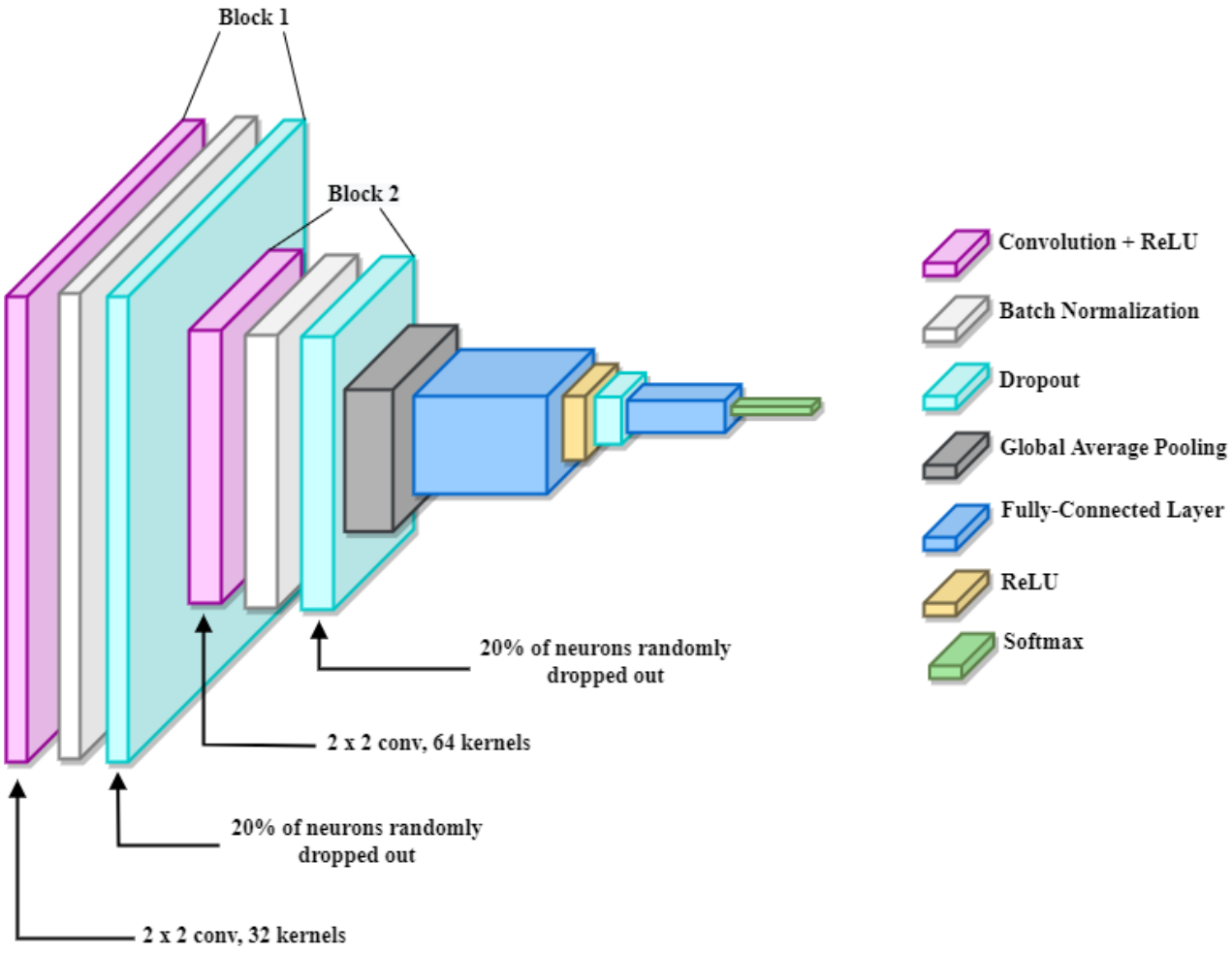

Figure 5.2: Visual representation of the layers in the 10-layer custom CNN classifier

Chapter 6

Implementation and result analysis

6.1 Model Implementation

The implementation of the four models for the detection of SJS, BP and EM is described in this section. The code for developing these models have been written and executed in the Jupyter Notebook. The implementation of these models consisted of several phases like preparing and processing our input data images, extracting features from those processed images, classification of those images using the extracted features, evaluating the models and doing a comparative analysis to find the best performing model.

6.1.1 Overview of the workflow

The combined dataset that we have created originally had 657 images which was not sufficient to train deep neural network models and gain a high accuracy. Thus, data augmentation has been applied on EM, BP and SJS images separately to acquire a total of 10,651 images. Before applying the data augmentation techniques, most of the images have been cropped at certain portions. Moreover, unwanted background have been removed from a few images. This cropping and background removal has been done to focus more on the lesions on the skin so that the model does not learn any redundant information. Then, all the augmented images have been resized according to the required input shape of the pre-trained model. After that, other necessary pre-processing (i.e. label encoding, rescaling etc.) have been applied on the images and the features have been extracted from these input images by using the pre-trained ImageNet weights of a transfer learning model. We had started our experiment by using the VGG16 model as our base model for transfer learning. Then those extracted features have been split into test and validation sets with a ratio of 80:20.

To classify the images, we have developed a 10-layered CNN architecture consisting of Convolution, Batch Normalization, Dropout, Global Average Pooling and Dense layers. The first block of this CNN classifier consists of a convolution layer followed by a batch normalization layer and lastly, a dropout layer. For the convolutional layer, 32 filters of size 2×2 have been used along with a ReLU activation function.

Then to optimize the performance, a batch normalization layer has been added on top of the convolution2D layer. We have also introduced a dropout layer with a value of 0.2 to reduce overfitting. The second block in this classifier architecture has been kept the same as the first block except the convolutional layer that now contains 64 filters here. Next, instead of using a flatten layer, a global average pooling layer has been used. This global average pooling layer is more robust than a typical flatten layer and can avoid overfitting. This suggests why the accuracy rate of this model had increased when we replaced the flatten layer with this global average pooling layer. Furthermore, we have used two fully connected layers for getting the desired output. The first fully-connected layer contains 128 neurons whereas the second fully-connected layer which is the last layer of our model contains 3 neurons as our dataset contains 3 classes of diseases. ReLU activation function and a dropout layer was also added between these two dense layers. Finally, a Softmax function has been used here as the activation function. Subsequently, we have also used Random Forest, Logistic Regression and SVM, which are machine learning algorithms to fit the extracted features and make predictions from the input images. The accuracy that has been received after applying these machine learning algorithms were then compared with the accuracy of our CNN classifier to evaluate how well a neural network classification model works with respect to these well-known machine learning classification models.

Lastly, all these steps have been repeated for three more pre-trained models namely InceptionV3, ResNet50V2 and InceptionResnetV2. Then a comparative analysis has been done on the performance of these four models to find out the model with the highest accuracy.

Each of these four models were trained for 300 epochs. During the training process of the models, Adam's optimizer has been used because this optimizer can perform very well by reaching a faster convergence.

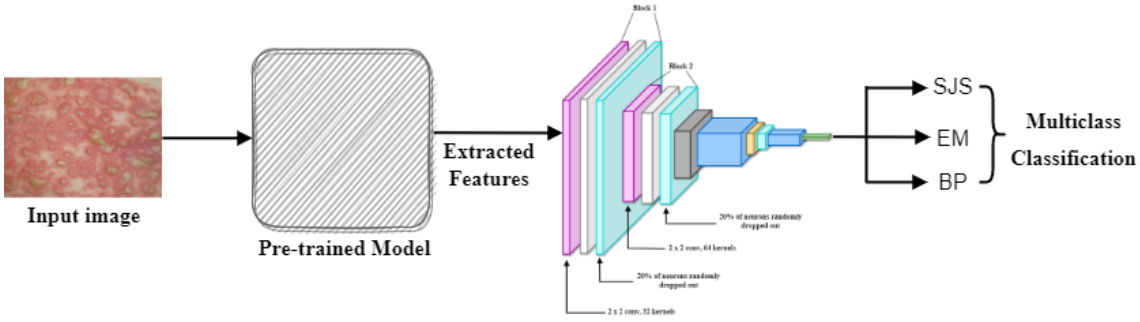

Figure 6.1: Model implementation

6.2 Performance Evaluation

The performance of the four DNN models has been evaluated using five performance metrics which are accuracy, sensitivity, specificity, precision and F1 score.

