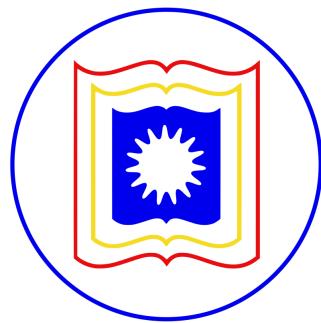


# Detection and Classification of Skin Cancer Using Convolutional Neural Networks



B.Sc. Engineering Project

A Project paper submitted for partial fulfilment of the requirements for  
the Degree of Bachelor of Science in Computer Science and Engineering

Submitted By

Anik Modak

ID: 1710476141

Session: 2016-17

Supervised By

Shamim Ahmad

Professor

---

Department of Computer Science and Engineering  
University of Rajshahi

# Abstract

Skin Cancer is the most predominant type of cancer. It occurs mainly due to the exposure of ultraviolet radiation emitted from the Sun. It is one of the fastest-growing diseases in the world. According to the Skin Cancer Foundation, the global incidence of skin cancer continues to increase. If a patient with a pigmented lesion could be identified as someone with or at a risk of developing skin cancer, then measures could be taken right way to lower their risk or destroy the cancer if developed at an early stage. If it is not diagnosed and treated early on, it is likely to spread to other areas of the body. In the recent past few years, some approaches have been proposed to develop automated skin cancer detection. In this context, our aim is to propose a method that uses a Convolution Neural Network (CNN) to accurately classify pigmented skin lesions in dermoscopic images to detect the malignant skin lesions. The MNIST HAM-10000 datasets are used in this work. In order to reduce the data imbalance in the dataset, undersampling, oversampling and class weights technique are used. Four different CNN models are applied to analyze and predict the pigmented lesions classes. The used models are VGG16, MobileNetV2, ResNet152V2 and finally, our own CNN architecture ÄM-12: Experimental results showed that performance of AM-12 was comparable with VGG16, MobileNetV2 and ResNet152V2, and closely competing with our tweaked VGG16 model which topped in all our scoring metrics, noticeably scoring an 84.87% weighted classification accuracy.

## Declaration of Academic Integrity

I hereby declare and confirm that the study, “*Detection and Classification of Skin Cancer Using Convolutional Neural Networks*” is entirely the result of my own work except where otherwise indicated.

April 17, 2022

---

Anik Modak

# Acknowledgement

Regarding the outcome of my project, I would like to express deepest sense of gratitude and respect to my supervisor **Shamim Ahmed**, Professor, Department of Computer Science and Engineering, University of Rajshahi for the complete supervision, advice, encouragement and continued support of my project. His patience, motivation, collaboration and interaction were key factors in the success of my project. Without his dedicate involvement in every step throughout the process, this project would have never been accomplished.

I would like to thank honorable teacher **Sangeeta Biswas**, Assosiate Professor, Department of Computer Science and Engineering, Rajshahi University for his valuable suggestions, inputs, and guideline to complete my project successfully.

I also like to thank all my friends for their encouragement and supports to complete my project successfully. I want to express my gratitude to my family for always supporting me and which have always been my source of motivation and happiness for me.

Finally, I would like to thank God from the deepest of my heart for making me able to finish this project work.

# Contents

<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>6</b>
<b>2 Literature Review</b>	<b>8</b>
<b>3 Background</b>	<b>10</b>
3.1 Skin Cancer . . . . .	10
3.2 State of the Art . . . . .	11
3.2.1 Deep Learning . . . . .	11
3.2.2 Deep Neural Network(DNN) . . . . .	12
3.2.3 Convolutional neural network (CNN) . . . . .	12
3.3 Transfer Learning . . . . .	18
3.4 Performance Metrics . . . . .	19
<b>4 Proposed Methodology</b>	<b>20</b>
4.1 Input images . . . . .	20
4.2 Pre-Processing . . . . .	20
4.3 Models . . . . .	21
4.3.1 AM-12 . . . . .	21
4.3.2 VGG16 . . . . .	21
4.3.3 MobileNetV2 . . . . .	21
4.3.4 ResNet152V2 . . . . .	21
4.4 Feature Extraction . . . . .	22
4.5 Classification . . . . .	22
<b>5 Dataset</b>	<b>23</b>
<b>6 Data Pre-Processing</b>	<b>29</b>
6.1 Directory Structure . . . . .	29

6.2	Data Cleansing . . . . .	29
6.3	Data Splitting . . . . .	30
6.4	Data Normalization . . . . .	30
6.5	Data Augmentation . . . . .	30
6.6	Data Resizing . . . . .	30
<b>7</b>	<b>Implementation</b>	<b>31</b>
7.1	Framework . . . . .	31
7.2	Models . . . . .	31
7.2.1	AM-12 . . . . .	31
7.2.2	VGG16 . . . . .	33
7.2.3	MobileNetV2 . . . . .	33
7.2.4	ResNet152V2 . . . . .	33
7.3	Hyperparameters . . . . .	34
7.4	Balancing Datasets . . . . .	34
<b>8</b>	<b>Result And Discussion</b>	<b>36</b>
8.1	Training the models with imbalanced dataset . . . . .	36
8.2	Training the models with balanced dataset . . . . .	38
8.2.1	Oversampling Technique . . . . .	38
8.2.2	Undersampling Technique . . . . .	39
8.2.3	Class Weights Technique . . . . .	40
8.3	Models Performance Comparisons for Testing Set . . . . .	42
<b>9</b>	<b>Conclusion</b>	<b>44</b>
<b>10</b>	<b>Future Scope and Limitations</b>	<b>45</b>
10.1	Future Scope . . . . .	45
10.2	Limitations . . . . .	45

# Chapter 1

## Introduction

SKIN cancer is defined as the abnormal growth of skin cells that is most often developed on skin exposed to the sun, primarily the areas of the scalp, face, lips, ears, neck, chest, arms, hands, and legs. It affects people of all skin tones, including those with darker complexions. It represents a major public health concern as it affects many people around the world, where it is the most common of all cancer types [1].

Cancer analysis is an important aspect of medicine. The healthcare industry collects an enormous amount of healthcare data from patients through medical studies, clinic appointments and hospital visits. This data can be analyzed to discover and realize hidden figures and statistics about an illness that could offer a breakthrough in care, treatment, or diagnosis of patients. Accurate analysis of data, such as malignant pigmented lesions, benefits early skin cancer detection, which in turn could lower the number of patients dying, as measures could be taken right away to lower the patient's risk or destroy the cancer. Malignant pigment lesions include melanoma, the deadliest type of skin cancer, basal cell carcinoma, and some vascular lesions. Automated analysis could be done via machine learning (ML), which would create a system and framework in the medical field that would help in providing contextual relevance, improving clinical reliability, helping physicians communicate objectively, reducing errors related to human fatigue, decreasing mortality rates, diminishing medical costs, and identifying diseases more readily [2]. A machine learning algorithm that can classify both malignant and benign pigmented skin lesions is a step in the right direction towards providing these benefits.

It is one of the most deathful of all the cancers that has spread worldwide. In 2018, melanoma accounted for about 22 per cent of skin cancer diagnoses, and non-melanoma tumours accounted for about 78 percent of skin cancer diagnoses.

According to the Skin Cancer Foundation, the global incidence of skin cancer continues to increase [3]. In 2019, it is estimated that 192,310 cases of melanoma will be diagnosed in the United States [4]. In 2018, there were over one million cases of skin cancer worldwide [5].

According to the latest WHO data published in 2018 Skin Cancers Deaths in Bangladesh reached 301 of total deaths [6].

Early diagnosis of skin cancer is a cornerstone to improving outcomes and is correlated with 99% overall survival (OS). However, once disease progresses beyond the skin, survival is poor [7].

In this project, Convolutional Neural Networks (CNN) were deployed to accurately classify pigmented skin lesions in dermoscopic images to detect the malignant skin lesions as early as possible. A CNN is a class of deep neural networks that use convolution in place of general matrix multiplication in at least one of their layers. It excels in analyzing visual imagery as they are fully connected (FC) feed forward neural networks that reduce the number of parameters very efficiently without losing out on the quality of models. The data set used in this paper is “The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions” [8], and is described in the Chapter 5. It is very common for data sets in the medical field to have major class imbalances, and this data set was no different. Four different CNN models were employed to analyze and predict the pigmented lesions classes. In order to reduce the data imbalance in the dataset, undersampling, oversampling and class weights technique are used. The used models were our own CNN architecture AM-12, VGG16, MobileNet and ResNet152V2. Experimental results showed that performance of AM-12 was comparable with VGG16, MobileNetV2 and ResNet152V2, and closely competing with our tweaked VGG16 model which topped in all our scoring metrics, noticeably scoring an 84.87% weighted classification accuracy.

## Chapter 2

# Literature Review

CNNs have been used frequently in the field of medical image processing image classification and so on [5]. CNNs have already shown inspiring outcomes in the domain of microscopic images classification, such as: human epithelial 2 cell image classification [9] and skin cancer detection [7]. There have been multiple AI studies on lesion diagnosis using dermoscopic skin lesion datasets, which are listed below:

1. Codella et al. [10] developed an ensemble of deep learning algorithms on the ISIC-2016 dataset and compared the performance of this network with 8 dermatologists for the classification of 100 skin lesions as benign or malignant. The ensemble method outperformed the average performance of dermatologists by achieving an accuracy of 76% and specificity of 62% versus 70.5% and 59% achieved by dermatologists.
2. Haenssle et al. [11] trained a deep learning method InceptionV4 on a large dermoscopic dataset consisting of more than 100,000 benign lesions and melanoma images and compared the performance of a deep learning method with 58 dermatologists. On the test set of 100 cases (75 benign lesions and 25 melanoma cases), dermatologists had an average sensitivity of 86.6% and specificity of 71.3%.
3. Brinker et al. [12] compared the performance of 157 board-certified dermatologists at 12 German university hospitals with a deep learning method (ResNet50) for 100 dermoscopic images (MClass-D) consisting of 80 nevi and 20 melanoma cases. Dermatologists achieved an overall sensitivity of 74.1%, and specificity of 60.0% on the dermoscopic dataset whereas a deep learning method achieved a specificity of 69.2% and a sensitivity of 84.2%.
4. Tschandl et al. [13] used popular deep learning architectures known as InceptionV3 and ResNet50 on a combined dataset of 7895 dermoscopic and 5829 close-up lesion images for diagnosis of non-pigmented skin cancers. The performance is compared with 95 dermatologists

divided into three groups based on experience. The deep learning algorithms achieved accuracy on par with human experts and exceeded the human groups with beginner and intermediate raters.

5. Maron et al. [14] compared the sensitivity and specificity of a deep learning method (ResNet50) with 112 German dermatologists for multiclass classification of skin lesions which includes nevi, melanoma, benign keratosis, BCC, and SCC (also solar keratosis and intraepithelial carcinoma). The deep learning method outperformed dermatologists at a significant level ( $p < 0.001$ ).
6. Rishu Garg et al. [15] used Decision Support System on a combined dataset of MNIST HAM-10,000 containing dermoscopy images. Dermatologists achieved weighted average precision of 0.88, a weighted recall average of 0.74, and a weighted F1 score of 0.77.

Mostly existing research considers binary classification, whether the cancer is melanomous or not and small work is performed for classification of general images. but their result is not very optimal. The existing algorithms used for the detection and classification of skin cancer disease uses machine learning and neural network algorithms.

# **Chapter 3**

## **Background**

### **3.1 Skin Cancer**

Cancer refers to the uncontrolled and abnormal growth of the specific type of cells in any part of the body. These cells could then spread to other body parts as well thus damaging them.

