UNIVERSITY OF
KWAZULU-NATAL
INYUVESI
YAKWAZULU-NATALI

School of Electrical, Electronic, and Computer Engineering

ENEL4ED

Electronic Design Project

Phase 3 Report

# Skin Cancer Diagnosis System

Student Name: Z. Mazibuko

Student No.: 214548660

## Abstract

This report shows the use of a deep learning-based skin diagnosis system through the use of raspberry pi with the intention of using the diagnose skin cancer in remote areas for those who cannot afford the make it to medical facilities. The MobileNet pre-trained model was incorporated for its lightweight framework and extensive capabilities in computer vision, the HAM10000 dataset was used to train, test and validate the model before implementation on the raspberry pi. The model performed well and exceeded the accuracy of 83% which is reported in literature as the accuracy of dermatologists, other metrics such as top 2 accuracy, top 3 accuracy and F1-score were used and exceeded expected results. This means the system could be used as a self-examination tool before consulting a dermatologist.

# Table of Contents

Declaration ..... **Error! Bookmark not defined.**
Acknowledgements ..... **Error! Bookmark not defined.**
Abstract ..... ii
Table of Figures ..... iv
1. Introduction ..... 1
2. Motivation and Problem Statement ..... 1
2.1 Problem Statement ..... 2
2.2 Complexity of Design ..... 2
2.3 Feasibility Study ..... 2
2.4 Development Platforms ..... 3
2.5 Scheduling Feasibility ..... 3
3. Theory ..... 4
3.1 Types of Skin Cancer ..... 4
3.1.1 Melanoma ..... 4
3.1.2 Non-melanoma ..... 4
3.2 Deep Learning ..... 4
3.2.1 Convolution Neural Networks ..... 4
3.2.2 MobileNet Architecture and Operation ..... 5
3.3 Deep Learning Based Skin Cancer Solutions ..... 8
4. System Model and Implementation ..... 8
4.1 Dataset ..... 8
4.2 Data Pre-processing ..... 10
4.3 Data Augmentation ..... 11
4.4 Training ..... 12
4.5 Model Evaluation ..... 13
5. Results ..... 14
6. Discussion and Improvements ..... 17
7. Conclusion ..... 18
References ..... 19
Appendix A ..... 22
Appendix B ..... 23
Appendix C ..... 24
Appendix D ..... 25

# Table of Figures

Figure 1: CNN Basic Architecture [18] ..... 4
Figure 2: Depthwise Convolution [22] ..... 5
Figure 3: Standard Convolution Layer vs Depthwise Separable Convolutions [23] ..... 6
Figure 4: MobileNet Architecture [20] ..... 7
Figure 5: MobileNet Model Breakdown [23] ..... 7
Figure 6: Sample images from HAM10000 dataset for cancer types (a) Actinic Keratosis (b) Basal Cell Carcinoma (c) Benign Keratosis (d) Dermatofibroma (e) Melanocytic nevi (f) Melanoma (g) Vascular Lesions [25] ..... 9
Figure 7: Metadata "head" Showing First 5 Entrances of HAM10000 Dataset ..... 10
Figure 8: Testing for Duplicates ..... 10
Figure 9:MobileNet Architecture used in the current study for the classification of skin lesion image among seven skin cancer types [25] ..... 11
Figure 10: MobileNet Setup Using Sequential Class of the Keras Library ..... 13
Figure 11: Testing Results ..... 14
Figure 12:Validation Results ..... 14
Figure 13: Accuracy vs Epochs for Training and Validation ..... 15
Figure 14: Training and Validation Losses ..... 15
Figure 15:Confusion Matrix ..... 16
Figure 16: Melanoma Result ..... 16
Figure 17: Nevi Result ..... 17
Figure 18: Benign Result ..... 17

# 1. Introduction

Human skin is the largest organ on the body and protects it from infections, injury and heat; it also stores fat, vitamins and water [1]. As global warming continues, the depletion of the ozone layer results in more solar ultraviolet radiation reaching the surface which increases the chances of getting skin cancer [2]. There are 3 types of skin cancers, namely, Squamous Cell Carcinoma (SCC), Basal Cell Carcinoma (BCC) and Melanoma. Melanoma is the deadliest of the 3 skin cancers and they all present as moles or skin marks that are irregular [1].

Over the years the number of both melanoma and non-melanoma cancer cases has been increasing worldwide and in South Africa the numbers are said to be extremely under reported to the National Cancer Registry (NCR) [3]. Over 20 000 South Africans were diagnosed with non-melanoma skin cancers (basal cell carcinoma and squamous cell carcinoma) in 2014 with more than 1500 being diagnosed with melanoma [4]. The American Cancer Society states that the average 5-year relative survival rate for skin cancer for all SEER (Surveillance, Epidemiology, and End Results) stages combined is 92% [5].

About 80% of South Africa's population is served by public health care facilities which are under-resourced and over-crowded because of the financial burden medical attention means for non-medical aid holders [6]. The remaining 20% of the population receives top-tier medical attention in the private sector as they hold medical aid and majority of these holders are white or coloured, this enables this population to visit the doctor frequently and receive world-class medical care. 60% of white/coloured adults visit the doctor frequently and 81% of them receives their health care at private hospitals whilst the African population rivals this with 44% doctor's visitation and 34% of these being in the private sector [7]. Cheaper ways to give diagnosis on serious illnesses are desperately needed in order to encourage the African population to visit doctor's and see value in this before dealing with the challenges of visiting a healthcare facility with all its troubles in the public sector.

Deep learning is a form of learning that enables the computer to learn from experience and understand the data that is fed to it [8]. Dermatologists use a non-invasive method called dermoscopy that is far superior to naked eye tests to check for skin cancer [9]. This method involves taking high resolution images and analysing them with greater depth. This method allows for artificial intelligence to be used in order to diagnose patients. Neural networks have been recorded with accuracy levels of up to 98% [9]. Although deep learning cannot replace dermatologists, yet, it can be used as the first line of skin cancer diagnosis in patients that are generally not keen on visiting the doctor.

This report proposes the use of deep learning to diagnose skin cancer through the use of various software and datasets to train and test neural networks for high accuracy diagnosis with low costs and easy operation.

# 2. Motivation and Problem Statement

With the rise of implementation and reliability of deep learning neural networks, we seek to exploit their high accuracy and use them to diagnose skin cancer. This involves the development of self-learning computer programs through which, when exposed to new data, they grow and change for the better. The advantage of using deep learning for skin cancer diagnosis is that once trained their models can be used in smaller scales where they can be applied in phone applications, embedded systems and many other portable devices that could aid in providing high accuracy diagnosis at the fraction of the price of visiting a dermatologist.

With low rates of doctor visits among African natives because of various reasons such as high travel costs to attend health care, out-of-pocket cost burden, long queues, perceived disrespectful treatment

by facility staff, medicine stock-outs, perceived ineffective care and a preference to see traditional healers, there is a need for software intervention [10]. As an African I saw an opportunity to build a cheap solution to aid the poorest of the poor access great technology even in the most remote places in South Africa.

The Independent Clinical Oncology Network in South Africa estimates that, depending on the type of cancer, treatment can cost anything between R10 000 and R1 million per patient, per year. Costs of new and innovative cancer treatment, advancing technologies and innovative surgical procedures are the leading cause for increasing cost in the oncology field [11]. These high costs are coupled with the reasons Africans have to not get medical attention, the need for a method to get medical help without the burden of high costs and the hardships that come with accessing healthcare facilities.

## 2.1 Problem Statement

Skin cancer diagnosis helps save lives and the earlier it is done the better the odds are that a patient will survive and the lower the cost of treatment, if any is needed. In a third world country the cost of getting medical aid could be the same throughout the world but will be more taxing because of the implications it has on the average person. The need for a way to diagnose skin cancer early and cost effectively is of great importance.