Accuracy: Accuracy is the metric that measures the number of correct predictions of the testing data set.

$$\text{Accuracy } = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{6.1}$$

Sensitivity: It measures how many positive data are predicted correctly out of the total actual positive values. It is also known as Recall. It is used in cases where false negative (FN) value needs to be reduced.

$$\text{Sensitivity } = \frac{TP}{TP + FN} \tag{6.2}$$

Specificity: This metric is used to determine how many negative values are correctly predicted out of the total actual negative values.

$$\text{Specificity } = \frac{TN}{FP + TN} \tag{6.3}$$

Precision: Precision is used to measure how many results are actually positive out of the total positive predictions. It is used in cases where false positive (FP) value needs to be reduced.

$$\text{Precision } = \frac{TP}{TP + FP} \tag{6.4}$$

F1 score: F Beta is the combination of precision and recall. This metric is used when both false positive and false negative value has a huge impact on the performance of the dataset. If both false positive and false negative values are equally important then β can be 1. When $\beta = 1$, it can be called an F1 score.

$$\text{Flscore } = 2\frac{\text{Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{6.5}$$

In the above equations,
TP = True positive which means the correctly labeled positive instances
FP = False positive which means incorrectly labeled positive instances
TN = True negative which means correctly labeled negative instances
FN = False negative which means incorrectly labeled negative instances

Confusion Matrix: A confusion matrix is a m×m matrix that is used for the evaluation of a classification model's performance, where m represents the number of target classes. It summarizes the predictions made by the classification model and compares the actual target values with those predicted by the classifier. It acts as a useful evaluation tool as it provides direct comparisons of values such as True Negative (TN), True Positive (TP), False Negative (FN) and False Positive (FP). Thus, it aids in calculating important performance metrics such as specificity, accuracy, sensitivity, precision and F1 score as all these metrics require TP, TN, FN and FP values for their calculation.

6.3 Experimental Results

6.3.1 Performance analysis of the different pre-trained models using CNN classifier

After extracting the features using the transfer learned base models, those features have been divided into train and validation sets, where 80% of the extracted features has been kept in the training set and 20% of them has been kept in the validation set. Then those training data have been fed into the CNN classifier to train it and along with that the performance of the model has been evaluated using the validation data. While training each of the four models, we have used the same batch size and same number of epochs to better comprehend the performance. The validation accuracy of each of the four models for different numbers of epochs have been demonstrated in the table given below.

Base Models	**Accuracy (%)**				
	10 epochs	**50 epochs**	**100 epochs**	**200 epochs**	**300 epochs**
VGG16	76.21	84.05	80.48	93.24	91.69
InceptionV3	84.34	87.35	88.58	89.45	87.16
Resnet50V2	96.48	96.76	97.37	97.75	97.04
InceptionResnetV2	93.20	97.89	97.89	98.12	98.50

Table 6.1: Validation accuracy at different number of epochs

From this table, we can see, as the number of epochs increases, the accuracy also increases. However, for some epochs such as at the 300th epoch of VGG16, it can be observed that the accuracy decreases. The accuracy function of training and validation for all four models has been visualized in the figure 6.2.

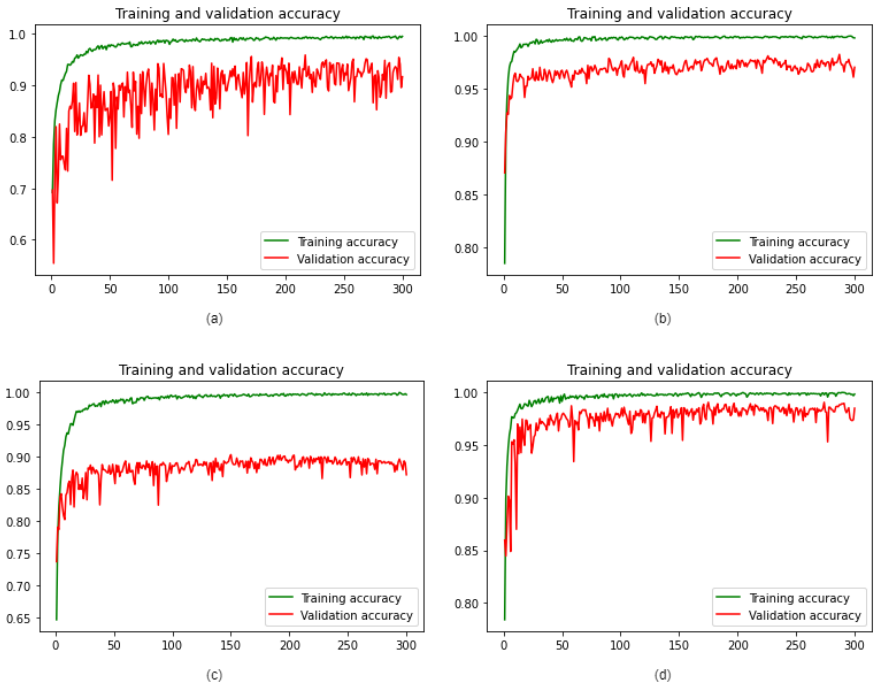

Figure 6.2: History of training and validation accuracy of (a) VGG16, (b)ResNet50V2, (c) InceptionV3 and (d) InceptionResNetV2

From the graphs in figure 6.2, we can observe that with an increasing number of epochs, the accuracy rate also increases. However, due to overfitting the validation accuracy is fluctuating and there is a gap between the training and validation accuracy. Here, we can see that this gap is lowest in case of InceptionResNetV2 model. Moreover, with InceptionResNetV2 we have obtained the highest validation accuracy of 99.06% at epoch 175, whereas ResNet50V2 has obtained the second highest accuracy of 98.26% at epoch 287, VGG16 has obtained 95.92% accuracy at epoch 217 and InceptionV3 has obtained 90.27% accuracy at epoch 150.

The loss function of training and validation for all four models has been visualized in the figure 6.3.

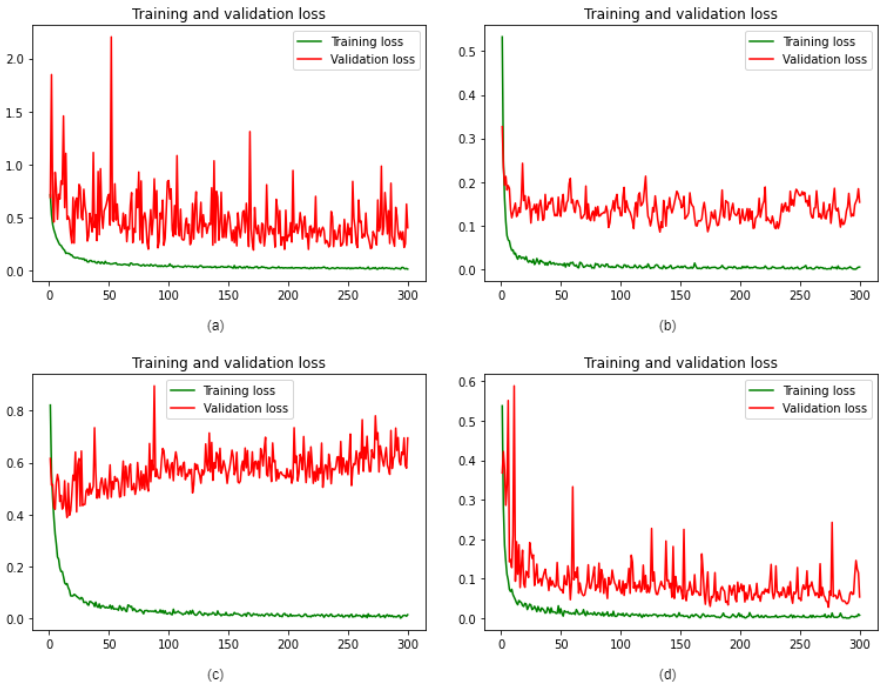

Figure 6.3: History of training and validation loss of (a) VGG16, (b)ResNet50V2, (c) InceptionV3 and (d) InceptionResNetV2

From the graphs of the figure 6.3, we can see that as the number of epochs increases, both the training and validation loss decreases in the case of VGG16, ResNet50V2 and InceptionResNetV2. However, we can observe from the history of training and validation loss graph of InceptionV3 that while the training loss of InceptionV3 decreases with increasing epoch, the validation loss tends to increase with increasing number of epochs and a large difference can be observed between the validation and training loss. This shows that the model which uses InceptionV3 suffers from overfitting. On the other hand, InceptionResnetV2 has been able to achieve the lowest loss rate compared to the other models. Therefore, it can be concluded that the InceptionResnetV2 model is performing better in terms of the other three models.

