DenseNet based Skin Lesion Classification and Melanoma Detection

by

Arifuzzaman Munaf
17301111
Ariful Hoque
17301107
Kazi Asif Jawwad
17301141

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

Department of Computer Science and Engineering
Brac University
June 2021

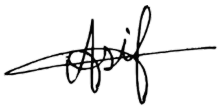
Declaration

It is hereby declared that

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

Student's Full Name & Signature:

Arifuzzaman Munaf
17301111



Kazi Asif Jawwad
17301141

Ariful Hoque
17301107

Approval

The thesis titled "DenseNet based Skin Lesion Classification and Melanoma Detection" submitted by

1. Kazi Asif Jawwad (17301141)
2. Ariful Hoque (17301107)
3. Arifuzzaman Munaf (17301111)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 02, 2021.

Examining Committee:

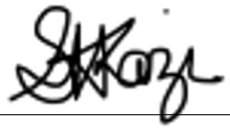
Supervisor:
(Member)

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

Co Supervisor:
(Member)

Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chairperson)



Sadia Hamid Kazi,Ph.D.
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In the language of medical science, the most harmful variant of skin cancer that may develop in human cells is distinguished as melanoma. The principal reasons behind developing melanoma in human skin are still unknown. However, scientists assume that the risk of developing melanoma increases due to exposure to ultraviolet radiation emitting from the sun. The increased rate of melanoma cancer is now a threat to the medical sector to cope with the increasing number of patients. Many scientists have already researched and tried to develop different projects to identify melanoma efficiently. Skin lesions are the best approach to find the symptoms of melanoma and predict the possibility of cancer growing in the skin. In this research paper, the main objective is to classify different types of lesions and find melanoma from skin lesion images using DenseNet-121 which is a densely connected CNN-based algorithm. We evaluated on 5066 imbalanced test images from ISIC 2019 Challenge dataset for initial classification of lesion images. We also organized the dataset into a balanced dataset by over sampling and downsampling where 600 test images were used for validation. The evaluation of imbalanced and balanced datasets results in respectively 80% and 84% accuracy for lesion images classification. Moreover, we normalized the dataset into two different classes which consists of melanoma and non-melanoma lesion images to perform binary classification. In this stage, we executed our model on 2000 test images and got an accuracy of 89% for classifying melanoma accurately.

Keywords: Cancer; Distinguished; Melanoma; Ultraviolet; Lesion; Densenet

Acknowledgement

On the beginning, we thank Allah because of His blessings, which has enabled us to continue our research without facing any major difficulties. In addition, we wanted to thank all the supportive faculty members and our supervisor in particular for tolerating our mistakes and providing continuous input to enhance our research. Moreover, we thank our parents as well as teammates who have given us tremendous support during the semester.

Table of Contents

Declaration i

Approval ii

Abstract iii

Acknowledgment iv

Table of Contents v

List of Figures vii

List of Tables viii

Nomenclature ix

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 1
- 1.3 Aims and Objectives . . . 2
- 1.4 Thesis Structure . . . 3

2 Related Work 4

3 Methodology 7
- 3.1 Dataset Description . . . 8
- 3.2 Data Pre-processing . . . 9
  - 3.2.1 Categorization of Original Dataset . . . 9
  - 3.2.2 Data Normalization . . . 13
  - 3.2.3 Training Dataset Augmentation . . . 13
- 3.3 Model Description . . . 14
  - 3.3.1 Convolutional Neural Network(CNN) . . . 14
  - 3.3.2 Convolution Layer . . . 15
  - 3.3.3 Padding . . . 17
  - 3.3.4 Pooling Layer . . . 17
  - 3.3.5 Activation function . . . 18
  - 3.3.6 Fully connected Layers . . . 19
  - 3.3.7 DenseNet . . . 19

4 Implementation **23**
4.0.1 Basic libraries for our model 23
4.0.2 Implementing the model 24
4.0.3 Training the model 25
4.0.4 Predicting a single image 25

5 Result Analysis **26**
5.1 Performance Evaluation using Original Dataset 26
5.2 Performance Evaluation using Balanced Dataset 28
5.3 Performance Evaluation using Normalized Dataset 29
5.3.1 Observation of Classification Report 31
5.3.2 Observation and Comparison of Accuracy 32

6 Conclusion and Future work **33**

Bibliography **36**

A Appendix **37**
A.1 Questions from the panel members: 37

List of Figures

3.1 Overall Overview of System Mechanism 7
3.2 Histogram of ISIC-2019 Dataset 8
3.3 Histogram Correction of a sample from HAM10000 dataset 9
3.4 Categorization of Original Dataset into Respective Sub-directory . . . 10
3.5 Workflow of Train-test Evaluation 11
3.6 Train-test Ratio of Dataset for all Classes 12
3.7 Train-test Ratio of Balanced Dataset for all Classes 12
3.8 Train-test Ratio for Binary Classification 13
3.9 Flattening of a 3x3 image matrix into a 9x1 vector 14
3.10 4x4x3 Matrix Representation of RGB Image 15
3.11 Convolution of a Image with a 3x3 kernel to Extract Feature_matrix . 16
3.12 Zero padding added to an Image 17
3.13 Max pooling , Average pooling 18
3.14 Activation functions: a)ReLU, b)tanh, and c)Sigmoid 18
3.15 Fully Connected Layers 19
3.16 5 -Layer DenseNet Architecture with 4 expansions 20
3.17 DenseNet with 3- Dense blocks 21
3.18 Interior view of a Dense Block 21
3.19 DenseNet Bottleneck Layer 22

4.1 DenseNet-121 for the proposed Lesion Classification and Melanoma Detection Model 25

5.1 Accuracy of Full Dataset 26
5.2 Loss of Full Dataset 26
5.3 Confusion Matrix of Full Dataset 27
5.4 Accuracy of Balanced Dataset 28
5.5 Loss of Balanced Dataset 28
5.6 Confusion Matrix of Balanced Dataset 29
5.7 Accuracy of Binary classification 30
5.8 Loss of Binary Classification 30
5.9 Confusion Matrix of Binary Classification 31
5.10 Comparison among a) Original Data b) Balanced Data c) Normalized Data 32

List of Tables

3.1 ISIC-2019 Dataset (number of images) 9
3.2 Categorization and Train-test Evaluation of ISIC-2019 Challenge . . . 11
3.3 Dataset Representation for Binary Classification 13
3.4 Augmentation Parameters and Their Values 14

4.1 Dense Block and Transition block 24

5.1 Statistical Overview of Model on Original Data 27
5.2 Statistical Overview of Model on Balanced Data 29
5.3 Statistical Overview of Model on Normalized Data 31
5.4 Comparison among the Accuracy of Observed Models 32

Nomenclature

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

ANN Artificial Neural Network

CNN Convolutional Neural Network

DNA Deoxyribonucleic Acid

MEL Melanoma

$ReLU$ Rectified Linear Unit

Chapter 1

Introduction

1.1 Motivation

The skin refers to an essential organic component of the human body that provides protection to the entire exterior of the body and interior organs from the virus, bacteria, and injuries of the environment. As it is located in the outer part of our body to shield our crucial organs, diverse radioactive and viral molecules tend to attack our skin and develop severe diseases in our skin. Biochemists mark skin cancer as one of those malicious diseases. Skin cancer is one of the most diagnosed cancers that result in the death of many people around the world. If it is detected in the early phase of cancer, it may be curable with proper treatment and care. So it is crucial to know the earliest phase to identify the symptoms of cancer to rescue the affected one. In most cases, doctors seem to use the visual part of the skin to identify symptoms that result in defective predictions by the doctors. It is very tough for doctors to identify symptoms with open eyes during the initial phase. In this regard, various deep learning models can help us with computer vision by training the model with thousands of images related to skin melanoma cancer to create an autonomous system to predict whether a person has symptoms of cancer or not. Moreover, this autonomous system can reduce the pressure on doctors by its melanoma predictability techniques. Finally, melanoma detection in the earliest phase is an urgent call in the health care systems to reduce the world's death rate.

1.2 Problem Statement

Melanoma cancer develops in the skin when an inconsistent rate of melanocyte cells is produced in our skin. Melanocytes are the cells that produce melanin to give color to our skin. In the human body, skin cells seem to develop in a consistent and controlled approach by pushing dead or old cells toward the skin surface to settle fresh cells in a cell lattice that prevents our skin from diseases. In the meantime, skin cells may grow in an uncontrolled way due to cell DNA damage leading to a higher probability of skin cancers. If the inconsistency is seen in melanocyte cells, melanin tends to grow at an increasing rate resulting in melanoma cancer. However, several chemical factors such as genetic and environmental factors are expected to develop melanoma in our body. Besides, researchers believe ultraviolet rays from the sun and tanning lamps are the fundamental cause of melanoma cancer. Melanoma cancer can result in massive death of population all over the world. In the USA, each year approximately

10,000 patients die annually due to melanoma cancer[1]. Nevertheless, the good news is that we can minimise this rate by detecting the symptoms in the earlier stage. The five-year survival rate is over 95% if we have successful attempts of diagnosing lesions with skin cancer in the early stage. However, the percentage goes down to 20% if we fail to detect it earlier[2]. Initially, we use dermoscopy to physically diagnose the melanoma developed in the skin. Dermoscopy examination classifies melanoma by assessing the radius, color, texture, and symmetry of moles. Mainly, dermatologists grab the information from the dermoscopy test and prepare a researched report compared with the existing template report of medical science to conclude. However, the dermoscopy report may result in defective classification in some cases as it is mainly evaluated by dermatologists[3]. On the contrary, image-based classification of lesion and melanoma can be optimized further with better accuracy with the help of deep neural networks such as MobileNet, GoogleNet, Inception, ResNet, deep CNN, AlexNet and DenseNet.Moreover, the traditional procedures of lesion classification for melanoma takes more time to predict the result as it needs more human resources on the eve of more tests. The addressed problem can be solved with the robust DenseNet algorithm, which is described with proper methodologies in this research paper for convenience.

1.3 Aims and Objectives

Melanoma is one of the most prevalent tumors in first-world nations, and it has the ability to be malignant. Melanoma may spread to the liver, lungs, brain, and abdomen. Bangladesh is the world's ninth most populated nation, with 142 million inhabitants. In Bangladesh, there are 13 to 15 Millions cancer patients, with around 2 Millions cancer sufferers diagnosed per year[4]. Every day, more than 9,500 individuals in the United States are diagnosed with skin cancer. More than two people die from skin cancer every hour. According to other statistics, 2333 people died from melanoma cancer in the United Kingdom between 2016 and 2018.[5] Melanoma is however discovered early, so it has the best possible chance of being cured.
By examining skin lesion images, we effectively-identified melanoma cancer, which is one kind of skin cancer. We intend to introduce different neural network models and how these models will lead to recognizing this dangerous cancer by performing this experiment.The contribution of our research work can be shortly pointed as below-

- Skin lesion classification of 8 different classes using DenseNet which results in better performance.
- Higher predictability rate of melanoma detection compared to traditional dermoscopy test.

1.4 Thesis Structure

We talked about previous work in this area at the beginning of Chapter 2. Then we go through our algorithm in detail. Then, we explain each part of our model. After that, in chapter 4, we illustrate the implementation part of our research work. We elaborately describe the data set used in this thesis report and how we pre-process this data set using various tools. In chapter 5, We present our model's outcomes and compare them with previous ones.

Chapter 2

Related Work

Skin cancer is considered to be a significant issue throughout the world for which many remarkable researches have been conducted by experts.We reviewed several papers to gather experience from those researches.Besides, we tried to compare best researches with our proposed methodology to contribute more in the field of skin cancer.