Deep learning can be used to diagnose skin cancer. Suspected cancerous areas will be taken in using a camera and then through a trained model to determine whether the patient has skin cancer or not. An embedded system is to be incorporated to allow for easy use of the software. The system's performance will be measured against the diagnosing accuracy of dermatologists found in literature and if possible actual dermatologists.

## 2.2 Complexity of Design

In order to successfully diagnose skin cancer using deep learning we must incorporate various fields of study. This is because the solution is not clear with a universal solution, fields such as deep learning, mathematics and image processing are needed. Images in the dataset should be pre-processed and split into their respective folders (training, testing and validation) before they can be passed on to the neural network. The project is also expected to be completed within the set time with the limited resources and fixed budget.

## 2.3 Feasibility Study

The focus of the project is to diagnose skin cancer using deep neural networks and use cheap methods to deliver it to the general public.

- **Model Architecture Design**

  The model is from pre-existing neural network architectures. The model is expected to take in a photograph of a skin abnormality and output a class in which it belongs (melanoma, benign, etc)

- **Training and Data Preparation**

  Data preparation involves acquiring the data and getting it ready for model training. For data preparation we consider skin cancer datasets that can be used for training, testing and model validation.

- **Validation and Testing**

  In this stage of the project we validate the model by using images the model has not seen before and checking if it accurately classifies the images after training.

- **Performance Measurement**

  Performance measurement is done by using subjective and objective methods to test how well the model performs.

## 2.4 Development Platforms

The system uses open source software that runs on home computers which means it is possible to run on most computers. The programs are downloadable online so with an internet connection one could have all the tools needed to run the program. Although training requires GPU which are both expensive and not generally pre-installed in non-gaming computers, Google Colab/Kaggle can be used to get through this problem as it offers GPU acceleration for free (12 hours and 40 hours respectively).

- **TensorFlow**

  TensorFlow is an end-to-end open source machine learning platform [12]. The TensorFlow platform has a comprehensive, flexible ecosystem of tools and libraries that allow for fast build and deployment of machine powered applications. TensorFlow is one of the best libraries for deep learning implementation.

- **Python**

  The choice of programming language that is easily compatible with TensorFlow is Python. For beginners, Python is the ideal option to focus on to leap into the field of machine learning and data science. It is a minimalist and considerably decreases the time needed to get from building to deploying deep learning models.

- **Keras**

  Keras is an open source deep learning python library that focuses on enabling fast experimentation. The Keras library is used to allow for fast prototyping, supports convolution neural networks and runs on both CPU and GPU platforms.

- **Jupyter Notebook**

  The Jupyter Notebook is an open-source web application that allows for the creation and sharing of documents that contain live code, equations, visualizations and narrative text [13].

- **Kaggle**

  Kaggle is a free cloud service that offers free GPU and TPU acceleration. It runs using Jupyter Notebooks and offers pre-loaded libraries such as Keras, TensorFlow, OpenCV etc. Kaggle also offers a wide variety of free datasets for use in data science.

- **OpenCV**

  OpenCV is a library of programming functions mainly aimed at real-time computer vision and can be used to access and manipulate images for viewing, editing and other various functions that may be needed in computer vision.

## 2.5 Scheduling Feasibility

The project is expected to run for a total of 14 weeks throughout the semester. The project is broken in to three phases with phase 3 being the final stage. Although phase 3 is the most important in terms of grades, all the other phases have a unique and key milestone that must be accomplished before moving forward. The proper scheduling of this design project can be seen on the Gantt Chart in Appendix A.

# 3. Theory

## 3.1 Types of Skin Cancer

### 3.1.1 Melanoma

Melanoma is a type of skin cancer that occurs when the pigment producing cells called melanocytes mutate and start to divide uncontrollably, although it is not the most common skin cancer, it is the most dangerous as it accounts for 80% of all skin cancer deaths [14] [15].

### 3.1.2 Non-melanoma

Non-melanoma cancers encapsulate all other types of skin cancers but they are generally used to describe BCC and SCC cancers, non-melanoma cancers account for 86% of all diagnosed skin cancers [16] [15].

## 3.2 Deep Learning

Deep learning is a combination of many non-linear functions to model the complex dependency between input features and labels. While neural networks have a long history, recent advances have greatly improved their performance in computer vision, natural language processing, etc. Compared to machine learning, deep learning has strong learning ability and makes better use of datasets. Although deep learning has progressive and stern dominance over machine learning, it still has limitations.

### 3.2.1 Convolution Neural Networks

Modern Convolutional Neural Networks (CNNs) obey similar laws of structure. They comprise primarily of several kinds of layers, including layers of convolution, activation layers, layers of pooling, layers of normalization and layers of complete connection [17]. Either these layers are concatenated or juxtaposed, creating complicated neural networks. Convolutional neural networks are usually composed by a set of layers that can be grouped by their functionalities.

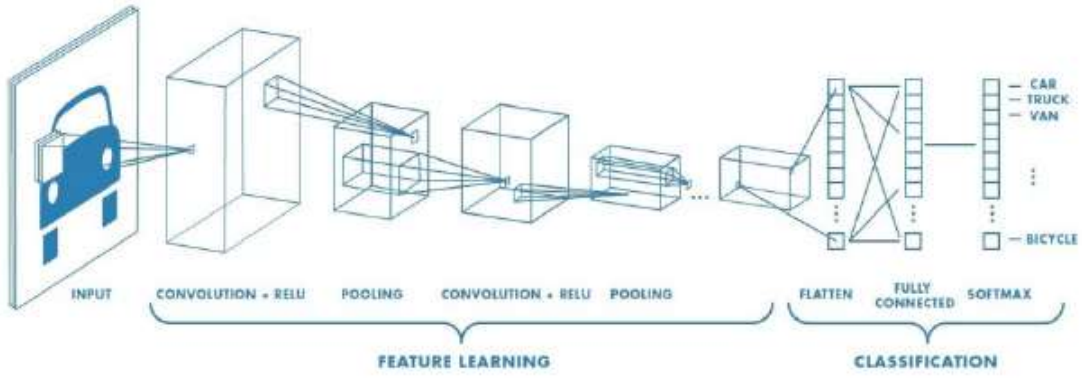



*Figure 1: CNN Basic Architecture [18]*

- **Convolution Layer**

Convolutional layer includes several filters that need to learn the parameters. The filters height and weight are usually lower than the quantity of the sample. Each filter is converted to the input quantity to calculate a neuron activation map. The input of a convolutional layer can be the input image or the output of a preceding layer. Convolutional layers in a CNN systematically apply learned filters to input images in order to create feature maps that summarize the presence of those features in the input [19]. The process is a 2D convolution on the inputs. Filter weights are shared across receptive fields [17]. The filter has same number of layers as input volume channels, and output volume has same depth as the number of filters.

- **Activation Layer**

Used to increase non-linearity of the network without affecting receptive fields of convolutional layers [17]. The commonly used activation function is the ReLU, which results in faster training.

- **Pooling Layer**

Since convolutional layers provide activation maps. Pooling layer applies non-linear down-sampling on these activation maps. To create a new set of the same number of pooled feature maps, the pooling layer operates separately on each feature map [17].

- **Fully Connected Layer**

The fully connected layer is a neural network that is typically used in a convolutional neural network as the final step. Mainly used for image classification, where the required result is an array of elements comprising picture probabilities of a specific class.

Fully convolutional means that the neural network consists of convolutional parts that are generally discovered at the end of the network without any fully connected parts, a CNN with completely link layers that can be learned as end-to-end as a completely convolutionary one. The primary distinction is that learning filters everywhere is the fully convolutional net. Even the levels of decision making at the network's start are filters.

A fully convolutionary neural network attempts to learn representations and create decisions depending on feedback from local input [17]. The addition of a fully linked layer allows the network to study something using global data in which the spatial structure of the entry drops and does not need to be applied.