The following table and bar chart show the summary of the implemented pre-trained models with our 10-layered CNN architecture as the classifier.

Feature Extractors	Accuracy (%)	Precision (%)	Sensitivity or Recall (%)	Specificity (%)	F1 Score (%)
VGG16	95.92	96.13	95.74	97.88	95.93
ResNet50V2	98.26	98.23	98.26	99.12	98.24
InceptionV3	90.27	90.44	90.14	95.05	90.29
InceptionResNetV2	**99.06**	**99.09**	**99.05**	**99.52**	**99.07**

Table 6.2: Summary of the performance of the pre-trained models with the CNN classifier

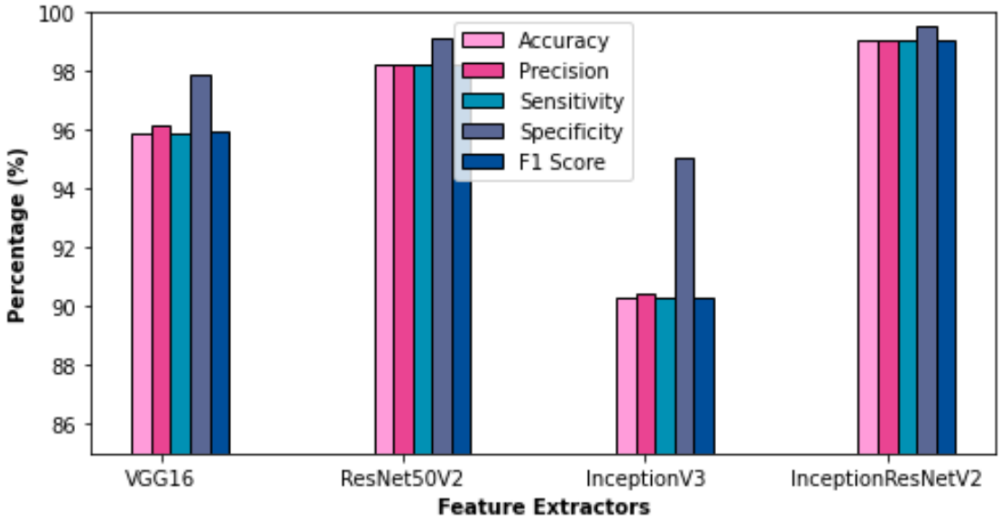

Figure 6.4: Summary of the performance of the four pre-trained models with the CNN classifier

If we analyze our experimental result, we can observe that the model that uses InceptionResNetV2 as the feature extractor proved to have the highest accuracy of 99.06 % on our own hybrid dataset. We can further notice that the ResNet50V2 architecture had a very close accuracy (98.26%) with the InceptionResnet50V2 model. However, if we compare the architecture of these two models, we can say that ResNet50V2 is less heavier and tends to train faster. We have also discovered that the InceptionResNetV2 model provides the highest precision of 99.09%, sensitivity of 99.05%, specificity of 99.52% and F1 score of 99.07% among all the other models with the ResNet50V2 model trailing extremely close behind it in terms of all these performance metrics. However, as mentioned before, the gap between the training and validation loss of InceptionResNetV2 is relatively smaller than that of ResNet50V2. Thus, overfitting is more under control in the case of InceptionResNetV2 than ResNet50V2. So, the decision regarding which model between the two should be used in the classification of SJS, EM and BP depends on whether we desire a model which has an overall highest performance or a model that performs very well and is also lighter and faster to train. On the other hand, upon further analysis, we can notice that the InceptionV3 model works the poorest among the 4 models experimented on our dataset by having the lowest performance in terms of all the evaluated performance metrics. Thus, even though InceptionV3 has been expected to work better on our dataset due to having more hidden layers and inception modules consisting of different sizes of kernels in the same level that aids it to learn more complex features, it works less well than VGG16 when used in our

particular dataset.

The confusion matrices given below helps to visualize how many predictions made by our custom CNN classifier are correct and how many of the predictions made are incorrect by using the extracted features from VGG16, InceptionV3, ResNet50V2 and InceptionResNetV2 respectively.

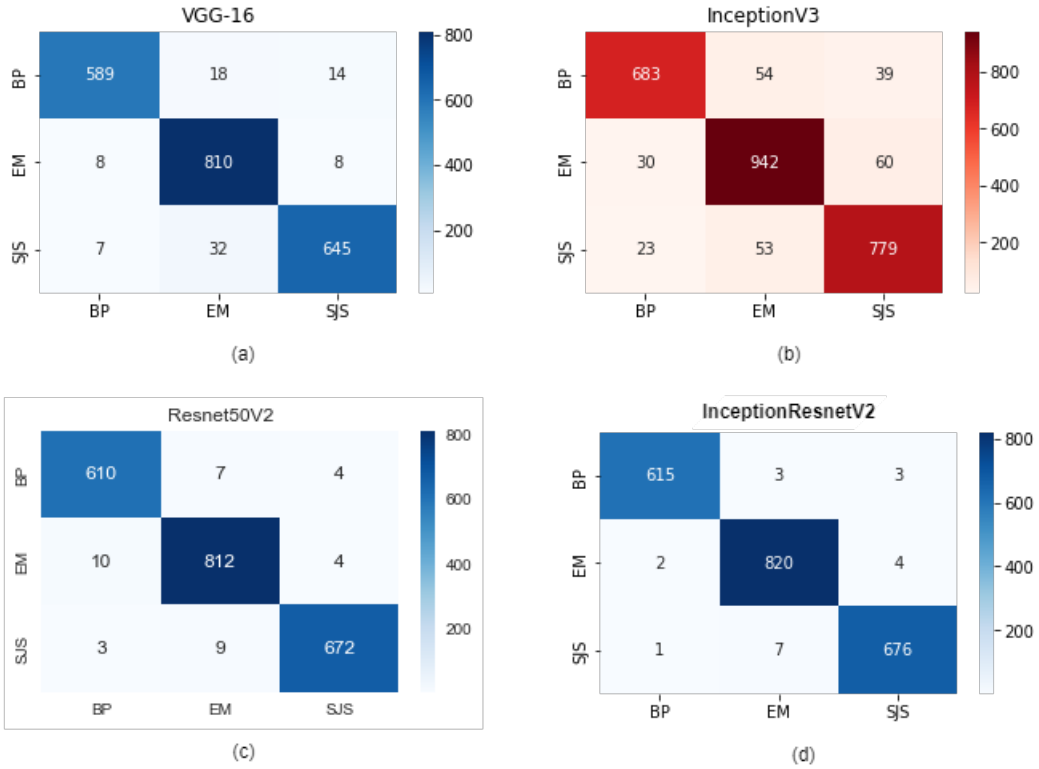

Figure 6.5: Confusion matrices of prediction results using (a) VGG16, (b) InceptionV3, (c) ResNet50V2, and (d) InceptionResNetV2 with our CNN classifier

The following figures demonstrate the fraction of incorrect predictions made by the VGG16, InceptionV3, ResNet50V2 and InceptionV3 models.