There are various types of cancers, few of them are as follows: (1) Carcinoma is a type of cancer that occurs in the skin and other parts of the body, such as the pancreas, lungs, and breasts. (2) Sarcoma is a type of cancer usually appears in the bone and muscles and even at the connective tissues of the body. (3) Leukemia occurs in the blood which forms tissues like bone marrow and leads to irregular blood cells. (4) Lymphoma occurs in the cells of the immune system. (5) The central nervous system cancer appears in the spinal cord and brain. (6) Skin cancer, which starts in the cell and then forms a pigment in the skin and finally spreads to other organs [16].

Skin cancer is one such silently killing disease. There could be many reasons contributing to skin cancer, the most common one being exposure to UV radiations from the sun. Due to ozone layer depletion, the UV exposure has increased drastically over the last few years and so has the possibility of developing skin cancer. Other reasons could be genetic inclusion, light skin thickness or lack of immunity. Majorly, skin cancer can be of two types namely melanoma skin cancer (MSC) and non-melanoma skin cancer (NMSC). Melanoma is a dangerous and very aggressive class of skin cancer which could be highly life threatening. Non-melanoma type majorly includes basal cell carcinoma (BCC) and squamous cell carcinoma (SCC). Non-melanoma types are generally not life threatening and can be cured by proper treatment methods such as a combination of surgery and chemotherapy. Early treatment of melanoma can also help in proper recovery and thus decreasing the mortality due to skin cancer [17].

## 3.2 State of the Art

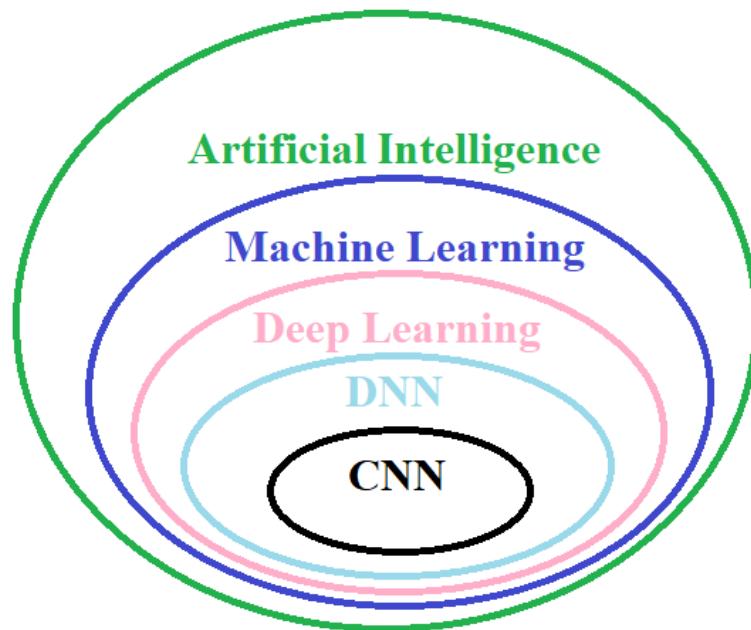


Figure 3.1: State of the art hierarchy

Artificial Intelligence(AI) is a super set, Machine Learning is a subset of Artificial Intelligence(AI), Deep Learning is a subset of Machine Learning.

### 3.2.1 Deep Learning

Deep Learning is a way to make a soft version of the functional part of brain in order to do tasks by computer which are time consuming and tedious for humans. It is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data.

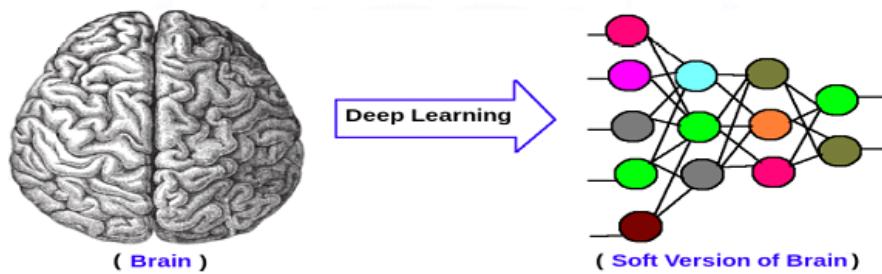


Figure 3.2: Deep Learning

### 3.2.2 Deep Neural Network(DNN)

Deep neural network (DNN) is one type of artificial neural network (ANN) with multiple hidden layers between the input layer and output layer. DNN models were originally inspired by neurobiology. A neuron (biological) takes numerous signals through the synapses contacting its dendrites and transmit a single stream of action potentials out by its axon. By categorizing its input patterns decrease the complexity of numerous inputs. Inspired by this system, ANN models are composed of units that integrate multiple inputs and generate a single output. These artificial neurons (AN) are collected into layers, and the outputs of one layer of neuron becoming the inputs of the next layer neuron in the sequence. There are three types of DNN are very popular nowadays:

1. Multi-Layer Perceptrons (MLP)
2. Convolutional Neural Networks (CNN)
3. Recurrent Neural Networks (RNN)

### 3.2.3 Convolutional neural network (CNN)

Convolutional neural networks were inspired by biological processes. In these, the connectivity pattern between neurons of a network resembles the organization of the animal visual cortex. The response of an individual cortical neuron in a restricted region of the visual field is known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. Each convolution operation outputs a numerical value by applying a filter (i.e., a matrix of weights) to a sub-region of an image. A pooling operation, on the other hand, is performed to merge semantically similar features into one, thus reducing the size of the image (a.k.a., sub-sampling) [18].

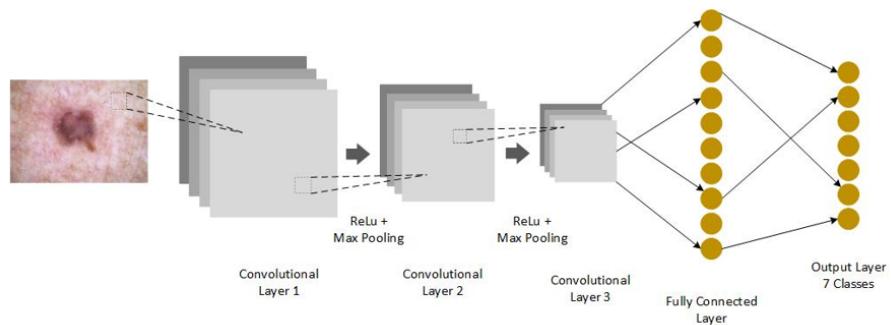


Figure 3.3: Convolutional Neural Networks (CNN) Architecture (collected from isysrg).

### Convolution Layer

In CNN the main layer is convolutional layer. In this layer the result of the output layer is gotten from the input by filtering in specific condition. This layer is constructed by the neurons which is in the shape of cubical blocks. A grayscale image is nothing more than a matrix of pixel values with a single plane, whereas an RGB image is nothing more than a matrix of pixel values with three planes. Take a look at this illustration to learn more.

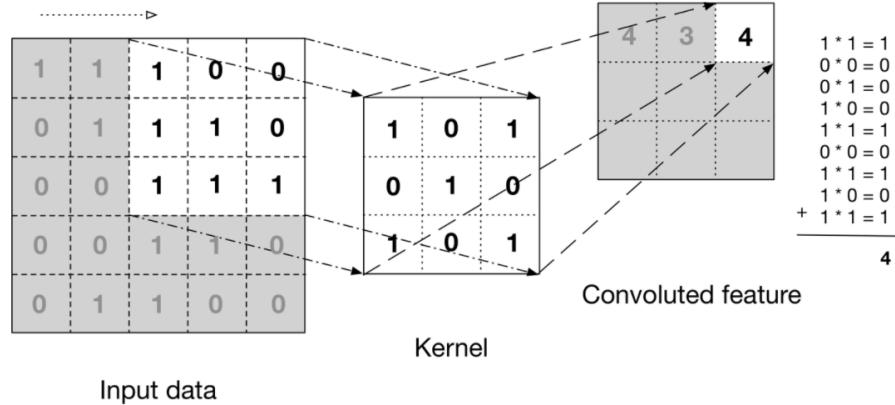


Figure 3.4: Convolution on gray scale image (collected from analyticsvidhya).

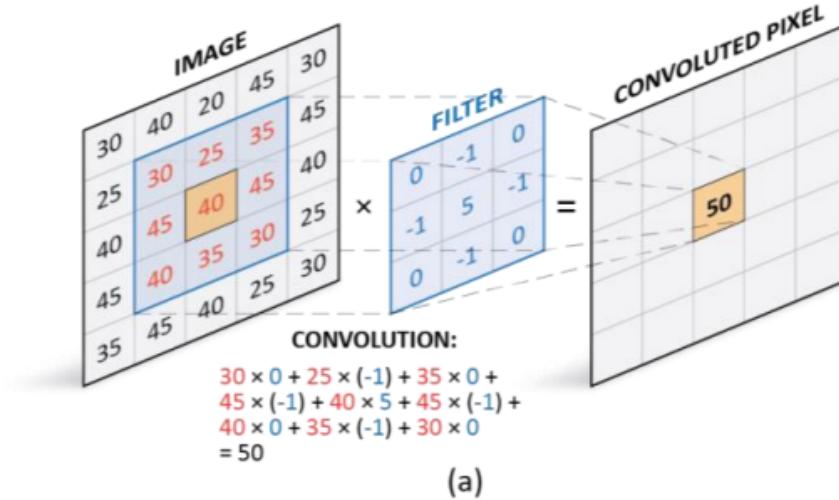


Figure 3.5: Convolution on RGB image (collected from analyticsvidhya).

### Pooling Layer

When the photos are too huge, the pooling layers portion would lower the number of parameters. Spatial pooling, also known as subsampling or downsampling, decreases the dimensionality of each map while preserving crucial data. Different types of spatial pooling exist :

1. Max Pooling
2. Average Pooling

The largest element from the corrected feature map is used in max pooling. The average pooling could be calculated by taking the average of all element.

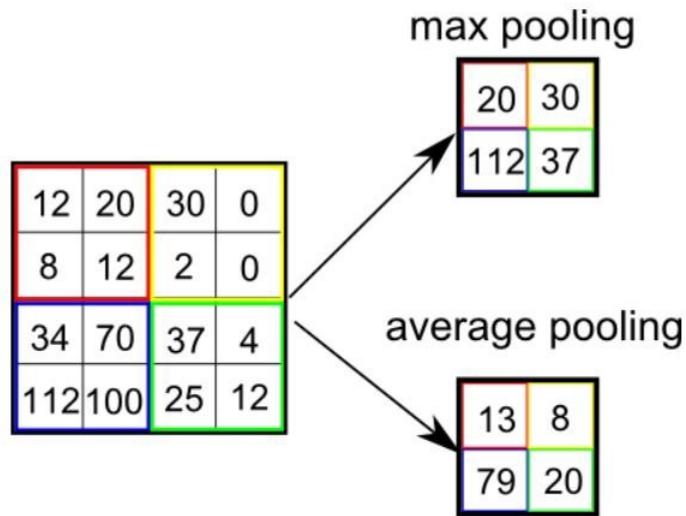


Figure 3.6: Example of pooling operation performed with a  $2 \times 2$  filter (collected from analyticsvidhya).

### Strides

The number of pixels shifted over the input matrix is referred to as the stride. When the stride is set to 1, the filters are moved one pixel at a time. We shift the filters two pixels at a time when the stride is two, and so on. Convolution would function with a stride of 2 as seen in the diagram below.

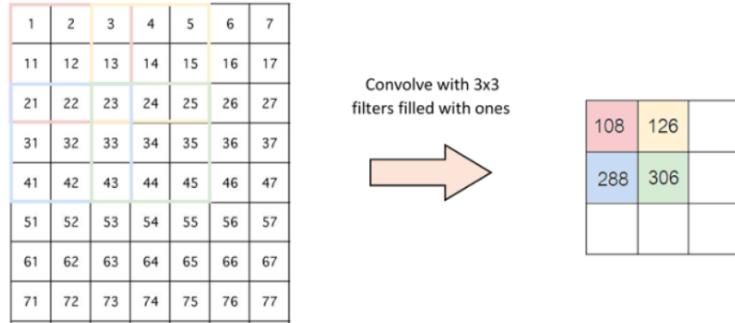


Figure 3.7: Stride of 2 pixels

### Padding

Sometimes filter does not fit perfectly the input image. We have two options:

1. Same Padding: Pad the picture with zeros (zero-padding) so that it fits
2. Valid Padding: Remove the portion of the image where the filter didn't work. This is known as valid padding, and it maintains only the image's valid parts.

