Segmentation and Classification of Melanoma

by

Vivek Verma

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2021 by the
Graduate Supervisory Committee:

Sebastien Motsch, Co-Chair
Spring Berman, Co-Chair
Houlong Zhuang

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

A skin lesion is a part of the skin which has an uncommon growth or appearance in comparison with the skin around it. While most are harmless, some can be warnings of skin cancer. Melanoma is the deadliest form of skin cancer and its early detection in dermoscopic images is crucial and results in increase in the survival rate. The clinical ABCD (asymmetry, border irregularity, color variation and diameter greater than 6mm) rule is one of the most widely used method for early melanoma recognition. However, accurate classification of melanoma is still extremely difficult due to following reasons(not limited to): great visual resemblance between melanoma and non-melanoma skin lesions, less contrast difference between skin and the lesions etc. There is an ever-growing need of correct and reliable detection of skin cancers. Advances in the field of deep learning deems it perfect for the task of automatic detection and is very useful to pathologists as they aid them in terms of efficiency and accuracy. In this thesis various state of the art deep learning frameworks are used. An analysis of their parameters is done, innovative techniques are implemented to address the challenges faced in the tasks, segmentation, and classification in skin lesions.

- Segmentation is task of dividing out regions of interest. This is used to only keep the ROI and separate it from its background.
- Classification is the task of assigning the image a class, i.e., Melanoma(Cancer) and Nevus(Not Cancer). A pre-trained model is used and fine-tuned as per the needs of the given problem statement/dataset.

Experimental results show promise as the implemented techniques reduce the false negatives rate, i.e., neural network is less likely to misclassify a melanoma.

ACKNOWLEDGMENTS

I would like to thank Dr. Sebastien Motsch for being such an amazing guide for my thesis work and giving me the opportunity to work with him. He motivated me to work harder and learn, but most important of all always believed in me. I refer to his notes on neural networks whenever I am solving problems related to deep learning. His help throughout this journey is invaluable.

I would like to thank Dr. Spring Berman for being such an amazing teacher and motivator. She always takes extra efforts to guide and help her students. I learnt a lot in her robotics class and am going to miss it.

I would like to thank Dr. Houlong Zhuang for teaching me the concepts of machine learning and always helping with questions related to coursework. His eagerness to help really shows.

I would also like to thank Arizona State University for providing me access to GPU at Agave and Mathematics clusters to perform simulations and pursue my research work. Without their support this work would not have been possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Melanoma is the most dangerous form of skin cancer and has the ability to spread to different parts of the body is left untreated. It results in approximately 75% of deaths related to skin cancer[1]. Therefore, it is crucial to correctly detect it at a much earlier stage, this results in a higher survival rate of patients. Clinical diagnosis of melanoma with an unaided eye is only about 60%[2].

Extensive research and advancements have been made in the field of deep learning in computer vision and they have been gaining a lot of dominance. Today, it can outperform humans in multiple areas such as detection, classification in digital images with less than 5%[3]. An automatic system that can be relied upon for melanoma detection is valuable for the pathologists as it increases their effectiveness and accuracy. These can be readily run on easily available hardware, hence increasing their reach.

Dermoscopy technique is a noninvasive technique in which magnified and clear images of cancer suspected skin regions are taken. This helps in enhancing the visual features of the skin lesion and aids in detection[4,5]. Nonetheless, the task of detecting melanoma using deep learning techniques poses several challenges. Few reasons are(not limited to): Subtle visual differences between melanoma and non-melanoma patches, less contrast difference between skin and lesions also, variations in the skin conditions, e.g., color of the skin, hair present around the patch[6]. This results in the melanoma patch having different type of characteristics, color, etc.

Segmentation is a fundamental step towards classification in a lot of approaches. A comprehensive algorithms study on automated skin lesion segmentation is available in [7]. Segmentation can help increase the accuracy of classification. A lot of studies have been done to achieve decent segmentation results[8,9,10,11,12,13]. On the basis of results obtained from segmentation, features can hence be extracted for melanoma detection.

Even though a lot of work has been carried out, there is still a lot of place for performance improvement in segmentation and classification of skin lesions. The International Skin Imaging Collaboration(ISIC): Melanoma Project is a focused towards facilitating the application of digital skin imaging to reduce mortality due to melanoma. They have developed and are expanding their data-set archive of skin images since 2016. It is an open-source public access archive to facilitate the development and testing of automated diagnostic systems. They have set new standards in the area of dermoscopic feature extraction.

The contributions of this thesis can be summarized as follows:

1. Diving into U-Net for segmentation and doing a comprehensive analysis. Also, experimenting with small architectures in order to better understand performance of complex and simple architectures.
2. Classification of the patches into Nevus(Non-Cancerous) and Melanoma(Cancerous) using a pre-trained model and fine tuning its parameters which better suits our problem statement/dataset. This is also known as transfer learning[14] and is an active field of study in deep learning.
3. Researching a strategy to include segmented data into the classification network for better performance.

For the task of classification, a pre-trained model was picked and fine-tuned with training to better serve us on our problem statement and dataset. This is known as transfer learning and is a widely used technique. It is also an active research topic in the field of deep learning.

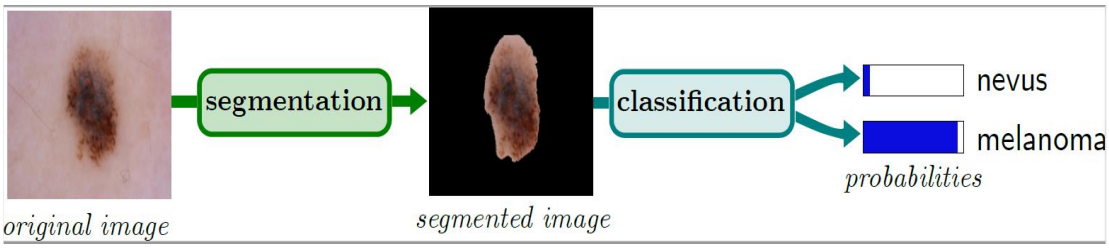

Figure 1.1: Overview of The Process

CHAPTER 2

SEGMENTATION

Image Segmentation is one of the key topics in the field of image processing and computer vision and plays a crucial role in applications areas such as medical image analysis, perception in robots, augmented reality, image compression and much more.

Segmentation involves dividing a visual input into segments in order to simplify its analysis. From all the different regions/objects that the networks segment we pick only the important ones for our analysis. The image is a collection of different pixels, we group together pixels which belong to the same category/class. It is generally done using a bounding-box method where we place a box around the region of interest or a pixel wise labelling resulting in different classes being highlighted with different colors.

Our goal using a segmentation algorithm is to extract the region of a mole from a given image and remove the background which is the skin. We are using the ISIC-2018 data set for the task of segmentation and it contains 2594 images and masks.

Mask is the ground truth of the input image, meaning it contains pixel level information of which pixel has a class mole and not a mole. In terms of numerical values, the mask is a binary one, which means it contains 0 and 1 as pixel values. 0 corresponds to that pixel not being a mole and 1 meaning it is a mole. It is a gray scale image with a black and white appearance and contains only a single channel. An example of an image and its mask is given in the figure 2.1.

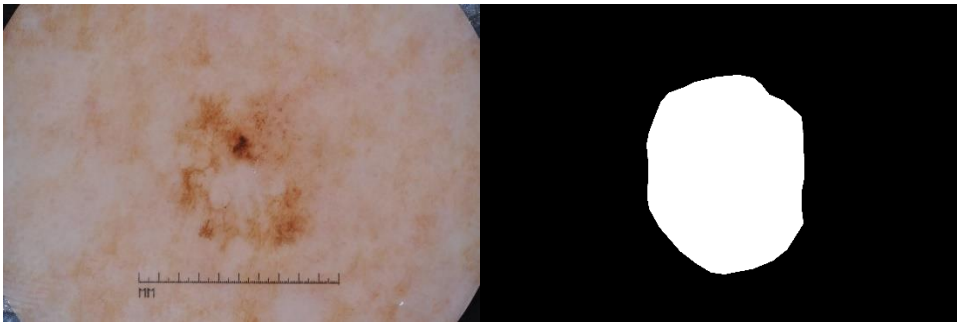

Figure 2.1: Image and Mask

2.1 Pre-Processing

The input images for training the deep learning model were normalized before the training process. This helps in getting it within a certain threshold range, it reduces the skewness[15] which helps the network learn better and faster. Mean pixel value and standard deviation of the three-color channels namely, Red-Green-Blue was estimated. Using the below values data was normalized and equation 2.1 was used for normalization.

- Mean r-g-b value: (0.708, 0.582, 0.536)
- Standard deviation: (0.0978, 0.113, 0.127)

$$Output[Channel] = \frac{Input[Channel] - Mean[Channel]}{Standard\ Deviation[Channel]} \quad (2.1)$$

An example of original image and normalized image can be found in Figure 2.2.