R.Rina , B. M. Achmad and P. P. Rachmadinna discussed where we could find melanoma cancer and the color description of this cancer. They also focused that Melanoma cancer is mainly detected by skilled doctors by analyzing the results of dermoscopy examination which is a manual process. They used CNN, which is a deep learning algorithm, and LeNet-5 architecture which is a multilayer network for classifying their image data which is deep learning technology. They collected 44 samples of dermoscopic images for the evaluation of training outcome. The main application is created using Python Programming Language and Keras Library as Tensorflow backend. They obtained the dataset from the ISIC-2018 website, which consists of 220 dermoscopic images. In their dataset, there were 110 samples melanoma images and 110 non-melanoma images.[6]

In another research ,Q. K. Muhammad and his teammates described Melanoma as a fatal type of screen cancer. They proposed an integrated system that can identify and cluster melanoma from nevus by using state-of-art techniques. They used a Gaussian filter for noise reduction from the image of skin lesions. Then, they extracted textual and color features from the lesion. They use a SVM for the classification of skin cancer into melanoma and nevus. For this system, they used the DERMIS dataset with 397 skin cancer images, and among them, 146 are melanoma, and 351 are nevus skin lesions. [7]

In recent times,melanoma skin cancer detection is conducted using deep neural networks and transfer learning architecture. S. Abhinav and Dheeba identified image glitches such as inconsistent brightness, blockage, and the assertion that numerous photos have the same form, appearance, and surface. They used the ISIC dataset, and 3000 samples were utilized for training and 600 photos were utilized for validation. They used the concept of transfer learning for categorization. They used three pre-trained models – Inception v3, InceptionResNet v2, and ResNet 152. They divided the dataset into two sections: a training samples set and an experimental samples, each comprising 80% and 20% of the statistical analysis.In the pre-processing

part, they augmented data by shearing, zooming, flipping, and brightness change and made almost double the original dataset. To measure efficiency, they utilized a 500% attrition and batch normalization layers within it. Two thick layers with 64 neurons and two neurons were employed.[8]

Another study gave us a brief idea about the analysis of melanoma lesions using a deep learning algorithm implemented on a database server with a graphics processing unit (GPU). The proposed model incorporates clinical photographs, which could assist a dermatologist in the early detection of this form of skin cancer. The proposed system preprocesses input clinical photos that could include light and noise artifacts in order to minimize these anomalies. After that, the improved images are fed into a deep learning algorithm that uses a pre-trained convolutional neural network (CNN). The CNN classifier identifies melanoma and benign instances using a wide range of training examples. In contrast to state-of-the-art approaches, experimental findings indicate that the proposed methodology is superior in terms of diagnostic accuracy.[9]

Moreover, we come to know that poor contrast between lesions and skin, visual similarities between melanoma and non-melanoma lesions, and other factors make correct melanoma identification extremely difficult. P. Fabio,A. Sandra and V. Eduardo introduced two deep learning approaches in their research to solve three essential tasks that have been developed in the sector of skin lesion image analysis, lesion dermoscopic feature extraction, and lesion classification. A supervised learning method built of two totally FCRN is given to develop the delineation and crude recognition accuracy at the same time. By measuring the distance heat-map, a lesion index measurement unit (LICU) is created to optimize the coarse classification effects. For the dermoscopic function extraction task, a simple CNN is suggested. On the ISIC 2017 dataset, the proposed deep learning frameworks were evaluated. [10]

With the number of architectures accessible, choosing one over the other is becoming more difficult. Existing research shows that the success of CNN architectures on ImageNet correlates strongly with their output on goal tasks while doing transfer learning.They test their argument for melanoma classification through nine CNN architectures, five sets of splits created on the ISIC Challenge 2017 dataset, and three repetitive steps, yielding 135 models. They evaluated two techniques for determining which models to ensemble: random selection vs. utilizing the validity range to decide the ones to choose first. They discovered that the second solution had a marginal benefit for limited ensembles, but that random option was still competitive. Despite not aiming for peak results, they were able to achieve AUCs that were equal to first position in the ISIC Challenge 2017.[11]

Computational innovations that are advanced and computer controlled are being developed for the identification of skin cancer, just as they are in other disciplines of medicine.Deep learning is a type of robotic practice that includes instructing a network model with large data sets and matching targeted labels. The existence of "good representations" of the input classification is measured on a pixel level within the network, which is made up of several sequential layers. As the number of training photos grows, the system deploys and evaluates picture elements that are important for identifying benign cysts from melanomas. As a result, deep learning can be defined as feature learning with a hierarchical structure. Deep learning convolutional

neural networks (CNNs) are a subcategory of deep learning algorithms that have demonstrated good image classification efficiency. Deep learning CNNs have shown diagnostic success comparable to that of seasoned physicians in the assessment of medical photos from dermatology, radiology, ophthalmology, and pathology to date. Histopathologic examination was used to confirm all melanoma reports, with supplementary detail on localization, Breslow thickness, and patient details required. To enable for later melanomas studies, the skin markings were manually placed on the melanoma photos using picture editing tools. Twenty nevi from the research set is used in a comparative comparison to show that electronically superimposed markings are equivalent to in vivo markings. The CNN's mean melanoma risk score for unmarked benign nevi was 0.15. Melanoma risk ratings vary from 0 to 1, with higher scores indicating a higher likelihood that the measured lesion is a melanoma. In situ markings raised the mean score to 0.52, while electronically superimposed markings yielded a similar mean score of 0.59 .[12]

Melanoma is the most serious form of skin cancer, although it may be treated if caught early enough. According to recent articles, AI technology has been found to be very effective at categorizing pictures of non-malignant cysts and malignant cysts with rheumatologist accuracy. However, no statistically meaningful change over dermatologist classification has been identified. 4204 biopsy-proven photos of malignant and benign (1:1) were used to train a convolutional neural network for this comparative analysis (CNN). Deep learning approaches were incorporated. An additional 804 biopsy-proven dermoscopic photographs of malignant and benign were anonymously provided to dermatologists from nine German university hospitals for the experiment, who measured the accuracy of each picture and stated their treatment recommendations. The key outcomes were three McNemar's experiments contrasting the performance of the CNN test runs in terms of sensitivity, reliability, and overall correctness. [13]

Many tumors are diagnosed by a board-certified pathologist using a tissue biopsy under a microscope. According to recent studies, human pathologists have a large level of disagreement. In the case of melanoma, there is a 25–26% discrepancy in classifying a healthy nevus and a malignant melanoma, according to the literature. To improve the accuracy of lung and breast cancer diagnosis, deep learning was effectively applied. This research aims to show how deep learning will aid human evaluation for histopathologic melanoma diagnosis. Six hundred ninety-five lesions were categorized according to existing standards by an expert histopathologist. Only the haematoxylin and eosin-stained (HE) slides of these lesions were digitized and then uniformly clipped using a slide scanner. Five hundred ninety-five of the images produced were used to train a CNN. The findings of the CNN were compared to the initial class labels using the additional 100 HE picture bits. [14]

Chapter 3

Methodology

Firstly, we categorized the ISIC 2019 Challenge Dataset into different directory for the ease of our work.We divided our working mechanism in three subsections.The first section contains 25331 images directly from ISIC 2019 dataset.Then we applied train-test split in 80 : 20 ratio in order to train and test our DenseNet-121 model for determining accurate classification. In the second section,we augmented our dataset and took 3600 images and applied train-test split with a ratio of 83 : 17 further experiment.Finally, in the last section, we generalized our dataset into two classes titled as MEL and NON-MEL for binary classification which will predict a image is MEL or not.The overall system model is described in short with a network flow diagram as given below in Figure 3.1.

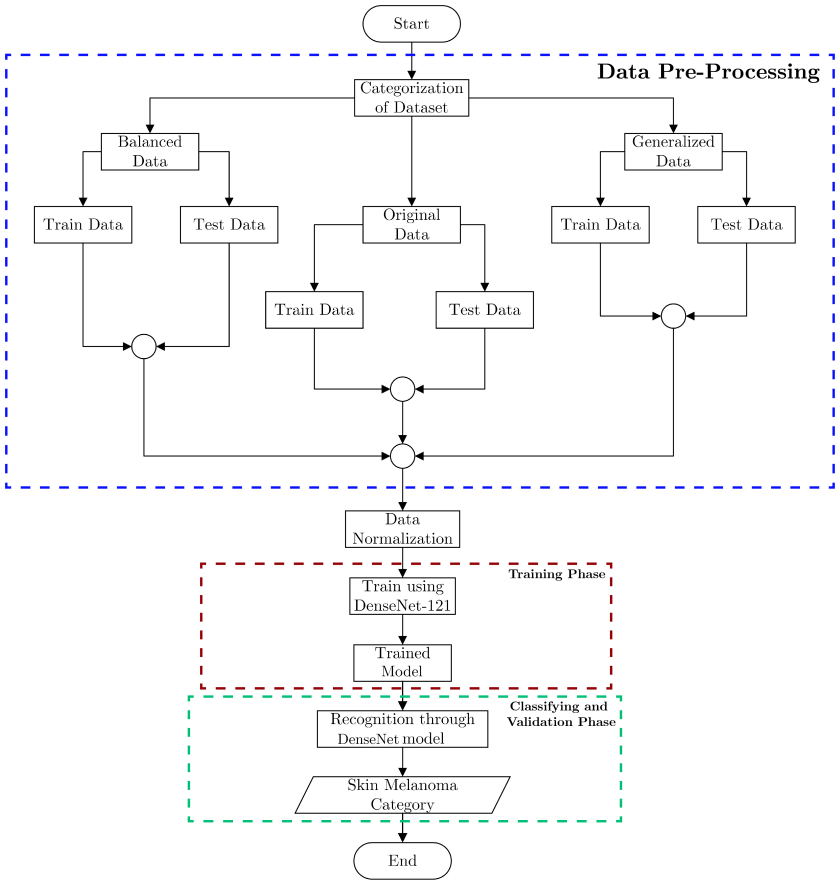


Figure 3.1: Overall Overview of System Mechanism

3.1 Dataset Description

In our research, we have used a large data set to train our model appropriately so that we can predict efficient result to detect skin cancer and lesion classification. We have used the ISIC-2019 challenge data set formally titled "Skin Lesion Analysis Towards Melanoma Detection" consisting of 25,331 images that are organized for classification of different skin lesions through dermoscopic images among 8 different categories.The distribution of lesion images for 8 different classes has been organised as shown in Figure 3.2.

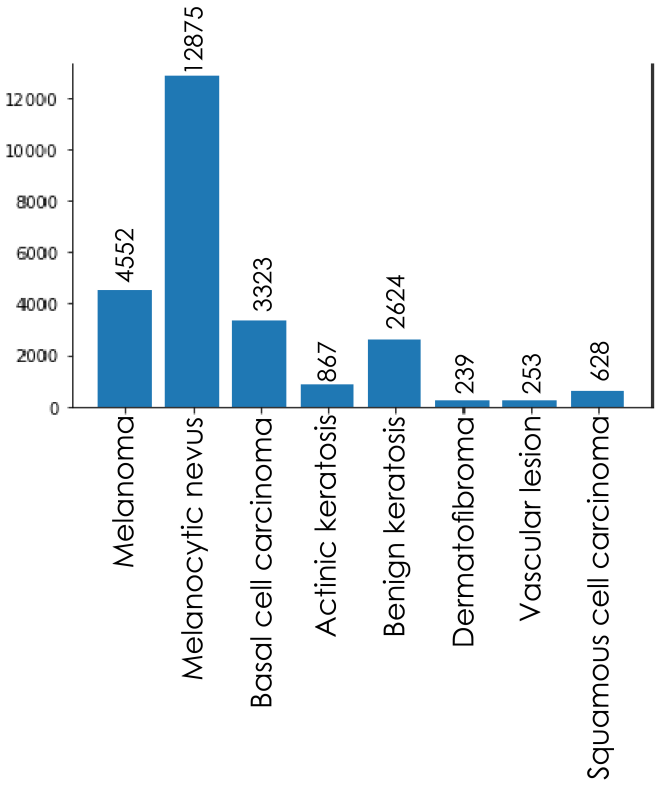


Figure 3.2: Histogram of ISIC-2019 Dataset

The International Skin Imaging Collaboration(ISIC) has introduced the ISIC archive as an international database of skin lesion images through organizing ISIC challenges. The ISIC-2019 data set is mainly formed for clinical training of diagnosis systems which is also used for research purpose of various automated deep learning algorithm interpretation. The ISIC-2019 data set is mainly a combination of HAM10000 Data set, BCN_20000 data set and MSK Dataset. HAM10000(Human Against Machine with 10000 training images) data set consists of 10000 lesion images of size 600×450 that are mainly centered to 800×600 px at 72DPI and compressed around the lesions. To improve strong contrast and pixel density, the data set curator performed manual histogram corrections[15]. A sample of a primary image is extracted from the HAM1000 data set for the comparison of pre-correction and after-correction of the image which is shown in Figure 3.3.