### Fully Connected layers

The final layer of a convolutional neural network (CNN) is a fully connected layer that is formed from the attachment of all preceding neurons. It reduces the spatial information as it is fully connected like in artificial neural network. It contains neurons beginning at input neurons to the output neurons skin cancer detection using CNN.

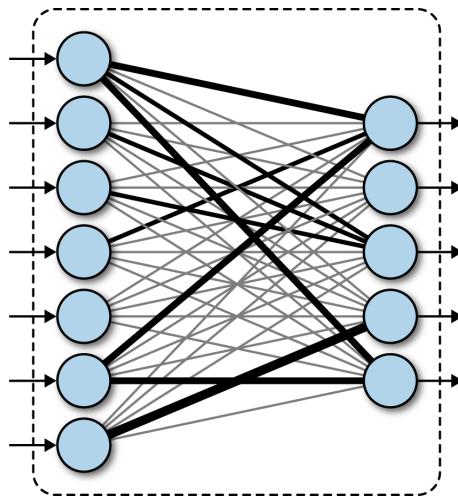


Figure 3.8: A fully connected layer in a deep network (collected from O'reilly).

### Activation Function

An Activation Function(AF) are function which decides whether a neuron should be activated or not. This means that during the prediction phase, it will determine whether the neuron's input to the network is essential or not. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).The primary role of the Activation Function is to transform the summed weighted input from the node into an output value to be fed to the next hidden layer or as output as shown in Fig 3.9. These AFs are often referred to as a transfer function in some literature.

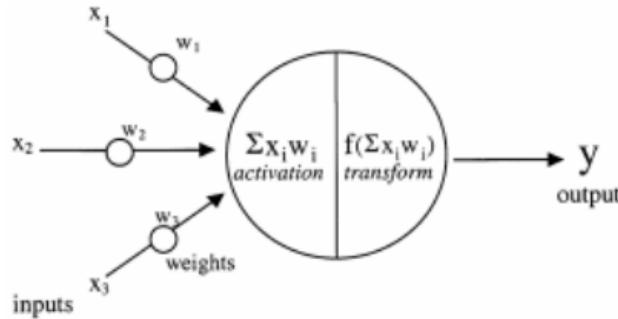


Figure 3.9: Neural Networks Architecture (collected from [19])

For a linear model, a linear mapping of an input function to an output, as performed in the hidden layers before the final prediction of class score for each label is given by the affine transformation in most cases [20]. The input vectors  $x$  transformation is given by

$$f(x) = w^T x + b \quad (3.1)$$

where  $x$  = input,  $w$  = weights,  $b$  = biases.

There are different type of non-linear activation function. Given some below:

#### Rectified Linear Unit(ReLU)

The Rectified Linear Unit is the most commonly used activation function in deep learning models. The function returns 0 if it receives any negative input, but for any positive value  $x$  it returns that value back. So it can be written as:

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (3.2)$$

The gradient value is 0 on the negative side of the graph Fig: 3.10. As a result, some neurons' weights and biases are not updated during the backpropagation process.

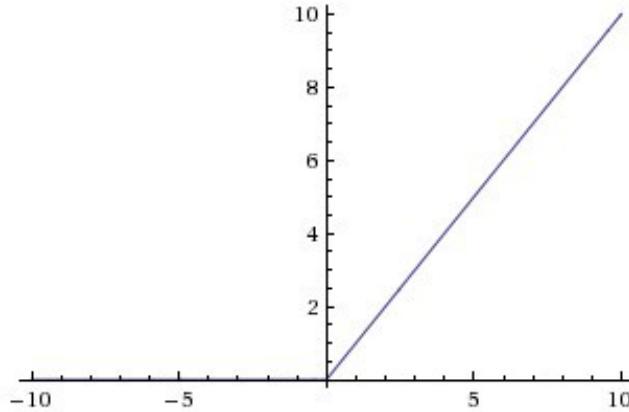


Figure 3.10: Rectified Linear Unit(ReLU) (collected from medium.com).

### Sigmoid Function

The sigmoid is a non-linear AF used mostly in feedforward neural networks. This function takes any real value

$$\text{real value} = [-\alpha, +\alpha] \quad (3.3)$$

as input and outputs values in the range of [0 to +1]. The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to 0.0, as shown below Fig: ?? and mathematically it can be represented as Equation 3.4

$$f(x) = \left( \frac{1}{1 + \exp^{-x}} \right) \quad (3.4)$$

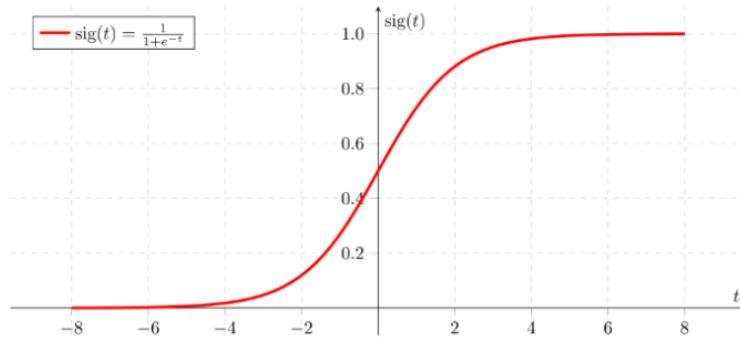


Figure 3.11: Sigmoid Function (collected from towardsdatascience.com).

### Softmax Function

Softmax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1.

### 3.3 Transfer Learning

Transfer learning is a technique in which a pre-trained model is used on another dataset. This technique is mainly used when there are not enough input data to properly train the model. In such cases, a different model, which is already trained in a different large dataset is used.

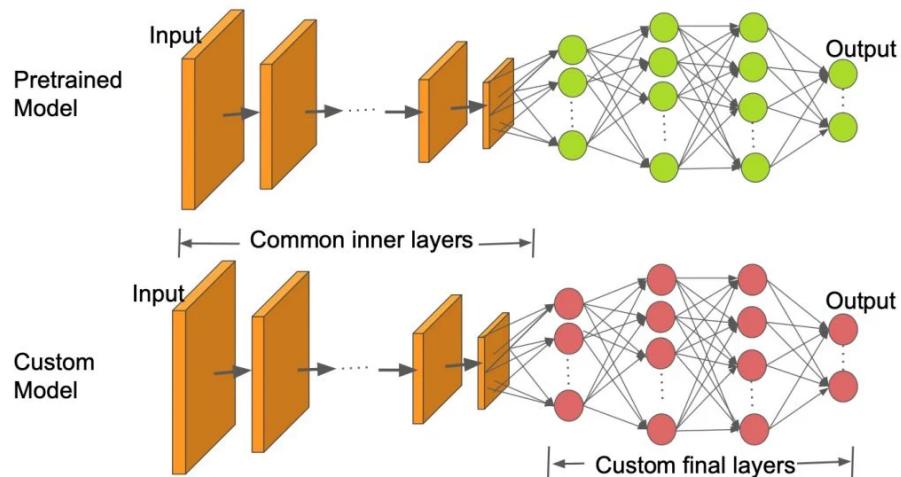


Figure 3.12: Transfer Learning (collected from learnopencv).

It is an excellent method to use the presentation of substantial pre-trained CNN models. With Transfer Learning, the last thick Layers of the pre-trained CNNs utilised for picture order are supplanted by another deep layer(s) where the new final layer actualises a Softmax initiation over the new classes to anticipate. This way, the incredibly solid element extractors of the pre-trained CNNs can be utilized to foresee the new classes. In this project, we used some of the models which were pre-trained in the ImageNet dataset that contain millions of images that are associated with 1000 classes [21]. These models are VGG16, MobileNetV2 and ResNet152V2.

### 3.4 Performance Metrics

The performance matrix can be evaluated by estimating the predicted image among four subsets: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP represents the number of positive cases classified correctly. TN represents the number of negative cases classified correctly. FP, the number of positive cases classified as inaccurate. FN, the number of negative cases classified incorrectly. Five different metrics were used in this project.

- **Confusion Matrix:** A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It provides insights into the true and false positives and negatives of a model. The goal is to maximize the values of the diagonal of the matrix which encompass the true positives and true negatives, while minimizing the values that are off the diagonal that encompass the false positives and false negatives.
- **Accuracy:** The accuracy of a model is defined as the ratio of true predictions to total predictions. The accuracy is expressed by using TP, TN, FP, FN as represented by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.5)$$

- **Recall:** The recall of a model is defined as the ratio of true positives to the total true positives and false negatives. The recall is expressed by using TP, FN as represented by:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.6)$$

- **Precision:** The precision of a model is defined as the ratio of true positives to the total true positives and false positives. The precision is expressed by using TP, FP as represented by:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.7)$$

- **F1-Score:** The F1-Score of a model is a balanced measure between precision and recall, utilizing harmonic mean to weigh the lower value heavier. The F1-Score is expressed by using Precision, Recall, TP, FP, FN as represented by:

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN} \quad (3.8)$$

## Chapter 4

# Proposed Methodology

The proposed methodology is shown in Fig 4.1. using a block diagram and each block is explained in detail below:

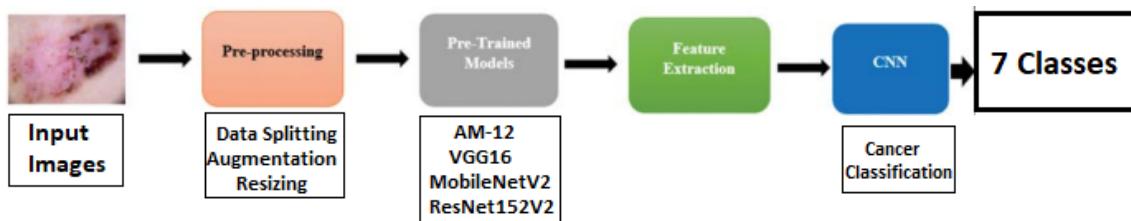


Figure 4.1: Block diagram of proposed methodology

### 4.1 Input images

The proposed system uses HAM10000 dataset consists of high-resolution dermoscopic images which are divided amongst seven classes, digitized and stored as JPEG images. In this proposed system, chapter 5 describes the datasets in detail.

### 4.2 Pre-Processing

The acquisition of images process must be non-uniform in several terms. Thus, the main goal of the pre-processing step is to enhance the image parameters such as quality, clarity, etc., by removing or reducing the unwanted parts of the image or the background. The main steps of the pre-processing are grayscale conversion, image enhancement, and noise removal. In this proposed system, Chapter 6 describes the pre-processing in detail.

## 4.3 Models

The CNN was applied in this work since in recent times, this technique is frequently used especially to solve computer vision-related problems. Typically, CNN is a combination of many layers, with each layer doing a particular subtask toward the main classification task. We used our self-made models AM-12 and the Transfer Learning methods using models like ResNet152V2, MobileNetV2 and VGG16 which was pre-trained with the ImageNet dataset.

### 4.3.1 AM-12

AM-12 is our own CNN architecture implementation. It consisted of 6 Convolutional layers, 5 Maxpool layers and 1 dropout layer. Hence the number 12 in the name to indicate the total number of layers. The letters AM indicate the first letter of author's name and title: "Anik Modak". The model has also one flatten layer and three dense layer but these are common layers of all models.