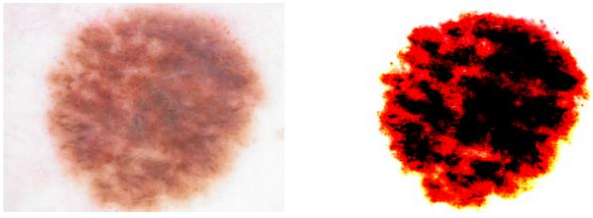

Figure 2.2: Original Image and Normalized Image

2.2 Data-Augmentation & Dice Index

Data augmentation is a set of techniques used to increase the amount of data available by adding moderately modified duplicate of data or create synthetic data newly from the already existing data. This helps to reduce overfitting[16] when training the model by acting as a regularizer[17]. There are a plethora of data augmentation techniques and were randomized with a probability of 50% of any of them happening. The performed techniques and visual examples are discussed in this section.

- Horizontal flipping: Horizontal flip augmentation is when the columns of the input image is reversed.

Figure 2.3: Original Image and Flipped Image

- Rotation: Rotation augmentation is done by rotating the image between -180º and 180º.

Figure 2.4: Original Image and Rotated Image

- Resized crop: Resized crop augmentation is when a random subset is created from the original image and scaled back to a given size.

Figure 2.5: Original Image and Resized Cropped Image

All images are all resized to 400x400 for training purpose. All the previous techniques combined generate an image the same as in figure 2.6.

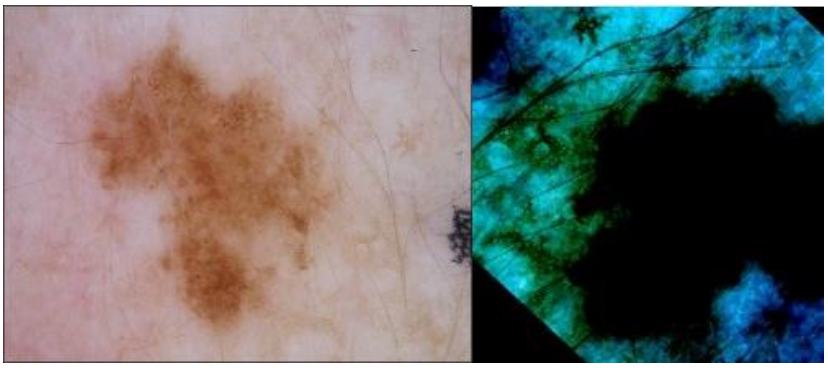

Figure 2.6: Original Image and Transformed Image

The same spatial data augmentation is applied to the mask as well. It keeps the image and mask both identical in terms of orientation. If both were to be aligned differently for

training, the model would not learn anything useful from it and prediction would be near random.

Dice index[18] is a statistical tool to find the similarity between two images and was used as the metric for accuracy. Formula of dice index can be found in equation 2.2.

$$2 \times \frac{|X \cap Y|}{|X| + |Y|} \tag{2.2}$$

Where X,Y are binary vectors. One signifies ground truth and the other signifies the model prediction. This is used to evaluate the similarity between the original mask and the predicted mask.

2.3 Deep Learning Architecture & Results

One Neuron Model: This architecture is the most basic one aimed at experimenting at how a model with as low as mere 8 parameters would perform at such a complex task of semantic segmentation. The r-g-b values from the image serves as the input for the two nodes. The two nodes or neurons[19] are responsible for giving out scores for the input pixel being skin and mole respectively, i.e., the model outputs a two-channel feature map of the score given to pixels from the input image. Visual representation of the architecture is given in the figure 2.7.

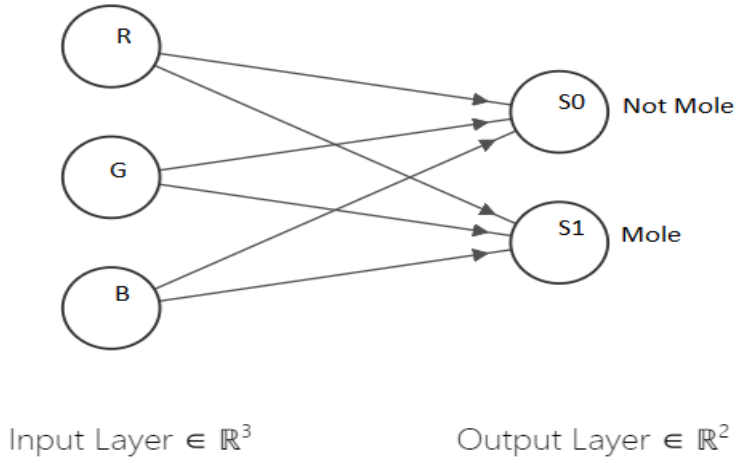

Figure 2.7: One Neuron Model

The train-test split[20] was kept as 90-10% along with a batch size[21] of 2. Learning rate of 10^{-3} was kept and the model was trained for 25 epochs. Since the number of parameters were small the model was quick to learn to its maximum limit. We could see that 3 epochs were enough for the model to learn, post which negligible changes were seen in the training and testing curves. The average dice achieved during testing was approximately 0.58. The results are not good and would lead in poor segmentation since it would contain a lot of misclassifications. The results show that 8 parameters do not serve the purpose of accurate segmentation. One Neuron training plot is given in the figure 2.8.

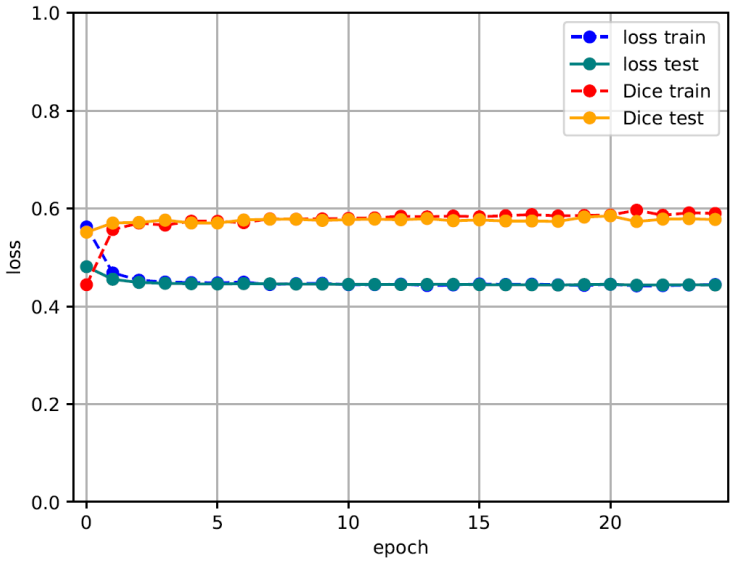

Figure 2.8: One Neuron Training Result

U-Net: The U-Net architecture[22] was specifically designed for biomedical Image segmentation[23]. U-Net architecture consists of two paths. The first path is contraction path(also called encoder) and is used to capture the features of the input images. Encoder contains convolution and max pooling layers. The second path is an expansion path symmetric to the encoder(also called decoder). Decoder contains transposed convolution[24] layers needed to up sample the image and convolution layers to keep extracting features. These layers reverse the standard convolution by dimensions thus creating a feature map having dimension greater than the input feature map.

This architecture also leverages the power of skip connections[25] at 4 different channel levels. Essentially, skip connections are connections from layers earlier in the architecture to the layers that come later via addition or concatenation. This has been experimentally proven that skip connections help the model converge faster. This is an end-to-end fully

convolution network and does not contain linear layers, hence it can accept images of any size. The output contains of 2 channels with each one representing the score of the that pixel being a skin and mole as predicted by the model. A SoftMax layer[26] is used on the score to get the class of the pixel. Formula for a SoftMax function can be found in equation 2.3.

$$\sigma(\mathrm{x}_i) = \frac{e^{x_i}}{\sum_{j=i}^{j=k} e^{x_j}} \tag{2.3}$$

ReLu[27] is the choice of activation function[28] in this architecture. Rectified Linear Units will output the input directly if its positive but will output 0 if the input is negative. ReLu activation can be described by the equation 2.4.

$$ReLu(x) = \max(0, x) \tag{2.4}$$

U-Net was divided into 4 different depth levels for studying how the performance changes with the number of parameters in the network. Every depth level has different number of channels the model is operating at from its original input of 3 channels. The hyper parameter's batch size and learning were kept as 1 and 10^{-4} for all different version of U-Nets.

The U-Net architecture can be found in the figure 2.9.

