Evaluation of Deep Learning Methods for Multi-modal Skin Lesion Classification

Mohamed Ramshad Basheer

Supervisors:
Dr. Wanqing Li & Dr. Philiip Ogonbuna

This thesis is presented as part of the requirements for the conferral of the degree:

Bachelor of Computer Science (Honors)

The University of Wollongong
School of Computing and Information Technology

June 2020

Declaration

I, Mohamed Ramshad Basheer, declare that this thesis submitted in partial fulfilment of the requirements for the conferral of the degree Bachelor of Computer Science (Honors), from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Mohamed Ramshad Basheer
September 16, 2023

Abstract

Skin cancer is the most common cancer but also one that is quite easy to prevent with early detection and following proper prevention techniques. Detecting skin cancer however is not an easy task as dermatologists need to be trained for years to be able to make accurate predictions that could have fatal consequences. To aid the dermatologists in this process, many computer aided methods have been proposed in the past. However, they do not meet the accuracy requirement for this sensitive problem. In this study, we focus on the deep learning based solutions to this classification problem. We systematically evaluate the various methods available in this field on a large dataset that contains images from various freely available sources (ISIC 2019, PH2 and Mednode) and study effectiveness of these solutions. Popular architectures such as ResNet, DenseNet and EfficientNet were trained on this dataset to find which performs best. Automatic segmentation techniques proposed before has been put into practice but did not prove to improve the performance. The dataset obtained had very high imbalance problem hence we evaluated the effectiveness of various popular techniques that were previously proven to be effective in tackling this issue such as creating synthetic image using GAN. We then propose a model fusion method which has proven to show the best result.

Acknowledgments

I am grateful to the God for the good health necessary to complete this thesis.
I would like to extend my sincere appreciation for my supervisors Dr. Wanqing Li and Dr. Philip Ogunbona for their support and guidance throughout this work. I would also like to thank the IT staff, especially Mr. Jun Hu for helping me resolve all the technical issues.
Finally, I would like thank my parents for their continued support in all aspects of my life and in acheiving my dreams.

Contents

Abstract **iii**

1 Introduction **1**
1.1 Structure 2

2 Background **3**
2.1 Skin biology 4
2.1.1 Skin Lesion 4
2.1.2 Imaging techniques 6
2.1.3 Diagnostic techniques 6
2.2 Deep Learning 8
2.2.1 Deep Neural Network 11
2.2.2 Testing 12
2.2.3 Convolutional Neural Network (CNN) 13
2.2.4 CNN Architectures 14
2.2.5 Transfer Learning 16
2.2.6 Autoencoders 17
2.2.7 Generative Adversarial Network (GAN) 18
2.3 Available Datasets 19
2.4 Evaluation metrics 20
2.4.1 Sensitivity 20
2.4.2 Specificity 21
2.4.3 Confusion Matrix 21
2.4.4 F1 Score 21
2.4.5 AUC-ROC Graph 22
2.4.6 Class Activation Map (CAM) 23

3 Methods **24**
3.1 Previous work 25
3.1.1 Segmentation 25
3.1.2 Data Imbalance 26
3.1.3 Real life applications 27

3.1.4 Critical Analysis 27
3.2 Overview 28
3.2.1 Data Preperation 28
3.2.2 Training and Evaluation 31
3.3 Experiments 32
3.3.1 Base Model Selection 33
3.3.2 Segmentation 34
3.3.3 Balance data using General Adversarial Network (GAN) . . . 36
3.3.4 Multiple classifier fusion training 37

4 Mobile application 43
4.1 Exploring options 44
4.2 Challenges 45
4.2.1 Lack of dataset 45
4.2.2 Image quality 45
4.2.3 Storage space 45
4.3 Deploying model 45
4.3.1 Optimization 46
4.4 Application functionality 47

5 Conclusion and Future work 49

Bibliography 51

A Appendix One 57
A.1 Importants snippets of code 57
A.1.1 Train Classifier 57
A.1.2 Custom callback for evaluation and saving model 59

B Appendix Two 61
B.1 Model training history 62

List of Figures

2.1 Structure of skin 4
2.2 Examples of each skin lesion classes 5
2.3 ABCDE skin lesion diagnosis technique[11] 7
2.4 Differenct types of ugly duckling diagnosis. Image a shows a lesion that is bigger and darker that stand out from its neighbors. Image b shows a very light lesion that is different and Image c shows a lone lesion with no neighbors. [13] 8
2.5 Comparison of a neuron and an early linear threshold system to showcase the similarity. 10
2.6 Common activation functions. [18] 10
2.7 Architecture of a deep neural network 11
2.8 Convolutional operation using 3x3 kernel on a 7x7 matrix [20] 13
2.9 Residual block in ResNet 15
2.10 DenseNet architecture [24] 15
2.11 EfficientNet scaling method. [25] 16
2.12 Structure of an autoencoder 17
2.13 Architecture of a GAN [27] 18
2.14 An example of how a confusion matrix looks for a muticlass classification model [36] 21
2.15 AUC-ROC graph. Green line shows the ROC graph that plots each threshold value and the shaded area corresponds to area under the curve [37] 22
2.16 An example of a Class Activation Map being applied to a dog image classifier [38] 23

3.1 Melanoma detection device MelaFind© [52] 27
3.2 Overview of system 28
3.3 Removal of black padding around skin lesion 30
3.4 Removal of foreign objects in images 30
3.5 Random cropping of images to resize to 224x224 while maintaining aspect ratio 31
3.6 Augmentation techniques applied to training dataset 32

3.7 Base model selection process 33
3.8 Confusion matrices of base model trained on binary classification . . 35
3.9 Confusion matrices of base model trained on multi-class classification 39
3.10 Grad-CAM 40
3.11 Image that requires segmentation to remove foreign objects [32] . . . 40
3.12 Architecture of BCDU-Net segmentation model used automatically segmenting image [55] 41
3.13 Confusion matrices for segmented image classification 41
3.14 Skin lesion image automatically segmented by BCDU-Net model . . . 41
3.15 Strange artifacts found in images generated by GAN 42
3.16 Comparison of images generated by GAN and image from original dataset 42
3.17 Groupwise training approach 42

4.1 Classification on server architecture illustration 44
4.2 DermLite - an example of dermatoscope attachment for mobile phones. [60] 46
4.3 Mobile application interface 48

B.1 Batch accuracy vs step count of all trained models 62

List of Tables

2.1 7-point Checklist method for skin lesion diagnosis 8

3.1 Count of each classification labels in the final training dataset 29
3.2 Count of each classification labels in the final validation dataset . . . 29
3.3 Count of each classification labels in the final testing dataset 30
3.4 Augmentation applied on dataset before training 31
3.5 Evalutation metrics of base model trained for binary classification . . 34
3.6 Evalutation metrics of base model trained for multi-class classification 34
3.7 Results from binary and multi-class segmented lesion classification . . 36
3.8 Model fusion score for binary classification 38
3.9 Model fusion score for multi-class classification 38

4.1 TfLite file size comparison 47

1 Introduction

Skin cancer is the most common form of cancer found among humans. Australia has the highest incidence of skin cancer in the world with more than 1,500 Australians losing their life to melanoma, the deadliest form of skin cancer each year [1]. Early detection through routine checkups is key to fighting skin cancer by preventing it from spreading it to other parts of the body and is easily curable through simple biopsy if found in its early stages [2]. Distinguishing between malignant and benign skin lesions, however, can be difficult, where even experienced dermatologists have differences in opinion when asked to classify the same skin lesions [2]. What this leads to is many unnecessary biopsies such as Punch, Excisional or Incisional biopsies where parts or entire areas of skin are removed for further analysis. This process can be painful and costly, discouraging patients from doing further checkups.

Computer Aided Diagnosis (CAD) especially deep convolutional neural network (CNN) based approaches has been proven to show great results in analysis of medical images that helps with diagnosis. As a result of many previous studies, some CAD systems have achieved sensitivity of up to 90% [3]. Unfortunately, CAD systems still has ways to go before it can replace human diagnosis completely this is due to the fact that the high sensitivity comes with the cost of high false-positive rate. Therefore at this point in time these systems are used mainly to give dermatologists a helping hand in diagnosis [3].

In this study, we compare and analyze existing state of the art CNN based approaches in skin lesion classification and how to improve them. We tackle the different challenges that exists within the problem of skin lesion classification such as imbalanced dataset and lack of data by evaluating different techniques that could aid to resolve such issues.

It is also important that these systems are accessible. It does not matter how good these systems are if it is only available to few dermatologists. Therefore, in this thesis, we also focus on how to bring these systems on to the hands of everyday

people in the form of a mobile application where they can do self-diagnosis regularly. We study and discuss the feasibility and challenges of such products by comparing the different latest technological advancements in this field.

1.1 Structure

This work is divided into different chapters. This section gives a breakdown of each chapter

- Chapter 2 : **Background** gives a brief explanation of the different component that makes up the skin. It then explains the different types of skin lesion that will be of interest in this study. Different processes of imaging and diagnosing skin lesion in practice is also discussed here. It is then followed by the background knowledge needed to understand the machine learning techniques that are applied in this study. It is not an extensive guide on machine learning as it is a very broad field but rather it focuses on working of selected techniques. Finally, we explore the various dataset that exists and looks at the different evaluation metrics that have been used in this field.

- Chapter 3 : **Methods** looks at the various research efforts that has gone into the problem of skin lesion classification over the years. Critical evaluation of different techniques that has been proposed is presented in this chapter. This chapter then presents the plan of action to tackle the research question at hand. It showcases the dataset, the pre-processing techniques applied and deep learning experiments. Along with the experiments, the results obtained for each of them is also presented. It also contains the review and comments on each result obtained

- Chapter 4 : **Mobile application** demonstrates the prototype mobile application designed to bring the deep learning technology to the hands of the user. It discusses the opportunities and challenges of designing the application and distributing them.

- Chapter 5 : **Conclusion and Discussion** gives brief overview of the results obtained and how it can be further developed in the future.

2

Background

In this chapter, different concepts that is required to understand the study are introduced. This chapter begins with a brief description of the anatomy of skin. This includes description of different types of lesion and how to differentiate among them. Few of the most popular dermatological processes involved in skin lesion diagnosis is also presented. It is then followed by the computer science aspect of this study where the machine learning concepts are described. Concepts such as deep learning, convolutional neural networks and other techniques used in machine learning classification are explored. This is then followed by an exhaustive list of all available datasets and their properties. Finally, evaluation metrics by which all of these methods will be compared is described in details as well.

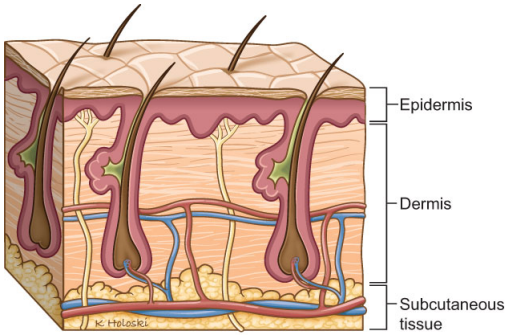


Figure 2.1: Structure of skin

2.1 Skin biology

Skin is the largest organ in the body and it mainly consists of 2 layers as it can be seen in figure 2.1. The top most layer is the **epidermis**, its purpose is to provide a barrier to the body by protecting from pathogens from entering the body and by keeping the water inside. The second layer is **dermis** which contains blood vessels, lymphatic vessels, hair follicles, sweat glands, oil glands, nerve endings and fibrous tissue. All layers of the skin are prone to cancer as it can start at the epidermis and then spread on to other layers or even to other parts of the body.

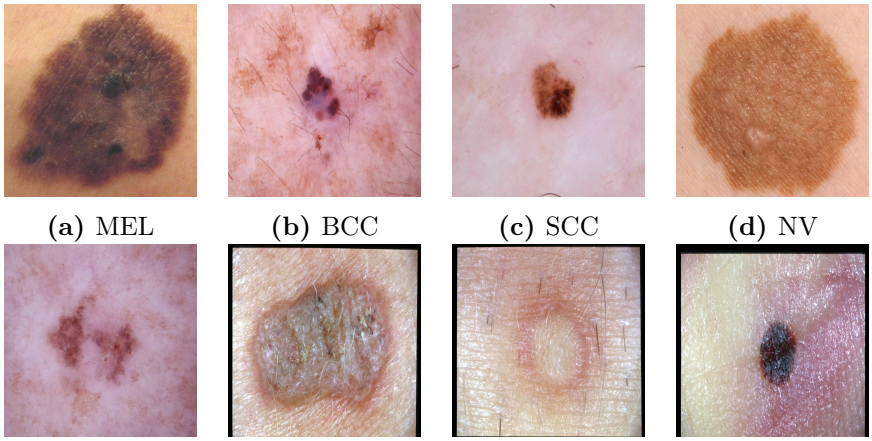
2.1.1 Skin Lesion

Skin lesions are parts of skin that has unusual growth or appearance to it compared to the rest [4]. When this abnormality is due to uncontrolled growth of uncommon cells in the outer layer of skin, dermis, then it is classified as skin cancer [5]. Skin cancer can either be nevus or malignant depending on whether the abnormal cells are confined to one area or whether it can spread to other areas respectively [5].

In this thesis, we look at the three main type of skin cancer; Basal Cell Carcinoma (BCC), Squamous Cell Carcinoma (SCC) and Melanoma (MEL) along with other categories such as Dermatofibroma (DF), Actinic Keratosis (AK), Seborrheic Keratoses (BKL), Vascular Lesions (VASC) and Nevus (NV) all of which are described below. Figure 2.2 shows examples of each of the classes of skin lesion.

Melanoma (MEL)

Melanoma occurs at the melanocyte cells which makes the pigment melanin that gives skin its color [2]. It is the least common amongst the three type of cancer mentioned but also the most dangerous due to its ability to spread to other regions in the body [5]. Melanoma can be caused by both intrinsic and extrinsic factors and can be identified with distinct characteristics such as its irregular edges, color (can be of more than one color) etc [2] [5].



(a) MEL (b) BCC (c) SCC (d) NV

(e) AK (f) BKL (g) DF (h) VASC

Figure 2.2: Examples of each skin lesion classes

Basal Cell Carcinoma (BCC)

Least deadly and most common form of cancer that occurs in the form of a raised, smooth bumpy tumor forming in the lower layer of the epidermis (outermost layer of the skin). Can sometimes bleed or become inflamed and grows slowly over months or even years [5].

Squamous Cell Carcinoma (SCC)

An early form of skin cancer that starts at the epidermis. It has the appearance of a "red, scaly patch" and could potentially spread to other parts if left untreated. SCC accounts for 30% of all non-melanoma skin cancers [5].

Nevus (NV)

Nevus is a general medical name for visible, circumscribed, chronic lesion of the skin. Nevus can be obtained by either congenitally or acquired [6]. They can be spotted by their irregular shape and uneven color [5].

Dermatofibroma (DF)

Dermatofibroma is a benign growth that appears as rounded bumps with varying colors. They occur more commonly in women and can be found anywhere in the body but more often on legs and arms.

Actinic Keratosis (AK)

Also known as sunspots are lesions that occurs due to frequent sun exposure and provides as a warning sign that the skin had too much sun exposure and is in risk

of cancer [5]. They can be noted by their sandpaper like texture in varying colors [7].

Seborrheic Keratoses (BKL)

Seborrheic Keratoses are benign lesions that do not require removal for medical reasons [8]. These lesions appear to be in round or oval shape. They are light brown to black in color and feels flat or slightly elevated.

Vascular Lesions (VASC)

Vascular lesions are common abnormalities that can form on skin, tissues or organs. They can occur anywhere in the body and are usually benign.

2.1.2 Imaging techniques

Dermoscopy

The use of dermatoscope, a device that consists of a magnifier, a polarised/non-polarised light source, a transparent plat and a liquid medium is called dermoscopy. This technique provides high quality images of skin lesions by allowing visibility of micro-structures that appear on top of the skin that is usually not visible to the naked eye. The images obtained through this technique is also void of any background noise making it ideal for studies. The dataset used for this study mainly involves around these images.

Clinical

Images captured by mobile or other portable cameras. These images have varying backgrounds, other body parts and different lighting conditions. These images are more difficult in classification as there is need to deal with the background noise. Hence, these images require much more heavy pre-processing compared dermascopic images. In designing systems that works in real world and not a clinical environment these images are very helpful.