### 4.3.2 VGG16

VGG16 is a convolution neural net (CNN) architecture. According to [22] VGG16 is a 16-layered CNN model that utilizes an architecture of increasing depths with very small convolution filters. The model won 1st place in the ImageNet recognition challenge in 2014. VGG16 support any input size greater than 32 x 32, with larger image sizes offering better performance.

### 4.3.3 MobileNetV2

MobileNet model is developed for efficiency and is designed for possibly running embedded devices or mobile devices. The depthwise separable convolution architecture is the primary layer of this model to reduce the feature number. MobileNet v2 was released in April 2017. It is very similar to the original MobileNet, except that it uses inverted residual blocks with bottlenecking features [23]. It has a drastically lower parameter count than the original MobileNet. MobileNetV2 also support any input size greater than 32 x 32, with larger image sizes offering better performance.

### 4.3.4 ResNet152V2

There are various models of ResNet, such as ResNet-50, ResNet-101, and ResNet- 152. ResNet152V2 is a 152-layered residual neural network (ResNet) developed by Microsoft's research team [24]. It builds on constructs known from pyramidal cells in the cerebral cortex, by skipping connections or jumping over layers. ResNet won the 1st place in the ImageNet recognition challenge in 2015. ResNet152V2 also support any input size greater than (32, 32) but (224, 224) to be compatible with this model.

## 4.4 Feature Extraction

Feature extraction is considered as the most crucial part in the entire process of classification. The extraction of relevant features from the given input dataset for performing computations such as detection and classification further is called feature extraction [25]. In this work, a unique feature extraction methodology was used to extract the feature. These features were extracted using three different pre-trained models, which were discussed in the above section. These pre-trained models were very effective as they reduced the time for developing the deep-learning neural network, especially during the training phase of the process. The prediction accuracy depends on the extracted features; if the feature is very significant, then the results are also considerably effective and significant. To achieve better accuracy, different features were extracted at different layers.

## 4.5 Classification

Convolutional Neural Networks (CNN) were utilized to accurately classify pigmented skin lesions in dermoscopic images. CNNs excel in analyzing visual imagery as they are fully connected feed forward neural networks that reduce the number of parameters very efficiently without losing out on the quality of models. Three different CNN models were employed to analyze and predict the pigmented lesions classes.

# Chapter 5

## Dataset

As mentioned earlier, the data set used in this paper was “The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions” [8]. It consisted of 10,015 dermoscopic skin pigmented lesion 600 by 450 pixel images which are divided amongst seven classes, digitized and stored as JPEG images.

They were initially manually cropped and centered around the lesion, as well as adjusted to enhance visual contrast and color reproduction.

The metadata set included 7 attributes associated with each image and patient, which were:

1. A Lesion ID [lesion id]
2. An Image ID [image id]
3. A Diagnostic Skin Lesion Category [dx]
4. A Technical Validation Field Type [dx type]
5. Patient’s Age [age]
6. Patient’s Sex [sex]
7. The Localization of the Skin Lesion [localization]

	<b>lesion_id</b>	<b>image_id</b>	<b>dx</b>	<b>dx_type</b>	<b>sex</b>	<b>localization</b>	<b>dataset</b>
<b>count</b>	10015	10015	10015	10015	10015	10015	10015
<b>unique</b>	7470	10015	7	4	3	15	4
<b>top</b>	HAM_0003789	ISIC_0027419	nv	histo	male	back	vidir_molemax
<b>freq</b>	6	1	6705	5340	5406	2192	3954

Figure 5.1: Descriptive statistics of the data set contents

We can see how there was a unique image id for each entry, but not a unique lesion id. This indicated that there were duplicate images for the same lesion id but at a different distortion, such as an angle, shear, or zoom distortion. In addition, unique lesion id has 7470 and class Melanocytic Nev[i]n[ny] dominated the skin lesion categories, by having a frequency of 6,705 out of the 10,015 images we got, directly signaling a class imbalance issue in the data set. So we need to balanced data performing various techniques. In the Chapter 7 describes the balancing techniques in detail. Before analyzing the skin lesion categories further, let us analyze the rest of the attributes first.

Fig: 5.2 shows the distribution of the patients' age. It can be seen that the majority of patients resided between the ages of 35 and 70.

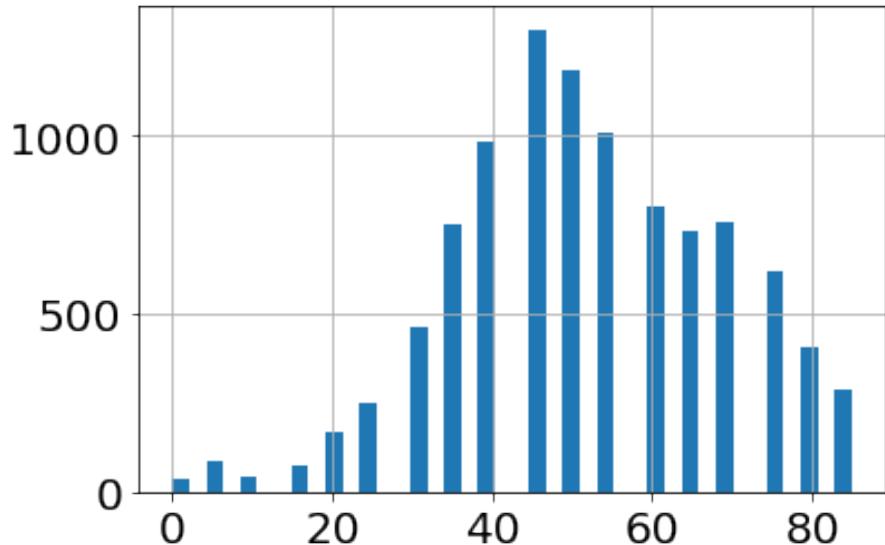


Figure 5.2: Patients' Age Distribution

Fig: 5.3 shows the distribution of the patients' sex. It can be seen that there was almost an equal amount of male to female patients.

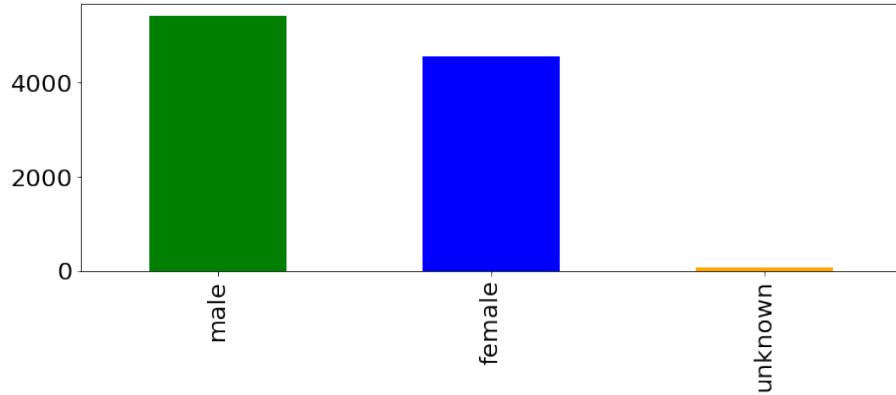


Figure 5.3: Patients' Sex Distribution

The technical validation field category represented the ground truth of the data set and indicated how the skin lesion diagnosis was made. The publishers defined four different types of ground truths which were:

- **Histopathology:** Diagnoses of excised lesions have been performed by specialized dermatopathologists [8]. We scanned all available histopathologic slides of the current ViDIR image set for later review.
- **Confocal:** Diagnoses of excised lesions have been based on reflectance confocal microscopy. Most cases were included in a prospective confocal study conducted at the Department of Dermatology at the Medical University of Vienna that also included follow-up for one year [26].
- **Follow-up:** This type is limited to nevi class only, where digital dermatoscopy did not show any changes during 3 follow-up visits or 1.5 years [8] we accepted this as evidence of biologic benignity.
- **Consensus:** Based on experts consensus. This type is defined for typical benign cases without histopathology or follow-up, and where two experts have provided same unequivocal benign diagnosis [8].

Fig: 5.4 shows the distribution of the technical validations. As shown in the figure, more than 50% of the skin lesion diagnosis were based on histopathology.

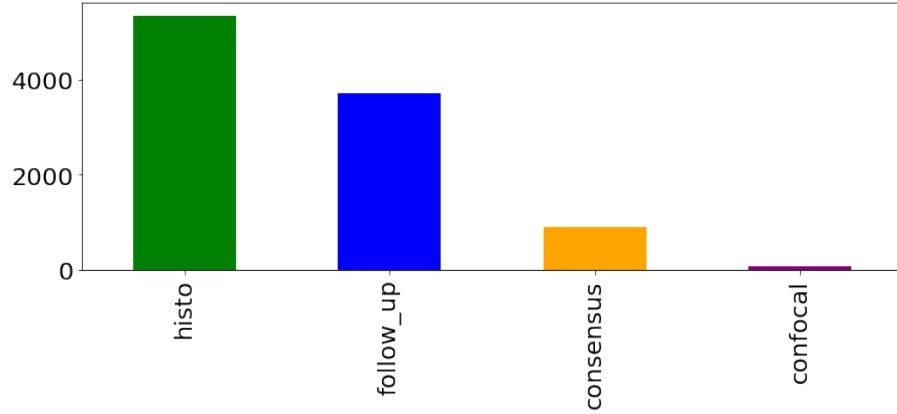


Figure 5.4: Technical Validation Field Distribution

Fig: 5.5 shows the localization distribution of the data set. It can be seen that the back, lower extremity, and trunk are heavily compromised skin cancer regions.

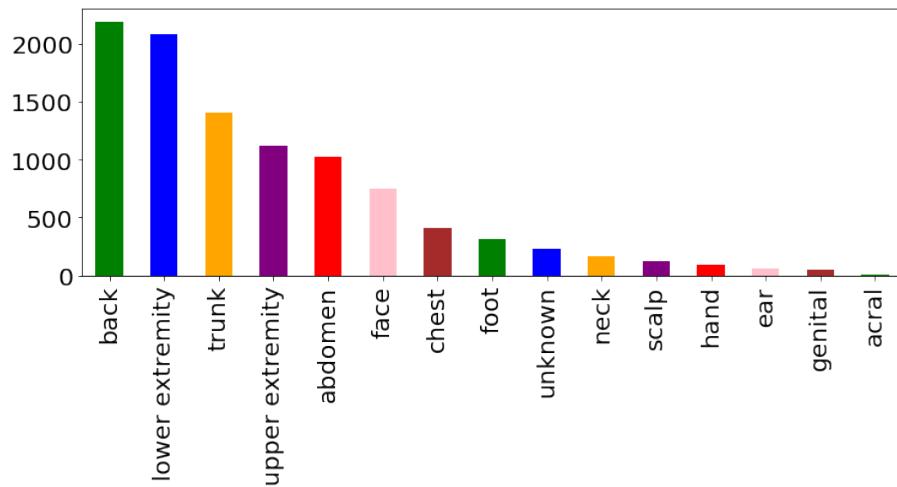


Figure 5.5: Localization Distribution

As for the diagnostic skin lesion categories, seven different classes were present in the data set. Fig: 5.6 shows the skin lesion categories distributions.