### 3.2.2 MobileNet Architecture and Operation

MobileNet is a lightweight deep convolution neural network with a streamlined architecture that incorporates depthwise separable convolutions and provides an efficient embedded system and mobile application range for computer vision. Depth wise separable convolution is composed of both point convolution filters and depthwise convolution filters [20]. Depthwise convolution saves computational power by transforming an image once and elongating it along its channels instead of transforming the image by the number of channels. Although there is the disadvantage to depthwise convolution if the neural network is too small, the MobileNet is sufficiently big enough to take advantage of its efficiency without the downside of losing effectiveness due to fewer parameters [21].

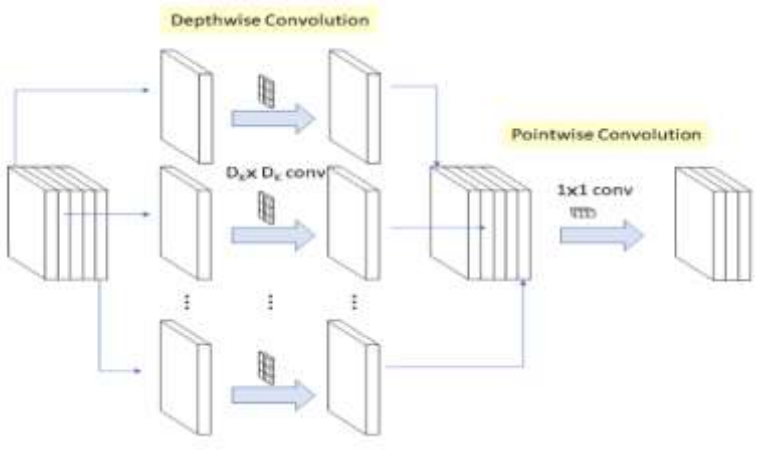



*Figure 2: Depthwise Convolution [22]*

Figure 2 shows the channel-wise $D_k$ x $D_k$ spatial convolution in the depthwise convolution layers. If the layers are thought to have 5 channels, we would view the $D_k$ x $D_k$ convolutions as 5 $D_k$ x $D_k$ spatial convolutions. The pointwise convolutions are dimensional changers and are performed by 1x1 convolutions. The operational cost for a depthwise convolution compared to a normal/standard convolution can be represented by

$$\frac{D_K * D_K * M * D_F + M * N * D_F * D_F}{D_k * D_k * M * N * D_F * D_F} = \frac{1}{N} + \frac{1}{D_K^2} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

Where:

$D_K$ = Kernel size

$M$ = Number of input channels

$N$ = Number of output channels

$D_F$ = Feature map size

And when we set $D_k$ x $D_k$ to 3x3, the computational power needed to perform a depthwise convolution compared to a normal one can be 8 to 9 times less with a small loss in accuracy [22].

The 25-layer neural network has all its convolution layers followed by batch normalization and ReLU nonlinearities with the exception of the final fully connected layer because it feeds into the softmax classification layer. The pattern of these layers can be seen on the right-hand side of Figure 2. The 3x3 depthwise convolution is fed into the batch normalization layer which standardizes the inputs so as to stabilize the learning process and reduce the number of training epochs required to train the network. From the batch normalization layer, we go to the ReLU activation function which is used to solve the problem of vanishing gradient which is an unstable behaviour that occurs during the training segment of the neural network because of useful gradient information not being passed from the output layer properly. The 1x1 convolution is the last layer before the repetition begins and it behaves as the dimensional changer before batch normalization occurs again.

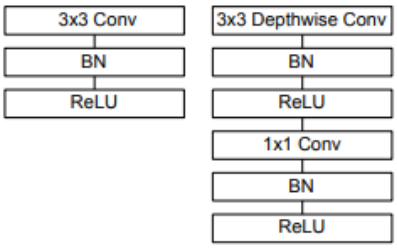



*Figure 3: Standard Convolution Layer vs Depthwise Separable Convolutions [23]*

Other layers exist within the MobileNet neural network that include global average pooling and softmax. The global average pooling is used to replace fully connected layers in classical convolution neural networks, to minimize overfitting by means of decreasing the number of parameters in the model and to smooth out the image for the final stage of classification. The softmax layer changes the vector generated by the neural network into a probability which can be used to train the neural network to know what probability matches the label fed into it and what it could be when we test. Both these layers can be seen in Figure 4.

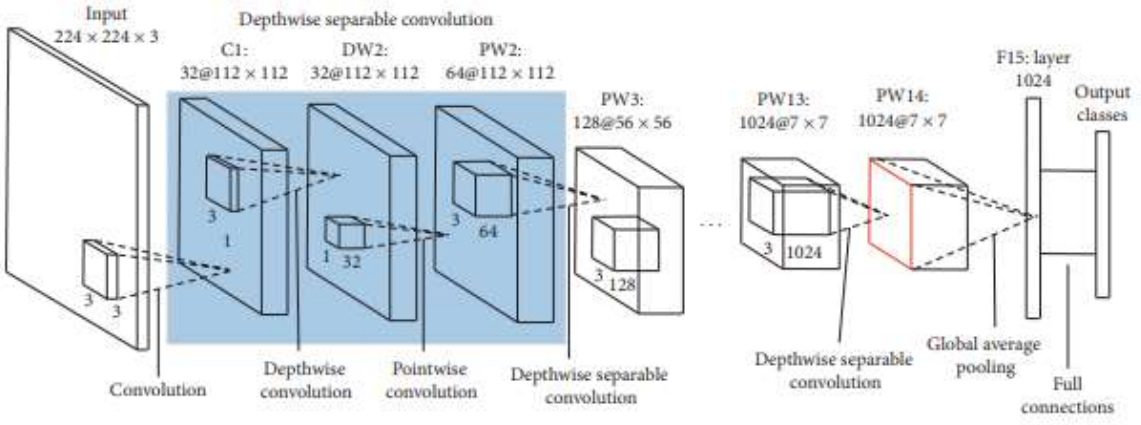



*Figure 4: MobileNet Architecture [20]*

The flexibility of MobileNet neural networks is advent by the ability to make the model thinner and reduce representation by changing the width and resolution multipliers, represented by α and $\rho$ respectively. The respective representation of the cost of computational power for are seen in equations (2) and (3).

$$D_K \text{ x } D_K \text{ x } \alpha M \text{ x } D_F \text{ x } D_F + \alpha M \text{ x } \alpha N \text{ x } D_F \text{ x } D_F \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

$$D_K \text{ x } D_K \text{ x } \alpha M \text{ x } \rho D_F \text{ x } \rho D_F + \alpha M \text{ x } \alpha N \text{ x } \rho D_F \text{ x } \rho D_F \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3)$$

Where α and $\rho \in (0,1]$. The α multiplier creates a smaller and a faster model for certain application requirements, the multiplier plays the role of reducing the width of the layers in the neural network by scales of 0.25, 0.5, 0.75 and 1; the default setting for this parameter is 1. This parameter reduces the computational power with the number of parameters by a factor of $\alpha^2$. As with the width multiplier, the resolution multiplier can reduce the computation power and the parameters by a factor of $\rho^2$. The resolution multiplier reduces the input resolution of the network to 224, 192, 160 or 128. The default MobileNet input is 224 when ρ = 1 [23].