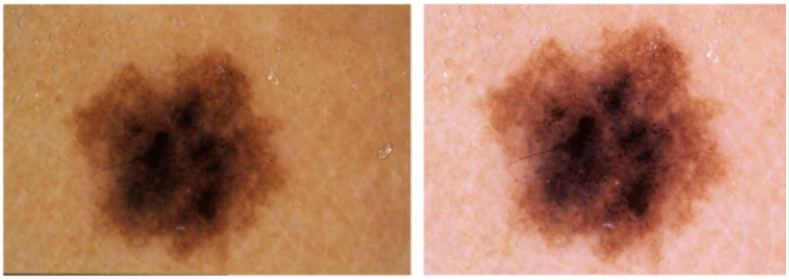


Figure 3.3: Histogram Correction of a sample from HAM10000 dataset

The BCN20000 dataset is comprised of 19424 dermoscopic photographs of skin lesions taken at the Laboratory in Barcelona between 2010 and 2016. This dataset is intended for use in tasks such as segmentation, diagnosis, and categorization that include lesion recognition. BCN20000 includes photographs with a resolution of 1024 * 1024 pixels. This dataset is especially complex to work with since many samples are uncropped and cysts are found in inaccessible and unusual places.[16] MSK Dataset can be used for research and development of skin melanoma identification through algorithm for automated diagnosis. The dataset contains images of various resolution that makes the dataset complex for the neural models to predict proper results.[17]

3.2 Data Pre-processing

The distribution of all the data are given in the Table 3.1 below:

Class	**Distribution of Images**
MEL	4522
NV	12875
BCC	3323
AK	867
BKL	2624
DF	239
VASC	253
SCC	628

Table 3.1: ISIC-2019 Dataset (number of images)

3.2.1 Categorization of Original Dataset

All the images in our dataset are given in one folder with different names for different image. There is a CSV file that contains the name and the category of each images. The categories are one hot encoded. There are many ways we can load the total data for our training. For example, we can create a CSV file of all the images and label them on a separate column, then we can read data from the CSV file or we can make a pickle file to save all the images with their respective label.

But this would be very difficult for us since we have limited RAM. A huge amount of ram would require to handle the total 25331 images. That is why this is not efficient approach to handle the input data.

The second way we can handle the data is by using batch size which means, if we can somehow select a particular number of images and train the model over all the selected images and continue it for rest of the images, we can avoid the limitation of our RAM. To execute this, we need to pre-process our data by dumping the images to their respective category's folder. For example, all the images that are under MEL category will be under MEL folder and so on.

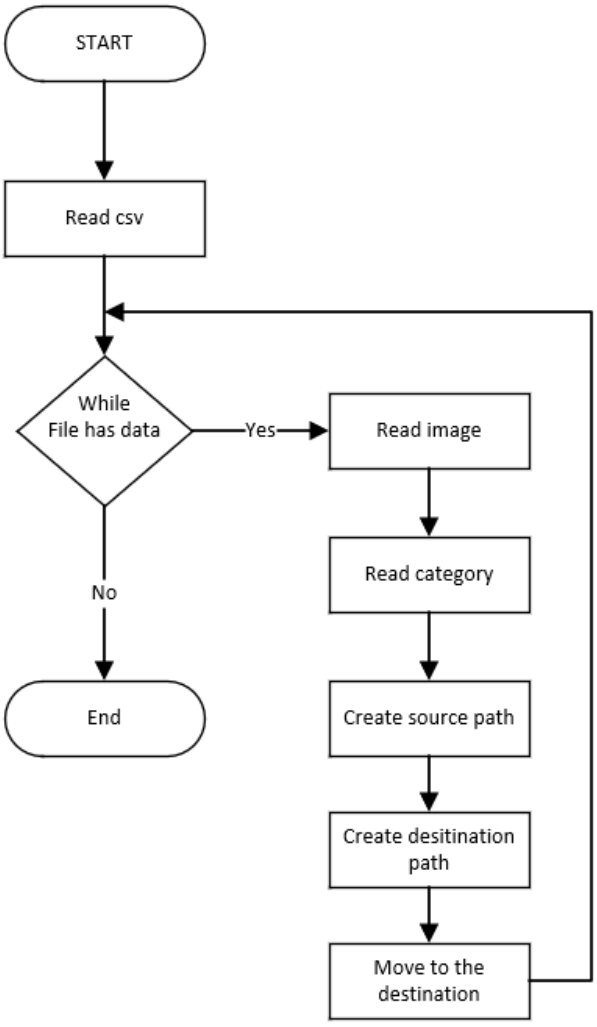


Figure 3.4: Categorization of Original Dataset into Respective Sub-directory

Since the data set did not define any train and validation data, we need to create the train and validation data by ourselves. For creating the training and validation data, we selected random 20% of data from each class and dump them to validation data folder. So the final validation data sets contains:

Class	Categorized Images	Train Images	Validation Images
MEL	4522	3351	904
NV	12875	10300	2575
BCC	3323	2659	664
AK	867	694	173
BKL	2624	2100	524
DF	239	192	47
VASC	253	203	550
SCC	628	503	125

Table 3.2: Categorization and Train-test Evaluation of ISIC-2019 Challenge

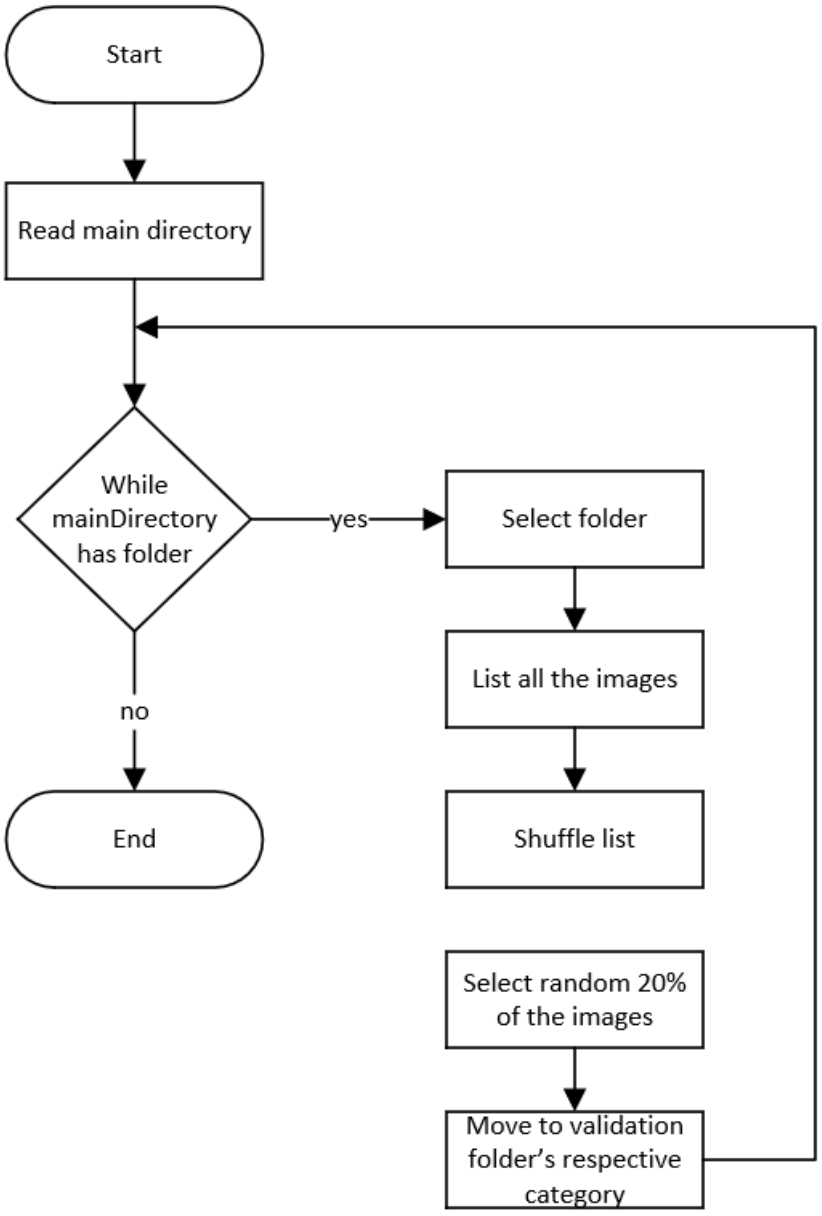


Figure 3.5: Workflow of Train-test Evaluation

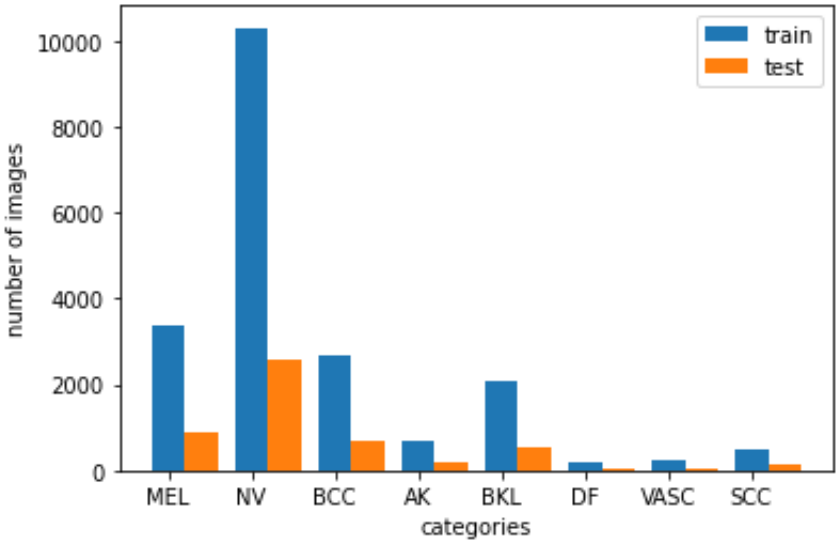


Figure 3.6: Train-test Ratio of Dataset for all Classes

In addition, the data set is highly imbalanced.We wanted to train our model in a balanced dataset and want to see the result as well .So, we had to balance the dataset by re-sampling. Some class has huge number of images and some classes has small number of images. We selected 375 images from each class as training dataset and 75 images for test dataset. First, we had to undersample the MEL, NV, BCC, AK, BKL, SSC classes since they have more than 375 images in each class and oversampled the DF, VASC classes as they have less than 375 images. For under sampling the classes, we randomly choose 375 images for train data and 75 images for test data. We used vertical flip, zoom in and zoom out, rotating for augment the images of DF and VASC classes. After the data augmentation steps finished, we randomly selected 75 images for test data and 375 images for training data.