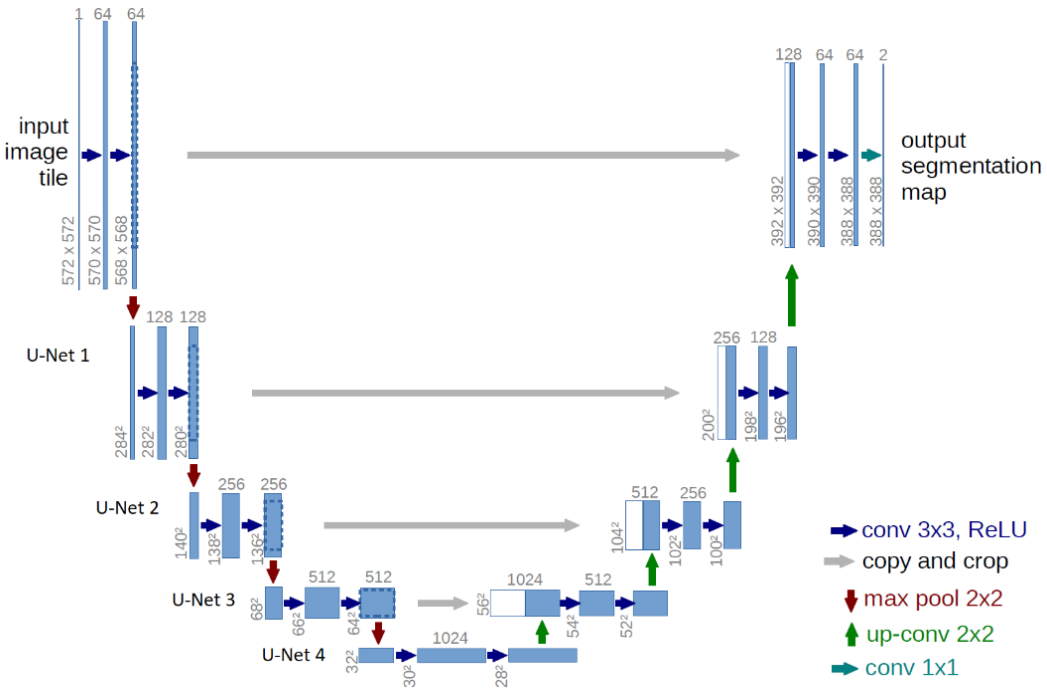

Figure 2.9: U-Net Architecture

We explore various modifications in the U-Net(see Figure 2.9) where we reduce the complexity of the neural network to see its effects and study the increase/decrease in performance with the change in number of parameters. For instance, in U-Net 1 we only keep one step of the architecture and remove all the others while in U-Net 4 we keep the entire architecture.

- U-Net 4: This network has 31,031,810 number of parameters in it making it quite heavy in terms of computation power. For training this network a train-test split was kept as 90-10%. The test loss follows train loss closely signifying no over fitting and the test dice curves show that the model is generalizing well on the test set. There is minor to negligible fluctuations throughout the training process. The model converges therefore the chosen hyper parameters combination prove to be good. The highest test dice attained during training was 0.834. So, on an average over the entire data set, there is

approximately 83.4% overlap between the original mask and the predicted mask. The achieved test dice value falls in the "good score" range. The training graph can is given in the Figure 2.10.

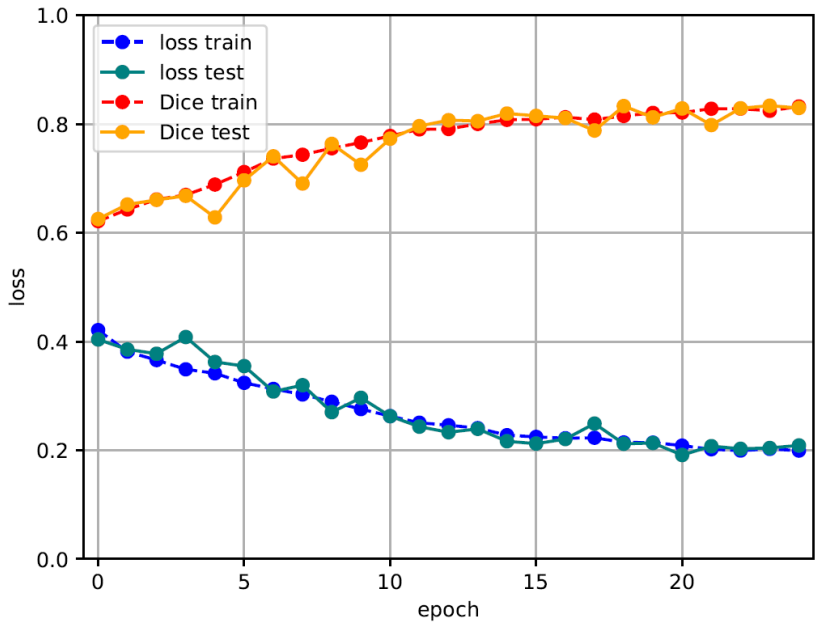

Figure 2.10: U-Net 4 Training Results

- U-Net 1: This network has 403,842 number of parameters and is the smallest out of all the U-Net variations. The effect of decrease in parameters is most visible in this network and it can be seen flattening out earliest in about 5 to 6 epochs, after which small change happens in the learning process. For training this network a train-test split of 60-40% was chosen. The test loss follows the train loss well which shows that there is no over fitting in the model and train and test dice index curves show that the model is generalizing well. There was almost no fluctuation during the training process which is a good indicator. The highest testing dice attained during training is approximately 0.72. So, on an average over the entire data set, there is approximately 72% overlap between the original mask and the predicted mask. There is approximately a 13% decrease in the

highest test dice achieved compared to the 4 step U-net. The training graph is given in Figure 2.11.

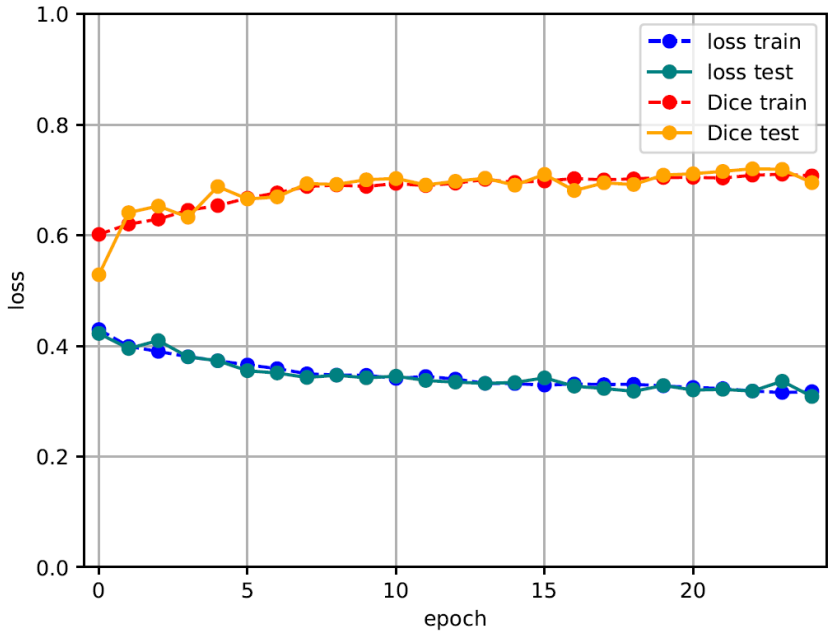

Figure 2.11: U-Net 1 Training Results

2.4 Performance vs. Number of Parameters

Parameters in a neural network architecture are the elements which are learn able. These are in general referred to the weights and biases that are learnt during the learning process. These contribute to the model's power of prediction and are altered during the back-propagation process. Although it seems logical that the more complex the architecture the results increase proportionally, the experimental results show a bit of a different story. A relationship between the highest test dice achieved during training vs the number of parameters in the architecture was conducted to answer this question. According to the computational results, as the number of parameters increases there is indeed a rise in the test dice index, but the proportion is marginal only up to a certain extent post which the

increment starts to flatten out. This shows that indeed a bigger model does produce better results, but the effect lasts only till a threshold value after which the increase in dice with increase in number of parameters is a lot slower.

Furthermore, if we look at the first two U-net models, i.e., U-Net 4 (Original) and U-Net 3. The difference in the highest dice achieved was a 3.4% even though the smaller model had almost 75% of the number of parameters removed. As the number of parameters increase so does the computational cost, it would be wise to select a model which is optimal in terms of both accuracy and cost. The performance comparison of the parameters can be found in the figure 2.12.

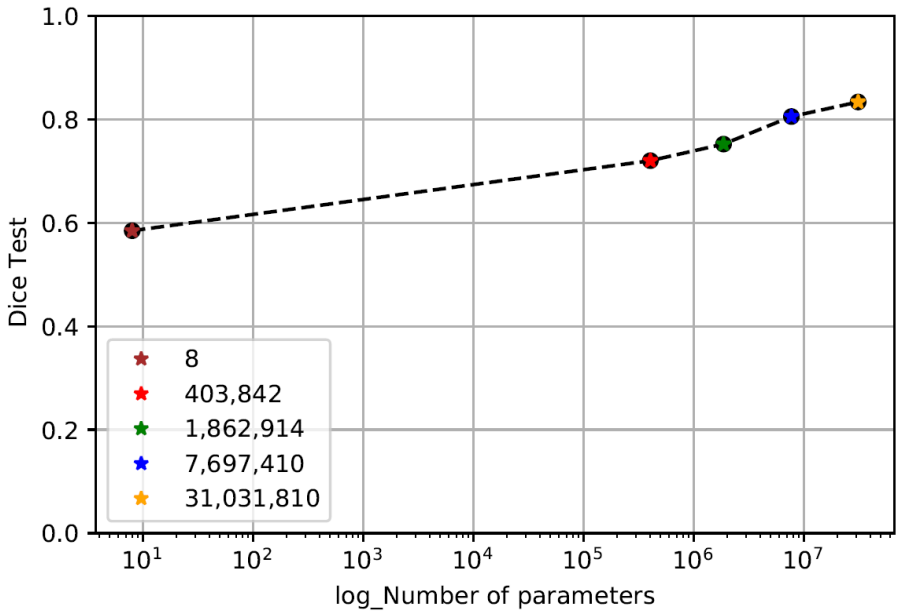

Figure 2.12: Test Dice vs. Number of Parameters(Log Scale)

2.5 Computational Time

Knowing the computational complexity of machine learning models is very important, it lets us know how fast or slow will the model perform for an input size. Since computation power is costly these days an efficient model is to be picked up to minimize the overall cost any given system while not compromising the efficiency of the model.