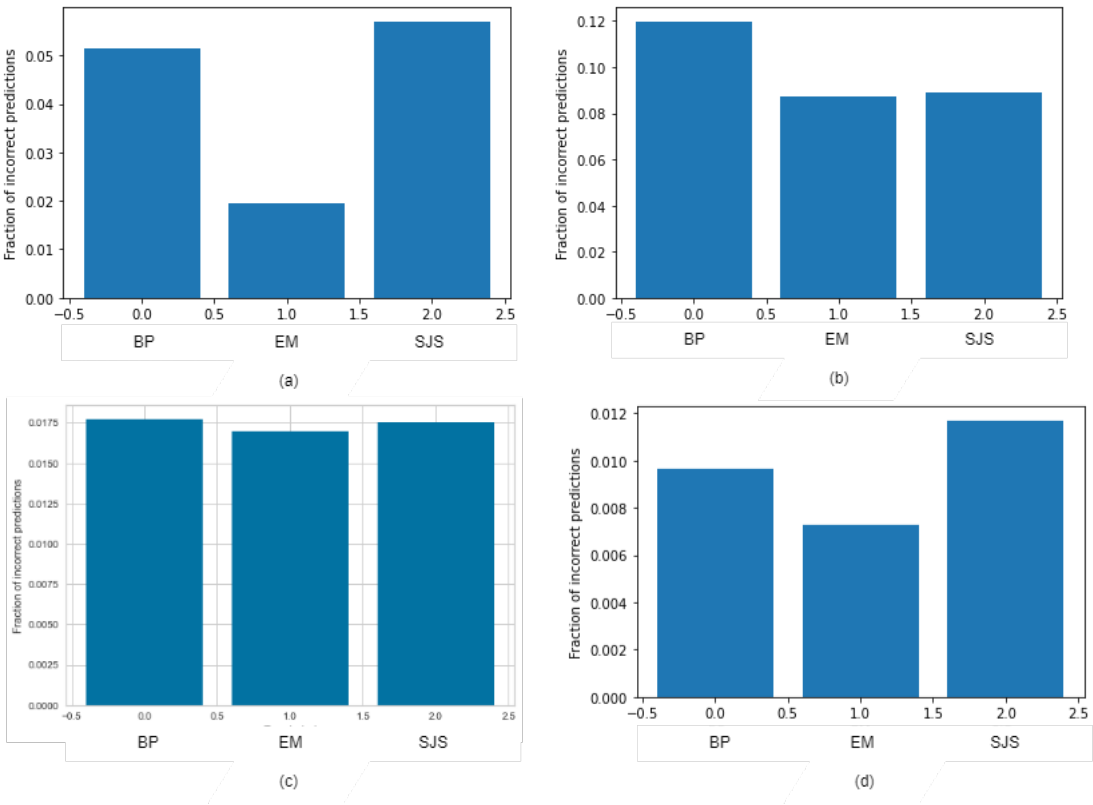

Figure 6.6: Fraction of incorrect predictions made by (a) VGG16, (b) InceptionV3, (c) ResNet50V2, and (d) InceptionResNetV2

The multiclass ROC (Receiver Operating Characteristics) curves of the VGG16, InceptionV3, ResNet50V2 and InceptionResNetV2 models are shown below. These ROC curves are probability curves and the area under these curves (AUC) show the separability of a classification model. A model with an AUC value close to 1 represents that it has a good degree of separability whereas a model with an AUC value close to 0 represents that it has a poor degree of separability between classes.

The figures below show that both the ResNet50V2 and InceptionResNetV2 models have AUC values for SJS, BP and EM that are close to one. This proves that both these models have a very good measure of separability among the three different classes of skin diseases.

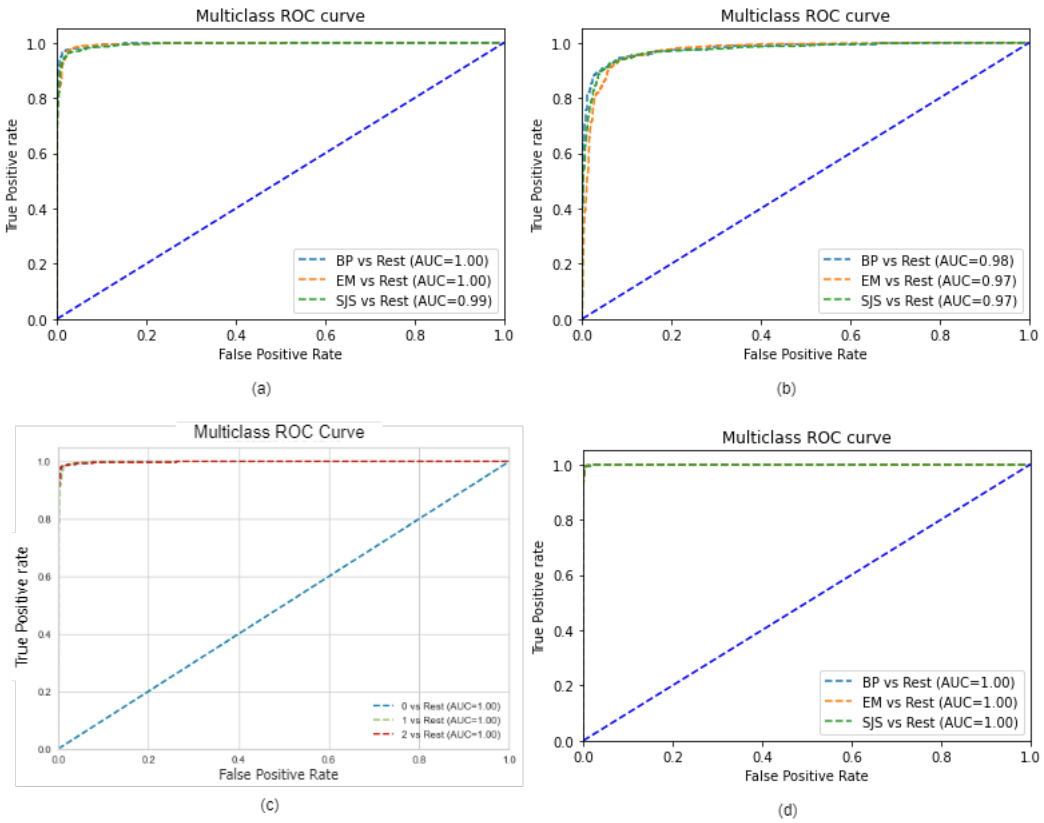

Figure 6.7: Multiclass ROC curves of (a) VGG16, (b) InceptionV3, (c) ResNet50V2, and (d) InceptionResNetV2

6.3.2 Performance analysis of the four pre-trained feature extractors with the different classifiers

In our experiment, apart from trying to discern and analyze the performance of VGG16, Inceptionv3, ResNet50V2, and InceptionResNetV2 as feature extractors on our newly created dataset, we have further extended our research to analyze the performance of three famous machine learning algorithms (SVM, Logistic Regression and Random Forest) in classifying SJS, EM and BP by using the features extracted from the four different feature extractors separately. The purpose of also experimenting on these machine learning algorithms was to develop a comparative analysis to see whether any of these machine learning algorithms perform better than our custom CNN classifier in classifying the three diseases in our hybrid dataset. After analyzing the result, we have found that the validation accuracy of each of the four pre-trained feature extractors when used with the 10-layer CNN head model on top for classification are significantly higher than the validation accuracy that we have got when we used the three machine learning classifiers. Furthermore, we have found that among all the implemented classifiers, CNN works best by providing a validation accuracy of 99.06% when used with InceptionResNetV2 as the feature extractor. From this analysis, it can be implied that neural network classifiers can perform significantly better than the machine learning classifiers for image classification, especially on our dataset.

The validation accuracy of all the classifiers for each of the four feature extractors

are displayed in the table and bar chart below:

Feature Extractors	Classifiers			
	CNN (%)	**Random Forest** (%)	**Logistic Regression** (%)	**SVM** (%)
VGG16	95.92	75.46	82.50	74.38
InceptionV3	90.27	76.08	84.27	85.73
Resnet50V2	98.26	79.49	92.02	90.29
InceptionResnetV2	99.06	77.29	91.13	90.43

Table 6.3: Comparison of the validation accuracy of the feature extractors using different classifiers

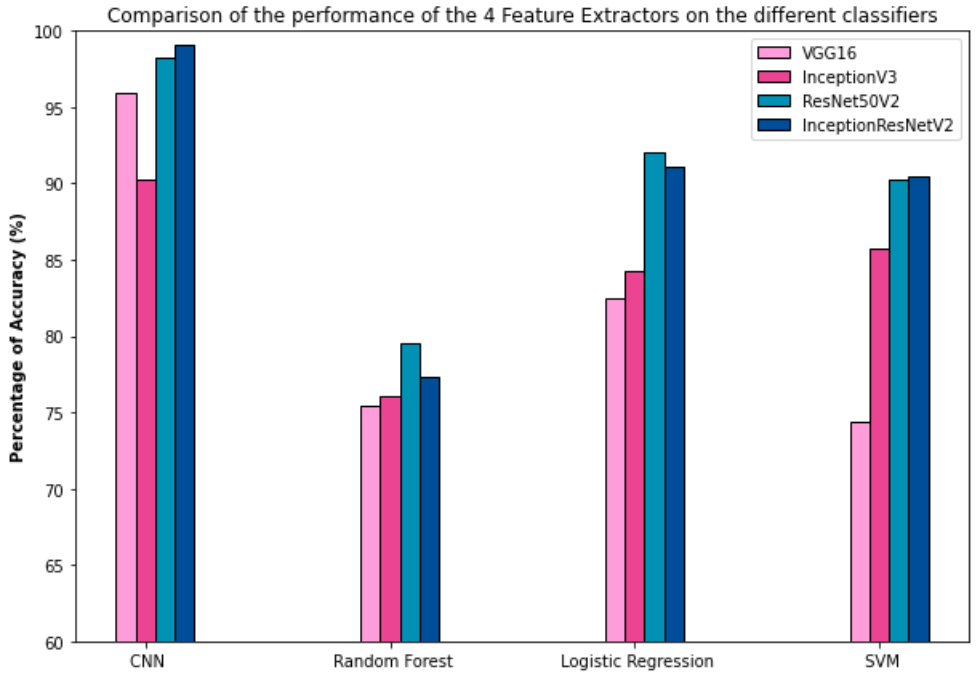

Figure 6.8: Summary of the validation accuracy of the four pre-trained models using different classifiers

6.4 Analysis of the predicted results using LIME

Over the years, the deep neural network (DNN) models have shown remarkable results in multiclass classification based problems. These models have a very good problem solving capability. Thus, DNN based AI systems are becoming very popular in medical sectors for the classification and prediction of diseases. However, the decision making process of these networks is like a black box. The internal processing techniques of these networks are very ambiguous. "How are these networks reaching the conclusion or by looking at what they are making this decision?" These questions remain unanswered because of the ambiguity of these models. As a result, physicians find it very difficult to completely rely on the decision of these DNN based diagnostic

assistance systems. Hence, to ensure the transparency of the decision making process of these models, the researchers have come up with the concept of Explainable Artificial Intelligence (XAI).