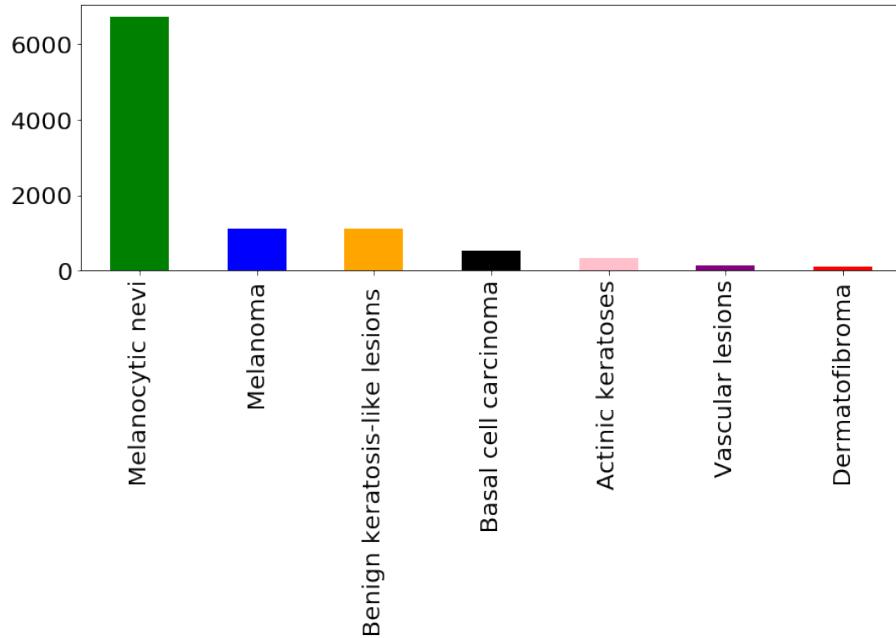


Figure 5.6: Skin Lesion Categories Distribution

Fig: 5.7 shows the images from the data set for each class. The seven categories were:

- **Melanocytic Nevi [nv]:** Benign neoplasms of melanocytes and appear in a myriad of variants. The variants may differ significantly from a dermatoscopic point of view. [6705 images]
- **Melanoma [mel]:** Malignant neoplasm derived from melanocytes that may appear in different variants. If excised in an early stage, it can be cured by simple surgical excision. [1113 images]
- **Benign Keratosis-like Lesions [bkl]:** Can be regarded as a flat variant of seborrheic keratoses and lichen-planus like keratoses (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression.[1099 images]
- **Basal Cell Carcinoma [bcc]:** A common variant of epithelial skin cancer that rarely metastasizes but grows destructively if untreated. [514 images]
- **Actinic Keratoses [akiec]:** Common non-invasive, variants of squamous cell carcinoma that can be treated locally without surgery. [327 images]
- **Vascular Lesions [vasc]:** These range from cherry angiomas to angiokeratomas and pyogenic granulomas, meaning they could be benign or malignant. [142 images]
- **Dermatofibroma [df]:** Benign skin lesions regarded as either a benign proliferation or an inflammatory reaction to minimal trauma. [115 images]

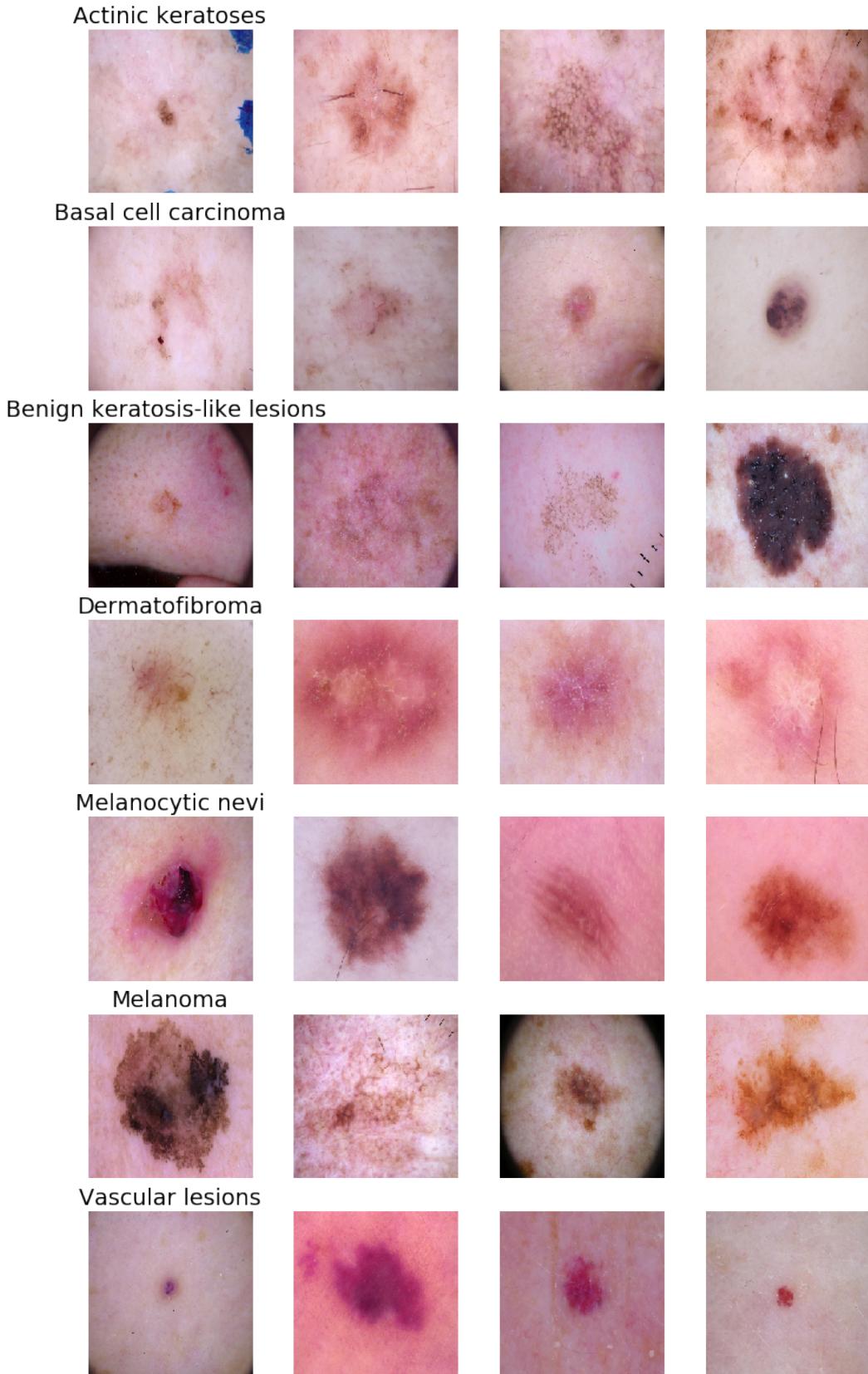


Figure 5.7: Sample images for the seven skin lesion categories

# Chapter 6

## Data Pre-Processing

### 6.1 Directory Structure

Created three directories for the images and their labels. The first directory name **train** directory, the second directory name is **validation** directory and the third directory name is **test** directory. Each directory has seven folder:

- **Melanocytic Nevi** [nv]
- **Melanoma** [mel]
- **Benign Keratosis-like Lesions** [bkl]
- **Basal Cell Carcinoma** [bcc]
- **Actinic Keratoses** [akiec]
- **Vascular Lesions** [vasc]
- **Dermatofibroma** [df]

### 6.2 Data Cleansing

This involved dropping lesion id's with duplicate images, keeping only one image for each lesion id. These images are of the same exact lesion but at a different angle, zoom etc.  
Unique numeric codes were created for each skin lesion category to assist with the predictions to be made later on, as integers were easier to handle than strings.

### 6.3 Data Splitting

Training 70 / Validation 10 / Test 20 split. The data was split 70:10:20 respectively across each class individually to ensure there was enough samples from each class in each split for accurate modeling. This was necessary due to the huge class imbalance demonstrated by the data set. After splitting 10015 images divided into 7009 train images, 1004 validation images and 2002 test images.

### 6.4 Data Normalization

The image feature was then normalized. Normalization is a scaling technique in which values are shifted and re-scaled so that they end up ranging between 0 and 1. The normalization of each image was done by subtracting it's values from the training's mean value and then dividing by the training's standard deviation.

### 6.5 Data Augmentation

All the original images were transformed and augmented every epoch and then used for training to avoid overfitting. This allowed the model to be more robust and accurate, as it was trained on different variations of the same image. The number of images in each epoch was equal to the number of original images. The images were:

- Randomly rotated by 20%
- Randomly shifted horizontally by 20%
- Randomly shifted vertically by 20%
- Randomly channel shifted by 10%
- Randomly sheared by 10%
- Randomly zoomed by 10%

### 6.6 Data Resizing

Images Width and height sizes would have been an issue when it came to training our CNN models, due to the huge amount of images on hand. Flattened images were reshaped back to width 64 height 64 to be fed into our models.

# Chapter 7

## Implementation

### 7.1 Framework

- Google Colab (Colaboratory is a product from Google Research.)
- Jupyter Notebook
- Python 3.7.4 environment.
  - Keras (Deep learning library for tensorflow).
  - Open CV (Open Source Computer Vision Library)
  - Scikit-learn (a Python module for machine learning built on top of SciPy)
  - Matplotlib (A comprehensive library for creating static, animated, and interactive visualizations in Python)
  - imbalanced-learn (This library is used to balance the dataset in python)

### 7.2 Models

#### 7.2.1 AM-12

Table 7.2.1 highlights the layers used in the network.

Layer (type)	Output Shape	Parameter
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
conv2d (Conv2D)	(None, 64, 64, 64)	1792
conv2d_1 (Conv2D)	(None, 64, 64, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 32, 32, 64)	0
dropout (Dropout)	(None, 32, 32, 64)	0
conv2d_2 (Conv2D)	(None, 32, 32, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 128)	0
conv2d_3 (Conv2D)	(None, 16, 16, 256)	295168
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 256)	0
conv2d_4 (Conv2D)	(None, 8, 8, 512)	1180160
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 512)	0
conv2d_5 (Conv2D)	(None, 4, 4, 64)	294976
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 32)	4128
dense_2 (Dense)	(None, 7)	231

Table 7.1: AM-12 Models convolution and max-pooling layer

- The first 2 layers consisted of 2 Convolutional 2D layers. Two layers were used rather than 1 as the model had more flexibility in expressing non-linear transformations without losing information, making it easier to learn. 64 filters were used to extract the simple features. ReLu activation function was used here and in the rest of the network layers (except in the last layer) as it solved the vanishing gradient problem and was computationally light. Same padding was used so the output size was the same as the input size.
- Layer 3 consisted of a MaxPool 2D layer, which acted as a down-sampling filter. The 2x2 pool size looked at 2 neighboring pixels and picked the maximal value.
- Layer 4 consisted of a Dropout layer. It is a regularization technique where randomly selected neurons are ignored during training, forcing the network to learn features in a distributed way while improving generalization and reducing overfitting. A quarter of the neurons were ignored in this layer

- 5-12 layers consisted of a Convolutional 2D layer and a 2x2 pool size MaxPool 2D layer. The Convolutional 2D layers are used 128, 256, 512 and 64 filters, to be able to extract more complex features from the images, as typically simple features are extracted first in a deep neural network before extracting the complex features. These 12 layers
- Layer 13 consisted of a Flatten layer, which flattened the input onto a single 1D vector.
- Layer 14-15 consisted of a Fully Connected Layer input Dense layer with 128 units and 32 units. A sigmoid activation function was used here because it exists between (0 to 1).
- Layer 16 consisted of a Fully Connected Layer output Dense layer with 7 units, the same number of lesion categories in the data set. A softmax activation function was used here because it mapped a vector to probability of a given output in binary classification, working well with our onehot encoding of our 7 target skin lesion categories.

### 7.2.2 VGG16

The model was pre-built and loaded via the keras.applications.vgg16 library package. After getting pre-train model taking output added extra one Convolutional 2D layer used 64 filters, One (2X2) Maxpooling layer and a dropout(0.25) layer. After flatten the model added two Dense layers were added two Fully Connected Layer input Dense layer with 128 units and 32 units. Finally added output Dense layer with 7 units, the same number of lesion categories in the dataset.

### 7.2.3 MobileNetV2

The model was pre-built and loaded via the keras.applications.mobilenet\_v2 library package. After getting pre-train model taking layer.output[-6] and added extra one Convolutional 2D layer used 64 filters, One (2X2) Maxpooling layer and a dropout(0.25) layer. After flatten the model added two Dense layers were added two Fully Connected Layer input Dense layer with 128 units and 32 units. Finally added output Dense layer with 7 units, the same number of lesion categories in the dataset.