All the models were run on the ISIC-2018 data set which contains 2594 images. This was done to get an estimate of total time required to process the images and make mask predictions. The results are a bit surprising to say the least. Even though there is an increase in the total time taken which was expected, the relative time difference between the models is less and gives us valuable insight.

Time taken by 31,031,810 parameters to process the data is 922 seconds while for 8 parameters, it is 820 seconds. Time taken by the rest of the parameter count falls in between these extremities.

The increase from the lowest to the highest amount of time taken is only about 12.43% which is inconsequential. So, when it comes to picking a desired model to perform computations the number of parameters does not seem to impact the time too much. The result graph of computational time vs number of parameter's(log scale) can be found in Figure 2.13.

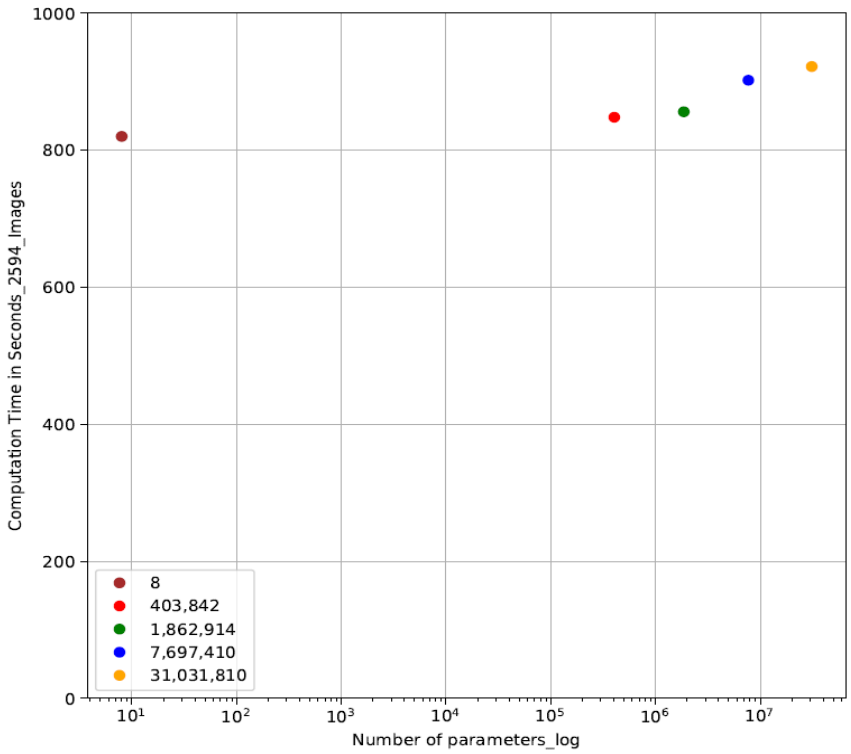

Figure 2.13: Computational Time vs Number of Parameters(Log Scale)

2.6 Density Estimation

After the completion of the training the model weights are saved for future computations. These weights can be loaded on to their respective models and once all the keys successfully match, the model is usable. All the trained models were run on the test set and a dice dictionary was created to keep track of all the dice indexes of the model prediction. This information was then used to mathematically calculate density of points on a 0 to 1 dice index range for analyzing model performance. Bin sizes of 0.1 were created and dice indexes lying in the appropriate range were put into the respective bin. This shows that out of any given amount of test images where does the maximum concentration of

predictions lie. This is a good indicator of model performance. The formula used for computing the density can be found in equation 2.5.

$$\frac{Number\ of\ Points\ in\ the\ bin}{Total\ Number\ of\ Points} \times \frac{1}{Bin\ Size} \tag{2.5}$$

From the results it can be seen that the density does not vary a lot for the One Kernel Model and is approximately constant over the entire range of dice index values with a small increase happening in the later half(Dice>0.5). The results of one kernel shows that is not fit for the task of semantic segmentation as the predictions are not accurate at all for the most part. For U-nets the concentration of points increases as we go up the scale from 0 towards 1. There is a heavy concentration of points in the bins 0.8-0.9 and 0.9-1 whereas bin sizes corresponding to lower dice index(Dice<0.5) have a smaller number of points. The density lines for U-nets increase in sort of an exponential manner which is good indicator of their predictive power.

The graph of the original U-net(4 step) shows that it has approximately the lowest density of points in the dice range 0-0.8, whereas has approximately the highest density of points in the dice range 0.8-1. The original U-net no doubt therefore performs the best here, the only model comparable to it, i.e., 3-step U-net lies very close to it.

All the models are compared in Figure 2.14.

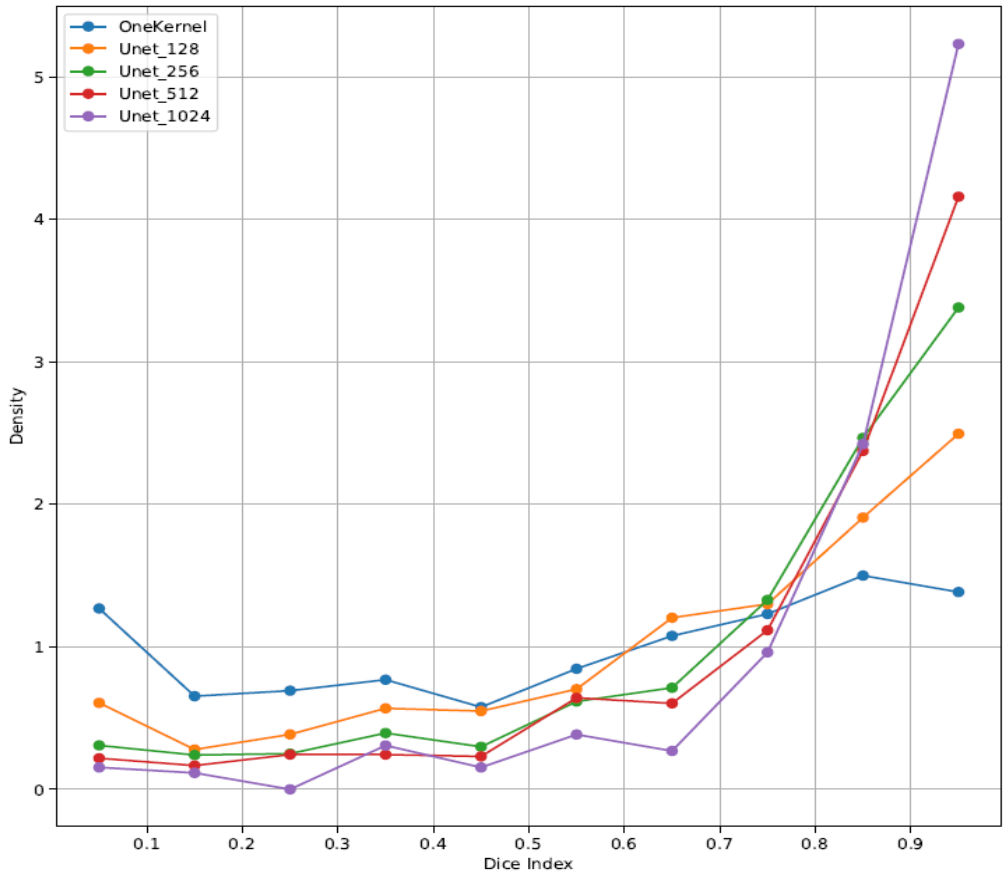

Figure 2.14: Density vs Dice Index

After evaluating the results, we decided to go with U-Net 4 for integrating into the classification architecture for classifying images into Nevus and Melanoma. A table comparing different model stats can be found in table 2.1.