In dermatology, ensuring the transparency of DNN has become a legal requirement to develop an AI software for diagnosing skin diseases. Researchers have developed various XAI methods to explain the decision making process of the deep neural network models. These methods usually mark down the important regions in the images that are used as features for the prediction or output an image that is similar to the predicted image [20]. Thus, by looking at these output images, physicians or users of the dermatological software can get an idea about the prediction results.

Since, the goal of this research work is to aid the dermatology sector by developing a DNN based skin disease detection system, it has become essential for us to interpret the prediction results so that the physicians do not hesitate to rely on the decision of the system. To explain the prediction results, we have used a LIME based explainable AI technique. LIME is one of the easiest XAI techniques. It is compatible with various types of classifiers. It creates a wide range of artificial data by manipulating the input data which contains only a specific instance of the original data. Then it classifies this new set of data and calculates the weight of these artificial data points. Next, it fits these weighted artificial data points into the classifier and explains the most important features [19]. With the help of LIME, based on what the classifier is reaching to a decision can be visualized.

6.4.1 Explanation of the VGG16 prediction results using LIME

From the images given below, the regions that determine the predictions made by VGG16 can be visualized.

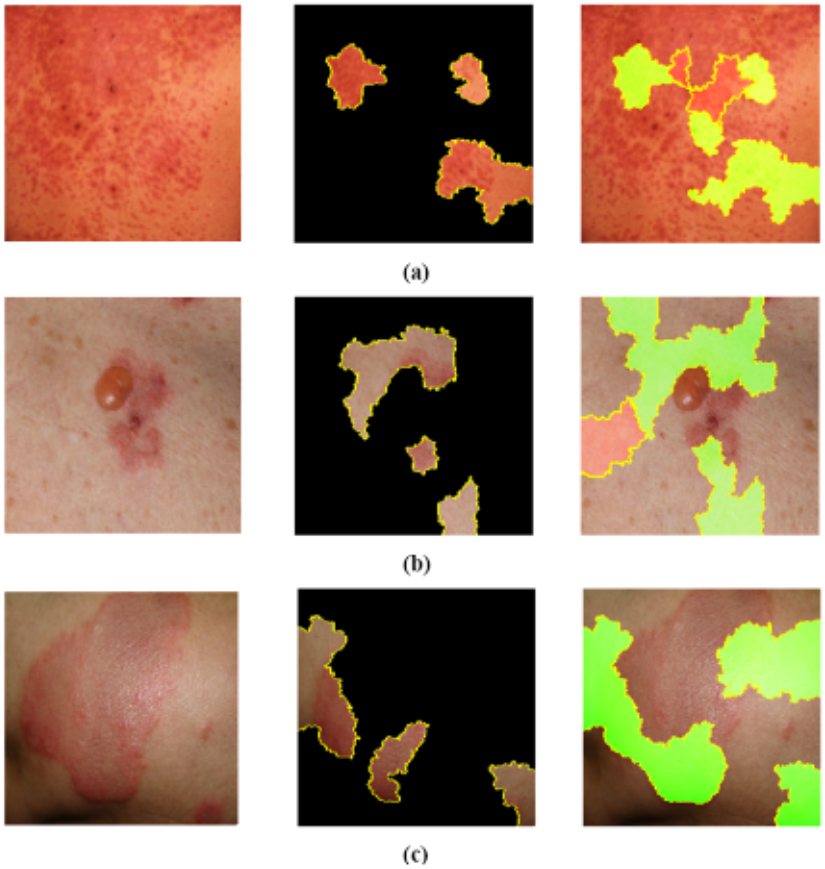

Figure 6.9: LIME interpretation of the prediction made by VGG16 when an image of (a) SJS, (b) BP and (c) EM is provided as input

The leftmost image of figure 6.9 (a) is an image belonging to SJS. The image in the middle of figure 6.9 (a) shows the regions of super-pixels that the VGG16 model uses to classify the image and the rightmost image highlights and shows the regions of super-pixels (the parts in green) that increase the probability of this image belonging to the class predicted by the VGG16 model. By analyzing this interpretation made by LIME, we can interpret VGG16's prediction as we can now see that it is looking at the correct regions (regions having lesions on them) while predicting that this image belongs to the SJS class. Similarly, we can see from figure 6.9 (c) that VGG16 is looking at the correct region while predicting the image belonging to the EM class as EM.

On the other hand, from the second image (figure 6.9 (b)) we can see that even though VGG16 is correctly classifying this image of BP as BP, the region it is looking at to make this prediction is not correct overall. It is looking at regions that contain normal skin and it is also missing out a crucial region of lesion while making this decision.

6.4.2 Explanation of the InceptionV3 prediction results using LIME

Similarly, Inception V3's prediction results can also be visualized using LIME. From the pictures given below, we can see which part of the skin is selected to make the decisions.

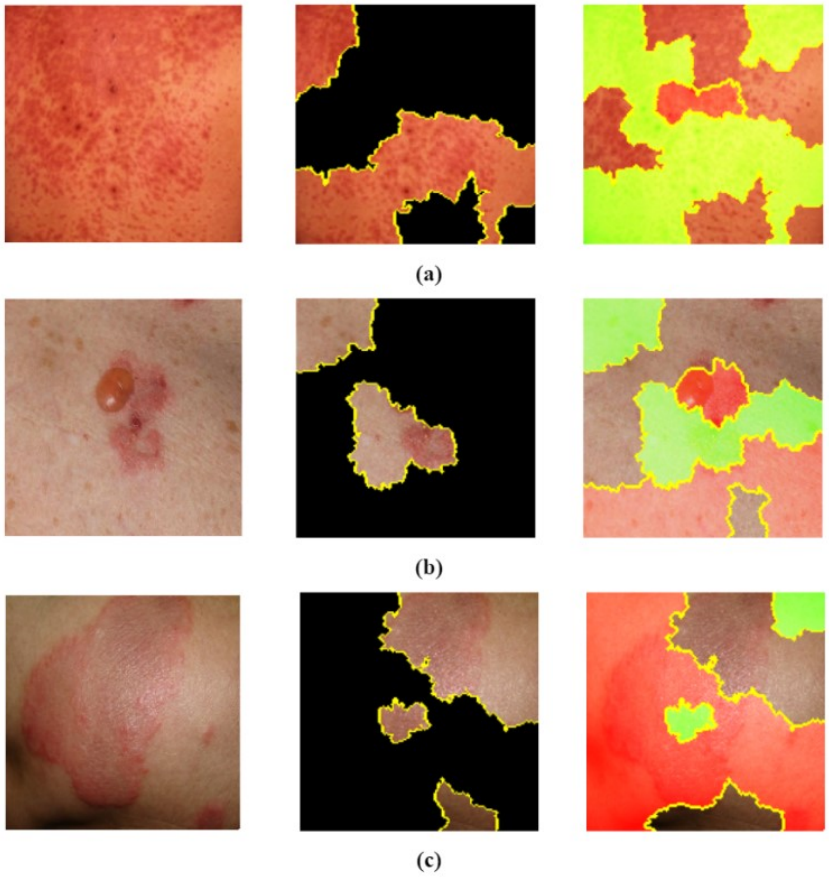

Figure 6.10: LIME interpretation of the prediction made by InceptionV3 when an image of (a) SJS, (b) BP and (c) EM is provided as input

In the case of InceptionV3, we can see from figure 6.10 (a) and (c), that the InceptionV3 model is making correct predictions of SJS and EM respectively by looking at regions containing lesions. However, the rightmost image of figure 6.10 (c) shows that a significant portion of the lesion (red colored region) in that image is decreasing the probability of this model in correctly classifying this image as EM. Moreover, figure 6.10 (b) shows that InceptionV3 is making a correct prediction of the BP image by overlooking a vital region of lesion and looking at regions that contain normal skin as well.