2.1.3 Diagnostic techniques

There exists several techniques that are used by dermatologists around the world to diagnose skin lesion. In this section, few of these techniques are explored to shed a light on the components of the skin lesion that stands out and helps dermatologists in classifying these lesions.

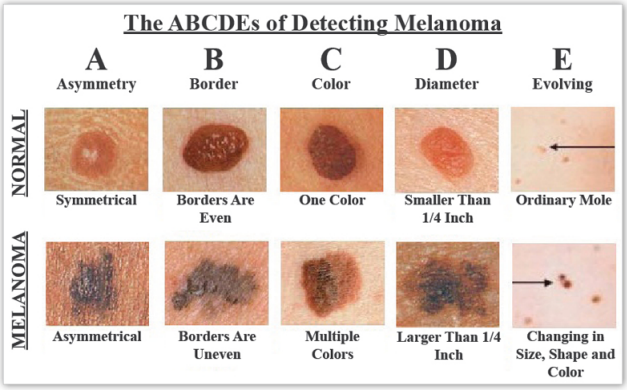


Figure 2.3: ABCDE skin lesion diagnosis technique[11]

ABCDE

One of the most commonly used in practice method is the one proposed by Friedman, Rigel, and Kopf [9] called the ABCD technique which was later advanced by Hazen, Bhatia, Zaim, *et al.* [10] to ABCDE. In this technique the lesion is observed carefully to look for different properties, each one for each letter.

- **A - Asymmetrical**: Melanoma lesions usually have the two halves that does not match.
- **B - Border**: The border of a benign lesion is quite even whereas the melanoma lesion can have irregular border.
- **C - Color**: Melanoma lesions consists of multiple colors rather than a single color in normal lesions.
- **D - Diameter**: Any lesion with more than 1/4 inch is suspicious to be melanoma.
- **E - Evolving**: This component requires monitoring the lesion over time to observe any changes size, shape or color.

7-Point Checklist

A score based system that is used to classify lesions by identifying certain features which are assigned points [12]. The amount of points for each feature is based on how much each feature increases the probability of the lesion being a melanoma. Table 2.1 shows the various features and their corresponding points.

Ugly Duckling

Another simple approach to detect melanoma is based on the fact that most normal lesions look alike where as the melanoma lesions sticks out [13]. These "ugly

Major ELM Criteria	**Score**
Atypical pigment network	2
Gray-blue areas	2
Atypical vascular pattern	2
Minor ELM Criteria	**Score**
Radial streaming (streaks)	1
Irregular diffuse pigmentation (blotches)	1
Irregular dots and globules	1
Regression pattern	1

Table 2.1: 7-point Checklist method for skin lesion diagnosis

ducklings" can be detected when any lesion that is different in size, shape, color compared to it's neighbors is found. Also, if a lesion is by itself not surrounded by any others then it is also to be recognized as a suspicious lesion.

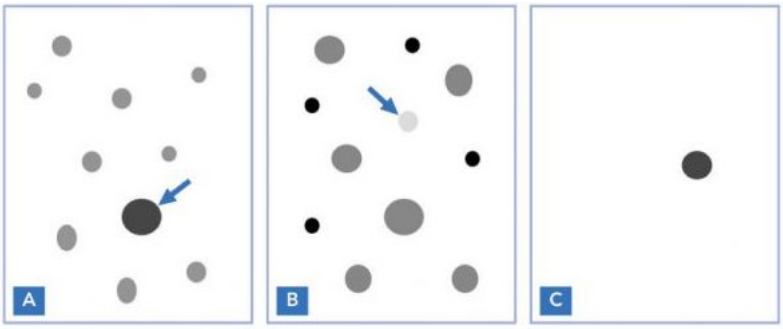


Figure 2.4: Differenct types of ugly duckling diagnosis. Image a shows a lesion that is bigger and darker that stand out from its neighbors. Image b shows a very light lesion that is different and Image c shows a lone lesion with no neighbors. [13]

2.2 Deep Learning

To understand what neural networks are and how it works, we first need to look into the inner working of neurons present in our own brain to allow ourselves to gain a better understanding of where the inspiration of neural networks comes from. A neuron mainly consists of three parts, dendrites, cell body and axon. Dendrites in the neuron are the many input branches that send signals to the nucleus which is the processing unit. It then sends electrical activity through the axon and this signal branches out through the axon terminals. These neurons, however, do not

touch each other instead they connect using either a chemical or electrical synapse. It is these connections that hold the key to the "connectionist" behaviour of the artificial neural network.

The very first breakthrough in neural networks could be linked to Mcculloch and Pitts [14] where they came up with the threshold model of biological neuron (see Fig. 2.5). This is a very basic model which takes the input in the form of a vector $X = [X_1, X_2 \ldots X_n\ ]$ where n is the length of the vector. Each of this input is then assigned a weight value w to be multiplied with the input. It can then be manipulated in order to suppress or incite an input. The sum of all these values is then fed onto an activation function that decides whether the neuron would fire given the input and gives out output y. A bias is often introduced at the activation stage in order to control the input fed to the activation. Hence, we can summarize this whole process by

$$y = \sigma(WX + b)$$

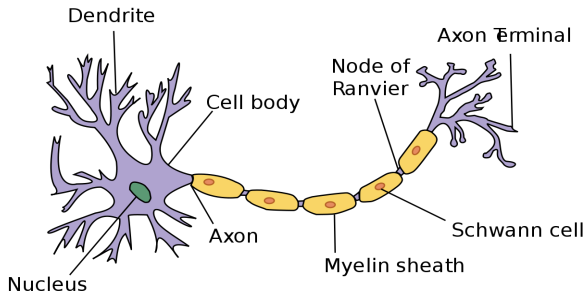
where σ is the activation function, W the weight vector, X the input vector and b the bias value. There are many choices for activation functions, Section 2.2 will describe few of the common ones that is used extensively throughout this study. The perceptron described above was able to perform simple classification tasks. However, it still lacked the ability to "learn" as the weights were added in manually. Van Der Malsburg introduced the concept of training this model to learn by punishing any misclassifications by adjusting the weight values in order to minimize some cost function.

The simple single layer neural network is limited in the sense that it is unable to learn non-linear classification tasks. This limitation lead to the invention of the multi-layer neural network. Due to the complexity of the problem in this study, we will be mainly focusing on different versions of multi-layer neural networks from now on.

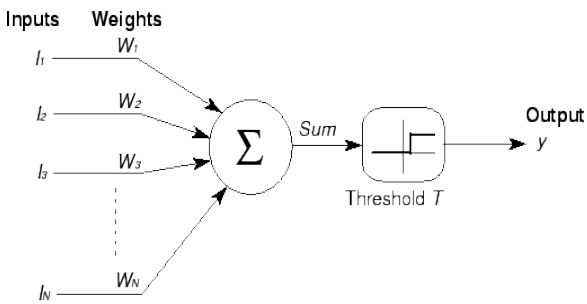
Activation functions

Activation functions are mathematical equations that determines the output of the model. An activation function is attached to each node in the model and is responsible for deciding whether that node would be activated or not. It is also does the job of normalizing the inputs by squashing the input values to a specific smaller range usually [0,1] or [-1,1] which makes the model more computationally efficient. Three of the activation functions used in this study are listed below

- Sigmoid: A smooth function that gives the output in range of 0-1 without any jumps in output. It has the issue of "vanishing gradient" where very low or



(a) Visual representation of a Neuron [16]



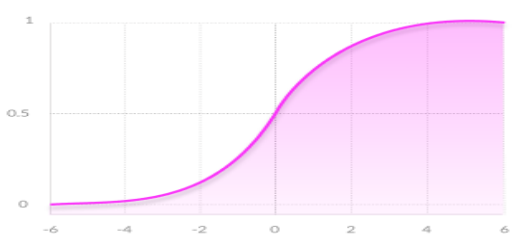
(b) Linear threshold system. [17]

Figure 2.5: Comparison of a neuron and an early linear threshold system to showcase the similarity.

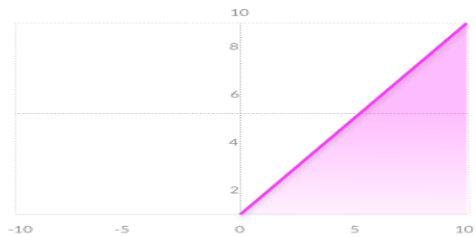
high values does not make much difference. It is also more computationally expensive.

- ReLu: A more computationally efficient function that allows quick convergence. It has the issue zero gradient when values are extremely close to zero preventing further learning however this is a very rare occurence in most cases hence ReLu is very widely used.

- Softmax: The only function listed that can handle multiple classes. It normalizes input to range of 0-1 and gives the probility of each class. Hence, this one of the most popular function to use for the output layer in any multi-class classfication problem.



(a) Sigmoid



(b) ReLu

Figure 2.6: Common activation functions. [18]

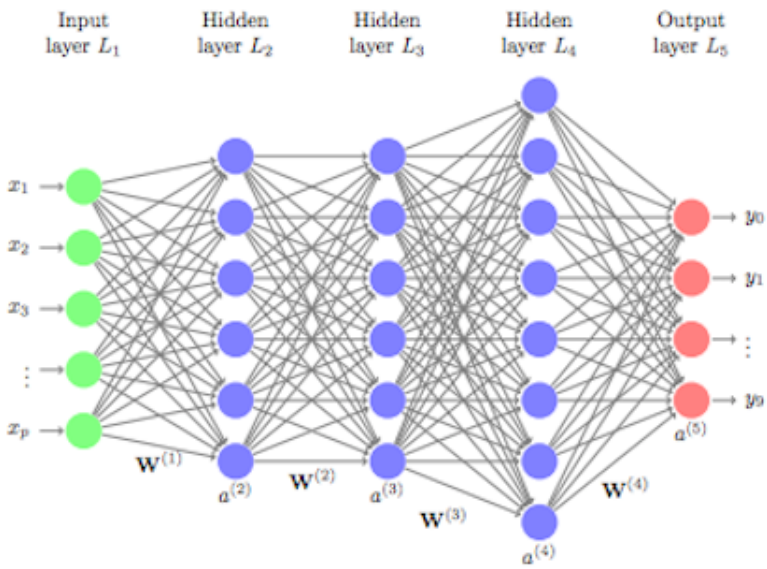


Figure 2.7: Architecture of a deep neural network

2.2.1 Deep Neural Network

So far we have seen the power of neural networks and it's limitation on complex classification tasks such as the problem of multi skin lesion classification that we are tackling in this study. In this section, we explore a more complex archtecture that is built on top of neural networks called Multilayer Neural Network or Deep Neural Network (DNN). The architecture of a DNN has 3 components (See Fig.2.7):

- Input Layer : This layer takes the input as a form of vector $x = [x_1, x_2, ..., x_n]$ where each of this input value can be thought of as a node. These nodes are then connected to other layers with each edge weighted by factors w_{ij}

- Hidden Layer : Every DNN has a number of layers in between the input and output layers. The number of hidden layers can be anywhere from few layers to hundreds of layers making the model "deep". Each node of each layer in the hidden layers are connected and get activated based on the activation function linked to each layer. The weights and biases effects which node gets activated and in turn effects the output and accuracy of the model.

- Output Layer : Final layer of the model that predicts which class the input belongs to. This can either be a binary classification or in our case the output would be the probability of the input belonging to each class.

Since, we have established the architecture of DNN, we can now dive deeper into how DNNs are trained to learn to solve complex problems. The process of learning which can be defined as finding the right weights and biases values in order to maximize the result, happens in three phases

Forward propagation

This phase starts off by assigning random values to the weights W. Then, the input data is propagated through the different layers all the while being multiplied with the weight values and passing through activation functions. Finally, it reaches the output layer where we obtain the result.

Estimation

In this phase the result obtained is compared to the ground truth to see how well the model performed. A loss function is used here to calculated the wellness. There are multiple candidates for the loss functions such as the mean squared error (MSE), Hinge etc but the one we use mainly for this study is the cross entropy as it performs well for multi-classification problems by reducing the distance between two probability distributions which is defined as

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Backward propagation

In this phase, the loss calculated is propagated to every single layer in the model. The weights and bias values are then tuned in order to bring the loss value as close to zero as possible. This is acheived using a technique called gradient descent. This technique works by slowly adjusting the values of a model by calculating derivatives of the loss function which lets us observe in which direction to make a step in order to reach global minima. The magnitude of the step taken is decided by the learning rate.

$$f'(m,b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N} \sum -2(y_i - (mx_i + b)) \end{bmatrix}$$

2.2.2 Testing

We have seen the components of a neural network and how to train one. Finally, after a model is trained it is important to test it to see how well it can perform on real life data. So after achieving preferable results in the training phase, the model is tested on data it has not seen in the training phase. This is to ensure that our model has not just memorized the training set (over-fitting) but has actually learned to generalize. Once we have similar result on the testing set, we can say that the model is ready.

2.2.3 Convolutional Neural Network (CNN)

Ever since AlexNet [19] won the ImageNet challenge; the biggest image classification competition in 2012, CNNs have been used for all sorts of image classification problems including medical image analysis. CNNs takes digital images which is nothing but a bunch of numbers and is able to identify the features that is unique to each class of images. This ability to identify features in an image is something that is so natural to us humans but extremely difficult for computers that sees the images in the form of pixels. However, CNNs overcome this issue by working in layers where each layer has a separate role.

Convolution

The first layer in a CNN is always a convolution layer. In this layer, a filter which is a bunch of numbers that represent a feature is slid across the image. The filter values are multiplied and summed up with image pixel values to obtain a single number. After the convolution, the input image is converted to what is known as a feature map.

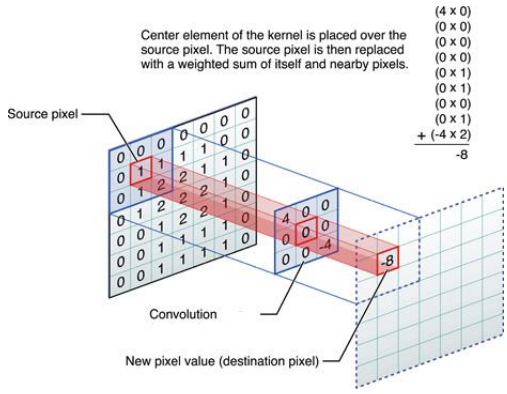


Figure 2.8: Convolutional operation using 3x3 kernel on a 7x7 matrix [20]

Normalization

To prevent the mathematical operations from breaking, we convert negative values to positive in this layer. In this layer, a function such as ReLU is used for conversion.

Pooling

In this layer, the feature map that we obtained from previous convolution layer is shrunk to a smaller size. It achieves this by taking a window of pixels i.e. 2 x 2 pixels and striding over the image taking the maximum or average value within the window. This results in a much smaller image that is a lot more computationally efficient.

Fully Connected

The three layers mentioned above are usually stacked together multiple times to form a deep neural network. At the end of this deep stack we have the final fully connected layer where each and every value of the final pooling layer is taken to form N dimensional vector where N is the total amount of classes to classify the image into.

2.2.4 CNN Architectures

In this section we present the different CNN architectures that are very powerful and has proven to be effective in multitude of classification tasks. We highlight the different brilliant advances made over the years that lead to the development of each of the models. We will be using all of these architectures for our experiments down the line.

VGG

In 2014, Simonyan and Zisserman created the VGG model for the popular ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [21]. They popularized the concept of block layers where each block consisted of 2 or 3 convolutional layers followed by a maxpool layer. By doing this, the authors claims that it simulates a large receptive field on the input while reducing the computational requirement. This model has two variations caleed VGG16 and VGG 19, each with 16 and 19 weight layers respectively.

ResNet

As the tasks became more and more challenging requiring models to learn complex features, there was need for a deeper networks. However, creating deeper networks lead to the "vanishing gradient" problem where the gradients from deeper layers becomes so small that they "vanish" due to the continuous multiplication operations. The breakthrough that solved this issue and lead to creation of very large models that can tackle difficult challenges was the use of residual blocks (see Fig 2.9) in ResNet [22]. This residual block has two paths, one normal path and another shortcut or skip connection path. The shortcut path allows the residual blocks to add the activation of an earlier layer onto a layer deeper in the model forcing the model to learn the residual (difference) between two layers. This concept allowed training models that are more than 100 layers deep!