Table 1. MobileNet Body Architecture

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | 3 × 3 × 3 × 32 | 224 × 224 × 3 |
| Conv dw / s1 | 3 × 3 × 32 dw | 112 × 112 × 32 |
| Conv / s1 | 1 × 1 × 32 × 64 | 112 × 112 × 32 |
| Conv dw / s2 | 3 × 3 × 64 dw | 112 × 112 × 64 |
| Conv / s1 | 1 × 1 × 64 × 128 | 56 × 56 × 64 |
| Conv dw / s1 | 3 × 3 × 128 dw | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 128 | 56 × 56 × 128 |
| Conv dw / s2 | 3 × 3 × 128 dw | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 256 | 28 × 28 × 128 |
| Conv dw / s1 | 3 × 3 × 256 dw | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 256 | 28 × 28 × 256 |
| Conv dw / s2 | 3 × 3 × 256 dw | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 512 | 14 × 14 × 256 |
| 5× Conv dw / s1 | 3 × 3 × 512 dw | 14 × 14 × 512 |
| 5× Conv / s1 | 1 × 1 × 512 × 512 | 14 × 14 × 512 |
| Conv dw / s2 | 3 × 3 × 512 dw | 14 × 14 × 512 |
| Conv / s1 | 1 × 1 × 512 × 1024 | 7 × 7 × 512 |
| Conv dw / s2 | 3 × 3 × 1024 dw | 7 × 7 × 1024 |
| Conv / s1 | 1 × 1 × 1024 × 1024 | 7 × 7 × 1024 |
| Avg Pool / s1 | Pool 7 × 7 | 7 × 7 × 1024 |
| FC / s1 | 1024 × 1000 | 1 × 1 × 1024 |
| Softmax / s1 | Classifier | 1 × 1 × 1000 |

*Figure 5: MobileNet Model Breakdown [23]*

Figure 5 shows the architecture of the MobileNet neural network, the input is defined by 224x224x3 which is an input image of 224x224 dimensions with RGB dimensions. The model follows a pattern showed and explained in Figure 3. The convolution layers are a combination of point and depthwise convolutions with varying strides. The last 3 layers are the final stages of the network and they serve the purpose of summarizing the weights that have gone through the network, retrieving weights from all the nodes and then converting the weights to a probability for classification.

### 3.3 Deep Learning Based Skin Cancer Solutions

Rezvantalab, Safigholi and Larimijeshni (2018) checked for the effectiveness and the capabilities of convolution neural networks in the classification of various skin cancers. They used 4 pretrained models on a dataset that has 10 135 dermoscopy images from 4 different datasets that include both melanoma and non-melanoma skin cancers. They compared the accuracy of the pre-trained models against highly trained dermatologists on the same set of images. They found that all four pre-trained models outperformed dermatologists and showed great promise for further application in dermatology practice [9].

Gupta et al. proposed the use of simple algorithms to increase the accuracy and efficiency of deep learning systems for skin cancer diagnosis. They used dermascopy images from one data set with a combination of pre-processing methods on the MATLAB platform. They trained their model to not only show what type of skin cancer is diagnosed, but also whether it is benign, to be melanoma or melanoma. They did not check their findings against dermatologists for accuracy testing [1].

Goyal et al. propose that although artificial intelligence systems outperform dermatologists, it is not as clear cut as it is made to seem. They propose that there be improvements made to neural network systems by having balanced datasets and selection of cases, data augmentation of and diverse datasets among other things. These proposed methods not only improve diagnosis chances on a wider range of skin cancers, but they show how deep learning is still far from being adopted for use by dermatologists in their everyday practice [24].

Chaturvedi, Gupta and Prasad made use of a pre-trained MobileNet network to create a multi-class skin cancer classifier using the HAM10000 dataset. They had high accuracy and even higher top2 and top3 accuracy. They used data augmentation and data pre-processing to balance the HAM10000 dataset to fit more general cases that are not in the dataset [25].

Due to time constraints and best implementation viability within that time, the method of Chaturvedi et.al. was selected for implementation.

## 4. System Model and Implementation

The chosen solution for this project is from Chaturvedi et.al. and is based on a multiclass classification neural network. They used a pre-trained MobileNet through the Keras platform. They used an unbalanced HAM10000 dataset that has duplicates which can alter the performance of the model. There needs to be data pre-processing and data augmentation to ensure a good dataset for optimal model performance.

### 4.1 Dataset

Tschandl et.al. sought to tackle the issue of there being fewer than necessary for dermatoscopy images for training neural networks for automated diagnosis of pigmented skin lesions. They did this by collecting data dermatoscopy images and releasing the “Human Against Machine with 10000 training images” dataset, commonly called the “HAM1000” dataset, the dataset consists of over 50% confirmed lesions by pathologists [26]. Although the initial goal of this project is to aid the darker skinned individuals diagnose skin cancer, the availability of datasets for such people is practically non-existent

as various reasons such as small number of cases within the community and lack of healthcare. The HAM10000 dataset has enough images to prove that the concept is viable and can be applied once the project has been completed. The dataset consists of 100015 dermoscopy images that are broken down to 7 groups that can be seen below.

**Table 1.** HAM10000 lesion groups and number of lesions per group

| Lesion Group | Number of Images |
|---|---|
| Melanocytic nevi | 6705 |
| Melanoma | 1113 |
| Benign keratosis | 1099 |
| Basal cell carcinoma | 514 |
| Actinic keratosis | 327 |
| Vascular | 142 |
| Dermatofibroma | 115 |

The images have 600x450x3 dimensions that were checked using the shape of the image which returns the height, width and dimensionality (RGB or B&W) and sample images can be seen below.

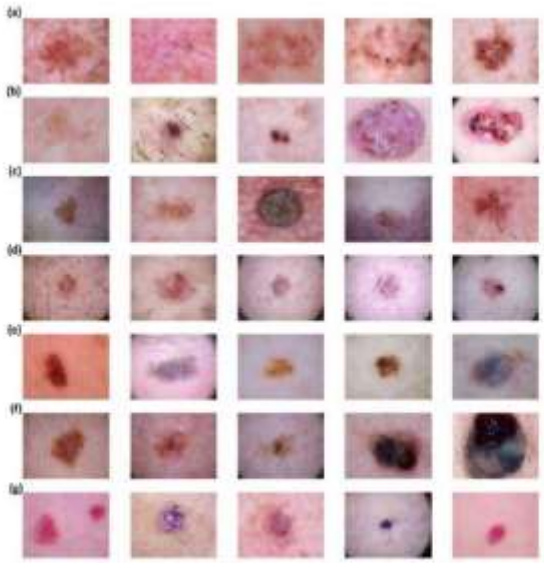


*Figure 6: Sample images from HAM10000 dataset for cancer types (a) Actinic Keratosis (b) Basal Cell Carcinoma (c) Benign Keratosis (d) Dermatofibroma (e) Melanocytic nevi (f) Melanoma (g) Vascular Lesions [25]*

This data is full of redundant images that alter the training of the neural network, so they need to be removed and this is done by using the metadata's "lesion_id" which is a unique combination of letters and numbers that identify each image in the dataset. Through the use of the pandas library we can check the metadata of the dataset and see the details such as lesion ID, image ID, diagnosis, diagnosis confirmation method, age, sex and localization (where the lesion was found on the body). We can check the metadata head by using the .head() method from the pandas library which shows the first 5 entries of the dataset when it is not given a specific integer parameter on where to start showing results. The default result of .head() for the metadata is shown in the next page.

| | lesion_id | image_id | dx | dx_type | age | sex | localization |
|---|---|---|---|---|---|---|---|
| 0 | HAM_0000118 | ISIC_0027419 | bkl | histo | 80.0 | male | scalp |
| 1 | HAM_0000118 | ISIC_0025030 | bkl | histo | 80.0 | male | scalp |
| 2 | HAM_0002730 | ISIC_0026769 | bkl | histo | 80.0 | male | scalp |
| 3 | HAM_0002730 | ISIC_0025661 | bkl | histo | 80.0 | male | scalp |
| 4 | HAM_0001466 | ISIC_0031633 | bkl | histo | 75.0 | male | ear |

*Figure 7: Metadata "head" Showing First 5 Entrances of HAM10000 Dataset*

The diagnosis in the dataset are distinguished as follows:

1. Melanoma (mel)
2. Melanocytic nevus (nv)
3. Basal cell carcinoma (bcc)
4. Actinic keratosis/ Bowens disease (akiec)
5. Benign keratosis (bkl)
6. Dermatofibroma (df)
7. Vascular Lesions (vasc)

The images were cleaned by the data collectors when they removed that we obscured by garments, tattoos and jewellery. The data is heavily skewed towards the melanocytic nevi class which has two thirds of the dataset within its class, this needs to be rectified in order to have a well-balanced multi-class classification model that does not overfit for one class. The duplicated images are removed by using a Boolean value of true if there is more than one lesion ID for a single image and a false when the it does not exist.

```
if lesion_id_cnt[id] > 1:
    return True
else:
    return False
```

*Figure 8: Testing for Duplicates*

The metadata has a “duplicate” mapping added to it for each image so we can identify which images need to be removed. Through this method the data has more unwanted images removed that will alter the performance of the model in a negative way.