6.4.3 Explanation of the ResNet50V2 prediction results using LIME

From the explanation of the figures given below, we can visualize how ResNet50V2 is classifying each of the diseases.

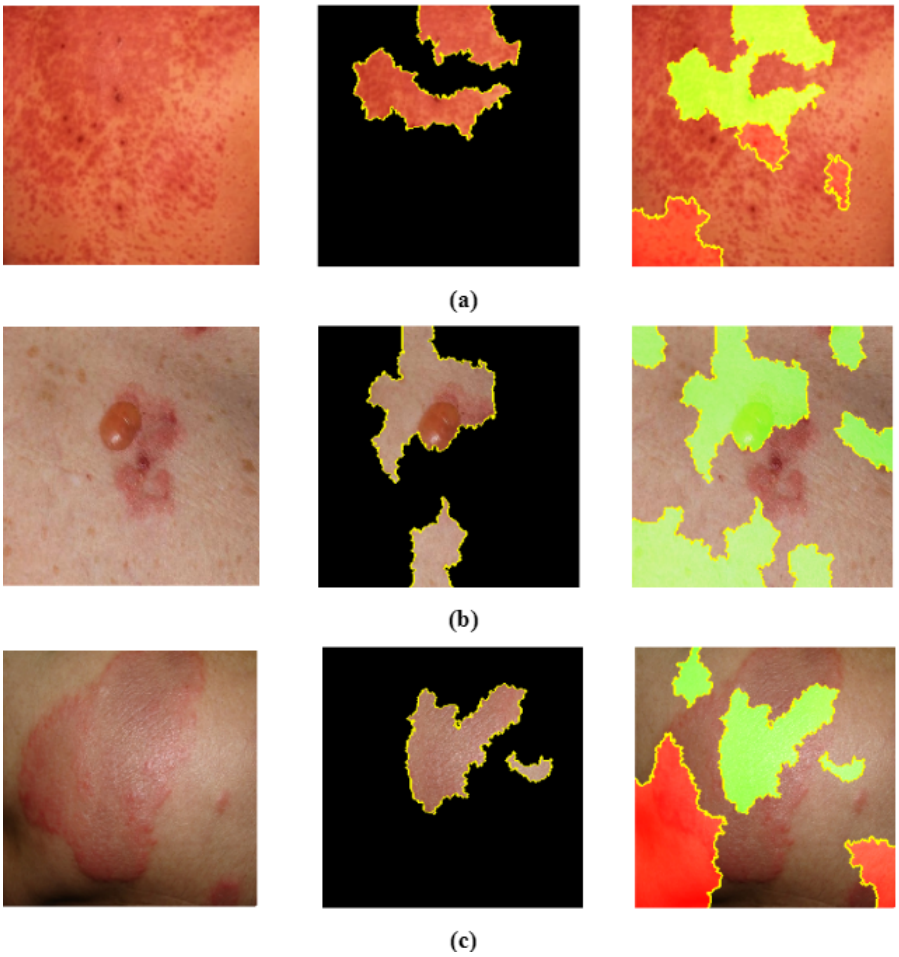

Figure 6.11: LIME interpretation of the prediction made by ResNet50V2 when an image of (a) SJS, (b) BP and (c) EM is provided as input

By analyzing the figures above, it can be observed that the ResNet50V2 model is making correct predictions of the three images belonging to the SJS, BP and EM class respectively by looking at regions of super-pixels that contain the most salient lesions. Here, the rightmost images also highlight the regions in red that decrease the probability of the images belonging to the class predicted by them.

6.4.4 Explanation of the InceptionResNetV2 prediction results using LIME

From the explanation of the figures given below, we can visualize how InceptionResNetV2 is classifying each of the diseases.

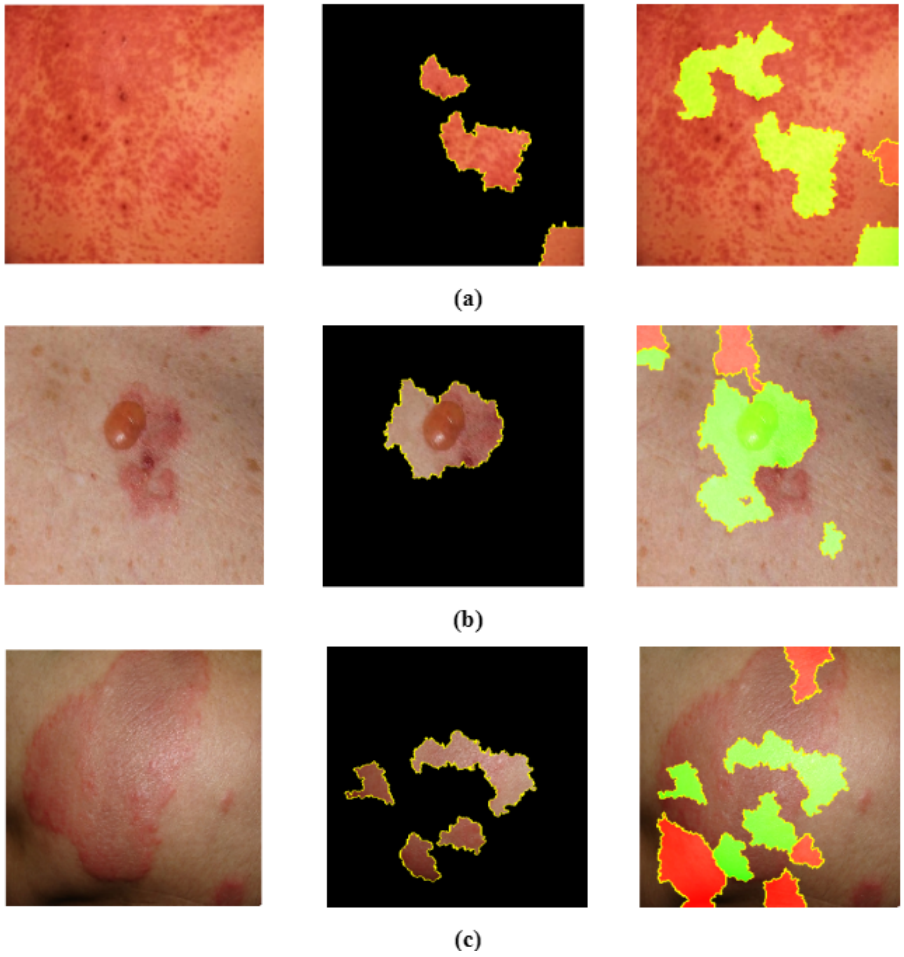

Figure 6.12: LIME interpretation of the prediction made by InceptionResNetV2 when an image of (a) SJS, (b) BP and (c) EM is provided as input

From the above visual interpretations, we can visualize that the InceptionResNetV2 model is correctly predicting the images in figure 6.12 (a), figure 6.12 (b) and figure 6.12 (c) as SJS, BP and EM respectively. Figure 6.12 (a), (b) and (c) show that this model is making these correct predictions by looking at areas of lesions. However, it can be seen in the rightmost image of figure 6.12 (c) that there are certain areas of lesions (the red regions) that are negatively affecting the prediction of this image as EM.

While interpreting the predictions made by these four models, we have further observed that both the VGG16 and InceptionV3 models made correct predictions of the image below by overlooking an essential area of lesion. However, both ResNet50V2 and InceptionResNetV2 made the right predictions by correctly identifying the lesion area. Thus, this observation that both ResNet50V2 and InceptionResNetV2

can detect the salient lesions in the image better than VGG16 and InceptionV3 gives a potential justification for the higher accuracies attained by them.