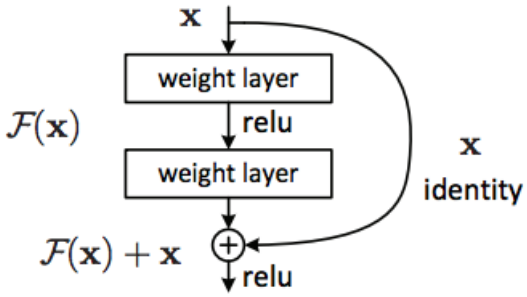


Figure 2.9: Residual block in ResNet

DenseNet

DenseNet adds onto the advances made by ResNet by shortening the connections closer to the input and output [23]. This is achieved by the concept of dense-blocks where each layer is connected to every other layer ahead, Figure 2.10. Hence, if other networks have L connections, DenseNet has L(L+1)/2 connections. Inside these dense-blocks, feature maps of all preceding layers are fed in as inputs to each layer. The number of feature maps learned after each convolution in this dense-block is set to a low number to avoid memory explosion. The main advantages of DenseNet are two folds. Firstly, DenseNet requires fewer parameters than traditional networks as there is no need to learn reduntant features. Secondly, DenseNet promises better flow of information and gradients throughout the model making it easier to train.

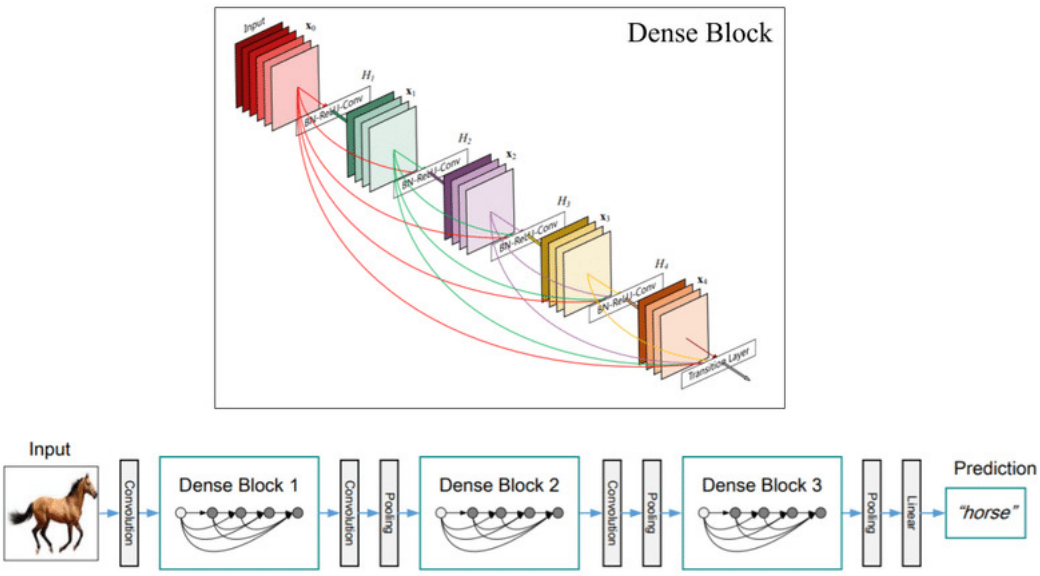


Figure 2.10: DenseNet architecture [24]

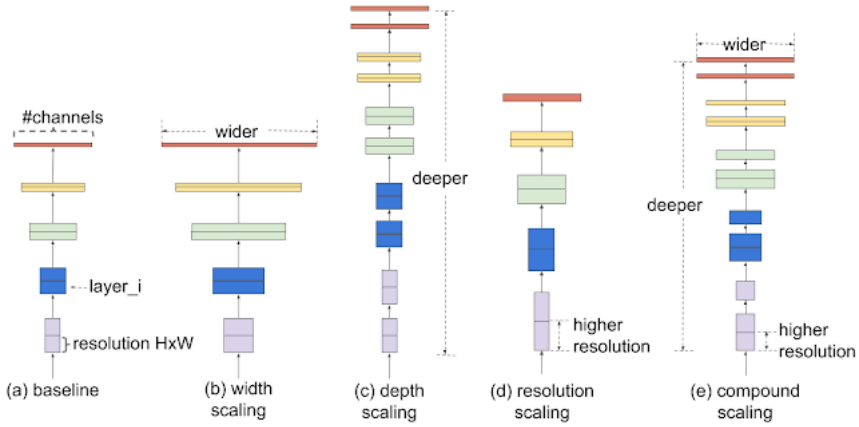


Figure 2.11: EfficientNet scaling method. [25]

EfficientNet

Finally, the latest model out of all the models we will test is the EfficientNet [25]. EfficientNet improves efficiency and accuracy by cleverly scaling the model. A model can be scaled 3 ways:

- Depth: This is the most popular way of adding layers to make the model deeper in order to learn more complex features.
- Width: The model can be made wider by increasing the channels in the convolutional layer. By increasing the width, a model is able to capture more fine-grained features and makes the model shallow which is often easier to train.
- Resolution: Feeding images of high resolution is another way of scaling a model as it makes the small detailed features more profound.

EfficientNet proposes a scaling method that uses a "compound coefficient" ϕ that uniformly scales all the three other variables, Figure 2.11. The authors then found an optimal network which is then optimized to scale at different levels. This resulted in 8 models, B0-B7 each larger than the one before with B7 achieving state of the art results.

2.2.5 Transfer Learning

One of the biggest issues with medical image classification problems is the lack of data. Hence, training a deep convolutional neural network from scratch becomes difficult [26]. To overcome this issue the transfer learning technique is used where a model that is already pre-trained with a large dataset of natural image is used. This model is then fine tuned to work with our own dataset of medical images which is quite different from what the model has seen before. However, the difference in

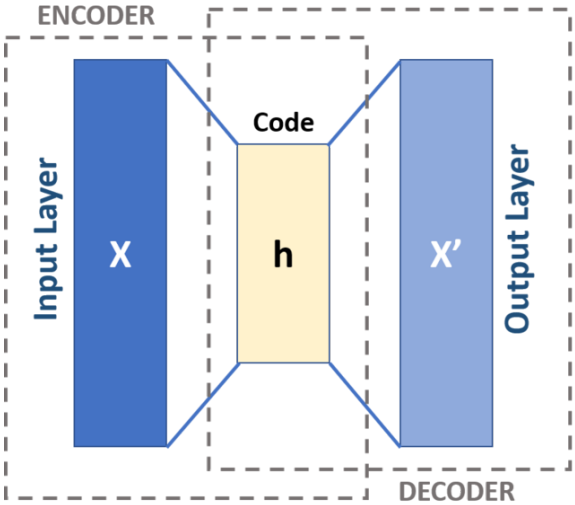


Figure 2.12: Structure of an autoencoder

medical images compared to natural images prevents this technique to work as well as it does with some other natural image classification but still the early layers that detects the general shapes and edges can be taken advantage of to speed up the training process.

2.2.6 Autoencoders

Autoencoders are a form of neural networks that consists of input, hidden and output layers. Figure 2.12 shows the common stucture of an autoencoder. It has two main parts, an Encoder and a Decoder.

Encoder

The encoder takes the input and maps it to a "code" usually after passing it through a set of hidden layers. By reducing the number of nodes in the code layer, autoencoder can be used for dimensionality reduction. The outputs of nodes in the hidden layers are defined by

$$z = \sigma(Wx + b)$$

where σ is activation function, usually **sigmoid** or **relu** it squashes the values passed to it in to a range. $\mathbf{x}$ is the input and it is multiplied by $\mathbf{W}$, the weight matrix and $\mathbf{b}$, bias vector both of which are learned during the backpropagation phase in order to minimize the error of output.

Decoder

The decoder tries to approximate the output which has the same dimension as input from the code which is of lower dimension of the input. A loss function such

as square error can be used to compare the output to the input in a process called backpropagation. In this process the weights and biases are tuned in order to learn the best approximation of the input in a smaller dimension.

Autoencoders are widely used in the development of skin lesion segmentation, wherein dataset of images of skin lesion and its corresponding segmentation mask performed by expert dermatologists is provided to an autoencoder. Here the image of the skin lesion acts as the data point and the segmentation the ground truth. The autoencoder model is then trained to generate an image of segmentation mask by reducing the difference in ground truth and the model output.

2.2.7 Generative Adversarial Network (GAN)

GAN in its simplest form has two components, a Discriminator and a Generator that works together in a competitive manner to generate data that mimics the real data. Both of these components works in phases where the discriminator tries to generate new data from random noise and the discriminator with the help of some real data, tries to predict whether the data created by the generator is fake or real. After each iteration, the weights of both of these components are updated. The discriminator's weights are updated in order to improve the classification accuracy and the generator's weights are updated in order to decrease the classification accuracy or in other words fool the discriminator. This process continues until it reaches a point where the discriminator cannot tell apart the fake data from the original. At this stage, we have a generator that is capable of generating synthetic data that looks like real data but is completely new. Fig 2.13 shows the architecture of a simple GAN.

This concept has been applied in skin lesion classification to create new data and reduce imbalance within a dataset. In this study, we will apply this technique and evaluate how well it performs in helping the classification task.

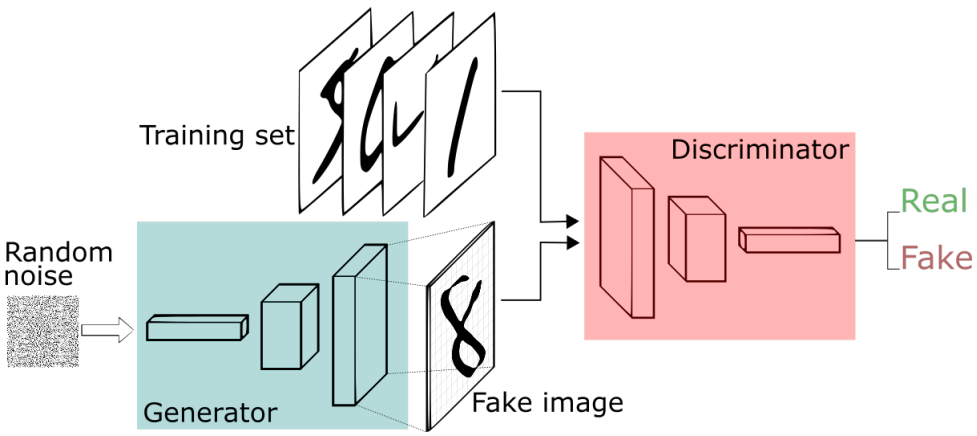


Figure 2.13: Architecture of a GAN [27]

2.3 Available Datasets

There exists many publicly available datasets that are either free to use or can be purchased for a fee to use for research purposes. Here, all the datasets that are found during the time of writing this thesis is listed. As these datasets are from complete different sources, it must be noted that they have varying properties. They differ in the resolution, preprocessing applied and quality of images etc. Hence, it is very important that careful attention is paid when combining multiple datasets.

1. ISIC 2019 Dataset: Largest available dataset with 25,331 images of common skin lesion classes. The images contained in this dataset is classified into 8 categories; Actinic Keratosis, Squamous Cell Carcinoma, Basal Cell Carcinoma, Seborrheic Keratosis, Dermatofibroma, Nevi, Melanoma, Vascular Lesions. [28]–[30]

2. DermNet NZ: Very large dataset not specific to skin lesion but various skin diseases. Includes mainly high quality clinical images. Can be used for research purposes but the dataset need to be purchased.[31]

3. PH2 Dataset: contains 160 nevus images and 40 melanoma images. This dataset also contains segmentation masks and contains other information such as asymmetry, pigment network, dots/globules, streaks, regression areas, blue-whitish veils and colors. [32]

4. Med-Node Dataset: A small datset containing 170 images (70 melanoma and 100 nevi cases). [33]

5. Asan and Hallym Dataset: Asan dataset contains more than 17000 images of 20 classes of skin diseases found amongst Asian people. The Hallym dataset adds upon this to include some more images of class BCC. The image resolution of these images are quite low, around 100x100 compared to other datasets. [34]

6. Seven-Point Checklist Dermatology Dataset: contains over 200 clinical and dermoscopy images along with metedata based on 7-point skin lesion malignancy checklist. The images are classified on many specific categories such as basal cell carcinoma, different types of nevus, dermatofibroma, lentigo, vascular lesion and different types and size of melanoma. [35]

2.4 Evaluation metrics

Choosing the right evaluation metric is key to identifying the right model that performs best with the data in hand. Accuracy measure is the most popular metric for evaluation models, however, in early experimentation of different architectures, it was soon found that accuracy is not an accurate form of evaluation in our case. This is due to the fact that our dataset is highly imbalanced. So a model could predict all images as nevus and if the dataset consists of 70% nevus images and 30% malignant images then the accuracy would be 70% but in reality the model has predicted all malignant images wrongly which are more important to be predicted correctly as it could have fatal consequences if predicted wrongly. Hence, in this section we look at different evaluation metrics that are used in practice that paints a better picture and allows a more accurate evaluation.

2.4.1 Sensitivity

Sensitivity is the measure of how well a model is able to classify a class of interest. In our case since misclassifying a case of Melanoma would have severe consequences we would like the model to maximise sensitivity to this class. Sensitivity is defined as

$$
\begin{aligned}
&Sensitivity = \frac{TP}{TP + FN} \\
&where, \\
&TP = \text{True Positives} \\
&FN = \text{False Negatives}
\end{aligned}
$$

Apart from measuring the sensitivity of each class, a mean sensitivity value will also be measured. This is defined as

$$
\begin{aligned}
&\hat{S} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i} \\
&where, \\
&C = \text{Classification classes}
\end{aligned}
$$

The mean sensitivity will be the most important evaluation metric for all the models.

2.4.2 Specificity

Specificity tell us out of all the data classified as relevant, how many were actually relevant. It is defined as

$$Specificity = \frac{TN}{TN + FP}$$

$where,$

$TN =$ **True Negatives**

$FP =$ **False Positives**

2.4.3 Confusion Matrix

Confusion matrix is a way to represent True Positive(TP), True Negative (TN), False Positive(FP), False Negative(FN) in an easy to understand and visually pleasing tabular format. Figure 2.14 shows an example of a confusing matrix.

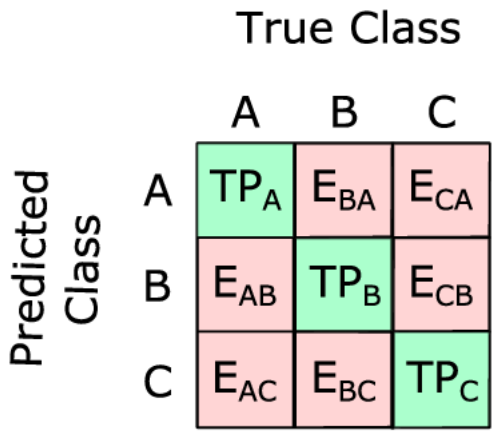


Figure 2.14: An example of how a confusion matrix looks for a muticlass classification model [36]

2.4.4 F1 Score

The F1 score combines both the sensitivity and specificity to give a value that is the balance between each of them. This is achieved by calculating the harmonic mean of the two values that punishes extreme values greatly. This is especially important as we are working with an imbalanced dataset and gives a more accurate metric. The F1 score is defined as

$$[h]F1 = 2 \times \frac{Sensitivity \times Specificity}{Sensitivity + Specificity}$$

2.4.5 AUC-ROC Graph

Classification outputs are generally given as probabilities and by decision to which class the input belongs to is made by setting a threshold, usually 0.5. However, in certain cases a higher or lower threshold results in high number of false negative or false positives than we prefer. For example, in our case, a high number of false negative means classifying cancer patient as healthy and this can have fatal consequences therefore we would like to minimize this as much as possible. That being said, drawing a confusion matrix for every possible threshold is not ideal either. The ROC (Receiver Operating Characteristics) graph can be used in this case to draw a graph that plots true positive rate against false positive rate for each and every threshold showing the ideal threshold in an easy to understand and compact manner. The AUC (Area Under the Curve) gives a single number that can be used to compare multiple ROC graphs easily.