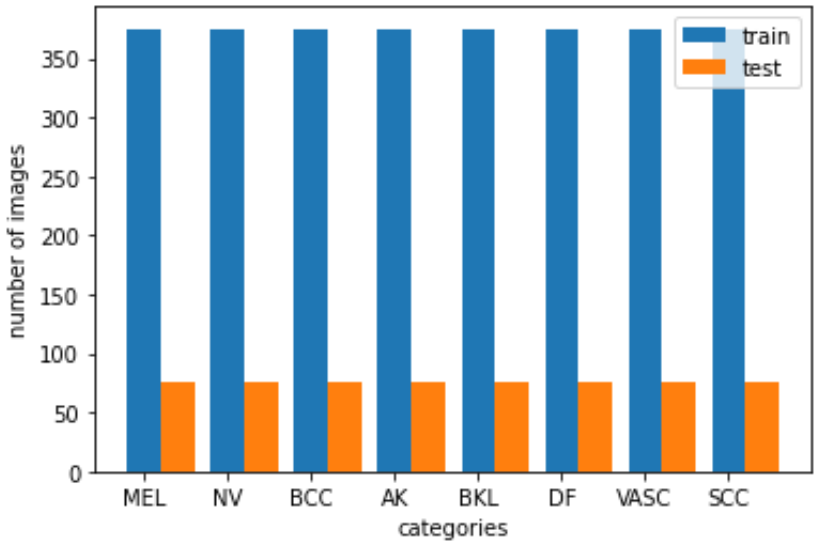


Figure 3.7: Train-test Ratio of Balanced Dataset for all Classes

3.2.2 Data Normalization

We also normalized all the classes into two categories based on melanoma images where categories are labeled respectively: MEL and Non-MEL. For this purpose, we took all the images of MEL class and randomly selected 4522 images from the other seven classes. Then we randomly selected 20% images from the MEL class and 22% image from Non-MEL class. All the steps that are followed are same as before. So the dataset of binary classification is:

Category	Total images	Train images	Test images
MEL	4522	3522	1000
Non-MEL	4522	3522	1000

Table 3.3: Dataset Representation for Binary Classification

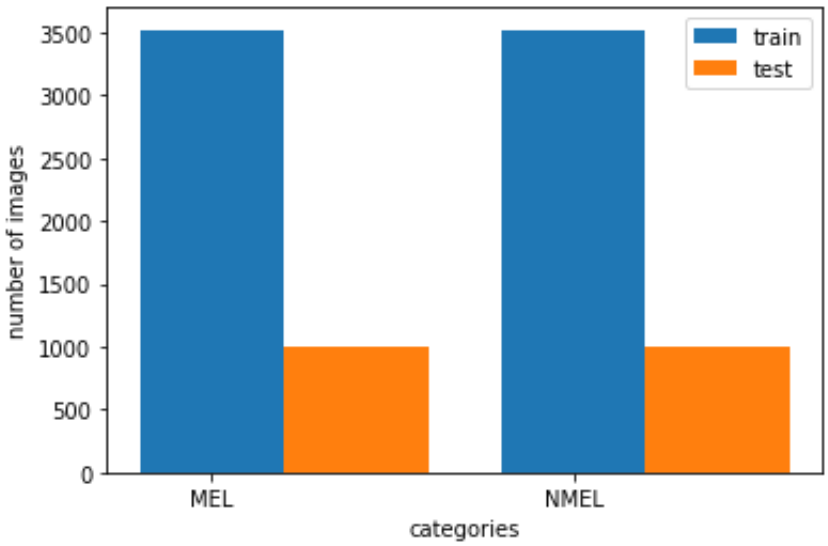


Figure 3.8: Train-test Ratio for Binary Classification

3.2.3 Training Dataset Augmentation

Before training the model, we augmented the entire training dataset. This augmentation is required to avoid overfitting. There are some pictures which a model can detect properly if the pictures are captured from a particular angle. However, if the picture's angle changes, a model won't detect that picture since the model did not learn how to classify the image in different angle. Same rules go for zoom in and zoom out. Some picture's special features are found at the center of the image. So, a model often fails to identify them if it was not trained to identify the zoom in image. If we augment our images before training, we can overcome these problems. Moreover, we rescaled all the images by 1/255. The augmentation information of our dataset is given below:

Parameter	Values
Re_scale	1 / 255
Rotation_range	40.0
Width_shift_range	0.20
Height_shift_range	0.20
Shear_range	0.20
Zoom_range	0.20
Horizontal_flip	On

Table 3.4: Augmentation Parameters and Their Values

Finally, we defined early stop by four epochs. Early stopping is mainly used to avoid overfitting our model. When model do not improve its validation accuracy for four consecutive times, it will stop the training.

3.3 Model Description

3.3.1 Convolutional Neural Network(CNN)

A CNN mainly works on images and it is an algorithm that allocate some values to different aspects in the image and separately identify them. If we use ANN for image classification we can find various difficulties like too much computation, ANN treats local pixel same as pixel far apart and sensitive to location of an object in an image. In CNN,low amount of pre-processing is needed compared to other classification images.
We can assume an image which is the combination of pixel values in matrix format. Here is an example of 3x3 image matrix into a 9x1 vector.

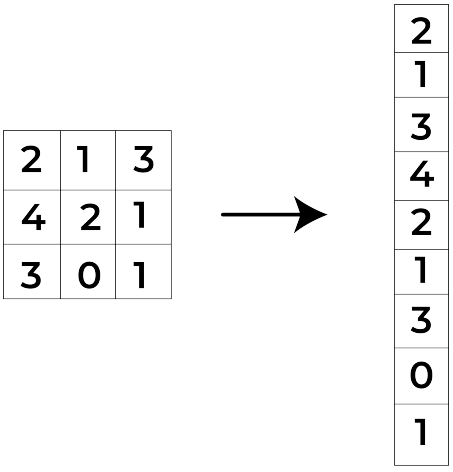


Figure 3.9: Flattening of a 3x3 image matrix into a 9x1 vector

We can perform class prediction on a binary image like identifying a number in image using multi-layer Perceptron which will give an average value but if we give a complicated image with lots of pixel incorporate with it, this method can not perform well. CNN can capture the spatial and temporal dependencies of an image.In CNN, lesser parameters are involved and weights of different parameters can be use repeated times.So,this model performs better on image dataset.

We can think an RGB image can be divided into three color planes as shown in the figure.There are others color space like RGB,CMYK,HSV etc.

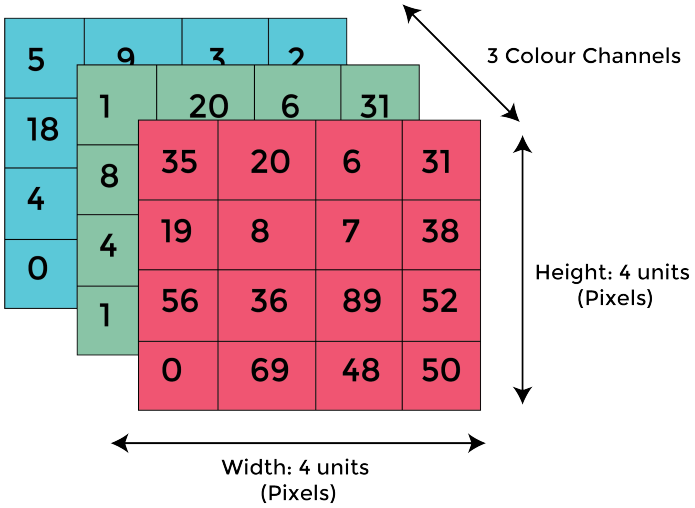


Figure 3.10: 4x4x3 Matrix Representation of RGB Image

ConVNet's performs reduction in image and make a simpler format that can easily processed also it preserve features of image that are important for image for performing a successful prediction.[18]

3.3.2 Convolution Layer

In convolutional neural networks, the essential building elements are convolutional layers. A convolutional layer is made up of multiple filters of a single dimension, and the network as a whole attempt to learn the values of these filters. The filter is introduced by moving it over the input picture and multiplying the dots. The dot multiplication's output values are determined by the shapes that appear on the image. The size of the filter is typically much smaller than the size of the input image. As an outcome, the model attempts to learn a filter that will adapt to a specific input field, allowing it to distinguish shapes and patterns in a specific image environment. In figure 3.11, we can find an input image of size 6×6, a 3×3 feature detector matrix and an output of 4×4 feature map matrix. The mechanism is given bellow-
3*1+1*1+2*1+0*0+5*0+7*0+1*-1+8*-1+2*-1 = -5.
0*1+5*1+7*1+1*0+8*0+2*0+2*-1+9*-1+5*-1 = -4.
1*1+8*1+2*1+2*0+9*0+5*0+7*-1+3*-1+1*-1 = 0.
2*1+9*1+5*1+7*0+3*0+1*0+4*-1+1*-1+3*-1 = 8.
After completing this calculation we can get a 4×4 feature map matrix by convolving the input and filter matrix.

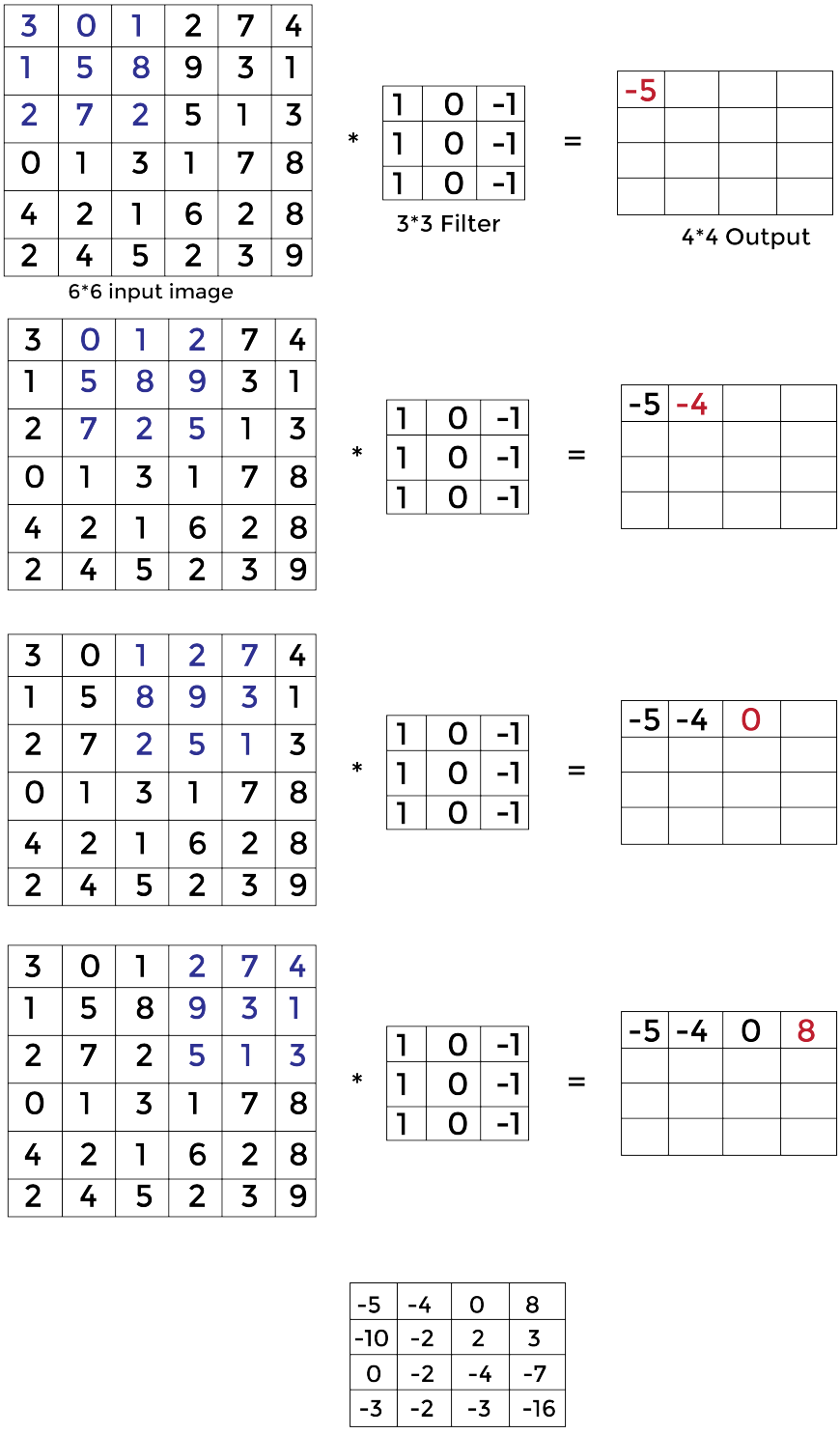


Figure 3.11: Convolution of a Image with a 3x3 kernel to Extract Feature_matrix

An input images has many high-level features. By doing this convolution operation, we can extract the high-level features. The 2D convolution layer is the most popular kind of convolution and is generally abbreviated as conv2D. In a conv2D layer, a filter or kernel "slides" through the 2D input data, executing element-wise multiplication. As a consequence, the findings will be summed into a single output pixel. For each point it slides over, the kernel will do the same procedure, changing a 2D matrix of features into a separate 2D matrix of features.[19]

3.3.3 Padding

Padding is basically the addition of zero layers to our input pictures. After padding, our $n \times n$ picture becomes $(n + 2p) \times (n + 2p)$ where p=number of layers of zeros added to the boarder of image. As a result, applying convolution (with the $(f \times f)$ filter) produces $(n + 2p - f + 1) \times (n + 2p - f + 1)$ images. To illustrate, after performing convolution on a (8×8) image with one layer of padding and a (3×3) filter, we will get a (8×8) output.
By taking the pixels at the original image's boundary into the center of the padded image, this enhances their contribution. As a result, both the details on the image's edges and the information in the center are retained.