Figure 6.13: An image of Bullous Pemphigoid

6.5 Accuracy Comparison on Related Works

In this section, the comparison of the performance among our models and other related works have been shown. The dataset used in this research work is a newly created hybrid dataset consisting of the colored images of three deadly skin diseases namely Stevens-Johnson Syndrome, Erythema Multiforme, Bullous Pemphigoid. Moreover, no other research work has been done on these three diseases altogether. However, similar kinds of methodologies that we have used in this research work have been used on different datasets of skin diseases in previous research works. For example, in this paper [21], the authors have used the transfer learning methodology. To elaborate, they have used MobileNetV2, InceptionResNetV2, ResNet50V2 and InceptionV3 which contain pre-trained Imagenet weights and biases. Similar to us, they have also customized the top layers of these models. However, their customized head model is different from ours. Using this customized head, the best accuracy that they have achieved is 85% with MobileNetV2. On the other hand, from our research work it has been found that InceptionResnetV2 provides a significantly high accuracy of 99.06% on our custom dataset. In another research work [8], the authors have also applied transfer learning but the pre-trained models that they experimented on were ResNet50, InceptionV3, InceptionResNetV2, Xception and DenseNet-121. There are some more research works where similar types of methods have been seen to be used by the authors. In table 6.4, a summary of the comparison among these similar kinds of works have been demonstrated.

S. No.	Topic	Author	Dataset used	Method(s)	Performance
1	Machine Learning Algorithms based Skin Disease Detection [6]	Bhadula, S., Sharma, S., Juyal, P., & Kulshrestha, C. (2019)	A dataset consisting of 3000 sample images of lichen planus, acne and sjs ten	CNN	Accuracy = 96%
2	Melanoma image classification based on MobileNetV2 network [21]	Indraswari, R., Rokhana, R., & Herulambang, W. (2022)	ISIC-Archive repository, the ISBI 2016 dataset, the MED-Node dataset, and the PH2 Database	Pre-trained MobileNetV2 + customized head model	Accuracy = 85% Specificity = 85% Sensitivity = 85% Precision = 83%
				Pre-trained ResNet50V2 + customized head model	Accuracy = 84% Specificity = 81% Sensitivity = 87% Precision = 79%
				Pre-trained InceptionV3 + customized head model	Accuracy = 81% Specificity = 82% Sensitivity = 79% Precision = 78%
				Pre-trained InceptionResNetV2 + customized head model	Accuracy = 78% Specificity = 91% Sensitivity = 63% Precision = 85%
3	Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images [8]	Wu, Zhe and Zhao, Shuang and Peng, Yonghong and He, Xiaoyu and Zhao, Xinyu and Huang, Kai and Wu, Xian and Fan, Wei and Li, Fangfang and Chen, Mingliang and Li, Jie and Huang, Weihong and Chen, Xiang and Li, Yi (2019)	A hybrid dataset consisting of 4394 clinical images of the six facial skin diseases - actinic keratosis (AK), lupus erythematosus (LE), rosacea (ROS), squamous cell carcinoma (SCC), seborrheic keratosis (SK) and basal cell carcinoma (BCC)	Pre-trained ResNet50	Avg. recall = 63.4% Avg. precision = 62.9%
				Pre-trained InceptionV3	Avg. recall = 66.6% Avg. precision = 64.0%
				Pre-trained DenseNet-121	Avg. recall = 68.2% Avg. precision = 68.8%
				Pre-trained Xception	Avg. recall = 70.6% Avg. precision = 68.1%
				Pre-trained InceptionResNetV2	Avg. recall = 77.0% Avg. precision = 70.8%
4	Analyzing the Diagnostic Efficacy of Deep Vision Networks for Malignant Skin Lesion Recognition [18]	Pranav, M V and Koushik, C and A V, Shreyas Madhav and Ganapathy, S (2021)	HAM1000 and ISIC	VGG16 + InceptionResNetV2 + DenseNet201	Accuracy = 90% Precision = 89.5% Recall = 89%
5	Our research work		A newly created hybrid dataset consisting of 657 images of Stevens-Johnson Syndrome (SJS), Erythema Multiforme (EM), Bullous Pemphigoid (BP)	Pre-trained InceptionResNetV2 + customized head	Accuracy = 99.06% Specificity = 99.52% Sensitivity = 99.05% Precision = 99.09% F1 Score = 99.07%
				Pre-trained ResNet50V2 + customized head	Accuracy = 98.26% Specificity = 99.12% Sensitivity = 98.26% Precision = 98.23% F1 Score = 98.24%
				Pre-trained InceptionV3 + customized head	Accuracy = 90.27% Specificity = 95.05% Sensitivity = 90.14% Precision = 90.44% F1 Score = 90.29%
				Pre-trained VGG16 + customized head	Accuracy = 95.92% Specificity = 97.88% Sensitivity = 95.74% Precision = 96.13% F1 Score = 95.93%

Table 6.4: Comparison of performance with the related works

6.6 Web Application

After concluding our research work, we proceeded to build a web application using the InceptionResNetV2 model with the 10-layered top head to classify the three aforementioned diseases. This particular model was chosen as it provided the best performance among all the other attempted models. Figure 6.14 displays the user interface of our developed web application.

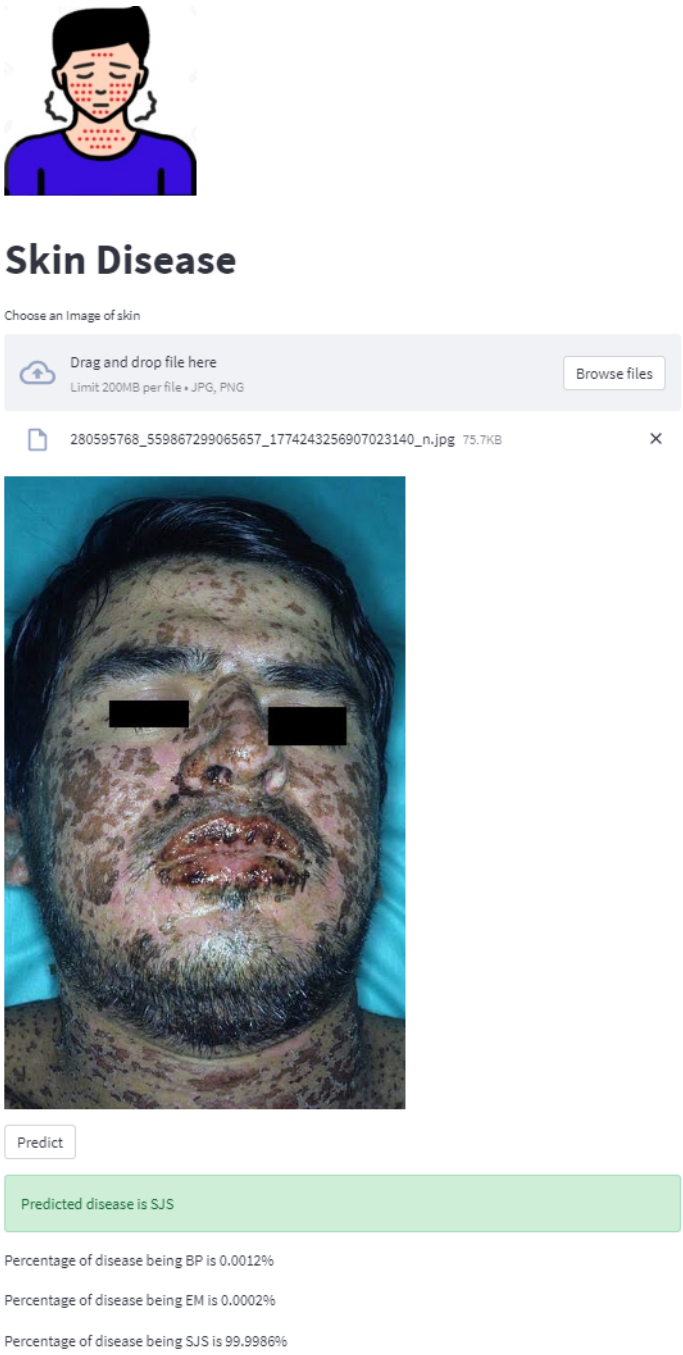

Figure 6.14: Screenshot of the user interface of our developed web application while predicting an image of SJS