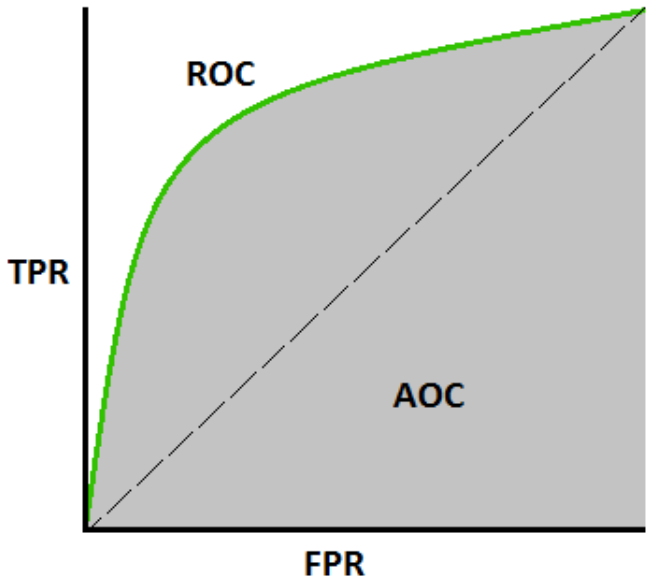


Figure 2.15: AUC-ROC graph. Green line shows the ROC graph that plots each threshold value and the shaded area corresponds to area under the curve [37]

2.4.6 Class Activation Map (CAM)

All the metrics discussed above lets us evaluate the performance of the models. However, why each of the model scored the score that it did is hard to determine as CNNs can be a black-box that just spits out the results. By using CAMs we have to some degree, knowledge of what parts of the image is being focused on by the model we are testing. This lets us evaluate if the performance it scored is in fact by focusing on the right parts of the image. CAM works by displaying a heat-map over the input image. This heat-map is a 2D grid of scores that is linked to the specific output class computed for every location in any input image. This shows how significant each location is in deciding the class that the model predicts. Figure 2.16 shows an example of CAM, in this example the model predicts the class based on the facial features of a dog which is clearly indicated over the image.

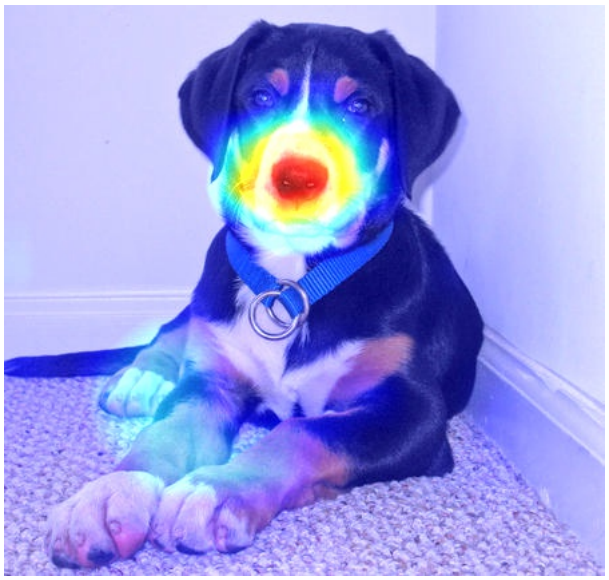


Figure 2.16: An example of a Class Activation Map being applied to a dog image classifier [38]

3

Methods

Now that we have obtained a deeper understanding of skin biology and deep learning methods, we now introduce the methods by which we plan to answer the research questions. This chapter begins by looking at the existing literature of related works through comprehensive analysis. Then details of the dataset used and how it is pre-processed and prepared for further study is given. After which the various methods of experimentation that we plan to conduct in respect to the research problem is presented. Namely, base model selection where models are trained to find which model works best for our problem in hand followed by three different techniques to handle the biggest issue of data imbalance that claws away the performance is also described.

3.1 Previous work

Deep learning has been shown promises in different medical imaging classification problems such as X-ray, CT, MRI etc [39] so it was just a matter of time until it was applied for skin lesion classification as well. Which is exactly what happened in 2017 when a team of computer scientists teamed up with dermatologists to solve this problem [40]. They put forward a deep CNN (InceptionV3) to solve the problem and as a result acquired better results than dermatologists in classifying lesions. The results as amazing as it was, it was still not enough to replace humans completely especially due to the high cautious nature of the medical industry. However, this opened up the door for many more research to take place in this field. Over the years, the data available for research has increased along with advancement in different deep learning techniques. This section discusses the various research and technological advancement taken place in this field.

Advancement in CNN architectures [22], [23], [25], [41] has lead to many interesting research work such as the work by Hosny, Kassem, and Foaud[42] where AlexNet architecture was used for binary skin lesion classification of melanoma vs nevus. They leveraged the use of transfer learning and augmentation techniques to overcome the lack of data. Their work achieved an accuracy of 97% despite using an architecture shallower than other new deeper architectures which raises the question of the significance of architecture deepness for this problem.

3.1.1 Segmentation

Image data used for classification can usually contain other foreign objects or background objects within the image which can negatively effect a model's performance. Segmentation is a popular technique used to overcome this issue. However, segmentation has its own challenges due to the irregularity of lesions, lack of ground truth data, low contrast between lesion and skin etc.

Early work on segmentation of skin lesion [43], [44] mainly involved intensity thresholding techniques such as Otsu's method. This involves selecting intensity threshold values based on the fact that the pixels of the lesion part of the image is darker than the rest of the skin. This method gave satisfactory results on most images. However, for images with artifacts such as hair, low contrast or lesions on dark skin, this method did not work very well.

More recently, deep learning methods has shown better results in segmentation. Yuan, Chao, and Lo [45] proposed a fully convolutional network (FCN) that contains two paths. First convolutional path extracts the contextual information from the input RGB image and then the deconvolutional path recovers the full image

resolution. This model performs pixel wise classification where each pixel in the output image is either classified as belonging to tumor or not. They use the Jaccard Distance as a loss function to check how similar the output is to the ground truth.

Tschandl, Sinz, and Kittler [46] proposed LinkNet that reuses encoder blocks from other network such as RestNet152 which they call LinkNet512. During training phase, the images are scaled to 512x512 and different augmentation techniques were used. The Jaccard-loss which has shown to be effective in segmentation were used along with Adam optimizer. The segmented binary image output is then taken through a post processing phase where conditional random fields(CRF) is used to refine the edges and restrict the segmentation to the largest area. The CRF post processing however decreases the overall performance and gave a more close but jaggered edges. The study achieves result on the same level as manual human segmentation. However, this study is strongly limited by the lack of data with segmentation ground truth and most data available are of pigmented (brown or black) skin therefore further research is needed to see if the result obtained holds for light skinned skin lesions.

In the more recent years, International Skin Imaging Collaboration (ISIC) has made tremendous efforts in attracting interests of researchers all over the world to this problem. They put up new challenges based on skin lesions every year since 2016, where researchers can submit their work and is ranked at the end of the year. They have also created a great website [47] that lets researchers view/download dataset and compare results of their work. This has lead to many great submission over the years. More notably, last years winner was the work of Gessert, Nielsen, Shaikh, *et al.* They used the state of the art classification model EfficientNet [25] which has proven to give better results than its predecessors to classify the same 8 classes of skin lesions tackled in this thesis. The dataset was pre-processed heavily with cropping, resizing and applying Shades of Gray color constancy and created an ensemble model consisting of best performing models from all the models trained. Hence, their work provides a benchmark for further work with the result of 72.5% mean sensitivity.

3.1.2 Data Imbalance

For any classification problem, ideally a well balanced dataset is key to generate a model that is not biased to a specific class. Unfortunately, when it comes to medical images, the data available are very limited let alone balanced. Since collecting more data is not always practical, there has been studies to reduce the data imbalance by using various techniques.

One of the approaches to tackle this issue is a sampling technique called under-

Figure 3.1: Melanoma detection device MelaFind© [52]

sampling or oversampling [49]. Using this technique, either the majority class is randomly sampled less or minority classes are sampled many times over respectively in order to reduce the difference in data samples passed to the model. This approach is prone to either overfit the data in case of oversampling as existing information is just copied over or lose valuable information in the case of under-sampling if not used carefully.

Another approach is introduced by Blagus and Lusa [50] where in synthetic images are created to balance the dataset in a technique called Synthetic Minority Oversampling TEchnique (SMOTE). Linear distance between feature vectors of two samples of the minority class is calculated. A new data point is then created by multiplying the result with a random number between 0 and 1. This process is repeated to created many more synthetic samples.

3.1.3 Real life applications

It is very important that these research development in classification can be transferred to meaningful products that can be used in practical environment. Hence, there has been efforts in the past to create solutions that can be used to detect melanoma such as MelaFind© 3.1 [51], a device that captures skin lesion image and outputs if it is melanoma or not by comparing it with a database of thousands of images. Even though this device does its job quite well, it is expensive only affordable to few dermatologists. Therefore, there is a need for a more affordable, more reliable device that is easy to use and is available to everyone over the world.

3.1.4 Critical Analysis

Looking back at all these work, we can see that there has been a significant contribution from the research community to this field and at quick glance, it may seem that the problem is a solved one due to the reported results. However, when looked closely, we can see that there is a lack of standardized evaluation which every study adheres to. Most results presented are on single dataset or a very controlled dataset collected from a single location by the authors. Hence, it is difficult to judge if the

results presented would generalize enough to be used in different real world situations. This is especially important in skin lesion analysis as the real world data varies from location to location due to different skin types. Therefore, in our study we apply these methods on the largest publicly available dataset and evaluate how they would perform.

3.2 Overview

In this section, we look at an overview of the system we use to run experiments and we also explore each of the component that makes up this system. Fig 3.2 shows the system as whole as a block diagram. The system starts with data preparation module where the dataset is compiled, pre-processed for experiments. The input data is then passed to a model that is trained to classify the images. The best model is obtained and used to evaluate how well the model performed. Each of these components are explained in further details below.

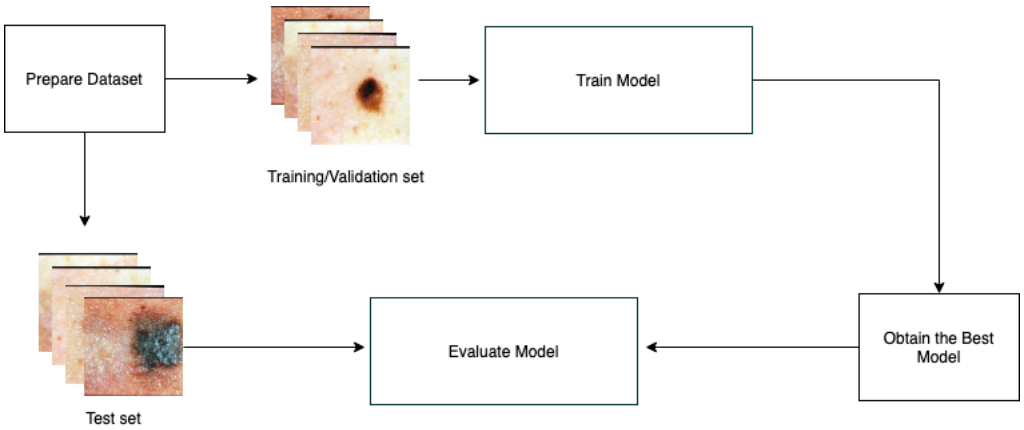


Figure 3.2: Overview of system

3.2.1 Data Preperation

Dataset compilation

There are many dataset available for academic research purposes that includes dermoscopic and clinical images of skin lesions classified over different categories. Out of all available datasets, those that are free to use have been downloaded and merged based on most available categories of classification to form a final combined dataset that will be used for further experiments. This dataset contains images that fall under the 8 categories of skin lesions mentioned in ISIC 2019 dataset. These labels are chosen as they have the highest available number of images within all the sources collected. Every class of the final dataset is then split into 3 categories train,

validation and testing with the ratio of 8:1:1 respectively to ensure the testing and validation dataset contains images of every class. Table 3.1, 3.2, 3.3 shows the count of each labels in the train, validation and testing dataset respectively.

Label	**Count**
Actinic Keratosis	867
Vascular Lesions	311
Squamous Cell Carcinoma	628
Basal Cell Carcinoma	3407
Seborrheic Keratosis	2714
Dermatofibroma	279
Nevi	14361
Melanoma	5133
Total	**27700**

Table 3.1: Count of each classification labels in the final training dataset

Data pre-processing

Pre-processing image data is an important initial step in any image classification problem. This is especially true when the dataset is a compilation of multiple sources and is not very large. Pre-processing the data can help in training models in this case. Hence, in this section, description of the techniques used to clean up and prepare the data as well as possible to aid training is described.

Firstly, many images contain thick black padding around the lesion. It would require the model a very long training time to learn that these black pixels do not contribute in lesion classification. So an algorithm was developed to detect and crop out these black padding.

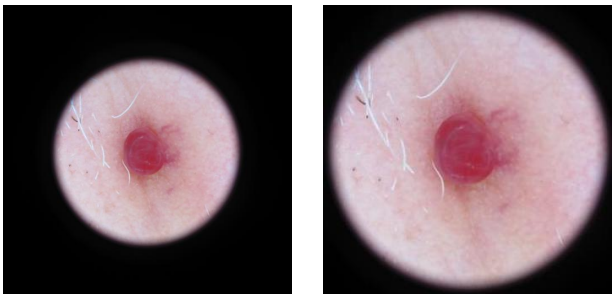
Secondly, foreign objects that exists within an image can throw off the model as well. This is mainly a problem for the clinical images where the images are not zoomed in on the lesion like dermatoscopic images. Luckily, these images constitutes

Label	Count
Actinic Keratosis	87
Vascular Lesions	31
Squamous Cell Carcinoma	63
Basal Cell Carcinoma	341
Seborrheic Keratosis	271
Dermatofibroma	28
Nevi	1436
Melanoma	513
Total	**2770**

Table 3.2: Count of each classification labels in the final validation dataset

Label	Count
Actinic Keratosis	86
Vascular Lesions	31
Squamous Cell Carcinoma	62
Basal Cell Carcinoma	340
Seborrheic Keratosis	271
Dermatofibroma	27
Nevi	1436
Melanoma	513
Total	**2766**

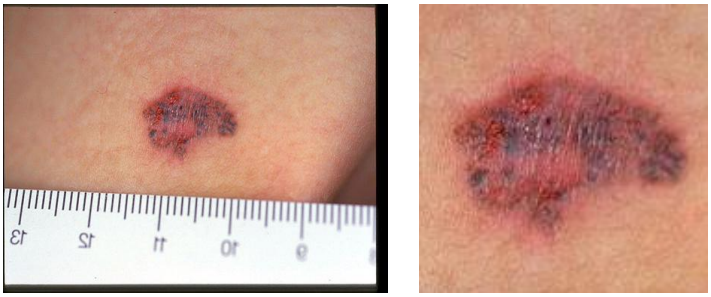
Table 3.3: Count of each classification labels in the final testing dataset



(a) **(b)**

Figure 3.3: Removal of black padding around skin lesion

only of a small percent of the whole dataset. Therefore, manual cropping approach was taken to clean up these images. For any larger dataset this approach would not be practical due to its time requirement but for our dataset it was feasible.

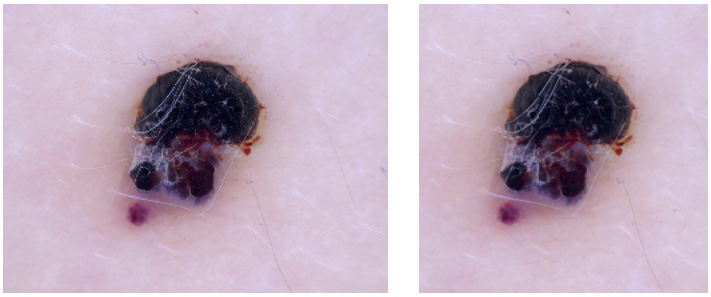


(a) **(b)**

Figure 3.4: Removal of foreign objects in images

Finally, CNN architectures used in this study are designed to work best with images of specific size. ResNet, DenseNet etc use 224x224 as the image resolution by default. The common practice when training a classifier is to just resize the image to fit into the resolution required using common interpolation algorithm such as nearest neighbor, bi-cubic etc. However, when dealing with skin lesion, the size and structure of the lesion is an important feature in classifying each lesion. Therefore, we opted to carefully scale down the image while maintaining aspect ratio. This is

done by random crop technique. To implement this, we first scale down the image by resizing the smaller side to 224 while maintaining aspect ratio. Then a random crop of 224x224 is done on the image.