## 4.2 Data Pre-processing

The HAM10000 dataset does not have a set training, testing and validation datasets so it must be generated through code. The dataset also has image redundancy, so the images had to be tested for duplicates so that the model does not see the same image multiple times in a single training session because this will alter its performance and reliability. An ImageDataGenerator was used to pre-process the images. The images had to be downscaled 224x224 in order to satisfy MobileNet conditions of operation. The data was split using an 80%, 10% and 10% split ratio between training, testing and validating respectively. The training dataset has 8029 images, the testing dataset has 993 images and the validation dataset has 993 images.

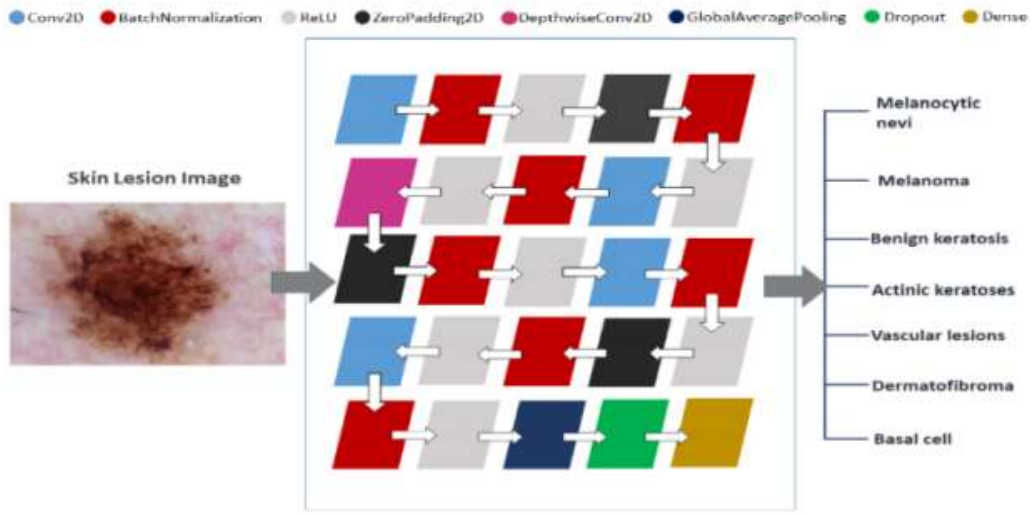



*Figure 9:MobileNet Architecture used in the current study for the classification of skin lesion image among seven skin cancer types [25]*

## 4.3 Data Augmentation

Before the dataset can be used, there needs to be a data split because the data is saved on two folders that are not partitioned as testing or training. The data is split as 80% training, 10% testing and 10% validation. Majority of the images go towards training because we want as much data to teach the neural network as possible whilst leaving enough to objectively check if it was trained well. Non-duplicated images are used, and the split is done by first adding a "has_duplicate" column to the metadata. This column is used to identify which images are added to the dataset that will be used for the final dataset to be used. The final dataset proportionalities after removing the duplicates become 80.2% training, 9.92% training and 9.92% validation.

The new images that are kept in new and separate directories so that there is no need to test for duplicates before each batch of images makes it through to the neural network for training, testing or validation. Each image is passed onto its respective folder with the respective labels and this process can be seen in the Appendix (Insert Appendix for copying images and labels). The images are split as 8029 images for training, 993 validation and 993 training data. From this stage we augment the data by randomly rotating, height and width shifting, zooming, horizontal and vertical flipping so as to cover a wider range of input images that will be taken by the Raspberry Pi. Since the environment in which the embedded system will be used cannot be controlled, this not only increases the number of images we can train, test and validate with, it widens the range for a higher accuracy.

The Keras library allows for there to be dynamic creation of data batches with desired data augmentation processes/combinations that are pre-defined. The data generator parameters are

i. rotation range: 180°
ii. width shift range: 0.1
iii. height shift range: 0.1
iv. zoom range: 0.1
v. horizontal flip: True
vi. vertical flip: True

An ImageDataGenerator is used to create new images that have variations from the original lesions and will aid the neural network diagnose lesions that are not well oriented, well-lit and overall, not well taken images. The data is mainly augmented to get the other classes to increase in size, so the neural

network does not overfit for the nevi class. The new combinational classes for the training and validation datasets have sizes that are close to the nevi class and can be seen below.

**Table 2.** New augmented class sizes

| Lesion Group | Number of Images |
|---|---|
| Melanocytic nevi | 6774 |
| Melanoma | 5928 |
| Benign keratosis | 5975 |
| Basal cell carcinoma | 5699 |
| Actinic keratosis | 5711 |
| Vascular | 5582 |
| Dermatofibroma | 4754 |

This allows for a better overall multi-classification model. In multi-class classification there needs to be balance within the training dataset so that no 1 class is seen more than the others because this would cause overfitting for that class only and the trained model would be useless for other classifications. HAM10000 has this imbalance and requires the data to be augmented not only for this reason but to improve the dataset to cover a wider range of images because the end goal involves using an embedded system which will be used in an uncontrolled environment. The images are randomly cropped, rotated, flipped horizontally and vertically, zoomed and filled. This is done on all the smaller groups of data (every class besides melanocytic nevi) and this not only increases (roughly 42000) the dataset but it improves it.

## 4.4 Training

The MobileNet neural network was used not only for its lightweight architecture, but for its easy access through Keras. The model is pretrained on 1 280 000 images with 1000 object classes from the 2014 ImageNet Challenge [27]. The use of generators is most suitable for applications that involve a large dataset because it saves memory, although GPUs are being used to train the neural network, it is still an efficient method of working through the data. Different flow parameters are setup for training, testing and validating. The augmented data is accessed through the data generator which is set through the ImageDataGenerator which is from the Keras pre-processing library.

The Keras library's Sequential feature is used to define and setup the pre-trained MobileNet neural network. The network is setup to have layers of:

i. Dropout
ii. Batch Normalization
iii. ReLU
iv. Dense layers
v. Softmax layer

```
model = Sequential()
model.add(mobilenet_model)
model.add(Dropout(dropout_dense))
model.add(BatchNormalization())
model.add(Dense(256, activation="relu"))
model.add(Dropout(dropout_dense))
model.add(BatchNormalization())
model.add(Dense(7, activation="softmax"))
```

*Figure 10: MobileNet Setup Using Sequential Class of the Keras Library*

The dropout layer is set to have a dropout rate of 0.1 and the dense layers are set to have a size of 256. The softmax layer is set to have a 7-unit activation layer because our dataset has 7 classes to classify. The input images are shaped to have dimensions of 224x224x3 before coming into the neural network, the shape is specified because the "include_top" parameter was set to false which means the top fully connected layer is disabled. Table 3 shows important training parameters used to train the MobileNet neural network.