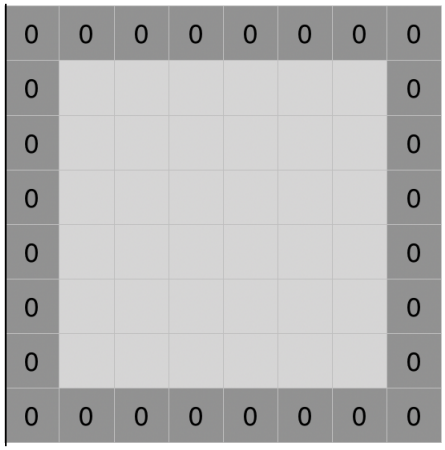


Figure 3.12: Zero padding added to an Image

3.3.4 Pooling Layer

Other than convolutional layers, CNN can use pooling layers to reduce the size of the representation to speed the computation as well as make some of the features it detects a bit more robust. We can find many types of pooling layer. From them Max pooling and Average pooling can plays a significant role in classifying of an image. By giving an abstracted representation of the convolved features, the max pooling layer helps minimize their spatial size and over-fitting. It's a discretization method based on sampling. It works in the same way as the convolution layer, but instead of taking the dot product of the input and the kernel, we take the maximum of the input overlapped by the kernel. Rather than taking max input we can take the average of each block. This process is known as Average Pooling. Average pooling differs from Max pooling because Average pooling stores a lot of information about

a block's or pool's "less significant" parts. Sometimes this can be beneficial in a variety of situations.

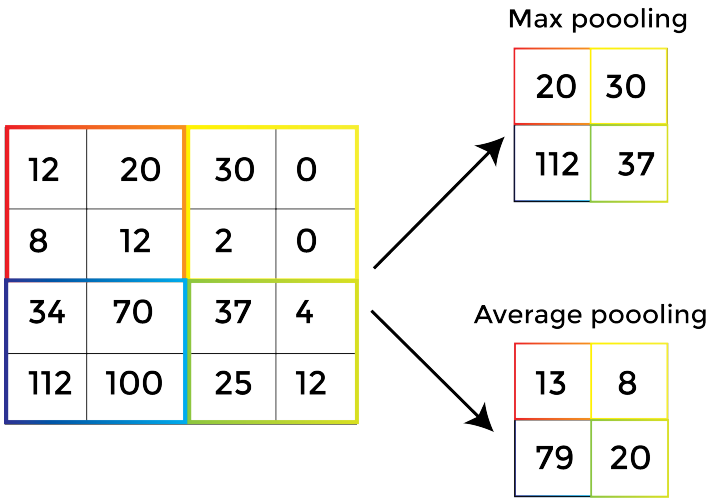


Figure 3.13: Max pooling , Average pooling

3.3.5 Activation function

To learn difficult functional mappings between the inputs and response variables ,non-linearity is introduced to a model by the Activation Function. There are many activation functions like Sigmoid, tanh, RelU etc. There are two major advantages of RelU function that is it is very simple to calculate because it depends only a comparison between its input and the value 0. Another advantage is it also has a derivative of 0 or 1, depending on whether the input is negative or positive.[20]

$$\text{ReLU}'(\text{x}) = \left\{ \begin{array}{ll} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{array} \right\} \tag{3.1}$$

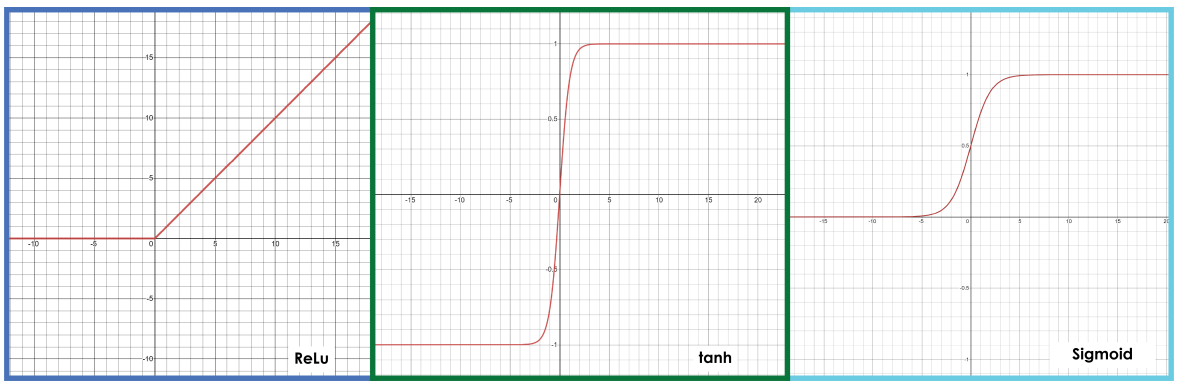


Figure 3.14: Activation functions: a)ReLU, b)tanh, and c)Sigmoid

3.3.6 Fully connected Layers

The input layer nodes are linked to every node in the second layer in a fully connected layer. By adding a fully connected layer, a model can learn different non-linear combinations of high-level features which is get from convolutional layers. To introduce non linearity and reduce over-fitting generally activation function and dropout layer are used between two consecutive fully connected layer. A feature space is provided by the convolution layers and Fully connected layers attempts to extract a pattern from that feature space.[21]

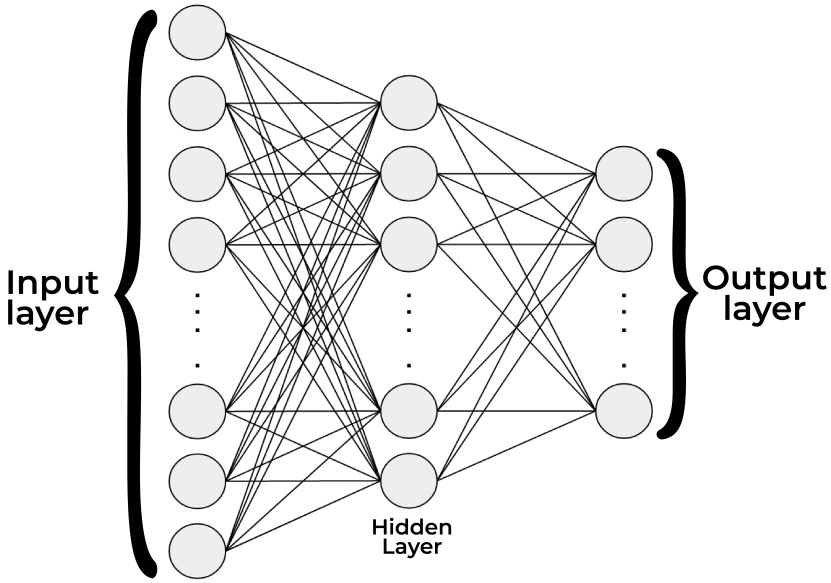


Figure 3.15: Fully Connected Layers

3.3.7 DenseNet

The architecture of standard Convolutional Neural Networks is very straight forward and problems with this architecture arise when the network tends to grow bigger for which the information from input layer to final output layer may vanish. They overcome the issue of vanishing gradients, facilitate feature propagation, and enhance feature reuse. Besides, CNNs have some other limitations that are overcome with DenseNets for better performance and accuracy.

DenseNets do not require learning from redundant feature maps for which they need fewer parameters compared to equivalent standard CNN networks. Moreover, we need to drop some layers in CNN when we find the contribution of those layers are barely noticeable. On the contrary, the layers of DenseNet are very narrow and it requires a very small set of feature maps compared to CNNs and its variations.

In standard feed-forward networks, the output of l^{th} layer connects with $(l+1)^{th}$ layer after executing a composite of several mathematical operations $h_l()$ This composite operation may include several convolution operation or pooling layers, batch normalization ,activation function and tuning parameters. The mathematical ex-

pression of this operation would be:

$$x_l = H_l\left(x_{l-1}\right) \tag{3.2}$$

ResNets extended the composite operations by including the feature of skip connections which generates a equation as below:

$$x_l = H_l\left(x_{l-1}\right) + x_{l-1} \tag{3.3}$$

DenseNets create some major differences with feed-forward and ResNets networks in the composite operations to accelerate its efficiency and credibility. Feed-forward and ResNets tend to sum the feature maps of a layer with the output from the previous layer. But DensNets deal with feature maps by concatenating them with each other which can be mathematically represented as follows:

$$x_l = H_l\left(\left[x_0, x_1, \ldots, x_{l-1}\right]\right) \tag{3.4}$$

In DenseNet, each layer has a connection with the other layers creating a dense network flow for which it is titled Densely Connected Convolutional Network or DenseNet. For a specific DenseNet architecture containing M layers can hold maximum $M(M+1)/2$ direct connections. Each layer captures the feature maps of all the immediate layers as input and its own feature maps are used as input for other successive layers as shown in figure 3.16.

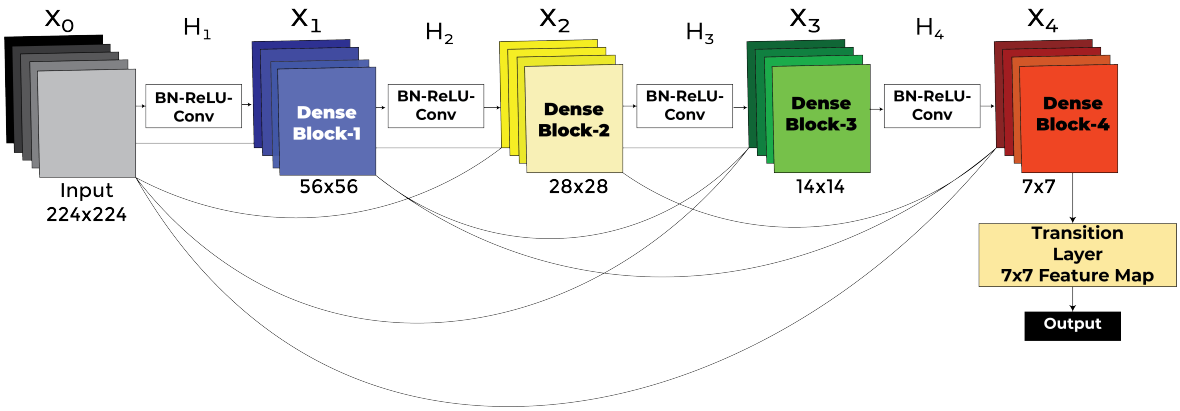


Figure 3.16: 5 -Layer DenseNet Architecture with 4 expansions

DenseNet is a combination of several DenseBlocks where the feature maps remain a constant shape but the filter quantity varies between blocks. The layers between blocks are known as transition layers. Since the main mechanism of DenseNet is to concatenate the feature maps, the dimension of each layer increases with the increasing number of data. We can generalize normal equation to find the feature maps for a specific layer with a given growth rate.

$$k_l = k_0 + k * (l-1) \tag{3.5}$$

The hyper parameter K denotes the growth rate which determines the quantity of new feature maps to be added to network each layer. K_0 is the initial number of feature maps for a respective layer and l denotes the layer for which we are going to evaluate the quantity of feature maps to be added. kl is the number of feature

maps used for the l^{th} layer. The compactness of the model can be further optimized by decreasing the number of feature maps.[22]

Down sampling the layers of networks becomes a significant methodology to change the shape of feature maps. To enable smoother downsampling, DenseNet partitions the network into several densely connected blocks as shown in Figure 3.17.