### 7.2.4 ResNet152V2

The model was pre-built and loaded via the keras.applications.resnet152\_v2 library package. After getting pre-train model taking output added extra one Convolutional 2D layer used 64 filters, One (2X2) Maxpooling layer and a dropout(0.25) layer. After flatten the model added two Dense layers were added two Fully Connected Layer input Dense layer with 128 units and 32 units. Finally added output Dense layer with 7 units, the same number of lesion categories in the dataset.

### 7.3 Hyperparameters

All the CNN models employed had so many parameters, meaning there were so many possible changes in the architecture. In addition, training them with the huge data set in hand took quite a long time. Performing hyperparameter optimization in our case would have been a total overkill, specially with the computing resources we had available to us. Table 7.2 highlights the hyperparameter values used across all four CNN models.

Hyperparameter	Value
Optimizer	Adam
Loss Function	Categorical cross-entropy
Epochs	50
Batch Size	32

Table 7.2: hyperparameter values

The following explains the reasoning behind the hyperparameters values selected:

- **Optimizer:** Adam is the most common optimization algorithm used today for training deep neural networks, as it is straightforward to implement, computationally very efficient, and is very effective in dealing with large data and parameters. Adam can be looked as a combination of RMSprop and Stochastic Gradient Descent with momentum.
- **Loss Function:** Categorical cross-entropy is a loss function that is used for single label categorization. This is when only one category is applicable for each data point. This worked perfectly here as one example could only belong to one of the seven skin lesion categories.
- **Epochs:** Upon multiple initial trials with values of 20, 50, 100 and 200, 50 epochs was sufficient to get to the most optimum results.
- **Batch Size:** Upon multiple initial trials with values of 8, 16, 32, and 64, a batch size of 16 produced the most optimum results.

### 7.4 Balancing Datasets

Since, our datasets in imbalanced. For imbalancing dataset predicting outcome in such cases becomes very difficult as data becomes biased towards one particular class of outcome. We need to balanced these dataset.

A balanced dataset is a dataset where each output class is represented by the same number of input samples. Balancing can be performed by exploiting one of the following techniques [27]:

- **Oversampling:** Oversampling is a technique which increases the number of samples of the smallest class up to the size of the biggest class. This is done by generating synthetic samples. Different techniques can be applied to oversample a class. We used RandomOverSampler. The imbalanced-learn package also provides other techniques, such as SMOTE, ADASYN and Rose. All the over samplers provide a function, called `fit_resample()`, which receives the `X_train` input variable and the `Y_train` output variable and return the over sampled input and output variables.
- **Undersampling:** Undersampling is a technique which decreases the number of samples of the biggest class down to the size of the smallest class. This is done by removing some samplers from the biggest class. Different techniques can be applied to undersample a class. We used RandomUnderSampler. The imbalanced-learn package also provides other techniques, such as Condensed Nearest Neighbour and NearMiss. Similarly to over samplers, all the under samplers provide a function, called `fit_resample()`, which receives the `X_train` input variable and the `Y_train` output variable and return the under sampled input and output variables.
- **Class Weights:** Setting the class weight constitutes another valid alternative for balancing. Each scikit-learn classification model can be configured with a parameter, called `class_weight`, which receives the weight of each class in the form of a Python dictionary. In this project, Weights were added to make the models more sensitive to the Melanocytic nevi [nv] skin lesion category due to the classes imbalance, as it represented around two-thirds of the data. Table 7.3 illustrates the weights associated with each skin lesion category:

Skin Lesion Category	Lables	Weight
Actinic keratoses [akiec]	0	4.086880466472303
Basal cell carcinoma [bcc]	1	2.3954203691045794
Benign keratosis-like [bkl]	2	1.1656411109263263
Dermatofibroma [df]	3	10.651975683890578
Melanoma [mel]	4	1.1656411109263263
Melanocytic nevi [nv]	5	0.2329654989031443
Vascular lesions [vasc]	6	9.357810413885181

Table 7.3: Class Weights

- **Threshold:** Adjusting threshold is a manual technique to balance a dataset. Conceptually, if a predicted value is greater than the threshold, it is set 1, otherwise, it is set to 0. We haven't use this technique in our project.

# Chapter 8

## Result And Discussion

The created models were trained by using the original imbalanced dataset and modified balanced dataset. As discussed before in the dataset in Chapter 5 and preprocessing in Chapter 6, we divided the whole dataset into training, validating, and testing with the ratio of 70:10:20, respectively. The number of epochs for which the models were trained was 50 and resized images size was (64X64). We then calculated the confusion matrix and evaluated the models using the overall accuracy of classification. Since our dataset was originally imbalanced; therefore, at first, we trained the models with this original imbalanced dataset. Following that, using some techniques we made the dataset balanced and then trained the models with this balanced dataset.

### 8.1 Training the models with imbalanced dataset

We trained models with imbalanced dataset and find out accuracy associated for each model.

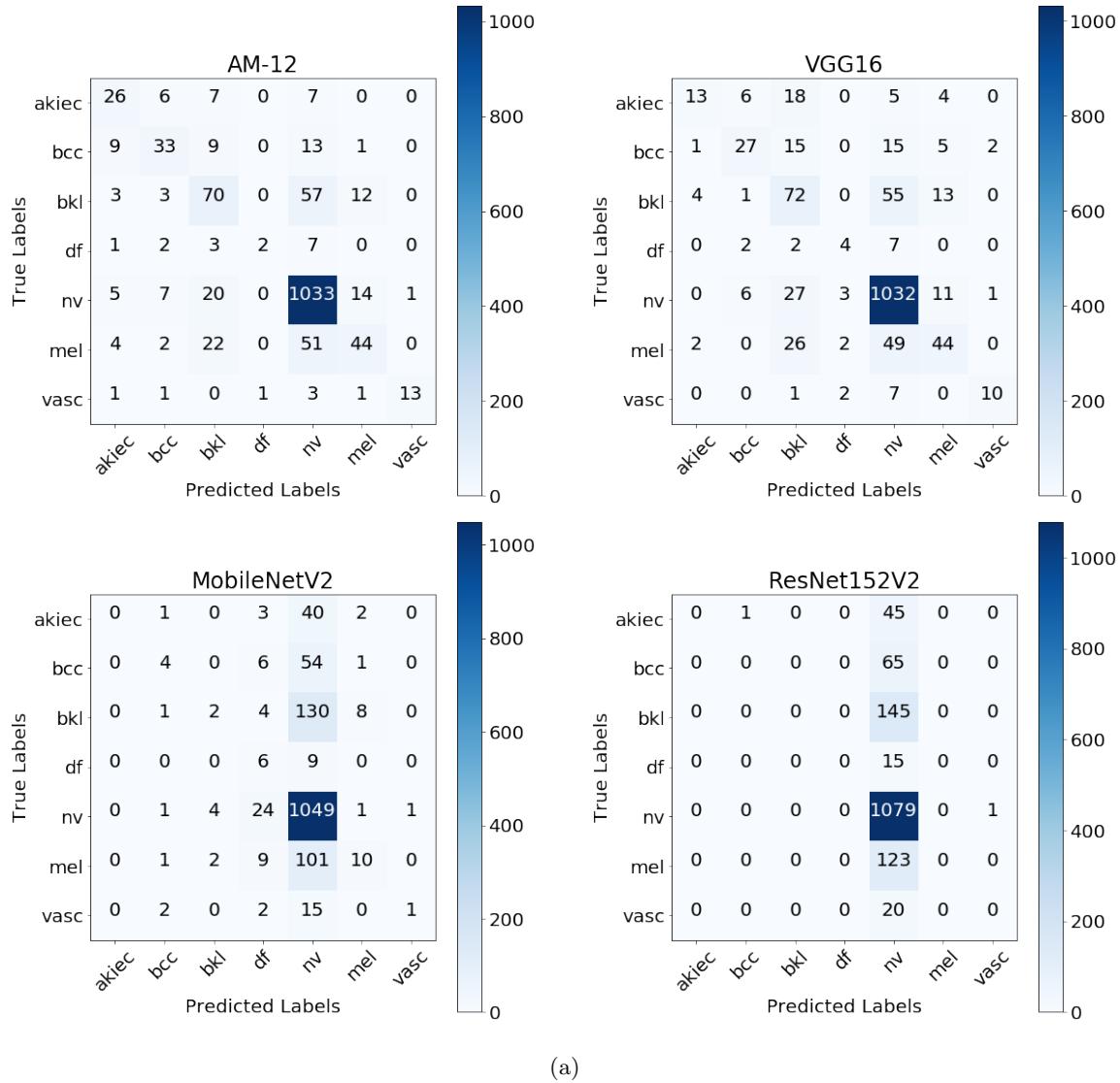
Table 8.1 shows the testing sets accuracy associated with each model.

Metrics	Models			
	AM-12	VGG16	MobileNetV2	ResNet152V2
Accuracy (%)	77.98	77.92	70.62	72.2

Table 8.1: Models Performance before balancing

Though test accuracy is quite satisfactory, however the classification of all class accuracy is not so satisfactory. The reason behind this result is that, in the dataset, the number of samples for the class Melanocytic Nevi is 4298, when some class has the number of samples less than 1000. For this reason classification may be biased by Melanocytic Nevi class. As a result, prediction for other classes is not so satisfactory.

Figure 8.1 shows the confusion matrices for true vs predicted labels for each model when run on the test set for imbalanced dataset.



(a)

Figure 8.1: Models Confusion Matrices before balancing

## 8.2 Training the models with balanced dataset

We trained models with balanced dataset and find out accuracy associated for each model. The data balancing can be performed by exploiting oversampling, undersampling and Class Weights techniques. We applied all of these technique and the performance were evaluated for testing set. All the models improvements with every epoch were verified as well through this visualization. The figures can be viewed in the code. Here we showing the accuracy and the confusion matrices for true vs predicted labels for each model when run on the test set.