(a) **(b)**

Figure 3.5: Random cropping of images to resize to 224x224 while maintaining aspect ratio

Data augmentation

Data augmentation is a pre-processing technique to increase the diversity of training data in a synthetic manner. This technique can be used to increase the variation in data helping the model to generalize better. There are many augmentation techniques such as flip, crop, scale etc that can be easily applied to the dataset in order to mimic the variability in data found in real life. Table 3.4 shows the augmentation technique used for this study and the parameters applied for each of them. Figure 3.6 shows the different techniques applied to an image.

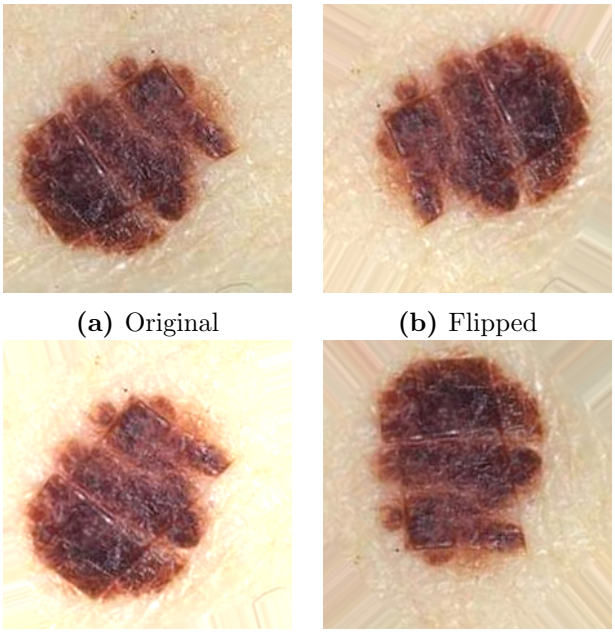
Augmentation Technique	Parameter
Rotation	0 - 90
Flip	Horizontal and Vertical
Brightness	0.7 - 1.3

Table 3.4: Augmentation applied on dataset before training

3.2.2 Training and Evaluation

Programming language and libraries

All experiments in this work is written in Python programming language due to its syntactic sugar and the plethora of well written libraries at disposal to implement concepts quickly. Data pre-processing and Model creations are done using Keras [53] library. Sklearn [54] library is also used extensively for evaluation purposes. Matplotlib and Seaborn libraries are used for plotting graphs and other visualization needs.



(a) Original **(b)** Flipped

(c) Brightness Changed **(d)** Rotated

Figure 3.6: Augmentation techniques applied to training dataset

Training strategy

During the training process, each model is run for 100 epochs, watching and saving the model that performs best at the end of each epoch. The training is stopped early when there is no performance improvement for 10 consecutive epochs to save time. All training process is done on a single NVIDIA-GeForce GTX 1080 Ti with 10GB memory.

As the dataset is quite large, it is not possible to fit all of it in memory. Therefore, training is done in batches. The batch size and other parameters such as resolution of images are all specified depending on how much data can be fitted in memory.

Evaluation

Finally, after training is complete, we select the best performing model and run evaluation on the test set that we have kept aside in the data preperation phase. To evaluate, we use Sensitivity, Specificity and F1 score as our evaluation metric as they give nice numbers that we can easily and accurately use to compare models.

3.3 Experiments

The objective of this study is to analyse the different techniques to perform and improve skin lesion classification. This is achieved two ways. First, the skin lesion images are classified into two categories of Benign (Nevus) and Melanoma. The purpose of this task is to built a model that can well identify the most dangerous

class amongst all category which is melanoma. This approach is then extended to a multi-class classification problem where each lesion is categorised into one of the 8 types of skin lesions mentioned. This section, states the different methods of experimentation conducted as part of this study.

3.3.1 Base Model Selection

Previously, we have explored many different CNN architectures. In this experiment, each of those architecture is trained using transfer learning technique to analyse which one performs best using our selected evaluation metrics. The best performing architecture is then used for further experimentation to save time. Fig 3.7 shows the structure of the model created for the experiments. Here the base components can be any of the VGG19, ResNet, DenseNet, EfficientNet variations. The FC component is the fully connected layers before the classfiication is made. There is an opportunity here for different combination of fully connected layers of varying parameters that will be tested. Prediction is then made by applying appropriate threshold value to the predicted probabilities.

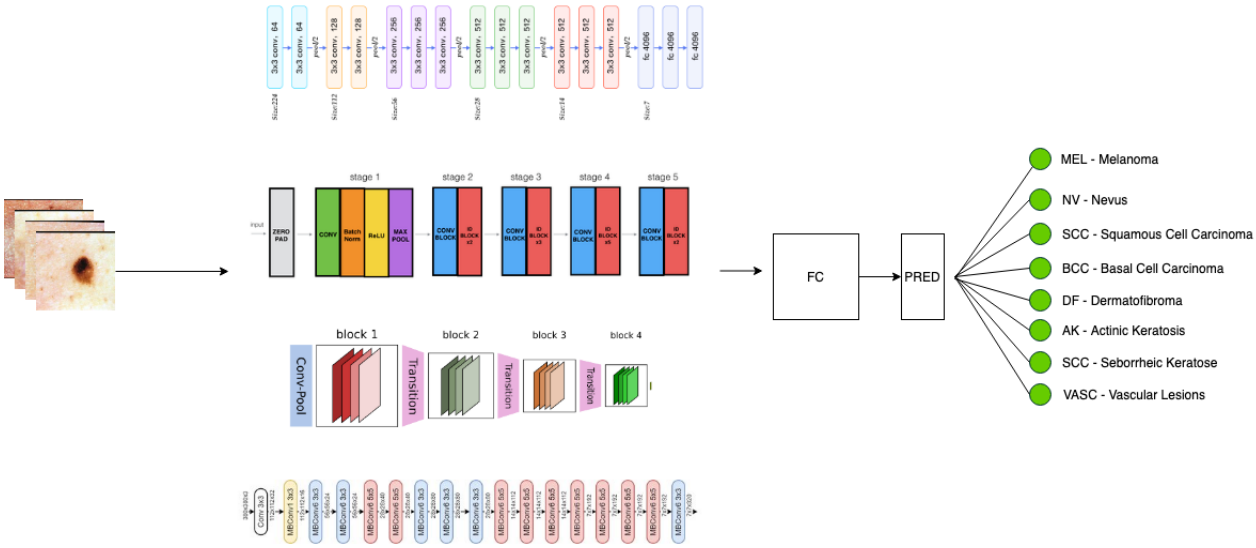


Figure 3.7: Base model selection process

To begin experiments, we first train a set of different popular architectures to investigate how well each one performs in classifying the dataset. Models are trained on both binary and categorical problem. The best performing model is then selected for further experiments.

Result

Training base model shows that deeper models performing better than shallower ones. Table 3.5 shows the evaluation metrics results for binary classification and

Figure 3.9 shows the confusion matrix. Sensitivity is our preferred evaluation metric since it is more important to detect melanoma than nevus. Therefore, we select DenseNet201 as the best performing model as it scored the best sensitivity.

Model	**Specificity**	**Sensitivity**	**F1**
VGG19	0.00	1.00	0.00
ResNet50	0.56	0.43	0.50
ResNet152	0.54	0.45	0.49
DenseNet121	0.63	0.39	0.51
DenseNet201	0.51	**0.48**	0.49
EfficientNetB0	0.78	0.24	0.51

Table 3.5: Evalutation metrics of base model trained for binary classification

Multi-class classification results

Surprisingly, for multi-class classification, ResNet152 gave the best sensitivity instead of DenseNet variant that worked better for binary classification. As for the binary classification, very small model such as VGG19 did do well at all as it ended up just predicting all samples as nevus with 0 sensitivity for melanoma.

Model	**Specificity**	**Sensitivity**	**F1**
VGG19	0.00	1.00	0.00
ResNet50	0.32	0.19	0.23
ResNet152	0.33	0.31	0.31
DenseNet121	0.32	0.25	0.28
DenseNet201	0.33	0.21	0.25
EfficientNetB0	0.33	0.29	0.31

Table 3.6: Evalutation metrics of base model trained for multi-class classification

GradCAM Evaluation

Gradient Class Activation Map (GradCAM) allows us to see which parts of the image the model is focusing on in making the prediction. This allows to see if the model is infact making decision by "looking" at the right regions (i.e the lesion) to make its prediction. Fig 3.10 shows the result of applying GradCAM on the predictions of all the models. As we can see most of the models struggle to focus on the right areas of the lesion.

3.3.2 Segmentation

Impurity within the dataset can have negative effects on the performance on the models. Images in our dataset contains impurities of different sorts (see 3.11) such as

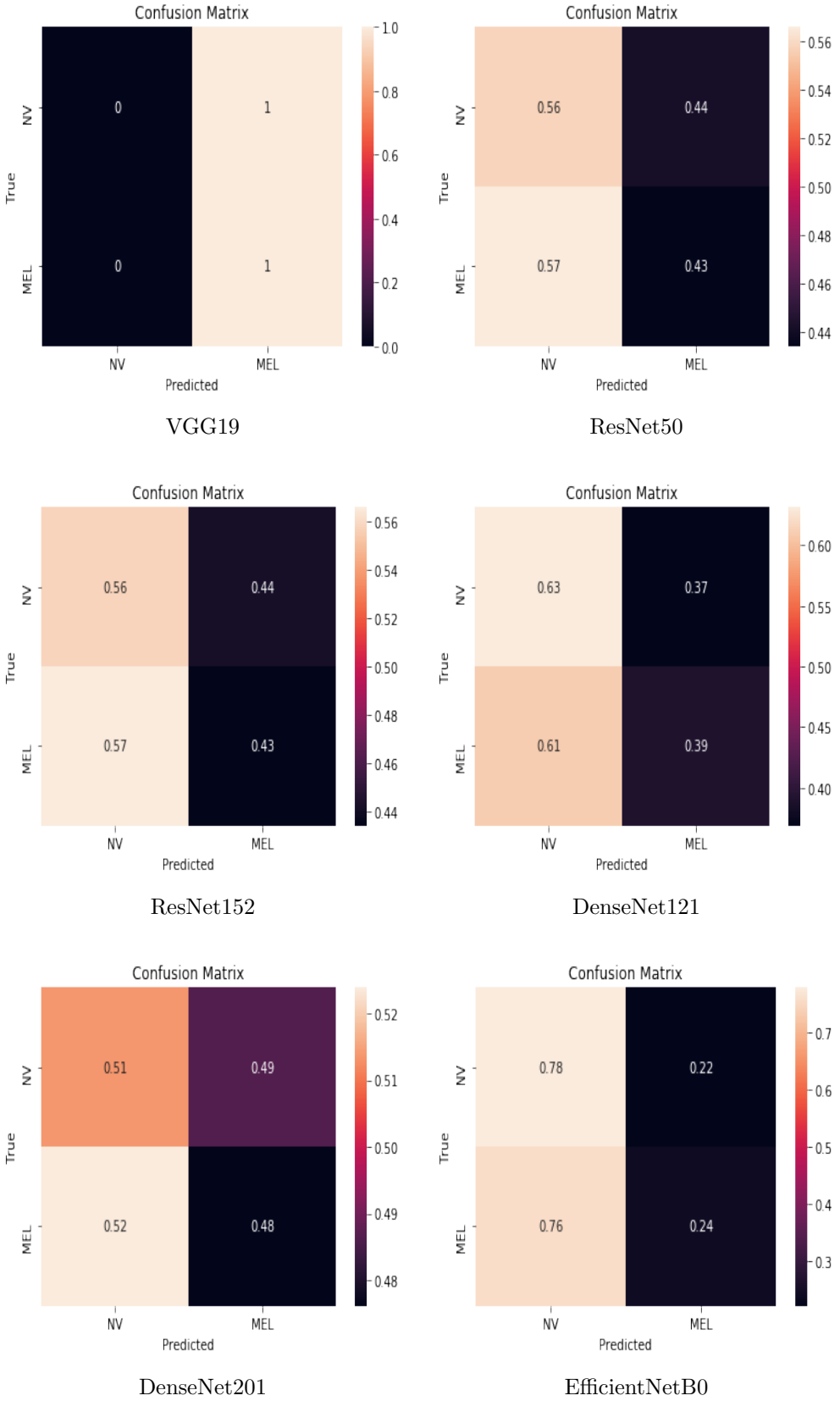


Figure 3.8: Confusion matrices of base model trained on binary classification

foreign objects, bright backgrounds, oil bubbles, thick black border etc. Skin lesion segmentation is a very time consuming and manual labour intensive process that requires high level of expertise. For this reason, segmentation masks for the dataset is much lower especially for the minority classes. This gave raise to development of automatic skin lesion segmentation techniques and for this study, we implement and analyse the affects of such techniques in our classification problem. To conduct this, we make use of an implementation [55] of deep learning based auto-encoder-decoder with U-Net architecture based on the works of Azad, Asadi-Aghbolaghi, Fathy, *et al.* [56]. The architecture of this model is presented in Fig 3.12.

For this experiment, all the images in the dataset were segmented using [55] to create a copy of the dataset with all images segmented and cropped around the segmentations. Fig 3.14 shows the image generated through this process. After the dataset is prepared, both the binary and multi-class classification in performed on this new dataset.

Result

Table 3.7 and Fig. 3.13 presents the results after testing the model. After evaluating the results, we can see that binary classification has benefited from the segmentation slightly with performance increase of 3% in sensitivity. However, the multi-class classification interestingly performed worse with the classification. By further analysis of the confusion matrix we can see that most of the minority classes have been poorly affected. This could be due to the fact that the segmentation tool used to segment the dataset was trained on a smaller dataset where there were no lesion these minority classes hence it does not generalize well to segment these lesions properly. This on top of the very few data already available could be the reason for poor performance. To completely discard the benefit of segmentation on multi-class segmentation it would be necessary to train a segmentation tool with images from all categories including the minority classes and repeat this experiment but due to lack of time this was not possible for this thesis.

Experiment	**Specificity**	**Sensitivity**	**F1**
Segmented (Binary)	0.52	**0.53**	0.52
Segmented (Multi-class)	0.19	0.22	0.21

Table 3.7: Results from binary and multi-class segmented lesion classification

3.3.3 Balance data using General Adversarial Network (GAN)

From these results it is clear that the imbalance in data affects our model's performance quite a lot. An approach to balancing minority classes is the method of

creating synthetic data that is based of the images in the original dataset. Here, we make use of the state of the art StyleGan2 [57] with the help of the pytorch implementation available at [58]. The GAN is trained for 100 epochs on each category saving results at each epoch. The best result is chosen after manual check of the results to create the synthetic images. These synthetic images are then added to the original dataset and treated as though it was always part of it. The motivation for this experiment is to create a model that learns the inherent property of each category and generate images that is indistinguishable from the real data which are then used to balance out the dataset.

Result

After training the model over all categories, the performance was not consistent throughout all categories. The biggest issue was the "mode-collapse" problem where despite the noise fed to the generator it always generates the same image. The classes with the lease amount of samples were affected by this problem the most. Another issue was the problem of strange artifacts being generated for most of the outputs, Figure 3.15 shows this issue. Due to this problem, careful manual selection of images were required before adding to the dataset.

On the other hand, the model performed quite well on the Melanoma class. Figure 3.16 shows an image generated. These images were quite indistinguishable from original images hence it showcases the potential of this technique given enough data.

Therefore, it was decided to use only the synthetic Melanoma images to balance the dataset and run binary classification. Training the model was also very time consuming with each category requiring few days to complete training. Hence, due to time limitation, there was less room for experimentation to improve the performance on minority classes. Therefore, it cannot be said that this method is not a viable option for multi-class classification problem.

3.3.4 Multiple classifier fusion training

Federated learning is a method that has been gaining popularity over the recent years [59]. This method trains many models with identical architecture on different remote devices. The trained model information is then sent to a server where all the individual models that were trained independently is aggregated by various techniques such as average weights. Note that only the model information such as weights are sent to the server while the data remains anonymous.