**Table 3.** Hyper parameters

| Parameter | Value |
|---|---|
| Batch Size | 32 |
| Epochs | 15 |
| Rescale | 1/255 |
| Input | 224x224x3 |
| Output | 7x1 |
| Learning Rate | 1 x e^-5 |
| Optimizer | Adam |

The Keras library offers tracking features such as history, early stopping and model checkpoint. These library features allow for progress analysis during training and also provide an efficiency measure because if the number of epochs is set too high, the "early stopping" feature stops the training if no change in accuracy is observed for a long time. The categorical crossentropy loss function was used with an Adam optimizer, accuracy, top2 and top3 accuracy to evaluate the model's performance after training.

## 4.5 Model Evaluation

The model was tested using accuracy, micro average precision, micro average of recall, and micro average of F1-score. The metrics are represented mathematically by the following expressions:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

$$Precision = \frac{TP}{(TP + FP)} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

$$Recall = \frac{TP}{(TP + FN)} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3)$$

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

Where: *TP* = true positive

*TN* = true negative

*FN* = false negative

*FP* = false positive

## 5. Results

The model was tested and validated using the respective datasets which the model had not seen before. The testing accuracy was found to be 88.32% with the top2 and top3 accuracy being 96.48% and 98.59% respectively.

```
Test accuracy: 0.8832
Test top-2 accuracy: 0.9648
Test top-3 accuracy: 0.9859
Test F1 score: 0.8842
```

*Figure 11: Testing Results*

The validation accuracy was found to be 86.91% with the top2 and top3 accuracy being 94.66% and 97.89% respectively.

```
Validation accuracy: 0.8691
Validation top-2 accuracy: 0.9466
Validation top-3 accuracy: 0.9789
Validation F1 score: 0.6415
```

*Figure 12:Validation Results*

The respective F1-scores are 88.42% and 64.15% for testing and validation respectively. The loss and accuracy curves can be seen below, and they serve the purpose of proving that the model is learning because they are changing with each epoch.

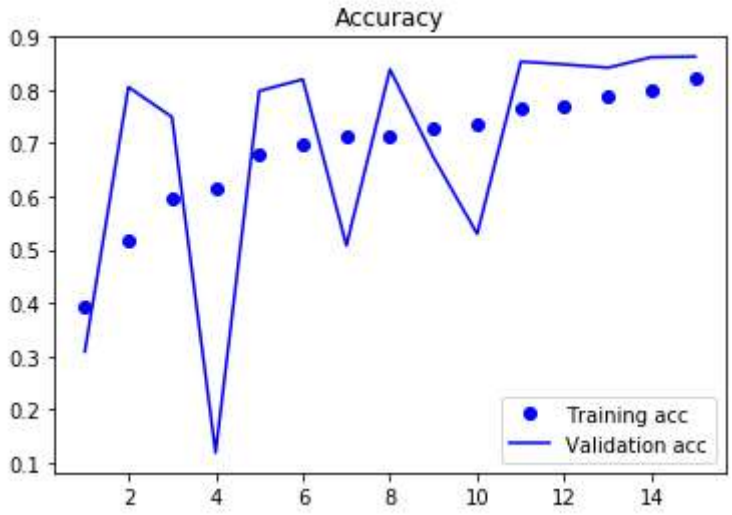



*Figure 13: Accuracy vs Epochs for Training and Validation*

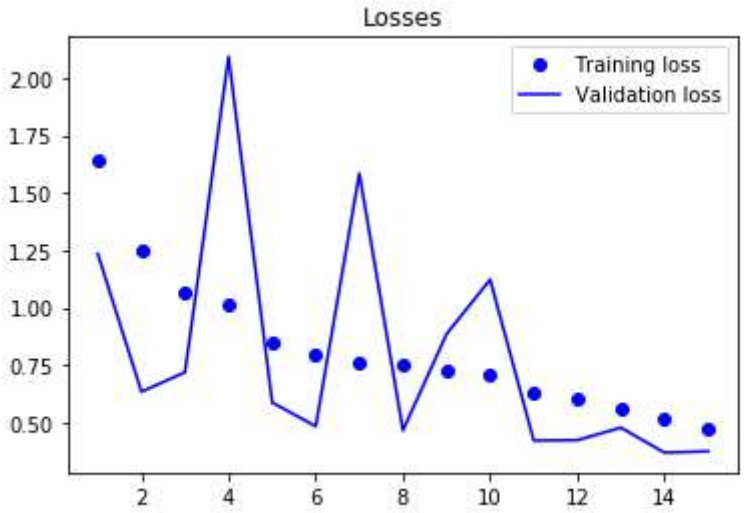



*Figure 14: Training and Validation Losses*

In order to check where the model went wrong, a confusion matrix was used as advised by the literature and the results can be seen below.

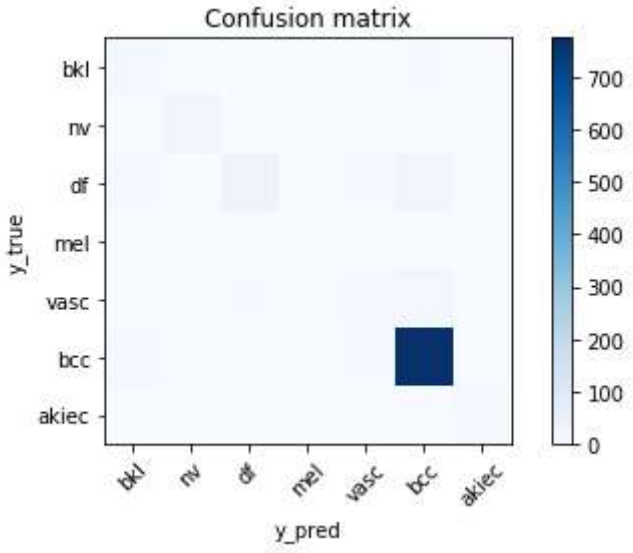



*Figure 15:Confusion Matrix*

For real world application this project is made to show a live video and place a rectangle in the centre where the lesion can be scanned and classified. The rectangle and video are shown on the monitor that is connected to the Raspberry Pi so that the person using it knows where to place the lesion and increase the chance of making the right diagnoses. The video is taken for 5 seconds so the person can orient their lesion correctly and the last frame of the video is stored as a JPEG file so that it can be fed into the model for classification. Since the real-world application includes reading in an image, we can use stored images to classify images from the internet because getting people to test it on is not possible due to Covid.

Firstly, a melanoma image was taken in and fed to the model through the model.setInput() method and the result comes back as 62% melanoma. The second image is of the NV group and comes back with a result of 71% NV and the last image read in is the benign group and has a 58% probability of being benign.


*Figure 16: Melanoma Result*

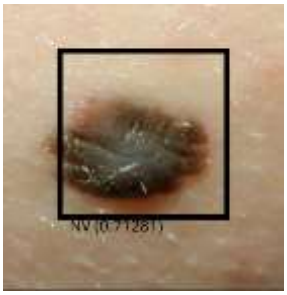


*Figure 17: Nevi Result*

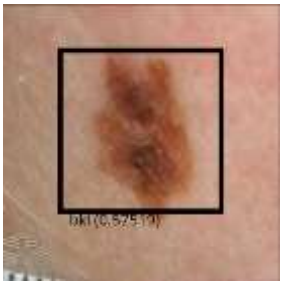


*Figure 18: Benign Result*

Every output of the model is given as a probability/confidence that ranges from 0 to 1 and because of this there is a chance to filter out results that are possible but are most likely not the case so because of this there needed to be a confidence/probability threshold set and it was set to values greater than 0.5. This is done so that the we only take the highest value that is most likely true.

## 6. Discussion and Improvements

Skin cancer is a growing and dangerous problem within society. The early detection of skin cancer is key to saving countless lives and increasing chances of exuberant amounts of money being saved from treatment costs. Deep learning shows promising results and has the chance to be the best diagnostic device in the dermatology field. High accuracy coupled with inexpensive technology is the best solution to not only the common population, but for help to reach the most remote places across the world.