Parameter	One Neuron Model	U-Net 1	U-Net 2	U-Net 3	U-Net 4
Highest Test Dice Index	0.58	0.72	0.75	0.80	0.83
Computational Time(seconds)	820	848	856	902	922
Density(>90%)	1.11	2.44	3.54	4.08	5.29

Table 2.1: Model Stats

CHAPTER 3

CLASSIFICATION

Classification involves predicting which class an item belongs to. Some classifiers are binary, which result in a yes or no decision and Others multi-class, which are able to categorize an item into one out of the several categories. Classification is a very common use case of machine learning and various classification algorithms are used to solve problems like email spam filtering, document categorization, speech recognition, image recognition, and handwriting recognition. A neural network is a type of a machine learning algorithms that can help solve classification problems with high efficiency. Its unique strength is its ability to dynamically create complex prediction functions, and emulate human thinking, in a way that no other algorithm can. There are many classification problems for which neural networks have yielded the best results. Manual classification is subjective and greatly depends on the person accessing the situation thus making it inconsistent in many conditions. Therefore, a computer aided technology is required to help the dermatologists perform the diagnosis. Research has indicated that classifiers based on convolutions neural network can classify skin cancer images at an accuracy equivalent to dermatologists which enables quick, accurate and lifesaving predictions. There is an ever growing need to improve the classification process and the work carried out in this part of the thesis aims to do the same with the help of techniques discussed in this chapter. For this purpose, a combination of ISIC-2019 and 2020 data set was used. It contains 57224 images, with melanoma having 52302 and Nevus having 4922 number of samples.

3.1 Pre-Processing, Data-Augmentation & Unbalanced Dataset

This follows the same procedure from the segmentation network in which we normalize the input data before training. The mean r-g-b and standard deviation values for the used dataset was estimated using these values the data was normalized.

- Mean r-g-b value: (0.74694,0.58144,0.56228)
- Standard deviation: (0.15022,0.13995,0.15327)

For training the classification network data-augmentation techniques were used and are listed below. Many of them are the same from the segmentation network and were already discussed earlier.

- Horizontal Flipping: Columns of the input image is reversed.
- Rotation: Image is rotated by an angle between -180º and 180.
- Resized crop: Resized crop augmentation is when a random subset is created from the original image and scaled back to a given size.
- Color Jitter: This is the type of augmentation in which rather than the location of pixels the brightness, contrast and saturation of an image is changed randomly.

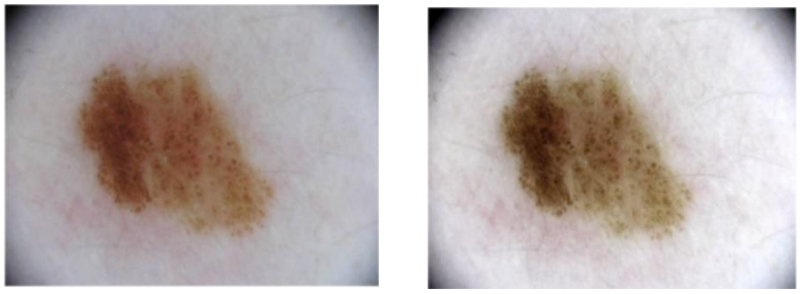

Figure 3.1 : Color Jitter Augmentation

Looking at the dataset for classification there are a total of 57224 Images. Out of the entire 57224 samples, 52302 are of Nevus while just 4922 are of Melanoma. There is a big

class imbalance here. To counter this problem a weighted loss function is used. Weights here are referred to as class weights. The ratio of number of samples of the class to the total amount of data in the entire dataset is calculated.

$$Class\ Nevus: \frac{4922}{57224} = 0.08601 \qquad Class\ Melanoma: \frac{52302}{57224} = 0.91399$$

The weights assigned are flipped, meaning Nevus gets the score of Melanoma while Melanoma gets the score of Nevus. This is done so that the model tunes itself slowly with respect to Nevus but for Melanoma it tunes faster(since the number of samples is small).

3.2 Deep Learning Architecture for Classification

Google in 2019 released a neural network architecture called Efficient net[28] thus making a new addition to the family of convolutions neural networks. This network proved to be very valuable since it provided better accuracy and also improved the efficiency of the model by reducing the number of parameters and floating-point operations per second by a large margin when compared to state-of-the-art models. Models with high accuracy while being efficient are of importance and hence was picked for the carrying out our study. While there are a range of B0-B7 models in the efficient-net family, we decided to go with pre-trained B1 model for conducting study, but the weights were un-freezed during training, this ensures the training happening on top of already trained model stays relevant to our problem domain. This is also called as transfer-learning. An illustration can be found in figure 3.2.

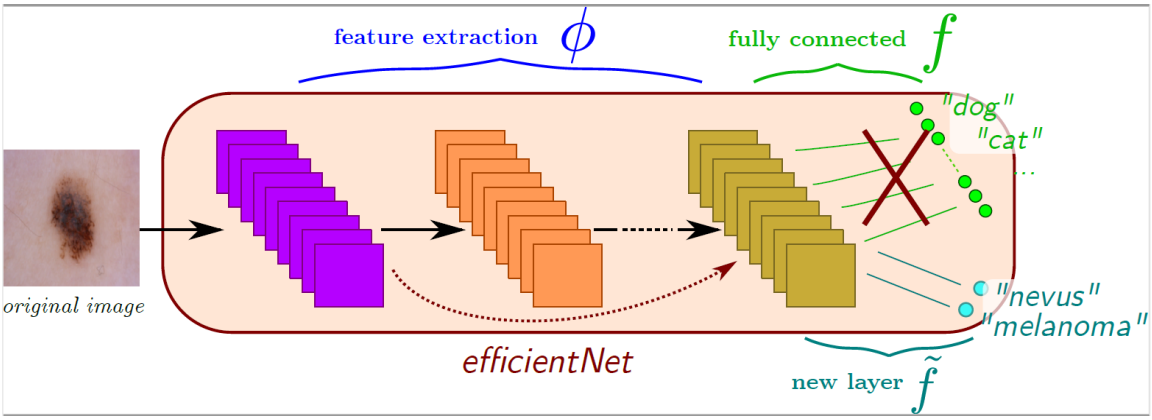

Figure 3.2: Transfer Learning Illustration

The main idea that sets the Efficient net apart from other classification network is the method of compound scaling. Compound scaling takes leverage of Depth(Number of Layers), Resolution(Size) and Width(Number of channels) of the network. It optimizes the value of these three parameters in order to gain the maximum accuracy of the network.

The architecture uses a concept of MBConv which is similar to the concept of inverted residual blocks. This is a type of a residual block for image classification that uses an inverted structure for efficiency reasons. Inverted residual blocks follow the narrow-wide-narrow approach and thus inversion happens. A 1x1 convolution filter first widens the input, then a 3x3 depth wise convolution and then a1x1 convolution to reduce the number of channels so that the input and the output can be concatenated. This is done because of the hypothesis that spatial and depth-wise information can be decoupled. There are numerical results to prove this theory and lately a lot of the architectures seem to use this technique, for example- Mobile-Net, Xception.

Efficient-Net B1 is a variation of the baseline efficient model B0 and was scaled in depth. There were in total of 23 MBConv blocks used in it. The resolution of input image is fixed at 224x224 and images are resized to that dimension before passing it as input. Though the efficient nets perform well on Image-Net data, they seem to transfer well to other data sets as well, which is precisely what has been done in this work as well. This deems this architecture useful as it can be used in a variety of tasks and can possibly be the backbone of many computer vision tasks in the future. It is an open sourced and can be accessed easily, which benefits the machine learning community. Architecture of the efficient net baseline can be found in figure 3.3.

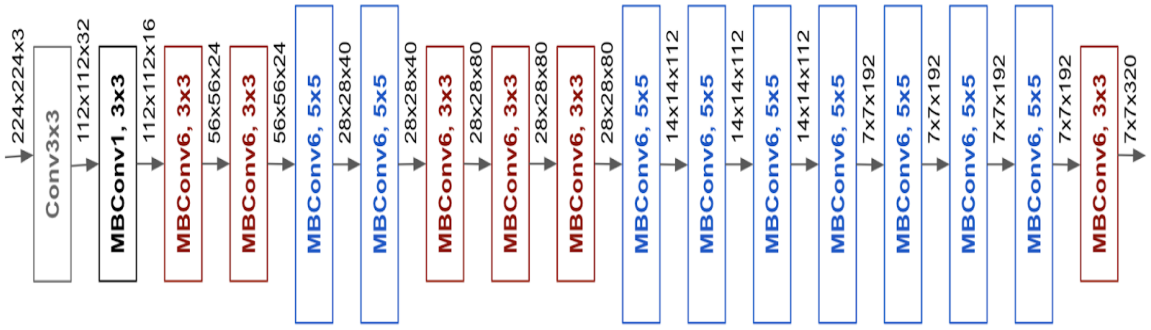

Figure 3.3: Efficient Net Baseline Model

3.3 Specific Architecture

While building machine learning models, there is a type of data called as meta data[29] which some-times goes overlooked. It does carry potential and carries additional information such as locations, data, quality, usage information, characteristics of the data etc, and can be leveraged to increase the overall efficiency of the model. It can also be thought of as data about data. Metadata enables the understanding of the origin of the data which can be crucial and help the model learn better. The meta features used for the thesis contains the information: Age, Gender, Location of the mole. All the information is combined to makes a 11 features vector used for training. Medical science also considers these factors in making informed decisions therefore these were included to make the prediction of the model better. The fully connected layer of the original efficient-net B1 takes input from the convolution layers with 7x7x1280 number of input neurons and outputs contains of 1000 neurons which predict score of the 1000 classes, respectively. This layer is removed, and a custom linear layer is created. This layer takes the 7x7x1280 activation output from the CNN layers of efficient-net and an additional 11 meta features. Output is a 2-class score of the image being a Nevus or Melanoma. But

the meta features are not directly used in the linear layer. These are first passed through another trainable linear layers which takes in the 11 meta features and outputs the same number of features back. A visual representation of CNN feature and meta-data concatenation and classification layer can be found in figure 3.4.