### 8.2.1 Oversampling Technique

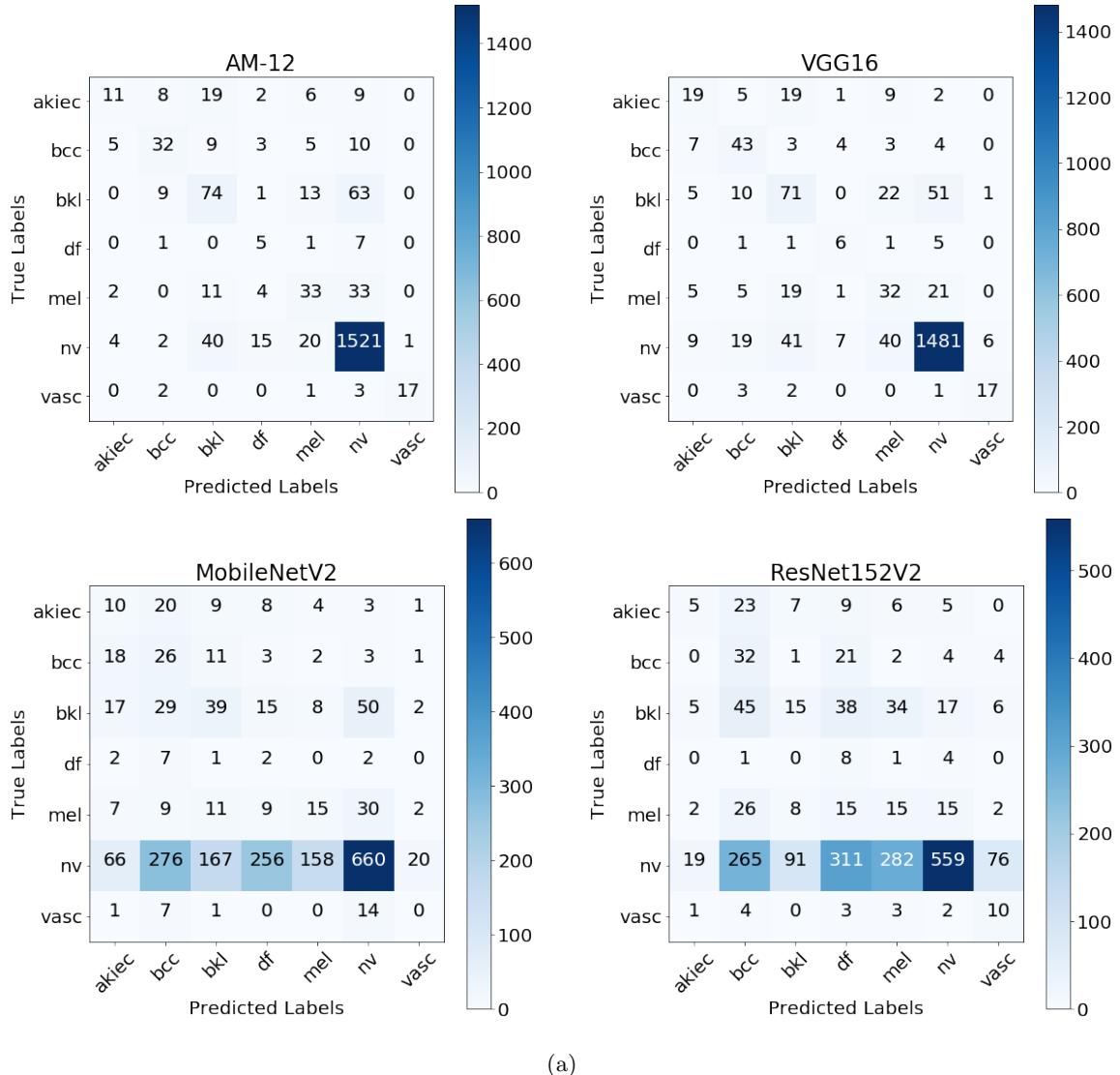
We applied oversampling technique with RandomOverSampler from imbalanced-learn package. In this technique we generated the augmented images and add them to the training set folders. These samples are of the same exact lesion but at a different angle, zoom etc. Though the data were increased but unique data are same. We got a good classification accuracy for testing set in this technique but it took huge time to training all model.

Table 8.2 shows the weighted testing sets accuracy associated with each model.

Metrics	Models			
	AM-12	VGG16	MobileNetV2	ResNet152V2
Accuracy (%)	82.06	83.02	53.47	45.5

Table 8.2: Models Performance after applying Oversampling Technique

Figure 8.2 shows the confusion matrices for true vs predicted labels for each model when run on the test set.



(a)

Figure 8.2: Models Confusion Matrices after applying Oversampling Technique

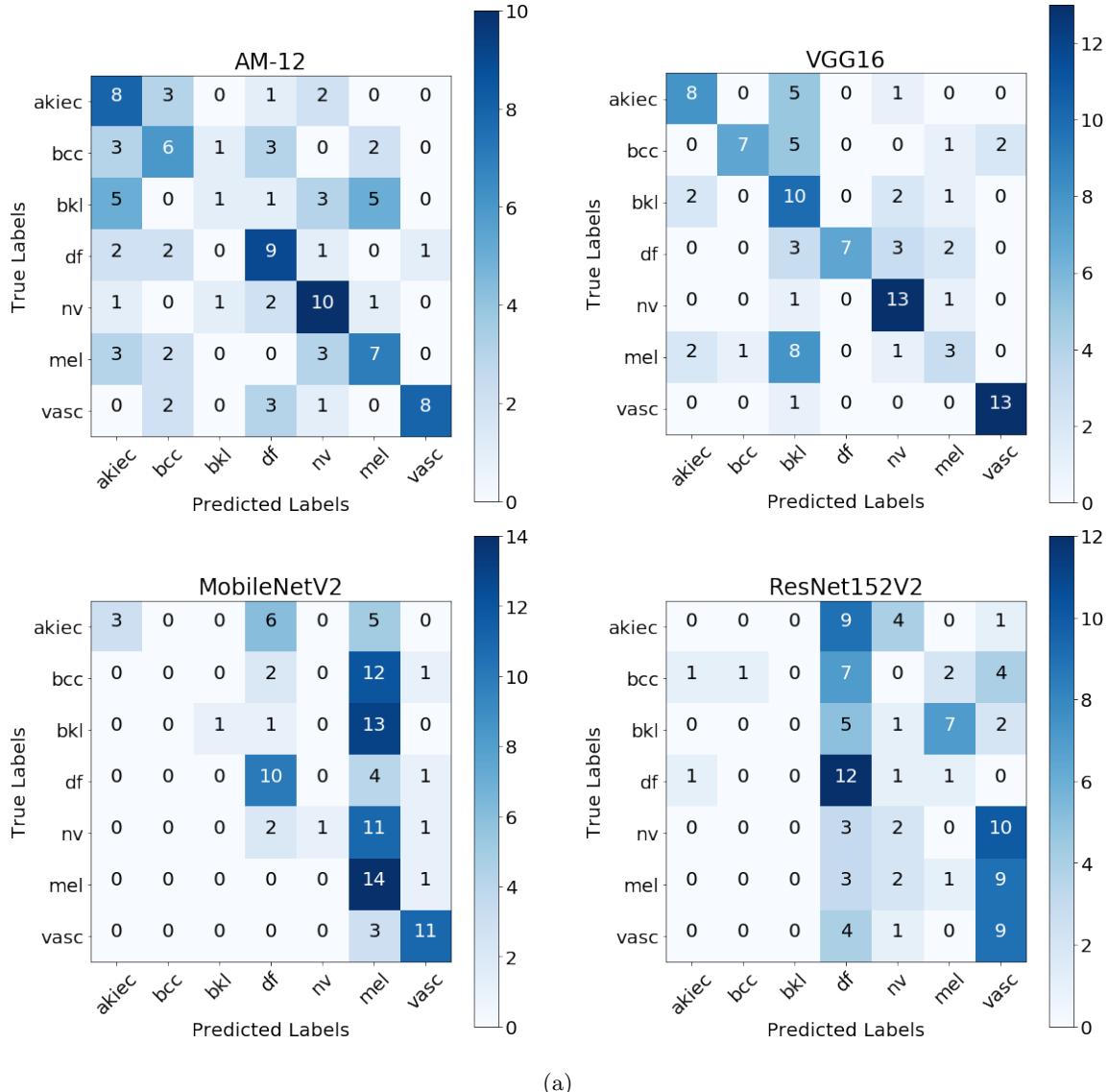
### 8.2.2 Undersampling Technique

We applied undersampling technique with RandomUnderSampler from imbalanced-learn package. After applying undersampling technique on unique data from dataset, we got a small number of train, test, valid data. For this reason we got very low classification accuracy for testing set after training the models with these small data. Table 8.3 shows the weighted testing sets accuracy associated with each model.

Figure 8.3 shows the confusion matrices for true vs predicted labels for each model when run on the test set.

Metrics	Models			
	AM-12	VGG16	MobileNetV2	ResNet152V2
Accuracy (%)	47.58	60.2	35	28.2

Table 8.3: Models Performance after applying Undersampling Technique



(a)

Figure 8.3: Models Confusion Metrics after applying Undersampling Technique

### 8.2.3 Class Weights Technique

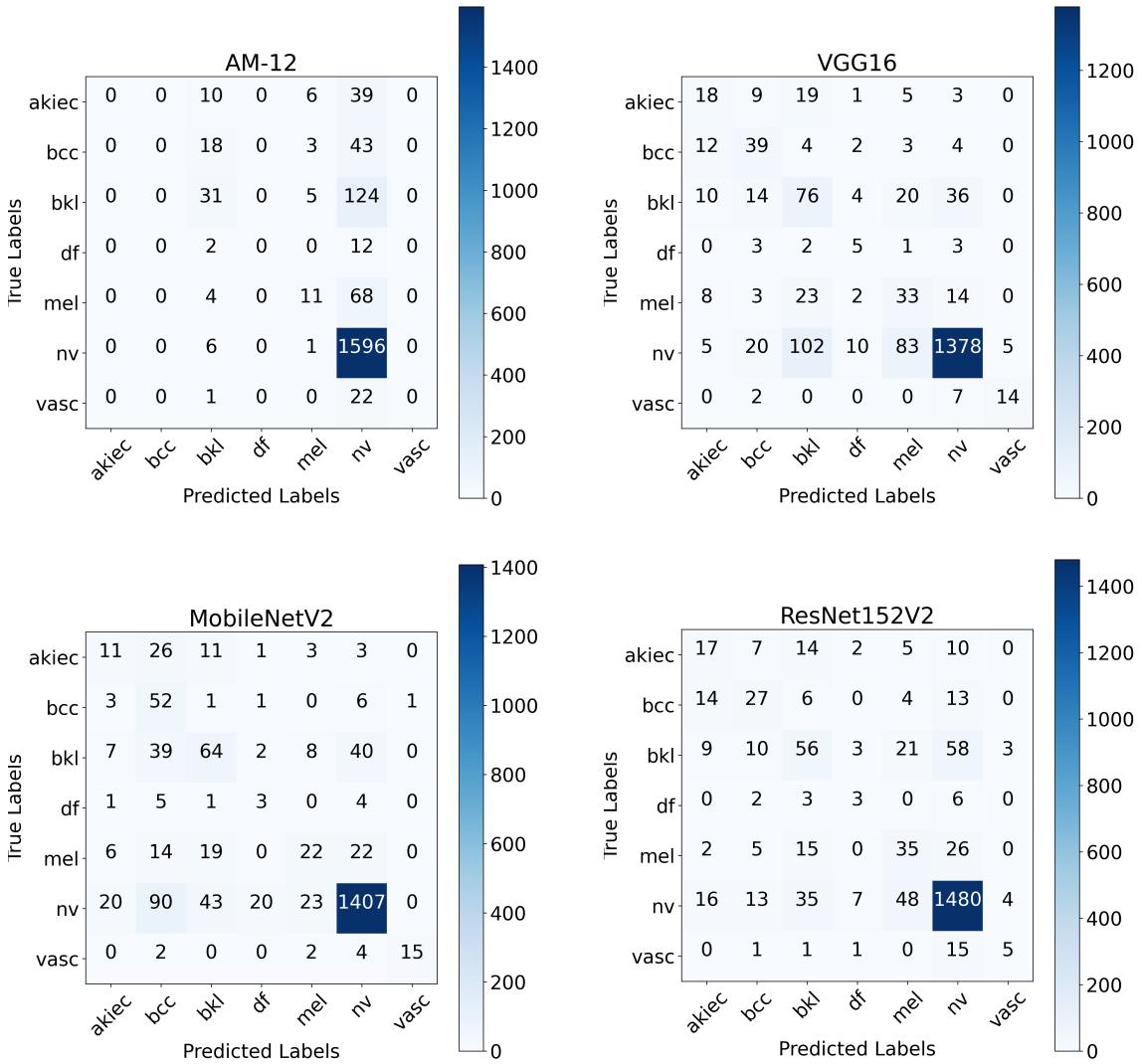
We applied class weights technique to set class weights using `sklearndatamodels` package. Table 7.3 shows the class weights of all Category Skin Lesion. Using class weights technique we got better accuracy for testing set all the model rather than other data balancing techniques.

Table 8.4 shows the weighted testing sets accuracy associated with each model.

Metrics	Models			
	AM-12	VGG16	MobileNetV2	ResNet152V2
Accuracy (%)	81.51	78.07	79.07	81.06

Table 8.4: Models Performance after applying Class Weights Technique

Figure 8.4 shows the confusion matrices for true vs predicted labels for each model when run on the test set.