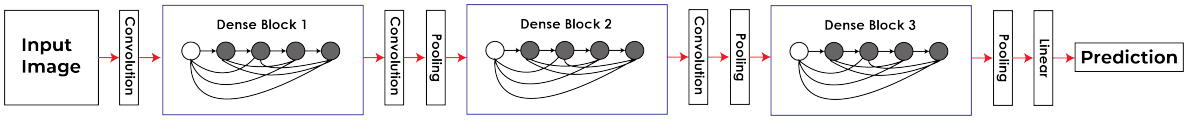


Figure 3.17: DenseNet with 3- Dense blocks

Layers between blocks are classified as transition layers and they are a combination of a batch-norm layer, 1×1 convolutional layer and a 2×2 average pooling layer. In a dense block, each layer of the block is connected with every other remaining layers with a constant size feature map. In Figure 3.18, we have an input feature map which is passed to Layer 0 with the green arrow and Layer 0 executed a non-linear transformation from which it generates some other features denoted by blue arrow and add those features with the green features. The resultant features then flow to Layer 1 as input execute its non-linear transformation mechanism to generate features for the processing of the next layer. The additional features of Layer 1 are denoted by a red arrow.[23]

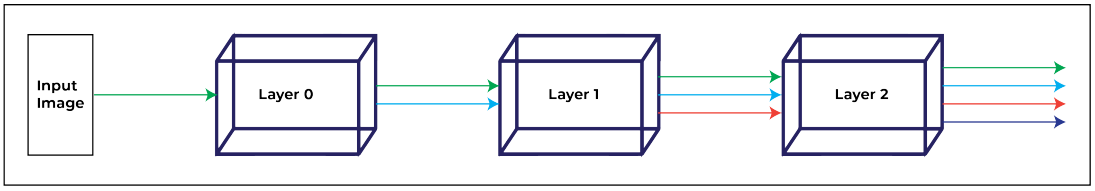


Figure 3.18: Interior view of a Dense Block

In the same way, Layer 2 takes input from Layer 1 which consists of resultant features from input image, Layer 0 and Layer 1 denoted consecutively by green, blue and red arrows. The number of features maps from Layer 2 can be generated with the help of growth rate K. As each layer produces a specific feature maps which is then concatenated with the output from the previous layer and passes to the next layer.

In this approach, the feature maps will become huge as the number of layers increases. The computational space for the huge computations may reduce the performance. To overcome the complexity, bottleneck layers have been introduced to make the computations easier and more efficient.

Bottleneck layer is mainly a 1×1 convolutional layer which can be placed before each 3×3 convolutional layer to decrease the number of feature maps to improve computational complexity as described in Figure 3.19. The bottleneck layer in DenseNet is distinguished by the fact that it creates a feature map of 4*growth rates using a

1×1 convolutional layer, which is then reduced to a feature map of a growth rate using a 3×3 convolution layers.[24]

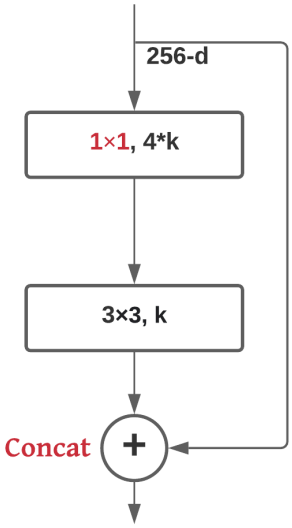


Figure 3.19: DenseNet Bottleneck Layer

Chapter 4

Implementation

4.0.1 Basic libraries for our model

We used Keras for developing our model. The main function that we used for implementing the model are given below

BatchNormalization()

It is used to normalize the input. It mainly converts the mean tends to zero and the standard deviation tends to one. In our model, we always need to batch normalize our input so that it maintains its mean to 0 and the standard deviation close to 1.

ReLu()

This is an activation function. This function will return max (0, x). If the value of x is greater than 0, it will simply return x. otherwise, this function will return 0.

Concatenate ()

This function concatenates all the input that are passed through the function by a list and returns a tensor with the concatenated value.

AvgPool2D ()

This function down samples the input by its height and weight. This averagePool2d () will downsample the input in all of its three channels with a window. The window will shift over the input and take the average of all the value that are found under the window. The stride defines how many units the window will shift to downsample the input.

MaxPool2D ()

This function down samples the input by its height and weight. This maxPool2d () will downsample the input in all of its three channels with a window. The window will shift over the input and take the maximum of all the value that are found under the window. The stride defines how many units the window will shift to downsample the input.

4.0.2 Implementing the model

We mainly applied DenseNet-121 in our image classification problem. First, we reshaped the input images to 224 by 224. We used 64 filters, strides of 2 in the first convolution layer. Then we used max pooling layer with pool size of 3, strides of 2 and keeping the same padding.

Then we created a general dense block which has two different convolution layers. The first convolution layer's kernel size is 1 and the second one's kernel size is 3. There are 4 times much filter in first convolution layer than the second convolution layer. The padding is same in both case. In each convolution layer, we first batch normalized the input features, applied a ReLu function, feed it into a 2D convolution layer. Besides, we concatenated both convolution layer.

Afterwards, we created a transition block. In our transition block, we first batch normalized the input features, applied ReLu function and feed it into a 2d convolution layer. Then we applied and average pooling layer with pool size of 2, strides of 2 and same padding.

Blockname	**Layername**	
Dense Block	1 x 1 convolution layer	Filters :128 Kernel :1 Strides :1 Padding :same
	3 x 3 convolution layer	Filters :32 Kernel :1 Strides :1 Padding :same
Transition block	1 x 1 convolution layer 2 x 2 average pool layer, strides = 2	

Table 4.1: Dense Block and Transition block

So far, we created two blocks. One is dense block and another one is transition block. In the same way, we repeated the dense blocks for six times, then concatenated the transition block for one time. Then we repeated the same steps with repetition of dense block twelve, twenty-four and sixteen times and added one transition layer after each repetition. Finally, we applied a global average pooling layer and fed it into a fully connected layer. The activation function of the last layer is softmax.

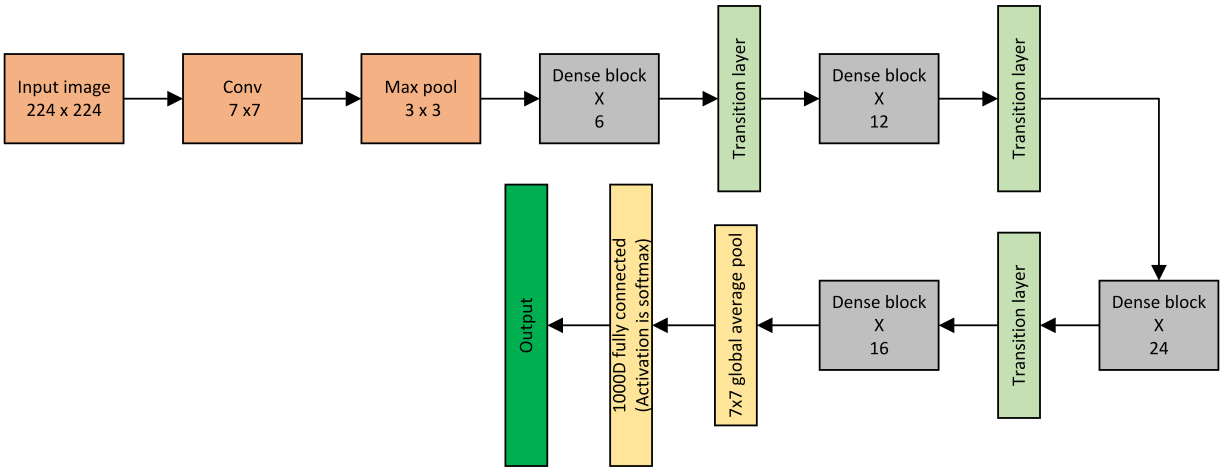


Figure 4.1: DenseNet-121 for the proposed Lesion Classification and Melanoma Detection Model

4.0.3 Training the model

For optimization, we used adam. Even though the DenseNet architecture uses SGD for optimization, we used adam since it gave us more accuracy in our validation data.
After creating the model, we trained our model. To avoid overfitting, we added early stop with patience of four. In addition, we also added reduce learning rate. This will decrease our learning rate if our model does not improve its accuracy.
During the running phase, we only saved the model if it gained a better accuracy then previous weight.

4.0.4 Predicting a single image

As our model were trained with the shape of $224 \times 224 \times 3$, we first reshaped the image to the same size . Then we normalized the image by dividing the image matrix with 1/255. Then we created a batch of image with batch size of one since keras always expect a batch of images for model prediction. As we are testing a single image, we are creating a batch with a size of one.Finally, we predicted the model that will return us a value in between 0 to 7 as our model will predict 7 different types of skin lesion/melanoma.

Chapter 5

Result Analysis

5.1 Performance Evaluation using Original Dataset

We trained our models in three different ways. At first, we took the total dataset and splitted the dataset into 20% for test data and 80% for train data. Then we trained our model. We got **80%** accuracy in this case.

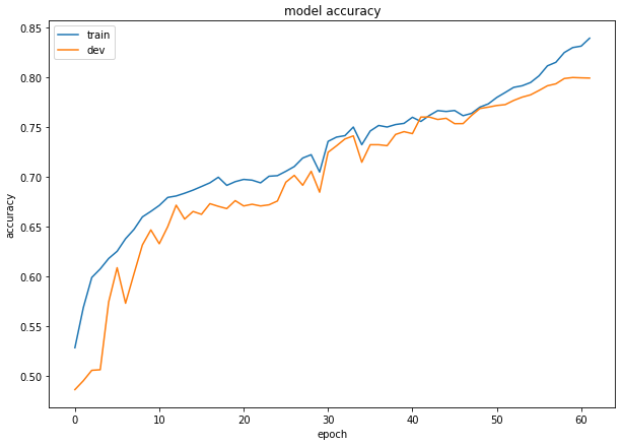


Figure 5.1: Accuracy of Full Dataset

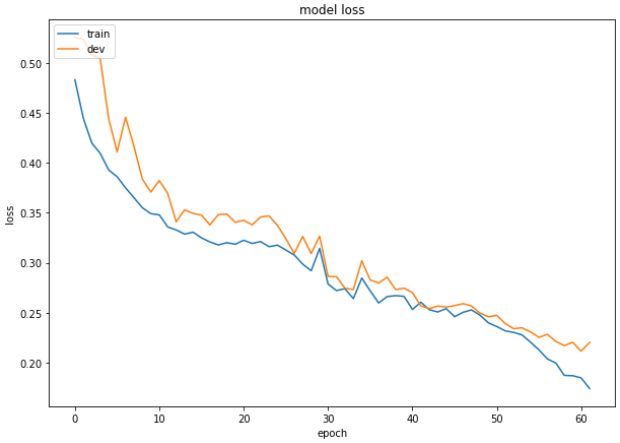


Figure 5.2: Loss of Full Dataset

In Figure 5.1, we can see that the training and validation accuracy line was close to each other for all the phases of training. However, there are some peak points where the validation accuracy is reduced for one or two epochs.
In Figure 5.2, the train loss and validation loss is reduced constantly. The loss of both train and test increased sometimes, but at the end of the training the validation loss was reduced to 0.25.
At the last moment of the training phase, the difference between training accuracy and validation accuracy were increasing. As the model did not improve for 50 consecutive epochs, we stopped our training of the system.