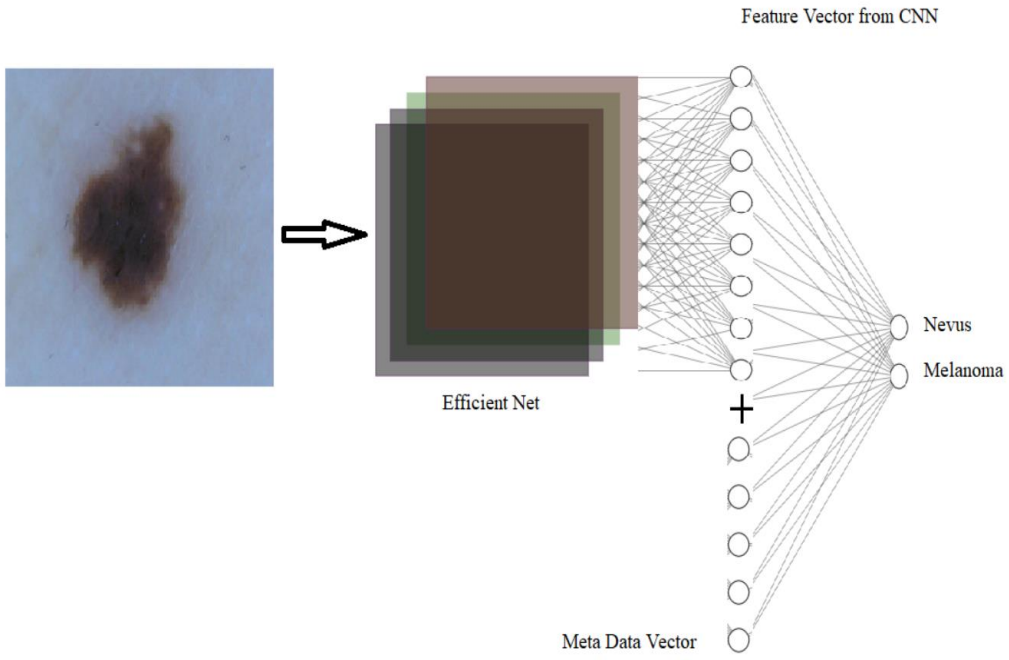

Figure 3.4: Example of CNN-Meta Data Fusion

3.4 Classification Results

There are two types of accuracy implemented to calculate how good or bad the model is doing. First one is accuracy and the other balanced accuracy, both of which are calculated from the confusion matrix. A confusion matrix, also known as an error matrix, is a table layout that helps to visualize the performance of an algorithm. The abbreviations in the confusion matrix will be discussed further in the chapter. An example of binary confusion matrix can be found in figure 3.5.

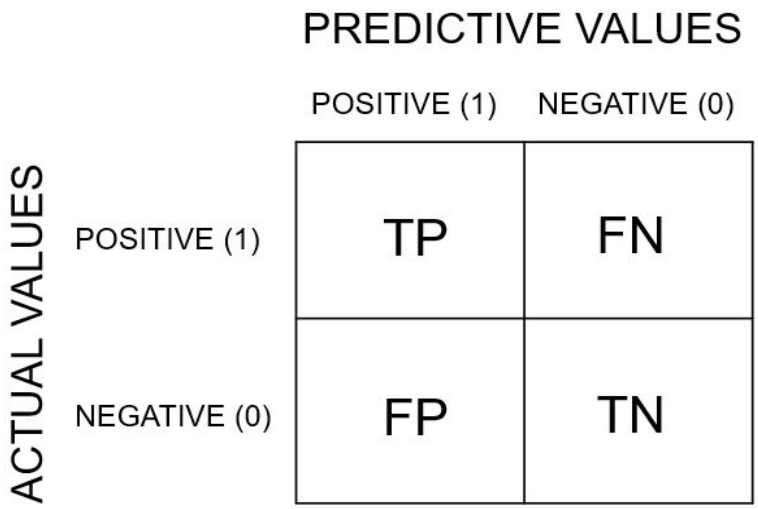

Figure 3.5: Binary Confusion Matrix

Formula for accuracy and balanced accuracy can be found in equation 3.1.

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

Formula for balanced accuracy can be found in equation 3.2.

$$\frac{\frac{TP}{TP+FN}+\frac{TN}{FP+TN}}{2} \tag{3.2}$$

Learning for CNN parameters, meta linear layer parameters and the final linear layer happens separately. CNN network learns with a rate of 10^{-4} while meta linear layer and the final linear layer learn with a rate of 10^{-3}. Batch size of 32 was used for training and weighted Adam was used as an optimizer. The model was able to achieve a test balanced accuracy of about 86%. There is not a lot of difference in the test and train accuracy curves meaning there is very less over fitting and the model is able to generalize well. The loss value decays steadily without many fluctuations. The training results of Efficient-Net B1 can be found in figure 3.6.

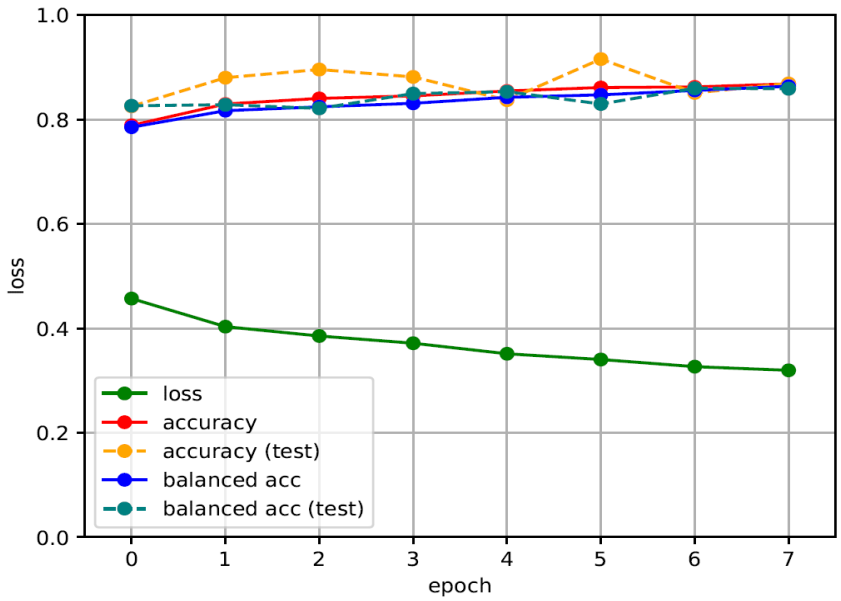

Figure 3.6: Efficient Net B1 Training Results

In machine learning it is essential to compare performance of models. When it comes to multi-class classification problems the method of AUC-ROC curve is used. AUC stands for 'Area Under the Curve' and ROC stands for 'Receiver Operating Characteristics'. ROC can be thought of as a probability curve while AUC represents the measure of separability. This tells how much capable the model is in distinguishing between classes. So, higher the AUC the better the model is a predicting 0s as 0s and 1s as 1s. Therefore, in our case it will be a measure of how good the model is at distinguishing between nevus and melanoma. Values of AUC range from 0 to 1. 1 is the best where the model exactly knows the difference between the classes while 0.5 means the model is making random predictions.

The curve is plotted with the true positive rate(TPR) on the y-axis and false positive rate(FPR)on the x-axis, the area under the ROC curve is the area under the curve and we

are concerned with that exact value. Before discussing how TPR and FPR are calculated we will first look at 4 terms used in machine learning in a binary test.

- True Positive: Is an outcome where the model correctly predicted the positive class. The model predicts the mole as being a nevus when it is indeed nevus.
- False Positive: Is an outcome where the model incorrectly predicted the positive class. The model predicts the mole as being a melanoma when it is not a melanoma but a nevus.
- True Negative: Is an outcome where the model correctly predicted the negative class. The model predicts the mole as a melanoma when it is indeed melanoma.
- False Negative: Is an outcome where the model incorrectly predicted the negative class. The model predicts the mole as nevus when it is melanoma. This is the most dangerous type of error when it comes to situations such as cancer detection. Wrong prediction can result in life threatening situations.

The true positive rate formula can be found in equation 3.3.

$$\frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3.3}$$

The false positive rate formula can be found in equation 3.4.

$$\frac{False\ Positive}{False\ Positive + True\ Negative} \tag{3.4}$$

The AUC graph from the Efficient-b1 model training and the test set confusion matrix can be found in figure 3.7. For the class Melanoma the area under the curve is 0.94, which means that the model has a 94% chance of distinguishing between the positive and negative class. A confusion matrix is basically plotting True positives(Top Left),

False Positive(Top Right), True Negative(Bottom Right)and False Negative(Bottom Left) in the same matrix.