This method motivates our next approach to reduce imbalance problem. In this proposed method, the whole dataset in divided into multiple random subset with

every image is contained in at least one group. Each of these sub groups are then treated like individual dataset. Figure 3.17 illustrates this process. The rationale behind this approach is to expose each model to a more balanced sub group of the dataset and by combining them we hope to bring the inherent properties learned by each model together to make a more accurate prediction.

As for the training, to ensure all groups start training on the same base, a parent model is created and saved. This model is loaded and trained on each of each of the dataset groups. Finally, after the training is complete, multiple aggregation techniques are explored in hopes of better results.

In order to make sub divisions it is important that we have enough groups to obtain a good balance. Hence, we started small with 5 groups which did not result in performance improvements. So the number of groups were doubled to 10.

To predict on the test set, different aggregation methods were used such as majority voting where each model is used to predict on the same test set and the most commonly predicted class is selected as the final prediction and average where average prediction from all of the model's prediction is taken.

Result

Experiment	**Specificity**	**Sensitivity**	**F1**
Majority voting	0.49	0.53	0.51
Average	0.45	**0.55**	0.49

Table 3.8: Model fusion score for binary classification

Experiment	**Specificity**	**Sensitivity**	**F1**
Majority voting	0.32	**0.36**	0.34
Average	0.33	0.34	0.33

Table 3.9: Model fusion score for multi-class classification

Fusion of models almost always performs better overall than it's single counterpart.

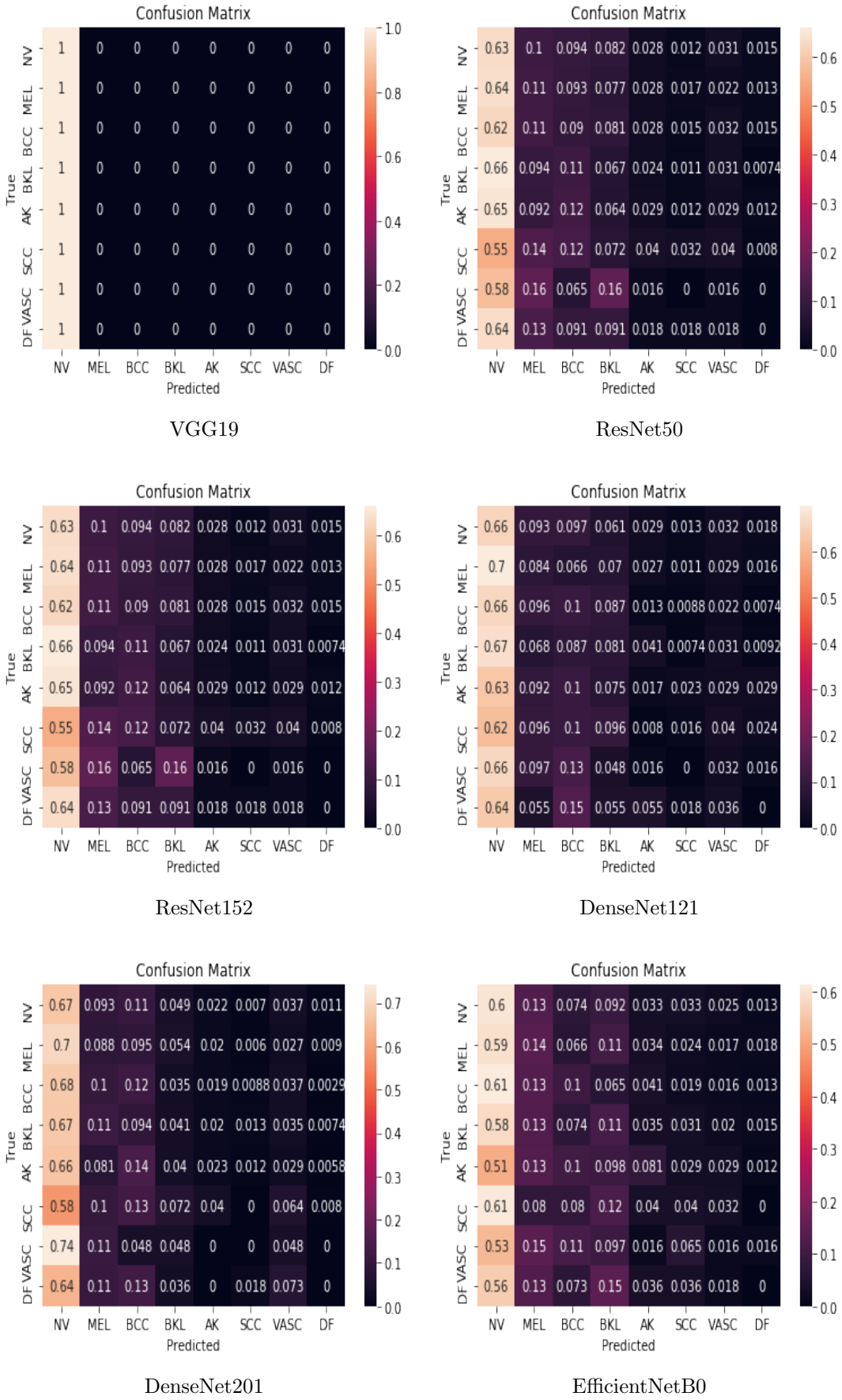


Figure 3.9: Confusion matrices of base model trained on multi-class classification

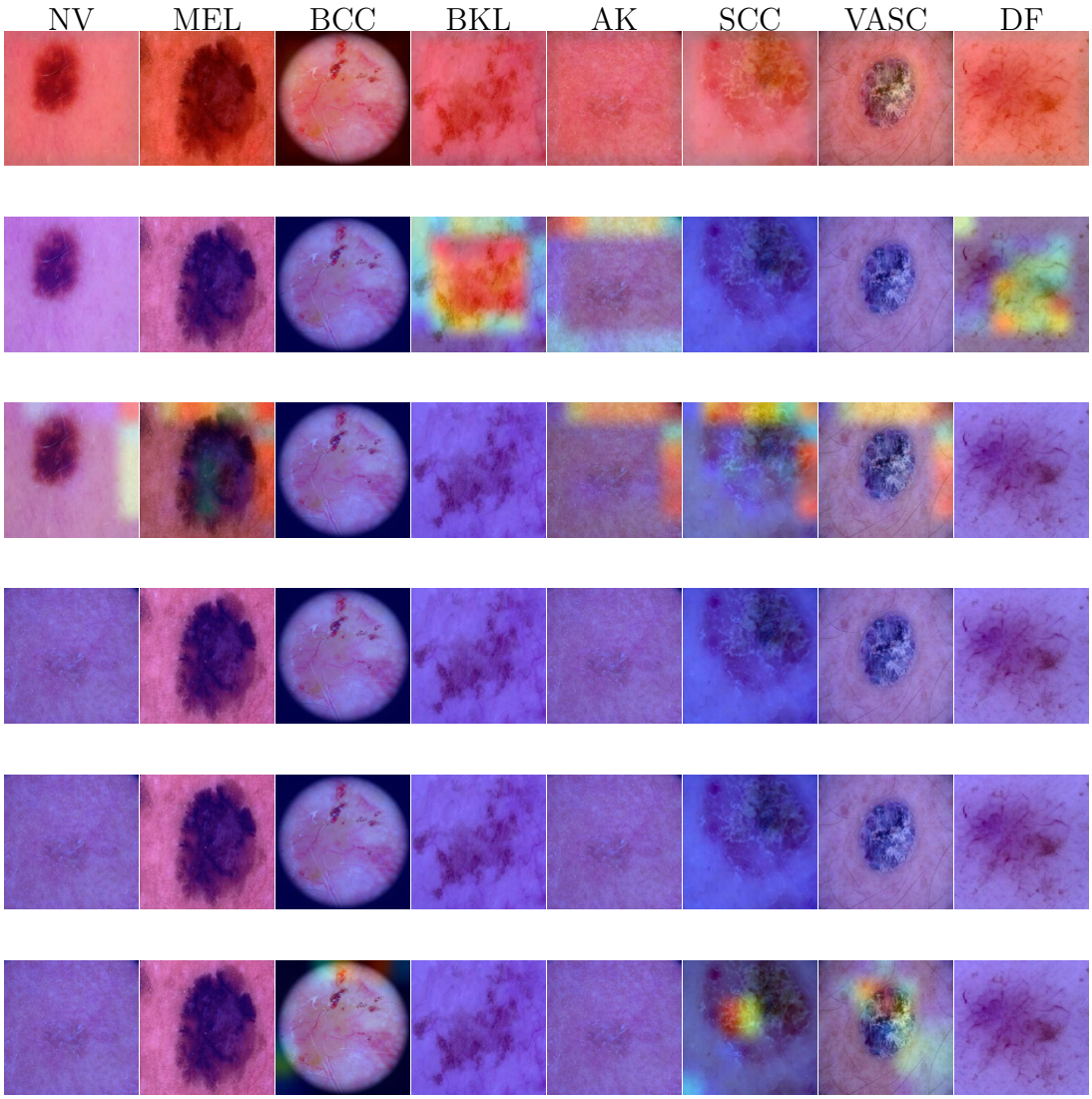


Figure 3.10: Grad-CAM

Figure 3.11: Image that requires segmentation to remove foreign objects [32]

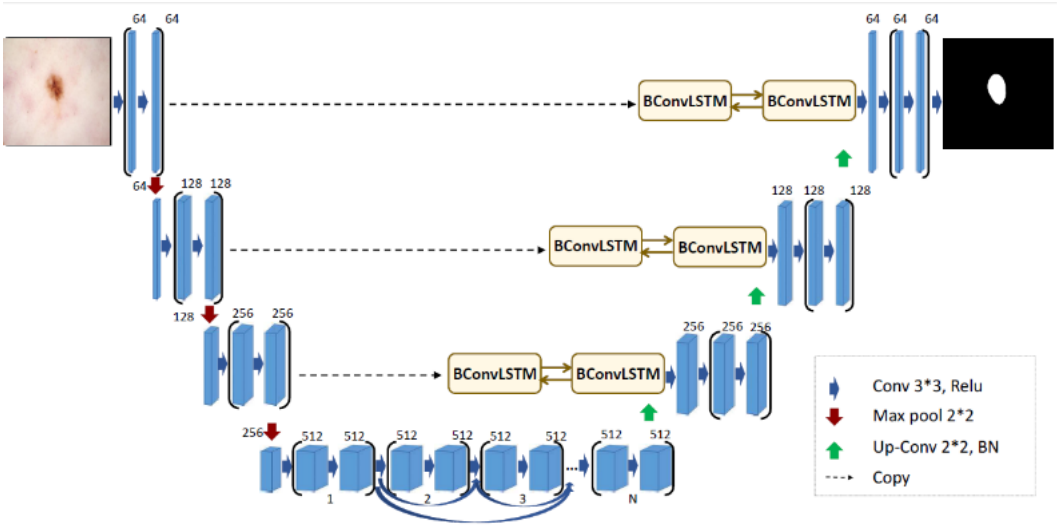
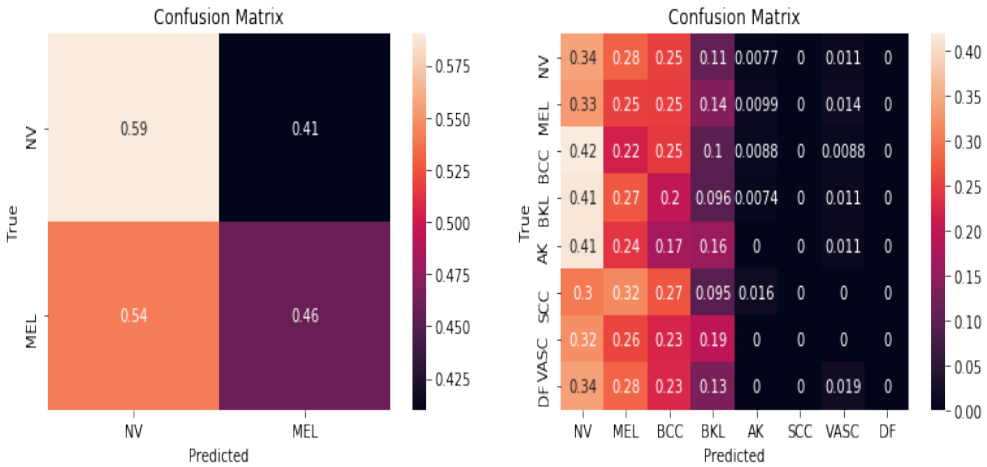
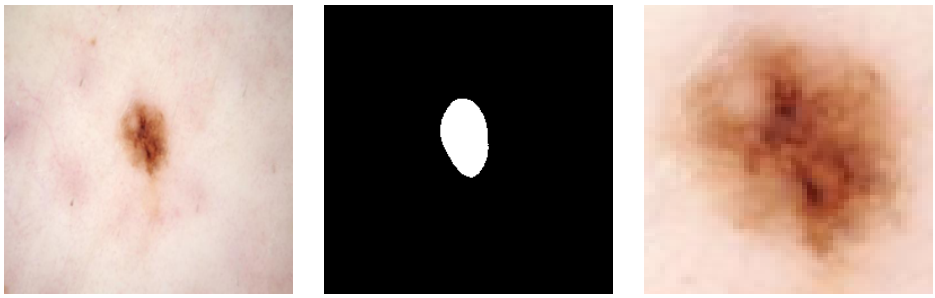


Figure 3.12: Architecture of BCDU-Net segmentation model used automatically segmenting image [55]



(a) Original **(b)** Segmentation

Figure 3.13: Confusion matrices for segmented image classification



(a) Binary **(b)** Segmentation **(c)** Segmentation cropped

Figure 3.14: Skin lesion image automatically segmented by BCDU-Net model

Figure 3.15: Strange artifacts found in images generated by GAN

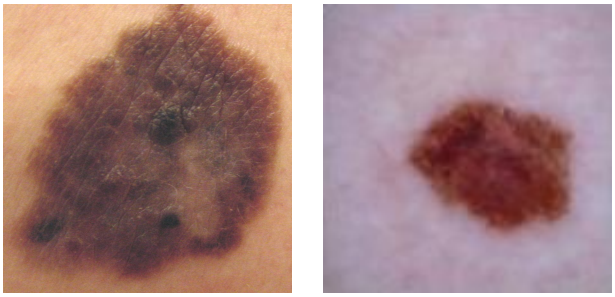


Image from original dataset | Image generated by GAN

Figure 3.16: Comparison of images generated by GAN and image from original dataset

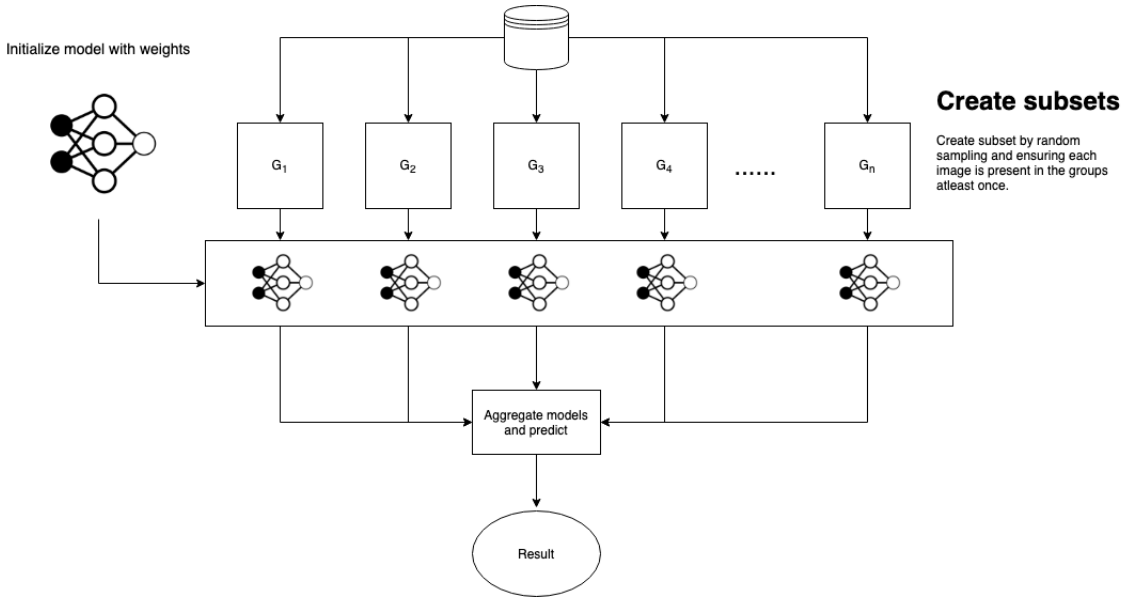


Figure 3.17: Groupwise training approach

4

Mobile application

So far we have been training and evaluating models on a very controlled environment. It is important that these efforts can be used in practical situations. Hence, in this chapter we explore the existing opputunities to bring this technology to the hand of the everyday user.