Chapter 7

Conclusion

7.1 Conclusion

In the modern world, skin diseases have become a regular phenomenon. In fact, skin diseases have been found to be the world's 4^{th} most common cause of all diseases in humans. It has also been reported that approximately $\frac{1}{3}$rd of the people in the world suffer from them [14]. These skin diseases can occur due to various reasons, ranging from chemical reactions to autoimmune reactions. Although the skin diseases that we have tried to classify in this research work are very unusual, they can prove to be lethal as they are usually misdiagnosed due to their similar symptoms. For this reason, it is crucial to identify these diseases properly with maximum reliability. To add to that, clinical processes for diagnosing skin diseases can be lengthier and painful. Therefore, we have tried to develop a system that would provide a painless and accurate diagnosis in a quick manner. For our research, we have constructed a hybrid dataset and this dataset was experimented on four deep learning pre-trained networks (ResNet50V2, VGG16, Inceptionv3 and InceptionResNetV2) with a custom 10-layer CNN classifier to distinguish the three rare skin diseases - Stevens-Johnson syndrome (SJS), Bullous Pemphigoid (BP) and Erythema Multiforme (EM). From this experiment, we have tried to find the DNN model that provides the highest accuracy and in the end, we have reached the highest accuracy of 99.06% using the InceptionResNetV2 network for transfer learning. The other accuracies that we have obtained are 90.27% using InceptionV3, 95.92% using VGG16 and 98.26% using ResNet50V2. The predictions made by these models have also been explained in this paper using LIME based explainable AI. Moreover, for the classification, 3 different machine learning algorithms namely Random Forest, Logistic Regression, and SVM were applied as well to compare their performances with the CNN architecture. Furthermore, to predict the skin diseases of the users, we have built a web application based on the InceptionResNetV2 architecture as it has shown the optimal accuracy with less erroneous predictions. By uploading skin lesion images into the system, the users can easily get an idea about which skin diseases they are suffering from. Along with that, they can also learn about the probabilities of their skin diseases being a symptom of SJS, EM and BP from this web application.

7.2 Limitations and Future Work

Even though our CNN classification model using InceptionResNetV2 has shown an incredibly impressive performance by having an accuracy of 99.06% on our hybrid dataset, there are still a few shortcomings or limitations in our work which we aim to resolve in the future.

Firstly, our initial intention was to collect image data of skin lesions belonging to SJS, EM and BP locally from various medical centers in Bangladesh along with the global images in order to embrace the local conditions of our country in our experiment. However, due to the Covid-19 pandemic, we had not been able to collect the desired data. Thus, in the future we aim to also include local data while training our models and ensure an unbiased deep learning.

Secondly, since our research focuses on reducing misdiagnosis of SJS, EM and BP, it is crucial that the models are trained on images belonging to every stage of these diseases. This is to ensure that even if an image of an early stage symptom or a late stage symptom is fed into the model or system, it can detect them correctly. Therefore, in future we aim to add more images belonging to all stages of these diseases to our hybrid dataset and enhance the versatility of our model.

Thirdly, as we have mentioned earlier, our dataset currently consists of three types of skin diseases. As a result, our model and web application can only classify three diseases at this point. Hence, our upcoming goal is to extend the scope of our research by incorporating more types of skin diseases and make our web application an effective tool that can detect a plethora of diseases.

Lastly, in future, we plan to work on decreasing the computational time of our models while making sure that the system is sturdy enough to detect all domains of skin diseases faster and with more precision.

Bibliography

[1] M. S. Arifin, M. G. Kibria, A. Firoze, M. A. Amini, and H. Yan, "Dermatological disease diagnosis using color-skin images," in *2012 international conference on machine learning and cybernetics*, IEEE, vol. 5, 2012, pp. 1675–1680.

[2] R. Yasir, M. A. Rahman, and N. Ahmed, "Dermatological disease detection using image processing and artificial neural network," in *8th International Conference on Electrical and Computer Engineering*, 2014, pp. 687–690.

[3] D. A. Shoieb, S. M. Youssef, and W. M. Aly, "Computer-aided model for skin diagnosis using deep learning," *Journal of Image and Graphics*, vol. 4, no. 2, pp. 122–129, 2016.

[4] *Bullous pemphigoid*, Sep. 2018. [Online]. Available: https://rarediseases.org/rare-diseases/bullous-pemphigoid/.

[5] M. ul Hassan, "Vgg16-convolutional network for classification and detection," *Neurohive. Dostopno na: https://neurohive. io/en/popular-networks/vgg16/[10. 4. 2019]*, 2018.

[6] S. Bhadula, S. Sharma, P. Juyal, and C. Kulshrestha, "Machine learning algorithms based skin disease detection," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 2, 2019.

[7] K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Classification of skin lesions using transfer learning and augmentation with alex-net," *PloS one*, vol. 14, no. 5, e0217293, 2019.

[8] Z. Wu, S. Zhao, Y. Peng, *et al.*, "Studies on different cnn algorithms for face skin disease classification based on clinical images," *IEEE Access*, vol. 7, pp. 66 505–66 511, 2019. DOI: 10.1109/ACCESS.2019.2918221.

[9] Z. Elhamraoui, *Inceptionresnetv2 simple introduction*, May 2020. [Online]. Available: https://medium.com/@zahraelhamraoui1997/inceptionresnetv2-simple-introduction-9a2000edcdb6.

[10] H. Janoria, J. Minj, and P. Patre, "Classification of skin disease from skin images using transfer learning technique," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020, pp. 888–895. DOI: 10.1109/ICECA49313.2020.9297567.

[11] A. Kaushik, *Understanding resnet50 architecture*, Jul. 2020. [Online]. Available: https://iq.opengenus.org/resnet50-architecture/.

[12] B. Raj, *A simple guide to the versions of the inception network*, Jul. 2020. [Online]. Available: https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202.

[13] S. Bandyopadhyay, A. Bhaumik, and S. Poddar, "Skin disease detection: Machine learning vs deep learning," 2021.

[14] C. Flohr and R. Hay, "Putting the burden of skin diseases on the global map," *British Journal of Dermatology*, vol. 184, no. 2, pp. 189–190, 2021. DOI: https://doi.org/10.1111/bjd.19704. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/bjd.19704. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/bjd.19704.

[15] R. L. Giesey, S. Mehrmal, P. Uppal, G. Delost, *et al.*, "The global burden of skin and subcutaneous disease: A longitudinal analysis from the global burden of disease study from 1990-2017," *SKIN The Journal of Cutaneous Medicine*, vol. 5, no. 2, pp. 125–136, 2021.

[16] *Introduction to explainable ai(xai) using lime*, Aug. 2021. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-explainable-aixai-using-lime/.

[17] M. Jose A Plaza, *Erythema multiforme*, Jul. 2021. [Online]. Available: https://emedicine.medscape.com/article/1122915-overview#:~:text=Erythema%5C%20multiforme%5C%20major%5C%20has%5C%20a,surface%5C%20area%5C%20of%5C%20sloughed%5C%20epithelium..

[18] M. V. Pranav, C. Koushik, S. M. A V, and S. Ganapathy, "Analyzing the diagnostic efficacy of deep vision networks for malignant skin lesion recognition," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, vol. 1, 2021, pp. 194–199. DOI: 10.1109/CENTCON52345.2021.9687979.

[19] R. Winastwan, *Interpreting image classification model with lime*, Jan. 2021. [Online]. Available: https://towardsdatascience.com/interpreting-image-classification-model-with-lime-1e7064a2f2e5.

[20] K. Hauser, A. Kurz, S. Haggenmüller, *et al.*, "Explainable artificial intelligence in skin cancer recognition: A systematic review," *European Journal of Cancer*, vol. 167, pp. 54–69, 2022, ISSN: 0959-8049. DOI: https://doi.org/10.1016/j.ejca.2022.02.025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095980492200123X.

[21] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on mobilenetv2 network," *Procedia Computer Science*, vol. 197, pp. 198–207, 2022, Sixth Information Systems International Conference (ISICO 2021), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.12.132. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921023565.

[22] D. M. Peraza, *Bullous pemphigoid - dermatologic disorders*, Apr. 2022. [Online]. Available: https://www.msdmanuals.com/professional/dermatologic-disorders/bullous-diseases/bullous-pemphigoid.

[23] *Stevens-johnson syndrome*, Apr. 2022. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/stevens-johnson-syndrome/diagnosis-treatment/drc-20355942.

[24] B. U. Correspondent, *Barishal university student dies of steven johnson syndrome*. [Online]. Available: https://www.daily-bangladesh.com/english/Barishal-University-student-dies-of-Steven-Johnson-Syndrome/59155.

[25] *Erythema multiforme*. [Online]. Available: https://rarediseases.org/rare-diseases/erythema-multiforme/.

[26] *Erythema multiforme: Pictures, causes, treatment, and more*. [Online]. Available: https://www.medicalnewstoday.com/articles/323801#diagnosis.

[27] *Stevens-johnson syndrome/toxic epidermal necrolysis - about the disease*. [Online]. Available: https://rarediseases.info.nih.gov/diseases/7700/stevens-johnson-syndrometoxic-epidermal-necrolysis.