The tested model has great results and was deployed on a Raspberry Pi in order to be used by the general public with ease. Creating a rectangle for correct orientation allows for there to be a higher probability of the user getting their lesion taken correctly and being predicted correctly as well. Although training and validation showed results in terms of accuracy and both top 2 and top 3 accuracies, no significant improvements were seen during training that justified increasing the number of epochs from 15. The accuracy is still great because being able to diagnose 4 out of 5 people (80%) is still a good achievement. Proof of the model learning can be seen in the loss vs epochs graphs because as the epochs increase, the loss decreases until it is stable.

Improvements can be made to the system as the quality of the camera is not great whilst the images taken by it can be auto focused on the lesions and not depend on the patient orienting their lesions correctly for it. The use of different methods of data cleaning to remove hairs, tattoos and any other contaminant on the lesion would require a great deal of time to incorporate but would increase the accuracy and reliability of the project.

Furthermore, live detection of lesions through videos would see the system make multiple diagnosis in a shorter period rather than taking a picture of each lesion before restarting for the next diagnosis.

# 7. Conclusion

Deep learning is a powerful tool that can help in the fight against skin cancer. Admittedly, the path for deep learning making definitive diagnosis is far, the current literature and present works are promising in it being a reality sooner rather than later. The use of neural networks to diagnose skin cancer is already welcome through web and cell phone applications for self-examination and this is due to the progress the field shows and how relatively cheap it is to access.

The model has high accuracy scores as they sit comfortably above 80% which means 4/5 people could be diagnosed correctly, this not only puts it in the region of the reported accuracy of dermatologists, 83%, but it makes it better. Although there remains a need to consult medical professionals on suspicious lesions, it is clear that more work within the field of deep learning promises easier access to medical grade assistance for a fraction of the price.

# References


[1] G. M, S. Gupta and N. S. Ramaiah, "Diagnosis of Skin Cancer Melanoma using Machine Learning," *SSRN,* pp. 1-3, 2019.

[2] " Mayo Foundation for Medical Education and Research (MFMER)," 20 February 2019. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605. [Accessed 6 October 2020].

[3] M. Steyn, H. F. Jordaan, R. Smith, S. Claasens and W. I. Visser, "An Analysis of Biopsies for Suspected Skin Cancer at a Tertiary Care Dermatology Clinic in the Western Cape Province of South Africa," *Journal of Skin Cancer,* vol. 2020, p. 1, 2020.

[4] "The Cancer Association of South Africa," 21 August 2020. [Online]. Available: https://cansa.org.za/be-sunsmart/. [Accessed 6 October 2020].

[5] T. A. C. Society, "American Cancer Society," Survival Rates of Melanoma Skin Cancer, 8 January 2020. [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer/detection-diagnosis-staging/survival-rates-for-melanoma-skin-cancer-by-stage.html. [Accessed 6 June 2020].

[6] "Health NDo. National Health Insurance Green Paper.," Health NDo, Pretoria, 2011.

[7] Z. McLaren, C. Ardington and M. Leibbrandt, "Distance decay and persistent health care disparities in South Africa.," *BMC Health Serv Res.,* vol. 14, p. 541, 2014.

[8] I. Goodfellow, Y. Bengio and A. Courville, in *Deep Learning*, Cambridge, The MIT Press, 2016, p. 1.

[9] A. Rezvantalab, H. Safigholi and S. Karimijeshni, "Dermatologist Level Dermoscopy Skin Cancer Classification Using Different Deep Learning Convolutional Neural Networks Algorithms," *arXiv,* 2018.

[10] B. Harris, J. Goudge, J. Ataguba, D. McIntyre, N. Nxumalo, S. Jikwana and M. Chersich, "Inequalities In Access To Health Care in South Africa," *J Pub lic Health Policy,* vol. 32, pp. 102-123, 2011.

[11] "Discovery," Discovery Limited, [Online]. Available: https://www.discovery.co.za/corporate/understand-cancer-costs-of-cancer. [Accessed 28 November 2020].

[12] "Tensorflow," [Online]. Available: tensorflow.org.

[13] "Jupyter Notebook," no. jupyter.org.

[14] M. MacGill and O. Kramer, "What To Know About Melanoma," Medical News Today, 2019 November 7. [Online]. Available: https://www.medicalnewstoday.com/articles/154322. [Accessed 6 October 2020].

[15] A. J. Miller and M. C. Mihm, "Melanoma," *The New England Journal of Medicine,* p. 1, 2006.

[16] V. Madan, J. T. Lear and R.-M. Szeimies, “Non-melanoma Skin Cancer,” *The Lancet,* vol. 375, no. 9715, p. 673, 2010.

[17] B. Jin, “Computational Aesthetics and Image Enhancements Using Deep Neural Networks,” *Infoscience,* p. 145, 2018.

[18] “Convolution Neural Networks for Visual Recognition,” CS231n, [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed 18 November 2020].

[19] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang and P. Zhou, “EnlightenGAN: Deep Light Enhancement Without Paired Supervision,” *arXiv,* 2019.

[20] W. Wang, Y. Li, T. Zou, X. Wang, J. You and Y. Lup, “A Novel Image Classification Approach via Dense-Mobile Models,” *Hindawi,* vol. 2020, p. 2, 2020.

[21] C.-F. Wang, “A Basic Introduction to Seperable Convolutions,” towards data science, 14 August 2018. [Online]. Available: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728. [Accessed 23 December 2020].

[22] S.-H. Tsang, “Review: MobileNetV1 — Depthwise Separable Convolution (Light Weight Model),” towards datascience, 14 October 2018. [Online]. Available: https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69. [Accessed 23 December 2020].

[23] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision,” *arXiv,* vol. 1, p. 1, 2017.

[24] M. Goyal, T. Knackstedt, S. Yan and S. Hassanpour, “Artificial Intelligence-Based Image Classification for Diagnosis of Skin Cancer: Challenges and Opportunities,” *arXiv,* pp. 6-7, 2019.

[25] S. Chaturvedi, K. Gupta and P. Prasad, “Skin Lesion Analyser: An Efficient Seven-Way MultiClass Skin Cancer Classification Using MobileNet,” *arXiv,* 2019.

[26] P. Tschandl, C. Rosendahl and H. Kittler, “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,” *Scientific Data,* pp. 1-8, 2018.

[27] J. D. H. S. J. K. S. S. S. M. Z. H. A. K. A. K. M. B. A. C. B. a. L. F.-F. Olga Russakovsky*, “ ImageNet Large Scale Visual Recognition Challenge,” *IJCV,* 2015.

[28] S. Grewal and E. Kaur, “Full Resolution Image Compression with Deep Neural Networks,” *International Journal for Research in Applied Science and Engineering Teachnology,* vol. 6, pp. 1125-1129, 2017.