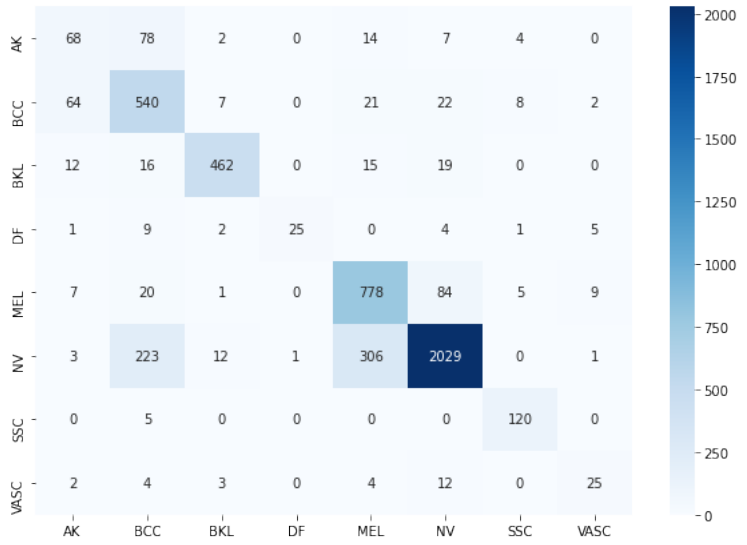


Figure 5.3: Confusion Matrix of Full Dataset

In Figure 5.3, the best prediction goes for the BKL class. The accuracy of this BKL class was 88%. The NV class has an accuracy of 79%. However, the VASC class has an accuracy of only 50% and the DF class has an accuracy of only 53%. The main reason behind this low accuracy is that both of these classes do not have enough images. The VASC class has only 203 images for training purposes and the DF class has only 192 images for training dataset. As the first approach was on an imbalanced dataset, the accuracy was lower of the classes that had comparatively less images.
Table 5.1 refers to an overall observation of performed classification on Original data which includes several statistical overview of our model:

Report	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**Acc.**
Precision	39.31	81.33	88.17	53.19	86.06	78.8	96.0	50.0	
Sensitivity	43.31	60.34	94.48	96.15	68.37	93.2	86.96	59.52	
Balanced Accuracy	70.58	78.68	96.56	97.86	82.58	87.14	93.43	79.51	79.95
F1-Score	41.21	69.28	91.22	68.49	76.2	85.4	91.26	54.35	
Specificity	97.86	97.02	98.64	99.56	96.79	81.07	99.9	99.9	

Table 5.1: Statistical Overview of Model on Original Data

5.2 Performance Evaluation using Balanced Dataset

As our dataset is imbalanced, we balanced the dataset by over sampling and down sampling. We took 375 images from each class for train data and 75 images from each class for test data. We again trained the model and got **84%**accuracy in validation data.

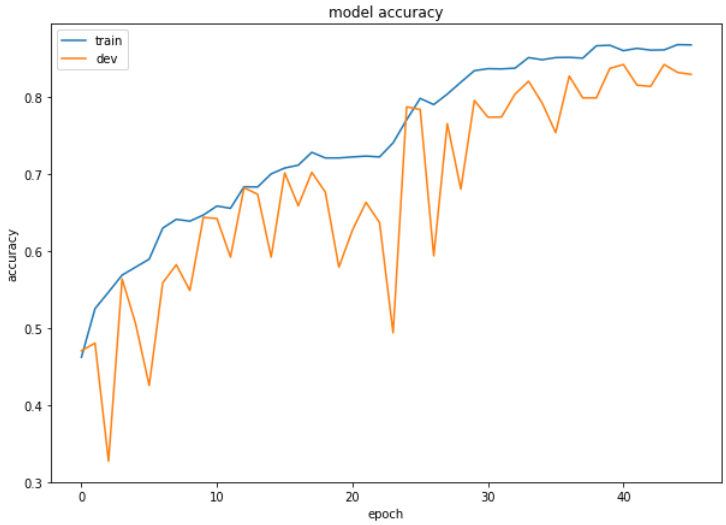


Figure 5.4: Accuracy of Balanced Dataset

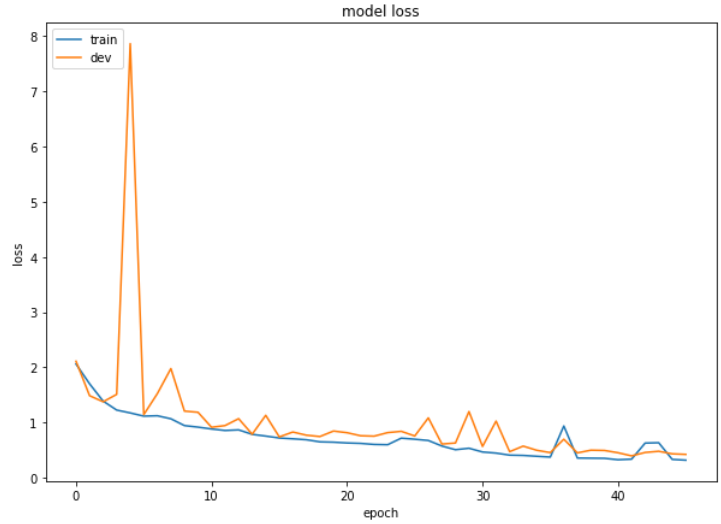


Figure 5.5: Loss of Balanced Dataset

In the above Figure 5.4, the training accuracy increased gradually, however the validation accuracy was inconsistent for some times between 10 to 25 epochs. After 25 epochs the validation rate increased gradually. The model stopped when the validation accuracy did not improve.Similarly the loss function was reduced as the train went on. However, at the second epoch the validation loss function increased very high but it eventually converged at the last epoch.

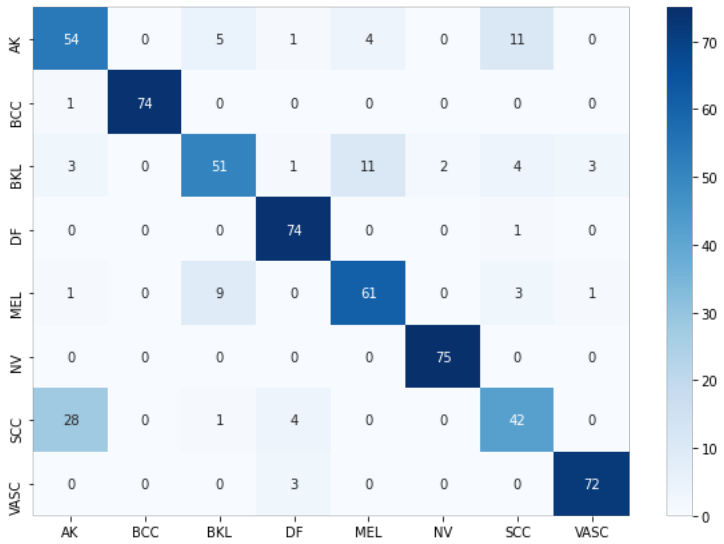


Figure 5.6: Confusion Matrix of Balanced Dataset

In Figure 5.6, All the images of NV were identified. The VASC class recovered the imbalanced problem in this case. Similarly, the DF class also gets better accuracy. However, the BKL and SCC class have a less accuracy than imbalanced data. This is because the random distribution of training and test data might be not perfect for this training.
Table 5.2 refers to an overall observation of performed classification on Balanced data which includes several statistical overview of our model:

Report	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**Acc.**
Precision	72.0	98.67	68.0	98.67	81.33	100.0	56.0	96.0	
Sensitivity	62.07	100.0	77.27	89.16	80.26	97.4	68.85	94.74	
Balanced Accuracy	78.99	99.9	86.39	94.48	88.79	98.7	81.36	97.08	83.83
F1-Score	66.67	99.33	72.34	93.67	80.79	98.68	61.76	95.37	
Specificity	95.91	99.81	95.51	99.81	97.33	100.0	93.88	99.43	

Table 5.2: Statistical Overview of Model on Balanced Data

5.3 Performance Evaluation using Normalized Dataset

Finally, we also did a binary classification based on the MEL class. Here we took 4522 images for both training and testing purposes. In this case, we got **89%** accuracy.

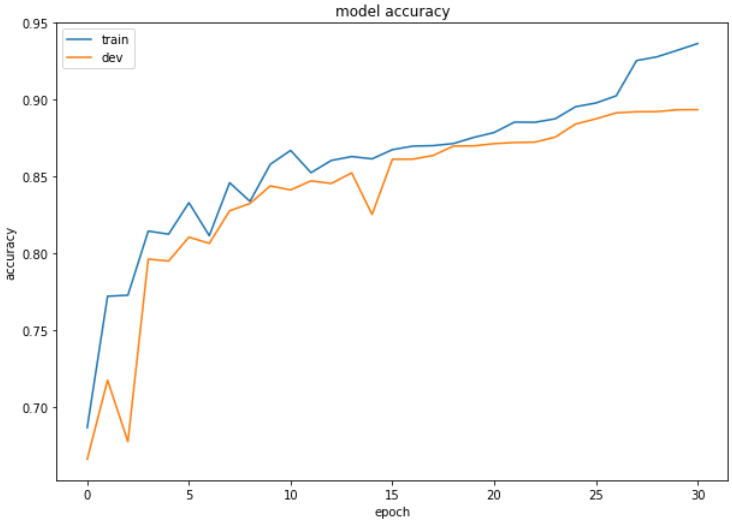


Figure 5.7: Accuracy of Binary classification

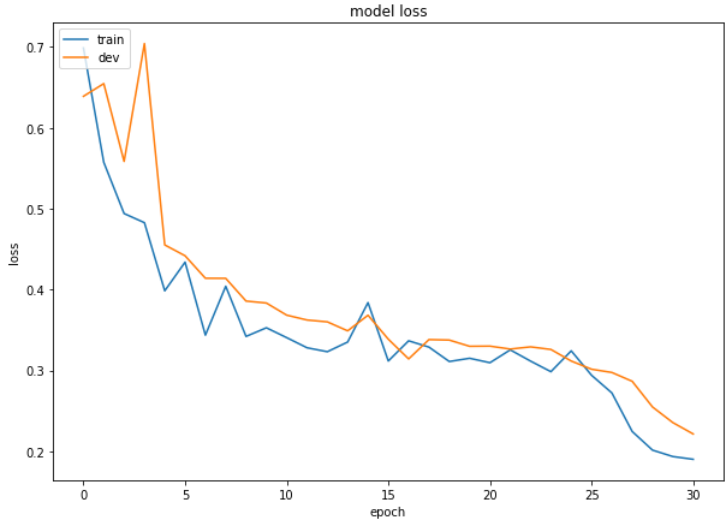


Figure 5.8: Loss of Binary Classification

In Figure 5.7, the training and validation accuracy were balanced for 25 epochs. The accuracy was improved till 25 epochs, then it stopped improving for five epochs. Then the model stopped running. The loss function also converged near to 0.2 at the end of the training.