(a)

Figure 8.4: Models Confusion Matrices after applying Class Weights Technique

### 8.3 Models Performance Comparisons for Testing Set

Firstly, all of the testing accuracies were comparable with the validation accuracies, further solidating the fact that all models were found neither underfitting or overfitting. All four models excelled in melanocytic nevi [nv] classification, each achieving a class accuracy of over 85%. However, classification accuracy results for all the other classes varied. The VGG16 model gave best average accuracy for all technique rather than other models. In the comparative study with the VGG16 Models, imbalanced dataset achieved 77.92%, the balanced undersampling technique achieved 53%, Oversampling technique achieved 83.02%, and class weights technique achieved 78.07% accuracy in 64X64 size images.

Above all of these technique class weights technique gave best performance for all model. In the comparative study with the class weights technique, AM-12 achieved 81.51%, VGG16 78.07%, MobileNetV2 79.07%, and ResNet152V2 81.06% accuracy. In Oversampling technique also gave high performance for two models AM-12 achieved 82.02% and VGG16 achieved 83.02% accuracy. In undersampling technique all of the models gave low performance for testing set.

Since, class weights technique and VGG16 model gave best performance of all cases. Thus we again trained VGG16 model with 128X128 dataset images and augmented images for balancing dataset we used class weights technique in this time. This time maximum attained accuracy was 84.87% the classification of dermoscopic images for testing set. Table 8.5 shows classification report for the weighted testing sets accuracy, recall, precision and f1-score associated with each class.

Skin Lesion Category	Class	Precision	Recall	F-score	Support
Actinic keratoses [akiec]	0	0.51	0.44	0.47	55
Basal cell carcinoma [bcc]	1	0.49	0.53	0.51	64
Benign keratosis-like [bkl]	2	0.55	0.42	0.48	160
Dermatofibroma [df]	3	0.38	0.43	0.40	14
Melanoma [mel]	4	0.34	0.41	0.38	83
Melanocytic nevi [nv]	5	0.93	0.95	0.94	1603
Vascular lesions [vasc]	6	0.88	0.65	0.75	23
accuracy				0.85	2002
macro avg		0.58	0.55	0.56	2002
weighted avg		0.85	0.85	0.85	2002

Table 8.5: Classification Report for VGG16 Model

Figure 8.5 shows VGG16 training history against the number of epochs. That was an important visualization to ensure the model kept learning with every epoch, improving its accuracy and reducing its losses and errors, while trying to optimize the objective function.

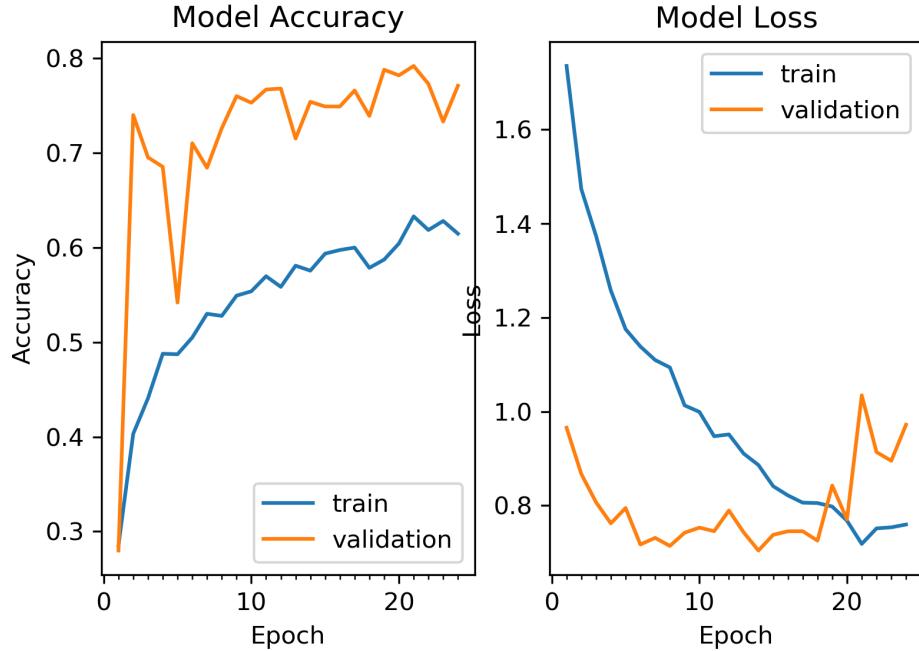


Figure 8.5: VGG16 Models training history against epochs

VGG16 performance scores were followed by our AM-12, then followed by ResNet152V2, and lastly, followed by MobileNetV2. Taking total training time into account however, AM-12's performance was very comparable to ResNet152V2 and MobileNetV2. Not only was AM-12 more time efficient than other models, but AM-12 model's memory consumption was also much smaller, due to a more compact network architecture.

# Chapter 9

## Conclusion

This work investigated the use of different CNN architectures to predict skin lesion categories based on skin lesion images. The data set was first pre-processed via data editing, and cleaning, then split into the feature and target values, before applying feature normalization and data augmentation. We built our own basic CNN architecture called “AM-12” to employ against the processed data set images to classify their respective skin lesion category, alongside VGG16, MobileNetV2 and ResNet152V2 which were pre-trained CNN on the ImageNet Dataset can be very helpful is the successful classification of cancer lesions in the HAM10000 Dataset. As for the results, VGG16 topped each metric score, scoring 84.87% accuracy, 85.0% precision, 84.87% recall, and 84.81% f1-score for the proposed technique for the classification of dermoscopic images on HAM 1000 Dataset. Results also indicated a promising future for AM-12, as its performance was comparable to MobileNet and ResNet152V2 performance, without any hyperparameter tuning. In our experiments, we could not use original images size due to the constraints of computational power. we had to use the reduced image size of 64 by 64, that might cause huge negative impact to get better performance. Encouraged by these outcomes, future work will include the improvement of prediction result and classification accuracy.

# **Chapter 10**

## **Future Scope and Limitations**

### **10.1 Future Scope**

As a future scope we intend to explore more on the datasets, try to capture real datasets from the hospital staff and get in-depth information about the real disease causes and treatment plans. The dataset preprocessing could also us another future scope as to assist the research community by providing them a unique training dataset exclusively for deep learning techniques which suit to perfectly for learning the features and classifying them further to detect the skin cancer. Another scope is explore other skin lesion data sets to obtain more training data to feed into our models, specially for the underrepresented and malignant skin lesion classes.

### **10.2 Limitations**

The work has some limitations as the Convolution Neural Network is applied it needs huge data. To increase the size if the training data we applied data augmentation. But to avoid data augmentation techniques, big balanced dataset is needed. The results of the work may suffer from better accuracy by the constraints of tools used to analyze and training the datasets. The original image size was 600 by 450 pixel, however, in our experiments, we could not use this images size due to the constraints of computational power. As a result, we had to use the reduced image size of 64 by 64, that might cause huge negative impact to get better performance.

# Bibliography

- [1] F. Santos, F. Silva, and P. Georgieva, “Transfer learning for skin lesion classification using convolutional neural networks,” pp. 1–6, 08 2021.
- [2] “Artificial intelligence in medicine:machine learning.” [Online]. Available: <https://www.ibm.com/topics/artificial-intelligence-medicine>. Accessed: 2022-03-28.
- [3] “Skin cancer facts and statistics.” [Online]. Available: <https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/general>. Accessed: 2022-03-28.
- [4] T. In, “Facts & figures 2019: Us cancer death rate has dropped 27% in 25 years,” *American Cancer*, 2019.
- [5] A. H. G. M. R. Gopi Kasinathan, Selvakumar Jayakumar, “Automated 3-d lung tumor detection and classification by an active contour model and cnn classifier,” *ABS Journal*, vol. 134, pp. 112–119, 2019.
- [6] “World health rankings.” [Online]. Available: <https://www.worldlifeexpectancy.com/bangladesh-skin-cancers1>. Accessed: 2022-03-28.
- [7] M. Goyal, T. Knackstedt, S. Yan, and S. Hass nanopour, “Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities,” *Computers in Biology and Medicine*, vol. 127, p. 104065, 2020.
- [8] H. K. Philipp Tschandl, Cliff Rosendahl, “The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,” *nature*, vol. 5, 2018.
- [9] Z. Gao, J. Zhang, L. Zhou, and L. Wang, “Hep-2 cell image classification with deep convolutional neural networks,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, 08 2014.
- [10] N. C. F. Codella, Q.-B. Nguyen, S. Pankanti, D. A. Gutman, B. Helba, A. C. Halpern, and J. R. Smith, “Deep learning ensembles for melanoma recognition in dermoscopy images,” *IBM Journal of Research and Development*, vol. 61, no. 4/5, pp. 5:1–5:15, 2017.

- [11] H. Haenssle, C. Fink, R. Schneiderbauer, F. Toberer, T. Buhl, A. Blum, and A. Kalloo, “Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists,” *Annals of Oncology*, vol. 29, no. 8, pp. 1836–1842, 2018. Immune-related pathologic response criteria.
- [12] T. J. Brinker, A. Hekler, A. H. Enk, J. Klode, A. Hauschild, C. Berking, B. Schilling, S. Haferkamp, D. Schadendorf, T. Holland-Letz, *et al.*, “Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task,” *European Journal of Cancer*, vol. 113, pp. 47–54, 2019.
- [13] P. Tschandl, C. Rosendahl, B. N. Akay, G. Argenziano, A. Blum, R. P. Braun, H. Cabo, J.-Y. Gourhant, J. Kreusch, A. Lallas, *et al.*, “Expert-level diagnosis of nonpigmented skin cancer by combined convolutional neural networks,” *JAMA dermatology*, vol. 155, no. 1, pp. 58–65, 2019.
- [14] R. C. Maron, M. Weichenthal, J. S. Utikal, A. Hekler, C. Berking, A. Hauschild, A. H. Enk, S. Haferkamp, J. Klode, D. Schadendorf, *et al.*, “Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks,” *European Journal of Cancer*, vol. 119, pp. 57–65, 2019.
- [15] R. Garg, S. Maheshwari, and A. Shukla, “Decision support system for detection and classification of skin cancer using cnn,” pp. 578–586, 01 2021.
- [16] R. Alabduljabbar and H. Alshamian, “Intelligent multiclass skin cancer detection using convolution neural networks,” *Computers, Materials Continua*, vol. 69, pp. 831–847, 01 2021.
- [17] A. Aggarwal, N. Das, and I. Sreedevi, “Attention-guided deep convolutional neural networks for skin cancer classification,” pp. 1–6, 11 2019.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [19] S. Agatonovic-Kustrin and R. Beresford, “Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research,” *Journal of pharmaceutical and biomedical analysis*, vol. 22, no. 5, pp. 717–727, 2000.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [22] A. Z. Karen Simonyan, “Very deep convolutional networks for large-scale image recognition,” *nature*, 2015.

- [23] R. A. Rill and K. Faragó, “Collision avoidance using deep learning-based monocular vision,” *SN Computer Science*, vol. 2, 09 2021.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” pp. 770–778, 2016.
- [25] M. Monika, N. Vignesh, U. Kumari, M. Kumar, and L. Lydia, “Skin cancer detection and classification using machine learning,” *Materials Today: Proceedings*, vol. 33, 08 2020.
- [26] E. Wurm, G. Pellacani, C. Longo, H. Soyer, S. Gonzalez, R. Hofmann-Wellenhof, V. Ahlgrimm-Siess, P. Guitera, C. Sinz, and H. Kittler, “The value of reflectance confocal microscopy in diagnosis of flat pigmented facial lesions: a prospective study,” *Journal of the European Academy of Dermatology and Venereology*, vol. 31, no. 8, pp. 1349–1354, 2017.
- [27] “How to balance a dataset in python.” [Online]. Available: <https://towardsdatascience.com/how-to-balance-a-dataset-in-python-36dff9d127041>. Accessed: 2022-03-28.