Machine learning has always been associated with heavy processing and hence, not something to be embedded into devices without heavy computing power. However, year by year new mobile devices are released to the market each one more powerful than the ones before. As popularity of machine learning increases and mobile devices gets powerful enough to handle heavy computing required for machine learning, there now exists solutions to bringing machine learning capabilities to mobile devices. In this chapter, an evaluation of different technologies related to in device machine learning approach that exist currently is made. The challenges of building such a product is described and finally, a mobile based application solution is proposed called Dermaquick, that is intended to replace bulky expensive hardware to aid the problem of early skin cancer detection and bring this technology to the hand of everyday user that can do diagnosis frequently and very easily.

4.1 Exploring options

When it comes to running image classification models there are few different solutions available to us. After exploring popular methods such as web based, "webapp" solution that would make image capture inconvenient using a webcam or a custom hardware solution that requires manufacturing of hardware, we have concluded that a mobile based application would be the most convenient and fastest to be able to build and test.

In the mobile application domain, there are couple of ways to built the application in terms of application architecture which are listed below

- Classification on server : In this method, the model is running on a cloud server where the image captured on mobile device is sent to. The server then does the classification and sends back the result, see Fig 4.1 for illustation. The benefit of this method is that, there is pretty much no computing limitations except for what is limited by affordability. The disadvantages are that it requires an internet connection to send image to the server which can make the classification time slower due to bad network issues. There is also the concern of data security and privacy issues as we are dealing with private medical information. There needs to be careful efforts to make sure that the data stored in server is secure and is safe from any hacking.

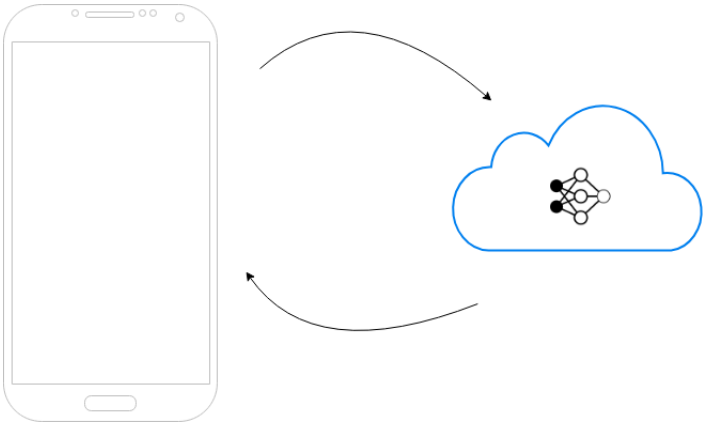


Figure 4.1: Classification on server architecture illustration

- In device classification : Another option is to run everything from start to finish on the device itself. The biggest advantage of this method is the convenience as the user can easily download the application and start doing tests right away. There is no need for internet connection as everything required in embedded in the device including the model. The disadvantage would be that storing the model and all the images in device would take up quite a lot of memory

space which could drive some users away especially ones with the lower end phones.

4.2 Challenges

This section describes the various challenges of running a skin lesion classifier on a mobile device.

4.2.1 Lack of dataset

As of writing this thesis, there is no dataset available that contains solely images of skin lesions captured by mobile devices. Hence, there is no option but to use models trained on clinical data which would obviously affect the prediction accuracy.

4.2.2 Image quality

The biggest issue with running image classification in a mobile device is the image quality. The model that is trained to make predictions is trained on high quality medical images captured in ideal lighting conditions and/or using dermatoscope which captures high level of details without any skin reflection. On the other hand, we cannot guarantee the quality of images in a mobile device as there could be many factors that could negatively the image quality such as poor lighting condition, poor focusing, cheap camera etc. However, we can to some extend make up for these shortcomings through careful pre-processing of images. There are also dermatoscope attachments available on the market that can be simply connected to the mobile phone to capture images that could be on par with the dermatoscopic images, see Fig 4.2.

4.2.3 Storage space

Classification models can be very large in size therefore increasing the size of the application by hundreds of megabytes. Luckily, mobile devices these days are equipped with huge amount of storage space range from 64GB to 128GB or even more. Furthermore, we use TensorFlow Lite to run the model on mobile devices which also reduces the memory footprint more on this will be explained later in the chapter.

4.3 Deploying model

To deploy the model onto mobile devices, we cannot just use the model trained on the powerful computer as it would be very slow in inference, very big in size

Figure 4.2: DermLite - an example of dermatoscope attachment for mobile phones. [60]

and would require more computing power than a small mobile device could provide. Thus, it is necessary to convert this model to a form that reduces the latency and decrease the memory requirement. For this purpose, we make use of TensorFlow Lite [61] that comprises of tools that can convert trained models to an optimized version that can be deployed to various devices.

Models trained using Tensorflow are saved in GraphDef file format that uses the concept of Protocol Buffers (protobuf) which defines the data structure of the model data. This includes all the information about the model including its architecture. Along with this is the checkpoint file that stores all parameter values after each training iteration.

A GraphDef file and checkpoint file can then be combined in a process called "freezing" where each of the values in a checkpoint file is converted to a constant. Finally, this frozen graph file is converted to a TFLite (.tflite) files using Tensorflow Optimization COnverter (TOCO) tool.

4.3.1 Optimization

TFLite file can be quite large for a mobile application. Therefore, we can use the method of quantization to reduce the size of this file by converting 32 bit float values to a reduced precision of numbers. Table 4.1 shows comparison of a quantized tflite file versus one without. The quantized version is one fourth of the size of the original. This, however, comes with the price of reduced accuracy which in most cases is negligible but is something to be tested for.

TFLite file	Size
ResNet152	232 MB
ResNet152 Quantized	58.5 MB

Table 4.1: TfLite file size comparison

4.4 Application functionality

In this section, description of the functionality of the mobile application is given. The application interface can seen at figure 4.3. The application is developed fully on Android Studio using Kotlin as the programming language.

User Login

The application begins with a user login page where each user securely logs into the application using their username and password. This ensures that the sensitive medical record of each user remains private.

Image capture

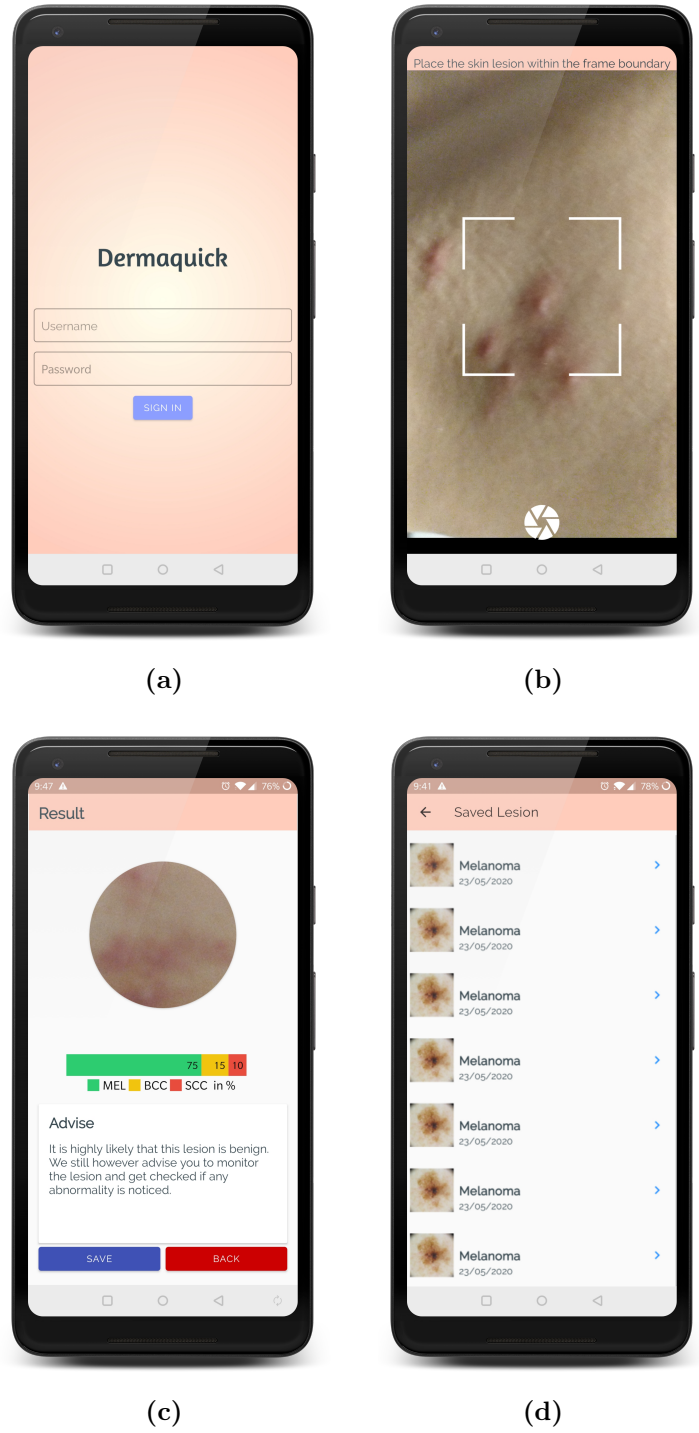
After a successful login, the user is presented with a camera view where the user can capture the image of the lesion. A bounding box is shown on screen with instruction to place the lesion within the bounding box. The user can capture the image by pressing the capture icon just like any other camera application. The captured image is then cropped around the bounding box to discard any background and other irrelevant information. It is then resized to 224x224 pixel dimension to feed into the model.

Classification results

Classification result from the model is displayed in this screen showing the percentage accuracy of the top 3 predictions. This page also gives advise based on the prediction so that the user does not blindly depend on the application. This is very important as the model accuracy is still not high enough to be trusted to be holy word when it comes to diagnosis therefore it should not replace frequent skin checkups. From this page the user can either go back or choose to save the result.

Saved Lesions

All the saved results are shown here for the user to come back to. This lets user to continuously monitor a lesion and compare results at different time.



(a) (b) (c) (d)

Figure 4.3: Mobile application interface

5

Conclusion and Future work

The problem of skin lesion classification is a complex problem with interest of the research world increasing more than ever. As more people fall victim to fatal consequences of skin cancer, it is evident that there is a need for a better solution to facilitate early detection and save lives. Machine learning especially deep learning techniques have shown promise in this kind of medical classification problems. However, the problem of skin lesion classification is especially challenging due to the lack of data and the problem of different skin tone which makes a solution that works well in some location not so much in others. Therefore, we chose this as our research topic to evaluate and study the various methods to understand what works best and why it does not.

We started with selecting a base model that works best for our problem among few of the popular CNN architectures such as ResNet, DenseNet and EfficientNet by training and systematically evaluating them. We also test the effects of applying available automatic segmentation to classification and observed slight improvements in performance of the majority classes as it was the type of data it was trained on.

The imbalance in dataset was clearly evident from initial experimentation, hence, we straight away focused on techniques to that could mitigate this issue as much as possible. We tested various techniques including synthetic data generation techniques using generative adversarial networks and training the data in groups inspired by the popular federated learning technique. Due to lack of data, the synthetic image data image generation did not perform well for minority classes. However, we observed the best results on the final group wise training which shows that classification really benefits from a combination models rather than relying on one single model.

To conclude, skin lesion classification still has long way to go to make it into the hands of everyday user to make accurate and reliable tests conveniently at the comfort of their homes. The plethora of research that is conducted in this area is the right step towards this ultimate goal. As for future work, without any time or

resource restrictions there are many improvements to be done with what we have stated in this study. If a better, more balanced dataset can be acquired by either purchasing the paid datasets or making deal with local dermatologists could help the most to reduce the imbalance problems.

We have seen that out of the models tested, the larger models always seemed to perform better. However, due to memory limitation we could not test even larger models or higher resolution of images. As for synthetic image data creation, instead of training a vanilla GAN model from scratch, a more focused implementation of this concept that is designed to work with skin lesions such as the work from Google [62], could be implemented.

Finally, a mobile based solution was presented for which due to time limitation could only create a prototype. The application has shown the ability to run model, classify and show results quickly. However, as we did not have access to real patients, we could not test if there is any drop in accuracy through testing which we hypothesize there would be since the image quality of a mobile camera would not be as great as an image taken in controlled clinical setting. In future, it would be worthwhile to test if this drop in accuracy if there is any is substantial. There are also many features to improve user experience and promote healthy practices that can be added to improve the effectiveness of the application such as

- Reminders : Provide alert notifications to user reminding them to conduct tests more often.
- Invitation : Invite friends and family to do frequent checkups
- Awareness : Educate users more on the consequences and steps to take care of the skin to prevent skin cancer in the first place using information from credible sources.
- Aid checkups : If a user is found to be cancer positive, then help user to find nearest dermatologist to do checkups and present information about the procedure to deal with it in such a way that the user takes the matter seriously yet does not panic.

Overall, there is still quite a lot of work to be done in this field but judging by the rate of progress made over just the past few years, it is possible to hope for a future where the fatality rates due to skin cancer is extremely low.

Reference List

[1] D. C. Whiteman, R. E. Neale, J. Aitken, L. Gordon, A. C. Green, M. Janda, C. M. Olsen, and H. P. Soyer, "When to apply sunscreen: A consensus statement for australia and new zealand.," *Australian New Zealand Journal of Public Health*, vol. 43, no. 2, pp. 171–175, 2019, ISSN: 13260200. [Online]. Available: http://ezproxy.uow.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=sih&AN=135665945&site=eds-live.

[2] P. A. T. E. Board, "Melanoma treatment (pdq®): Health professional version," in *PDQ Cancer Information Summaries*. National Cancer Institute (US), 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK66034/.

[3] L. Ferrante di Ruffano, Y. Takwoingi, J. Dinnes, N. Chuchu, S. E. Bayliss, C. Davenport, R. N. Matin, K. Godfrey, C. O'Sullivan, A. Gulati, and et al., "Computer-assisted diagnosis techniques (dermoscopy and spectroscopy-based) for diagnosing skin cancer in adults," *The Cochrane Database of Systematic Reviews*, vol. 12, 2018.

[4] K. Holland, *Whats causing this skin lesion?* https://www.healthline.com/health/skin-lesions, 2016.

[5] J. Mothoneos, "Understanding skin cancer: A guide for people with cancer, their families and friends," 2018. [Online]. Available: https://www.cancercouncil.com.au/skin-cancer/#common.

[6] R. Happle, "What is a nevus? a proposed definition of a common medical term," *Dermatology (Basel, Switzerland)*, vol. 191, no. 1, pp. 1–5, 1995, ISSN: 1018-8665.

[7] R. L. Moy, "Clinical presentation of actinic keratoses and squamous cell carcinoma," *Journal of the American Academy of Dermatology*, vol. 42, no. 1 Pt 2, pp. 8–10, Jan. 2000.

[8] J. M. Jackson, A. Alexis, B. Berman, D. S. Berson, S. Taylor, and J. S. Weiss, "Current understanding of seborrheic keratosis: Prevalence, etiology, clinical presentation, diagnosis, and management," *Journal of drugs in dermatology: JDD*, vol. 14, no. 10, pp. 1119–1125, Oct. 2015, ISSN: 1545-9616.

[9] R. J. Friedman, D. S. Rigel, and A. W. Kopf, "Early detection of malignant melanoma: the role of physician examination and self-examination of the skin," *CA Cancer J Clin*, vol. 35, no. 3, pp. 130–151, 1985.