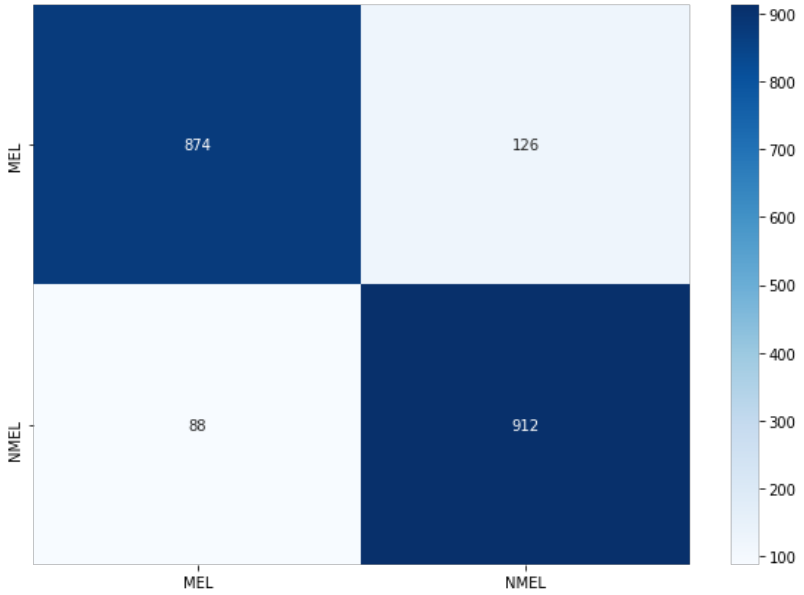


Figure 5.9: Confusion Matrix of Binary Classification

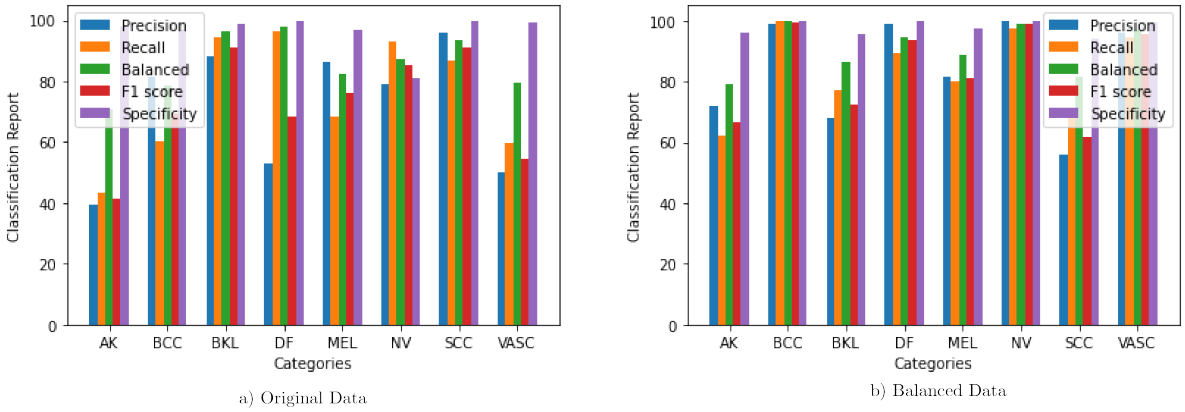
In figure 5.9, 912 images were classified as NMEL correctly and 874 images were classified as MEL . However, 88 images were incorrectly classified as MEL and 126 were incorrectly classified as NMEL.
Table 5.3 refers to an overall observation of performed classification on Normalized data which includes several statistical overview of our model:

Report	**MEL**	**Non-MEL**	**Acc.**
Precision	87.4	91.2	
Sensitivity	90.85	87.86	
Balanced Accuracy	89.36	89.36	89.3
F1-Score	89.09	89.5	
Specificity	87.86	90.85	

Table 5.3: Statistical Overview of Model on Normalized Data

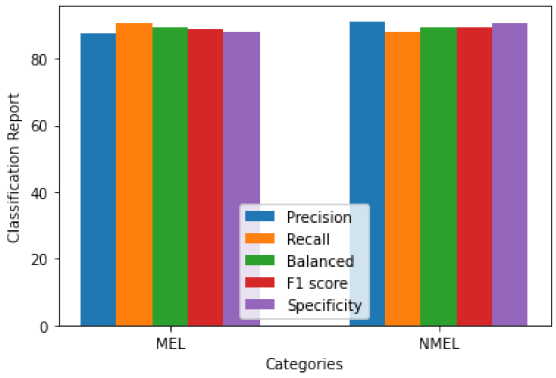
5.3.1 Observation of Classification Report

Figure 5.10 refers to the classification report of 3 different aspects on all the classes to visualize the model performance. We have plotted individual bar chart for each section describing different accuracy parameters to compare the best classification. In the figure , X-axis represents the categories of all classes in pre-processed data and Y-axis represents the report of different approaches of theoretical accuracy determination .



a) Original Data

b) Balanced Data



c) Normalized Data

Figure 5.10: Comparison among a) Original Data b) Balanced Data c) Normalized Data

5.3.2 Observation and Comparison of Accuracy

Previously some reseach work conducted using ISIC-2019 Dataset.In one research work, they ensamble some model and make their own model.They use a loss balancing strategy to solve the common problem of extreme class imbalance in skin lesion classification.They use a variety of EfficientNets, with input cropping strategy and resolution.And Finally,they got balanced accuracy of 74.2% using five-fold cross-validation.[25]

Models	**Accuracy**
Densenet(Original Dataset)	80%
Densenet(Balanced Dataset)	84%
Ensemble[25]	74.2%

Table 5.4: Comparison among the Accuracy of Observed Models

Chapter 6

Conclusion and Future work

In this paper, we worked with ISIC 2019 Challenge dataset containing lesion images classified into 8 different classes which are highly imbalanced to evaluate. We worked on both the classification of lesion images and melanoma detection with the help of DenseNet-121.As the dataset is imbalanced, we preprocessed the dataset by oversampling and under sampling to extract better accuracy.We researched on different CNN based models to choose the best model which will perform best in the classification of lesion images.Besides, we generalized our dataset into two classes titled MEL and NON-MEL to perform a binary classification.
The dataset was not separated into train and test segment for which we had randomly selected lesion images for model training and validation.Furthermore, some photos have not been adequately pre-processed.Many photos have a black circle around them, which makes picture categorization difficult.It would be beneficial to enhance the accuracy if the photos were sufficiently preprocessed.
The objective of our image is to classify image lesions and melanoma detection.Melanoma cancer is becoming more common, posing a challenge to the medical system's ability to cope with the rising number of patients.The main factors for the development of melanoma in human skin remain unclear.However, experts believe that exposure to UV light emitted by the sun increases the chance of acquiring melanoma.Skin cancer or more specifically melanoma cancer can be cured if it is detected in early stage of developing cells. So we want to classify melanoma with more accuracy in the future with the help of lesion image classification which may predict melanoma in short time. Furthermore, we wish to extract features from the DenseNet's second last layer and feed them to a variety of machine learning models, such as SVM, to improve accuracy in a more comprehensive and efficient approach.

Bibliography

[1] Howard W Rogers et al. "Incidence estimate of nonmelanoma skin cancer (keratinocyte carcinomas) in the US population, 2012". In: *JAMA dermatology* 151.10 (2015), pp. 1081–1086.

[2] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *nature* 542.7639 (2017), pp. 115–118.

[3] Rina Refianti, Achmad Benny Mutiara, and R Poetri Priyandini. "Classification of melanoma skin cancer using convolutional neural network". In: *IJACSA* 10.3 (2019), pp. 409–417.

[4] Syed Md Akram Hussain. "Comprehensive update on cancer scenario of Bangladesh". In: *South Asian journal of cancer* 2.4 (2013), p. 279.

[5] Gabriella Liszkay et al. "Changing Trends in Melanoma Incidence and Decreasing Melanoma Mortality in Hungary Between 2011 and 2019: A Nationwide Epidemiological Study". In: *Frontiers in Oncology* 10 (2021), p. 3236. ISSN: 2234-943X. DOI: 10.3389/fonc.2020.612459. URL: https://www.frontiersin.org/article/10.3389/fonc.2020.612459.

[6] Rina Refianti, Achmad Benny Mutiara, and Rachmadinna Poetri Priyandini. "Classification of Melanoma Skin Cancer using Convolutional Neural Network". In: *International Journal of Advanced Computer Science and Applications* 10.3 (2019). DOI: 10.14569/IJACSA.2019.0100353. URL: http://dx.doi.org/10.14569/IJACSA.2019.0100353.

[7] M. Q. Khan et al. "Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer". In: *IEEE Access* 7 (2019), pp. 90132–90144. DOI: 10.1109/ACCESS.2019.2926837.

[8] Abhinav Sagar and J Dheeba. "Convolutional Neural Networks for Classifying Melanoma Images". In: *bioRxiv* (2020). DOI: 10.1101/2020.05.22.110973. eprint: https://www.biorxiv.org/content/early/2020/05/23/2020.05.22.110973.1.full.pdf. URL: https://www.biorxiv.org/content/early/2020/05/23/2020.05.22.110973.1.

[9] E. Nasr-Esfahani et al. "Melanoma detection by analysis of clinical images using convolutional neural network". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016, pp. 1373–1376. DOI: 10.1109/EMBC.2016.7590963.

[10] Yuexiang Li and Linlin Shen. "Skin Lesion Analysis towards Melanoma Detection Using Deep Learning Network". In: *Sensors* 18.2 (2018). ISSN: 1424-8220. DOI: 10.3390/s18020556. URL: https://www.mdpi.com/1424-8220/18/2/556.

[11] Fabio Perez, Sandra Avila, and Eduardo Valle. "Solo or Ensemble? Choosing a CNN Architecture for Melanoma Classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[12] Julia K. Winkler et al. "Association Between Surgical Skin Markings in Dermoscopic Images and Diagnostic Performance of a Deep Learning Convolutional Neural Network for Melanoma Recognition". In: *JAMA Dermatology* 155.10 (Oct. 2019), pp. 1135–1141. ISSN: 2168-6068. DOI: 10.1001/jamadermatol.2019.1735. eprint: https://jamanetwork.com/journals/jamadermatology/articlepdf/2740808/jamadermatology_winkler_2019_oi_190038.pdf. URL: https://doi.org/10.1001/jamadermatol.2019.1735.

[13] Titus J. Brinker et al. "Deep neural networks are superior to dermatologists in melanoma image classification". In: *European Journal of Cancer* 119 (2019), pp. 11–17. ISSN: 0959-8049. DOI: https://doi.org/10.1016/j.ejca.2019.05.023. URL: https://www.sciencedirect.com/science/article/pii/S0959804919303491.

[14] Achim Hekler et al. "Pathologist-level classification of histopathological melanoma images with deep neural networks". In: *European Journal of Cancer* 115 (2019), pp. 79–83. ISSN: 0959-8049. DOI: https://doi.org/10.1016/j.ejca.2019.04.021. URL: https://www.sciencedirect.com/science/article/pii/S0959804919302758.

[15] Philipp Tschandl. *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. Version V3. 2018. DOI: 10.7910/DVN/DBW86T. URL: https://doi.org/10.7910/DVN/DBW86T.

[16] Marc Combalia et al. *BCN20000: Dermoscopic Lesions in the Wild*. 2019. arXiv: 1908.02288 [eess.IV].

[17] Noel C. F. Codella et al. *Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)*. 2018. arXiv: 1710.05006 [cs.CV].

[18] Rasool Iranpoor et al. "Skin lesion segmentation using convolutional neural networks with improved U-Net architecture". In: *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. 2020, pp. 1–5. DOI: 10.1109/ICSPIS51611.2020.9349577.

[19] Zbancioc Marius-Dan and Feraru Silvia Monica. "Emotion Recognition for Romanian Language using Deep Learning with DL-CNN convolutional layers". In: *2020 International Conference and Exposition on Electrical And Power Engineering (EPE)*. 2020, pp. 762–765. DOI: 10.1109/EPE50722.2020.9305543.

[20] Maria Pavlova. "Comparison of Activation Functions in Convolution Neural Network". In: *2020 28th National Conference with International Participation (TELECOM)*. 2020, pp. 65–67. DOI: 10.1109/TELECOM50385.2020.9299559.

[21] Adekanmi A. Adegun and Serestina Viriri. "FCN-Based DenseNet Framework for Automated Detection and Classification of Skin Lesions in Dermoscopy Images". In: *IEEE Access* 8 (2020), pp. 150377–150396. DOI: 10.1109/ACCESS.2020.3016651.

[22] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[23] Yiming Zhang and Chong Wang. “SIIM-ISIC Melanoma Classification With DenseNet”. In: *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. 2021, pp. 14–17. DOI: 10.1109/ICBAIE52039.2021.9389983.

[24] Adar Elad et al. “Direct Validation of the Information Bottleneck Principle for Deep Nets”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 758–762. DOI: 10.1109/ICCVW.2019.00099.

[25] Nils Gessert et al. *Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data*. 2019. arXiv: 1910.03910 [cs.CV].

Appendix A

Appendix

A.1 Questions from the panel members:

Md.Saiful Islam Sir

- Is the size of your dataset small? How did you make it(dataset) balance?

Shahnewaj Sir

- Did you use any pre-processing mechanism to make the imbalanced dataset balance?

Dewan Ziaul Karim Sir

- Could you please explain why did you use batch normalization?
- Do you think you would get better accuracy if you use SVM in your model?

Md. Ashraful Alam, PhD Sir

- What was your individual contribution in your thesis?