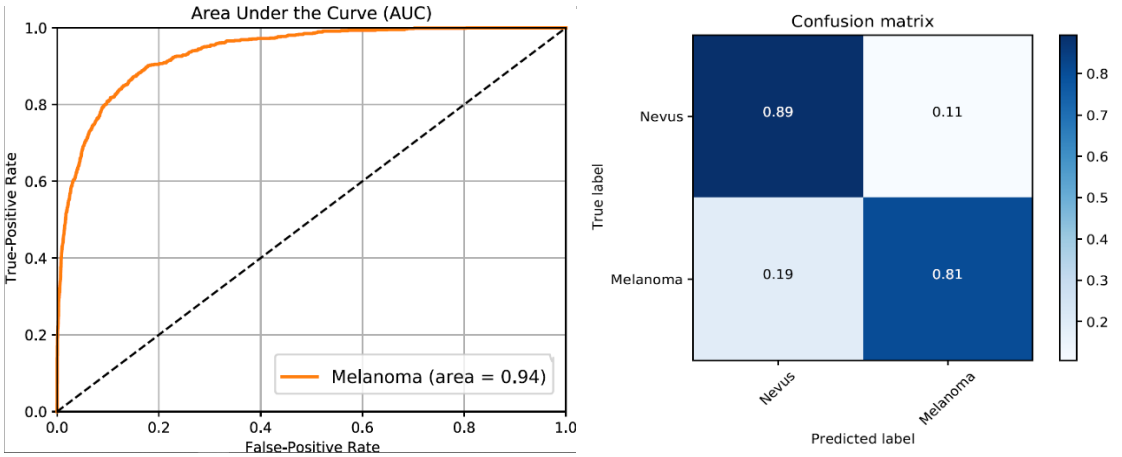

Figure 3.7: Efficient-Net B1 AUC-ROC Curve and Test Confusion Matrix

3.5 Enhance Classification Results

Increasing the accuracy of model's prediction is critical and is always a work in process. Below techniques were tried and experimented with to increase the prediction accuracy.

- Black Background: The 2-channel output score from U-Net is converted into a binary mask using an Argmax function which gives the position of the higher score output, i.e., returns the true class of the label. This binary mask is concatenated three times in tandem to form a 3-channel mask stack. Now, a channel wise multiplication between the image and the mask stack is performed. This is segmenting the image with respect to the mask values, since only the pixels predicted by U-net retain their original values and other pixels get zeroed out.

Initially while training this model a good loss function convergence was not observed. To tackle this problem a learning rate decay was used which decays the learning rate by a factor of 1.25 if the current loss values is more than 0.99% of the last loss value. An example of the image and the image with blackened background can be found in figure 3.8.

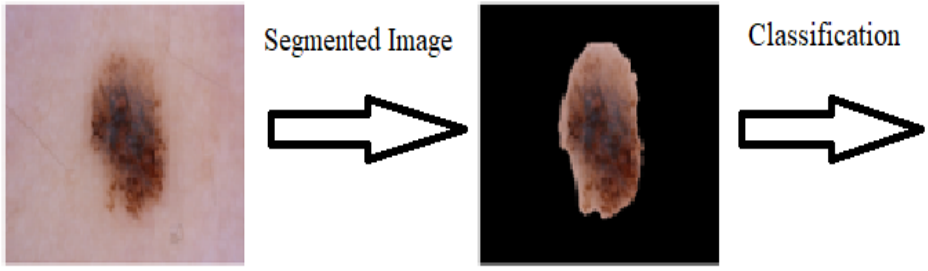

Figure 3.8: Original Image and Blackened Background Image

Experimental results show that the method is not effective. The training loss does not change much signifying that the model learns very less, and the balanced test accuracy is lower as compared to the results of the original architecture. The model could achieve an AUC of 0.92 and a balanced test accuracy of about 83%. The training results can be found in figure 3.9.

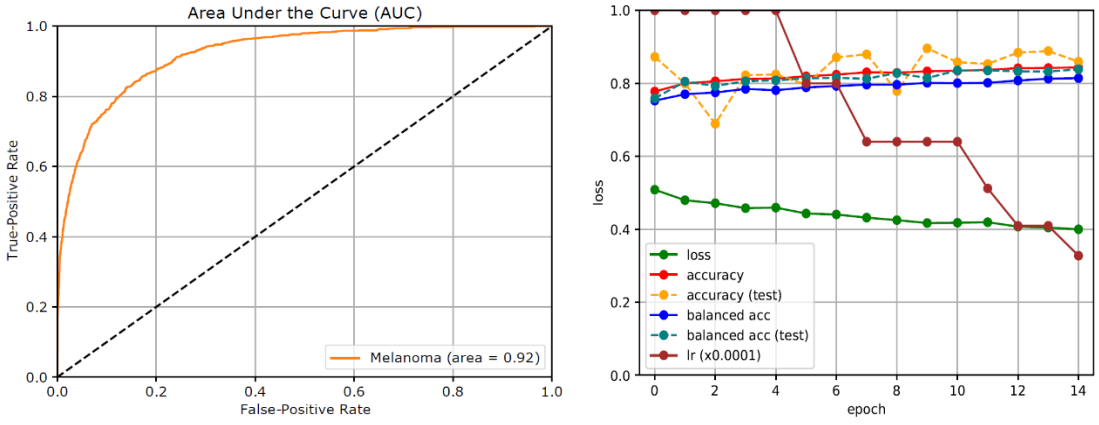

Figure 3.9: Black Background Image Results

- Bounding Box Approach: This method focuses on cropping out exactly the region of mole from the input image and use that as classification. The binary mask generated from the U-net score is passed through an algorithm which calculates the co-ordinates of the mole. These coordinates are used to cut the original image and resize it back to the given input size. What this essentially does is zooms in around the mole and reduces any skin area sent in classification. An important thing to note here is that zooming in the image has chances of distorting the original image based on the size of the mole. Experimental results show that this did not yield any significant results. Learning rate decay was used in this method as well since the loss was not converging. The model could achieve an AUC of 0.91 and a balanced accuracy of about 82%. The input image, cropped image and training results can be found in figure 3.10 and 3.11.

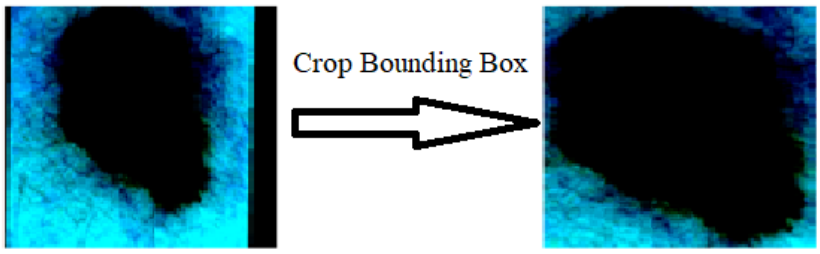

Figure 3.10: Original Image and Image Cropped with Bounding Box Coordinates

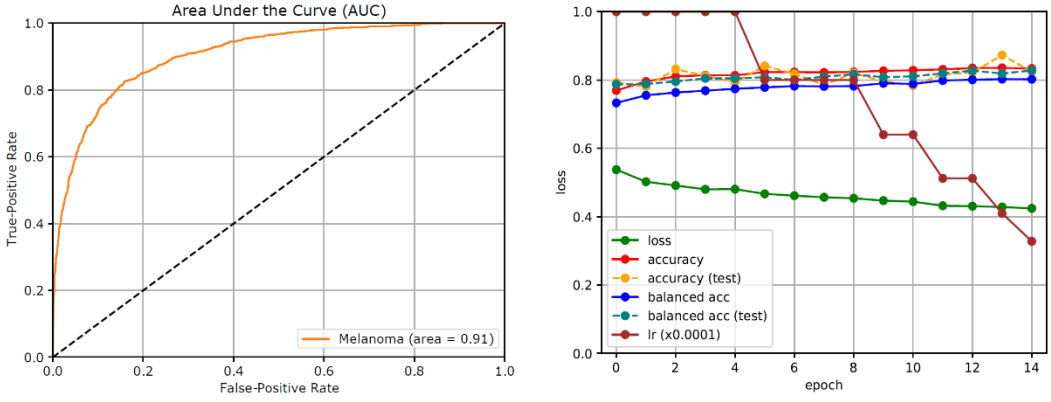

Figure 3.11: Bounding Box Results

- Integrate mole probability in Image: In this method the 2-channel score output from U-net is passed through a soft-max layer which gives the probability of the pixels being a mole or not a mole on the 2-channel output. We pick the feature map with the pixel probability of it being a class 0 or mole and concatenate that with the input RGB image. Now, our input consists of a 4 channel(red-green-blue-mole probability feature map). But the Efficient-net convolution filters only contain three channels and would not allow a 4-channel input. So, an additional channel is included in the first convolution block layer and the weights in the filters are initialed with Xavier random weights[30]. The goal of Xavier initialization is to initialize the weights in such a way that the across every layer the variance of the activation's remains the same. This helps the gradient from exploding/vanishing. Since other filters in the pre-existing CNN are already optimized, we might want to be careful in initializing the 4th channel's weight. The results of this method are interesting. The modified model was able to achieve an AUC of 0.95 and balanced accuracy of about 87%. The results can be found in figure 3.12.