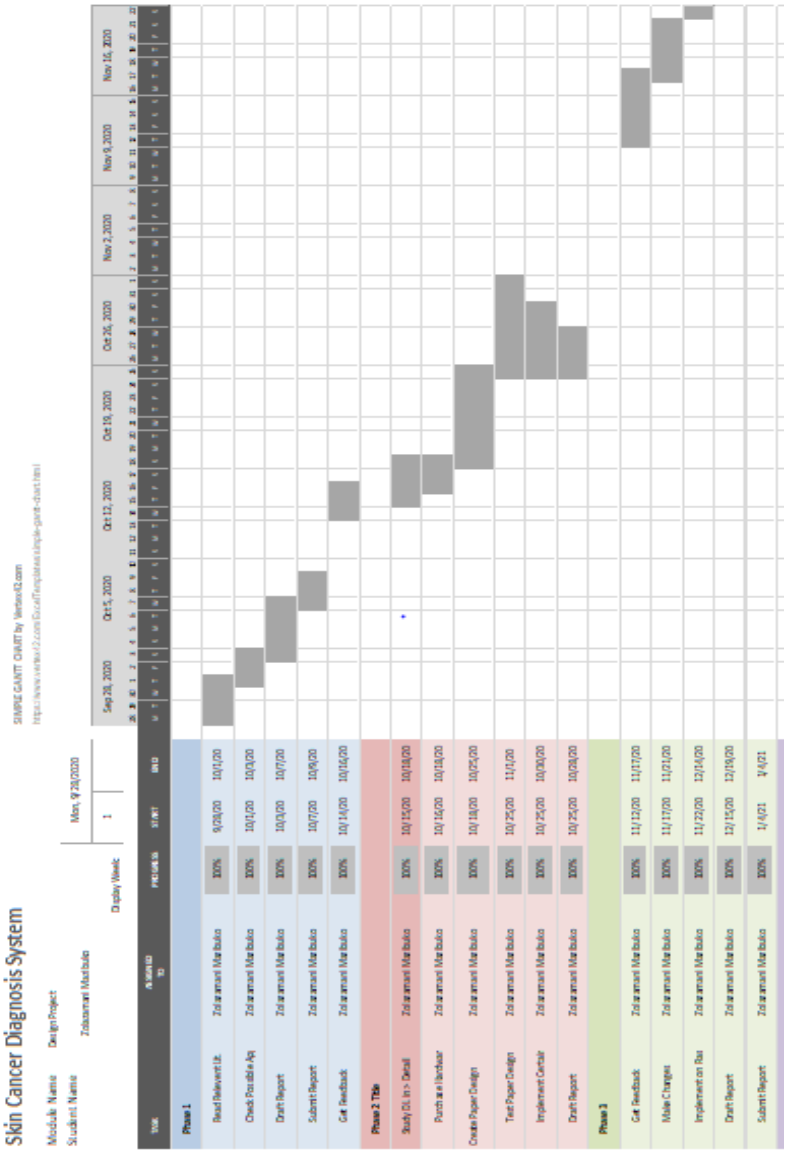
[29] G. E. Hinton and Salakhudinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science,* vol. 313, pp. 504-507, 2006.

[30] Z. Liu, T. Liu, L. Jiang, J. Xu, Y. Wang and G. Quan, “DeepN-JPEG: A Deep Neural Network Favorable JPEG-based Image Comopression Framework,” *AeXiv: 1803.05788v1,* 2018.

[31] F. Culfaz, “Transfer Learning using Mobilenet and Keras,” towards data science, 6 November 2018. [Online]. Available: https://towardsdatascience.com/transfer-learning-using-mobilenet-and-keras-c75daf7ff299. [Accessed 22 November 2020].

# Appendix A

Gantt Chart



# Appendix B

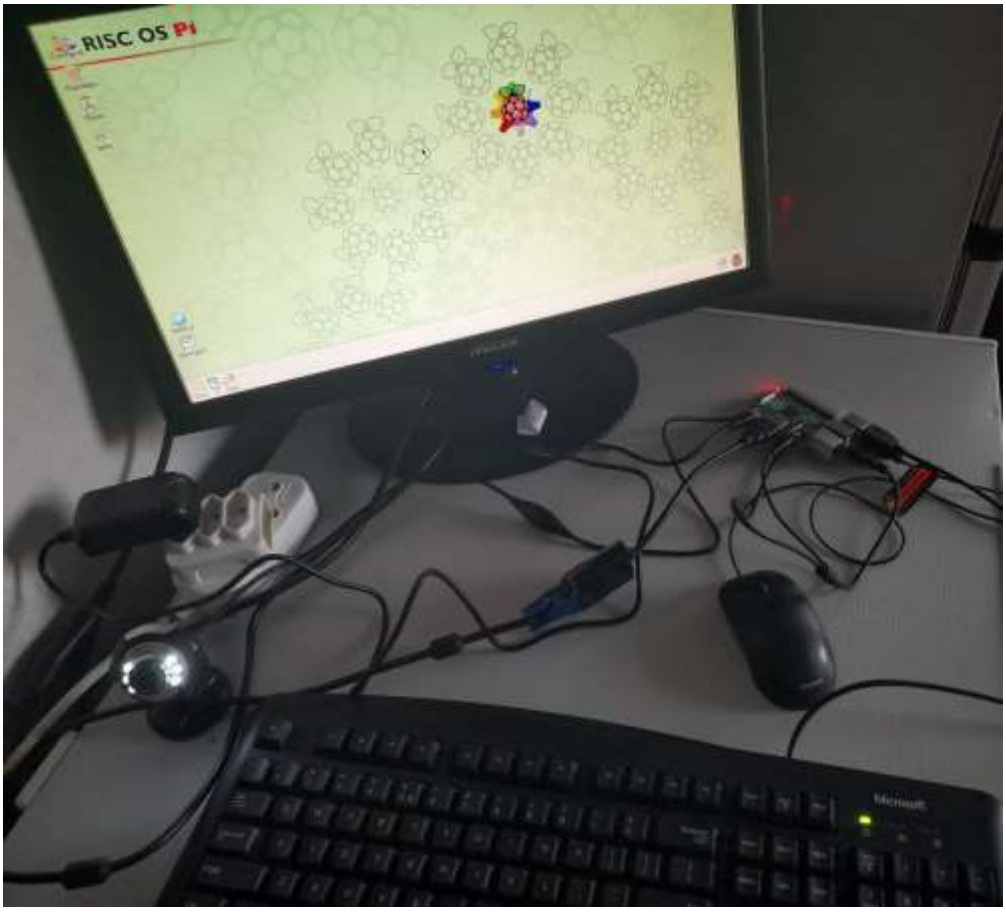
Proof of Training

```
Epoch 00001: val_categorical_accuracy improved from -inf to 0.30847, saving model to model.h
5
Epoch 2/15
250/250 [==============================] - 146s 583ms/step - loss: 1.2470 - categorical_accu
racy: 0.5168 - top_2_acc: 0.7290 - top_3_acc: 0.8668 - val_loss: 0.6343 - val_categorical_ac
curacy: 0.8054 - val_top_2_acc: 0.9147 - val_top_3_acc: 0.9625

Epoch 00002: val_categorical_accuracy improved from 0.30847 to 0.80541, saving model to mode
l.h5
Epoch 3/15
250/250 [==============================] - 150s 599ms/step - loss: 1.0685 - categorical_accu
racy: 0.5948 - top_2_acc: 0.7910 - top_3_acc: 0.9016 - val_loss: 0.7194 - val_categorical_ac
curacy: 0.7492 - val_top_2_acc: 0.8762 - val_top_3_acc: 0.9511

Epoch 00003: val_categorical_accuracy did not improve from 0.80541
Epoch 4/15
250/250 [==============================] - 142s 568ms/step - loss: 1.0180 - categorical_accu
racy: 0.6149 - top_2_acc: 0.8085 - top_3_acc: 0.9146 - val_loss: 2.0929 - val_categorical_ac
curacy: 0.1186 - val_top_2_acc: 0.4631 - val_top_3_acc: 0.7752

Epoch 00004: val_categorical_accuracy did not improve from 0.80541
Epoch 5/15
250/250 [==============================] - 143s 571ms/step - loss: 0.8528 - categorical_accu
racy: 0.6786 - top_2_acc: 0.8581 - top_3_acc: 0.9416 - val_loss: 0.5876 - val_categorical_ac
curacy: 0.7981 - val_top_2_acc: 0.8866 - val_top_3_acc: 0.9573
```

# Appendix C

Components used

1. Raspberry Pi
2. 10 Megapixel Webcam
3. Keyboard
4. Monitor
5. Mouse



# Appendix D

**Conducted Searches**

1. Skin Cancer
2. Skin Cancer in South Africa
3. Deep Learning Concepts
4. Deep Learning Based Skin Cancer Solutions
5. Modelling Deep Learning Models Using Python
6. HAM10000 dataset/HAM10000 explained (towards datascience)
7. MobileNet Explained/MobileNet towards datascience
8. Application of MobileNet on Raspberry Pi