[10] B. P. Hazen, A. C. Bhatia, T. Zaim, and R. T. Brodell, "The clinical diagnosis of early malignant melanoma: expansion of the ABCD criteria to improve diagnostic sensitivity," *Dermatol. Online J.*, vol. 5, no. 2, p. 3, Nov. 1999.

[11] Arta Dermatology, *Skin cancer at arta dermatology, newport beach*, [Online; accessed May 18, 2020]. [Online]. Available: `http://www.artaderm.com/images/abcdes-of-melanoma-large.png`.

[12] M. F. Healsmith, J. F. Bourke, J. E. Osborne, and R. A. Graham-Brown, "An evaluation of the revised seven-point checklist for the early diagnosis of cutaneous malignant melanoma," *Br. J. Dermatol.*, vol. 130, no. 1, pp. 48–50, Jan. 1994.

[13] Allan, C., Halpern and Ashfaq, A., Marghoob and Ofer, Reiter, *Melanoma warning signs.what you need to know about early detection*, 2019. [Online]. Available: `https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/#abcde`.

[14] W. Mcculloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 127–147, 1943.

[15] C. Van Der Malsburg, "Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms," in *Brain Theory*, G. Palm and A. Aertsen, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 245–248, ISBN: 978-3-642-70911-1.

[16] [Online; accessed May 08, 2020]. [Online]. Available: `https://simple.wikipedia.org/wiki/Neuron`.

[17] [Online; accessed May 08, 2020]. [Online]. Available: `http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/img18.gif`.

[18] *7 types of neural network activation functions: How to choose?* [Online; accessed May 08, 2020]. [Online]. Available: `https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/`.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., 2012, pp. 1097–1105.

[20] *6 basic things to know about convolution*, [Online; accessed May 08, 2020]. [Online]. Available: `https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411`.

[21] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: `1512.03385`. [Online]. Available: `http://arxiv.org/abs/1512.03385`.

[23] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. arXiv: `1608.06993`. [Online]. Available: `http://arxiv.org/abs/1608.06993`.

[24] D. Sengupta, "Deep learning architectures automated image segmentation," PhD thesis, Mar. 2019.

[25] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019. eprint: `arXiv:1905.11946`.

[26] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[27] Chris Nicholson, *A beginner's guide to generative adversarial networks (gans)*, [Online; accessed Jun 01, 2020]. [Online]. Available: `https://pathmind.com/images/wiki/GANs.png`.

[28] 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*, vol. 5, no. 1, Aug. 2018. [Online]. Available: `https://doi.org/10.1038/sdata.2018.161`.

[29] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. K. Mishra, H. Kittler, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (ISIC)," *CoRR*, vol. abs/1710.05006, 2017. [Online]. Available: `http://arxiv.org/abs/1710.05006`.

[30] M. Combalia, N. C. F. Codella, V. Rotemberg, B. Helba, V. Vilaplana, O. Reiter, C. Carrera, A. Barreiro, A. C. Halpern, S. Puig, and J. Malvehy, "BCN20000: Dermoscopic Lesions in the Wild," *arXiv e-prints*, Aug. 2019. eprint: `1908.02288`.

[31] *Dermnet nz*, https://www.dermnetnz.org/.

[32] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira, "Ph2 - a dermoscopic image database for research and benchmarking," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2013, pp. 5437–5440.

[33] I. Giotis, N. Molders, S. Land, M. Biehl, M. F. Jonkman, and N. Petkov, "Mednode: A computer-assisted melanoma diagnosis system using non-dermoscopic images," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6578–6585, 2015, ISSN: 0957-4174. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415002705.

[34] S. S. Han, *Asan and hallym dataset (thumbnails)*, Sep. 2017. [Online]. Available: https://figshare.com/articles/Asan_and_Hallym_Dataset_Thumbnails_/5406136/1.

[35] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, "Seven-point checklist and skin lesion classification using multitask multimodal neural nets," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 538–546, Mar. 2019, ISSN: 2168-2194.

[36] A. Tharwat, "Classification assessment methods: A detailed tutorial," Aug. 2018.

[37] [Online; accessed May 08, 2020]. [Online]. Available: https://miro.medium.com/max/722/1*pk05QGzoWhCgRiiFbz-oKQ.png.

[38] [Online; accessed May 08, 2020]. [Online]. Available: https://raw.githubusercontent.com/nickbiso/Keras-Class-Activation-Map/master/readme_images/output1.jpeg.

[39] M. I. Razzak, S. Naz, and A. Zaib, *Deep learning for medical image processing: Overview, challenges and future*, 2017. eprint: arXiv:1704.06825.

[40] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, Jan. 2017. [Online]. Available: https://doi.org/10.1038/nature21056.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, 2014. eprint: arXiv:1409.4842.

[42] K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Classification of skin lesions using transfer learning and augmentation with Alex-net," *PLOS ONE*, pp. 1–17, 2019.

[43] J. Humayun, A. Malik, and N. Kamel, "Multilevel thresholding for segmentation of pigmented skin lesions.," *2011 IEEE International Conference on Imaging Systems and Techniques, Imaging Systems and Techniques (IST), 2011 IEEE International Conference on*, pp. 310–314, 2011, ISSN: 978-1-61284-894-5.

[44] H. Fan, F. Xie, Y. Li, Z. Jiang, and J. Liu, "Automatic segmentation of dermoscopy images using saliency combined with otsu threshold.," *Computers in Biology and Medicine*, vol. 85, pp. 75–85, 2017, ISSN: 0010-4825.

[45] Y. Yuan, M. Chao, and Y. Lo, "Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance.," *IEEE Transactions on Medical Imaging, Medical Imaging, IEEE Transactions on, IEEE Trans. Med. Imaging*, vol. 36, no. 9, pp. 1876–1886, 2017, ISSN: 0278-0062.

[46] P. Tschandl, C. Sinz, and H. Kittler, "Domain-specific classification-pretrained fully convolutional network encoders for skin lesion segmentation," *Computers in Biology and Medicine*, vol. 104, no. 2019, pp. 111–116, 2019.

[47] *Isic archive*. [Online]. Available: `https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main`.

[48] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, *Skin lesion classification using ensembles of multi-resolution efficientnets with meta data*, 2019. eprint: `arXiv:1910.03910`.

[49] R. Dubey, J. Zhou, Y. Wang, P. M. Thompson, and J. Ye, "Analysis of sampling techniques for imbalanced data: An n=648 adni study," *NeuroImage*, vol. 87, pp. 220–41, Feb. 2014.

[50] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data.," *BMC Bioinformatics*, vol. 14, no. 1, pp. 1–16, 2013, ISSN: 14712105.

[51] G. Monheit, A. B. Cognetta, L. Ferris, H. Rabinovitz, K. Gross, M. Martini, J. M. Grichnik, M. Mihm, V. G. Prieto, P. Googe, R. King, A. Toledano, N. Kabelev, M. Wojton, and D. Gutkowicz-Krusin, "The Performance of MelaFind: A Prospective Multicenter Study," *Archives of Dermatology*, vol. 147, no. 2, pp. 188–194, Feb. 2011, ISSN: 0003-987X. [Online]. Available: `https://doi.org/10.1001/archdermatol.2010.302`.

[52] *Melafind® (melanoma detection)*, [Online; accessed May 08, 2020]. [Online]. Available: `https://www.cancerquest.org/patients/detection-and-diagnosis/melafindr`.

[53] F. Chollet *et al.*, *Keras*, `https://keras.io`, 2015.

[54] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[55] rezazad68, *Bcdu-net : Medical image segmentation*, `https://github.com/rezazad68/BCDU-Net`, 2019.

[56] R. Azad, M. Asadi-Aghbolaghi, M. Fathy, and S. Escalera, "Bi-directional convlstm u-net with densley connected convolutions," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[57] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *arXiv preprint arXiv:1912.04958*, 2019.

[58] lucidrains, *Stylegan2-pytorch*, `https://github.com/lucidrains/stylegan2-pytorch`, 2019.

[59] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, *Towards federated learning at scale: System design*, 2019.

[60] Dermlite, *Dlua; universal smartphone adapter*, [Online; accessed Jun 08, 2020]. [Online]. Available: `https://dermlite.com/products/universal-smartphone-adapter`.

[61] *Tensorflow lite*, [Online; accessed Jun 08, 2020]. [Online]. Available: `https://www.tensorflow.org/lite`.

[62] A. Ghorbani, V. Natarajan, D. Coz, and Y. Liu, *Dermgan: Synthetic generation of clinical skin images with pathology*, 2019. eprint: `arXiv:1911.08716`.

A

Appendix One

A.1 Importants snippets of code

A.1.1 Train Classifier

```
from comet_ml import Experiment
import numpy as np
from matplotlib import pyplot as plt
import os
import time
import tensorflow as tf

CATEGORIES = ["NV", "MEL", "BCC", "BKL", "AK", "SCC", "VASC", "DF"]

# Set seed for replicability
np.random.seed(1234)

base = ResNet152(input_shape=(IMG_CROP_SIZE,IMG_CROP_SIZE,3),
    weights="imagenet",include_top=False)
x = base.output
x = Flatten()(x)
predictions = Dense(8, activation='softmax',name='output_layer')(x)
model = Model(base.inputs, predictions)
model.compile(loss='categorical_crossentropy',optimizer='adam' ,
    metrics=['accuracy'])

# Freeze layers
nLayers = len(model.layers)
freeze_layers = int(nLayers * 0.20)
for l in range(freeze_layers):
    model.layers[l].trainable = False


```

```
datagen = ImageDataGenerator(rescale=1./255.,
                             rotation_range=90,
                             horizontal_flip=True,
                             vertical_flip=True,
                             brightness_range=(0.8, 1.2),
                            validation_split=0.2)

val_datagen = ImageDataGenerator(rescale=1./255.)

train_generator = datagen.flow_from_directory(
    target_size=(IMG_SIZE, IMG_SIZE),
    directory=train_path_isic,
    batch_size=BATCH_SIZE,
    shuffle = True,
    classes=CATEGORIES,
    subset="training",
    class_mode="categorical")

val_generator = datagen.flow_from_directory(
    batch_size=BATCH_SIZE,
    target_size=(IMG_SIZE, IMG_SIZE),
    directory = train_path_isic,
    shuffle = True,
    subset="validation",
    classes=CATEGORIES,
    class_mode = "categorical")


# Run model

# Custom callback for evaluation and saving model
metrics = CustomMetricsAndEarlyStop(10)

# Track training using Comet.ml
experiment = Experiment(api_key="ufQQBm4PhfVWxcwobrFapECw8",
                        project_name="skin-lesion-classification",
    workspace="rsd96")
import time
import multiprocessing
t=time.time()

experiment.set_name(model_name)
print(train_generator.samples)
hist = model.fit(train_crop_generator,
                           validation_data = val_crop_generator,
                           epochs=100,
                           shuffle=True,
```

```
                                workers = multiprocessing.cpu_count(),
                                callbacks = [lr_on_plateau_callback, metrics],
                                steps_per_epoch=train_generator.samples // BATCH_SIZE,
                                validation_steps=val_generator.samples // BATCH_SIZE)

experiment.end()
print('Training time: %s' % (time.time() - t))
```

A.1.2 Custom callback for evaluation and saving model

```
class CustomMetricsAndEarlyStop(tf.keras.callbacks.Callback):

    def __init__(self, patience=5):
        self.patience = patience
        # best_weights to store the weights at which the minimum loss occurs.
        self.best_weights = None

    def on_train_begin(self, logs={}):

        # The number of epoch it has waited when loss is no longer minimum.
        self.wait = 0
        # The epoch the training stops at.
        self.stopped_epoch = 0
        # Initialize the best as infinity.
        self.best = 0

    def on_epoch_end(self, epoch, logs={}):
        print(f"Log: {logs}")
        val_acc = logs['val_accuracy']


        current = val_acc
        if np.greater(current, self.best):
            save_path = f"/home/mrb685/Saved Models/{model_name}.h5"
            print(f"val_accuracy improved from {self.best} to {current}. Saving model!")
            self.model.save(save_path)
            self.best = current
            self.wait = 0
```

```
            # Record the best weights if current results is better (less).
            self.best_weights = self.model.get_weights()
        else:
            self.wait += 1
            if self.wait >= self.patience:
                self.stopped_epoch = epoch
                self.model.stop_training = True
                print('Restoring model weights from the end of the best epoch.')
                self.model.set_weights(self.best_weights)

    def on_train_end(self, logs=None):
        if self.stopped_epoch > 0:
            print('Epoch %05d: early stopping' % (self.stopped_epoch + 1))
```

B

Appendix Two

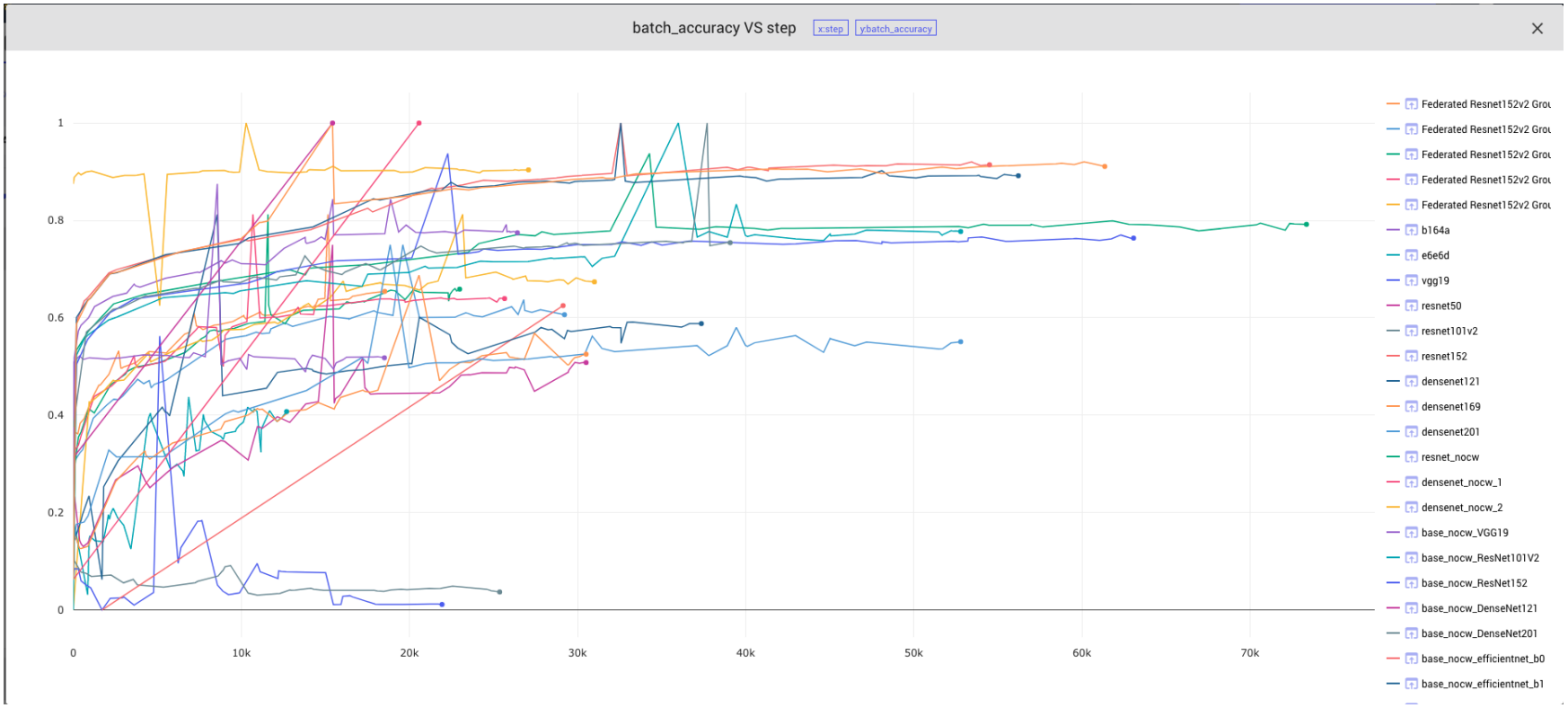


Figure B.1: Batch accuracy vs step count of all trained models

B.1 Model training history