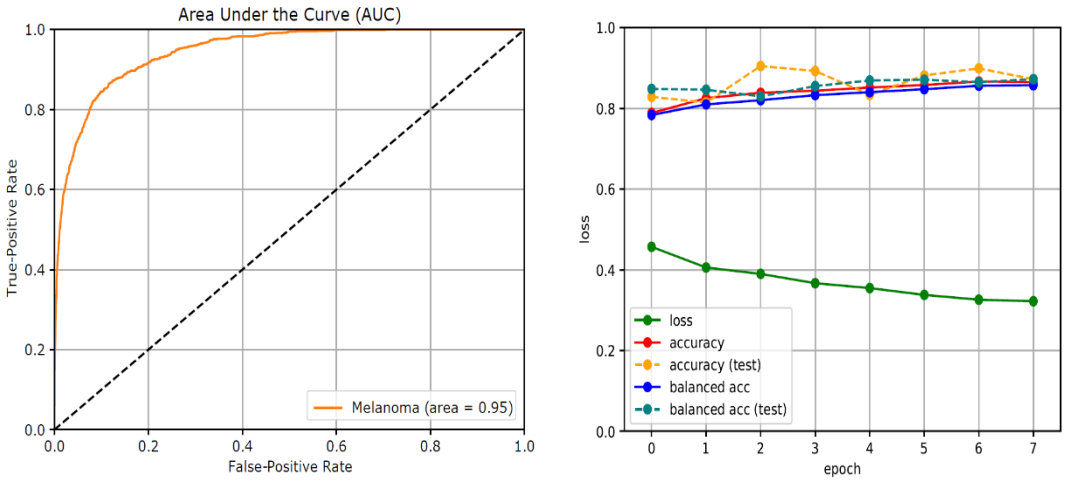

Figure 3.12: Integrate Mole Probability Results

While on the AUC front there does not seem to be many changes when compared to the original Efficient-Net, when we look at the false-negative in the test confusion matrix, much improvement can be seen. Both, the original architecture, and the modified architecture(4 channel input model) were run for 8 runs and in all the runs the modified model has a smaller number of false negatives prediction than the original architecture. Comparison results between the modified model(4 channel input) and the original architecture depicting the results of all the 8 runs can be found in Table 3.1.

Trial	Original Architecture	Architecture with Mole Probability
1	0.16	0.12
2	0.16	0.14
3	0.15	0.10
4	0.15	0.13
5	0.23	0.10
6	0.17	0.08
7	0.19	0.13
8	0.26	0.10
Average	0.1837	0.1125

Table 3.1: Test Set False Negative Comparison

After the induction of mole probability as the 4^{th} channel in the image, we see that the neural network is on an average(in the 8 runs) making 38.7% less misclassifications of melanoma.

CHAPTER 4

FUTURE WORK

Future work of this thesis is:

- Implementing different architectures: For example: Mask RCNN is a state-of-the-art model when it comes to detection and classification. It is very accurate and effective when it comes to task of object detection and classification. One of the main advantages of this model is that it is an end-to-end model, which means that it has detection and classification units built into it by design and it takes on a bounding box approach for detection and not a pixel wise approach.
- Real time implementation: The end goal is to create a complete software application which can be used by people for keeping track of any spots they think are suspicious. People can create their accounts on the application, click and upload pictures on the application which will then be processed by our Deep learning framework and return results in a matter of minutes. This would also serve as a repository to keep track of the evolution of the mole over time which also helps a lot in assessing to get it checked or not, since melanoma changes shape at a fast rate.

REFERENCES

[1] Early detection and treatment of skin cancer. American Family Physician, 62(2),July 2000.

[2] Malvehy, J., et al. "Clinical performance of the Nevisense system in cutaneous melanoma detection: an international, multicentre, prospective and blinded clinical trial on efficacy and safety." British Journal of Dermatology 171.5 (2014): 1099-1107.

[3] A. Bhattacharya, A. Young, A. Wong, S. Stalling, M. Wei, and D. Hadley. Precision Diagnosis Of Melanoma And Other Skin Lesions From Digital Images.

[4] M. A. Marchetti, M. Fonseca, S. W. Dusza, A. Scope, A. C. Geller, M. Bishop, A. A. Marghoob, S. A. Oliveria, and A. C. Halpern. Dermatoscopic imaging of skin lesions by high school students: a cross-sectional pilot study. Dermatol Pract Concept, 5(1):11–28, Jan 2015.

[5] Ivan Bristow and Jonathan Bowling. Dermoscopy as a technique for the early identification of foot melanoma. Journal of foot and ankle research, 2:14, 06 2009.

[6] A. K. Gupta, M. Bharadwaj, and R. Mehrotra. Skin Cancer Concerns in People of Color: Risk Factors and Prevention. Asian Pac J Cancer Prev, 17(12):5257–5264, 12 2016.

[7] M. Emre Celebi, Quan Wen, Hitoshi Iyatomi, Kouhei Shimizu, Huiyu Zhou, and Gerald Schaefer. Astate-of-the-art survey on lesion border detection in dermoscopy images. pages 97–129, 09 2015.

[8] S. R D and S. A. Deep Learning Based Skin Lesion Segmentation and Classification of Melanoma Using Support Vector Machine (SVM).Asian Pac J Cancer Prev, 20(5):1555–1561, May 2019.

[9] S. N. Hasan, M. Gezer, R. A. Azeez, and S. G¨ulse¸cen. Skin lesion segmentation by using deep learning techniques. pages 1–4, 2019.

[10] K. Zafar, S. O. Gilani, A. Waris, A. Ahmed, M. Jamil, M. N. Khan, and A. Sohail Kashif. Skin Lesion Segmentation from Dermoscopic Images Using Convolutional Neural Network. Sensors (Basel),20(6), Mar 2020.

[11] H. M. ?nver and E. Ayan. Skin Lesion Segmentation in Dermoscopic Images with Combination of YOLO and Grab Cut Algorithm. Diagnostics (Basel), 9(3), Jul 2019.

[12] Saban¨Ozt¨urk and Umut¨Ozkaya. Skin lesion segmentation with improved convolutional neural network.J. Digit. Imaging, 33(4):958–970, 2020.

[13] Oludayo Olugbara, Tunmike Taiwo, and Delene Heukelman. Segmentation of melanoma skin lesion using perceptual color difference saliency with morphological analysis. Mathematical Problems in Engineering, 2018:1–19, 02 2018.

[14] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. A survey of transfer learning. Journal of BigData, 3, 05 2016.

[15] Aisyah Larasati, Apif M Hajji, and Anik Dwiastuti. The relationship between data skewness and accuracy of artificial neural network predictive model.IOP Conference Series: Materials Science and Engineering, 523:012070, 07 2019.

[16] Xue Ying. An overview of overfitting and its solutions. Journal of Physics: Conference Series,1168:022022, 02 2019.

[17] Connor Shorten and Taghi Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of Big Data, 6, 07 2019.

[18] Kelly Zou, Simon Warfield, Aditya Bharatha, Clare Tempany, Michael Kaus, Steven Haker, William Wells, Ferenc Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. Academic radiology, 11:178–89, 02 2004.

[19] Sarat Kumar Sarvepalli. Deep learning in neural networks: The science behind an artificial brain. 102015.27

[20] Yun Xu and Royston Goodacre. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. Journal of Analysis and Testing, 2, 10 2018.

[21] Pavlo Radiuk. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Information Technology and Management Science, 20:20–24, 12 2017.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

[23] Intisar Rizwan I Haque and Jeremiah Neubert. Deep learning approaches to biomedical image segmentation. Informatics in Medicine Unlocked, 18:100297, 2020.

[24] X. Lu, H. Huo, T. Fang, and H. Zhang. Learning deconvolutional network for object tracking. IEEE Access, 6:18032–18041, 2018.

[25] Hyeongyeom Ahn and Changhoon Yim. Convolutional neural networks using skip connections with layer groups for super-resolution image reconstruction based on deep learning. Applied Sciences,10:1959, 03 2020.

[26] B. Ding, H. Qian, and J. Zhou. Activation functions and their characteristics in deep neural networks. pages 1836–1841, 2018.

[27] Chaity Banerjee, Tathagata Mukherjee, and Eduardo Pasiliao. An empirical study on generalizations of the relu activation function. pages 164–167, 04 2019.

[28] Mingxing Tan and Quoc Le. Efficient-Net: Rethinking model scaling for convolutional neural networks.05 2019.

[29] Ciro Castiello, Giovanna Castellano, and Anna Fanelli. Meta-data: Characterization of input features for meta-learning. pages 457–468, 07 2005.

[30] Xavier Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research - Proceedings Track, 9:249–256, 01 2010.