

University of Mississippi

eGrove

---

Honors Theses

Honors College (Sally McDonnell Barksdale  
Honors College)

---

Spring 5-1-2021

## Using Deep Learning to Automate the Diagnosis of Skin Melanoma

Akhil Reddy Alasandagutti

Follow this and additional works at: [https://egrove.olemiss.edu/hon\\_thesis](https://egrove.olemiss.edu/hon_thesis)



Part of the [Artificial Intelligence and Robotics Commons](#), [Data Science Commons](#), and the [Skin and Connective Tissue Diseases Commons](#)

---

### Recommended Citation

Alasandagutti, Akhil Reddy, "Using Deep Learning to Automate the Diagnosis of Skin Melanoma" (2021). *Honors Theses*. 1928.

[https://egrove.olemiss.edu/hon\\_thesis/1928](https://egrove.olemiss.edu/hon_thesis/1928)

This Undergraduate Thesis is brought to you for free and open access by the Honors College (Sally McDonnell Barksdale Honors College) at eGrove. It has been accepted for inclusion in Honors Theses by an authorized administrator of eGrove. For more information, please contact [egrove@olemiss.edu](mailto:egrove@olemiss.edu).

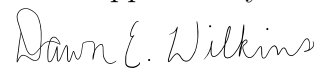
USING DEEP LEARNING TO AUTOMATE THE DIAGNOSIS OF SKIN MELANOMA

by  
Akhil Reddy Alasandagutti

A thesis submitted to the faculty of The University of Mississippi in partial fulfillment of  
the requirements of the Sally McDonnell Barksdale Honors College.

Oxford  
May 2021

Approved by



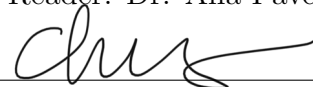
---

Advisor: Dr. Dawn Wilkins



---

Reader: Dr. Ana Pavel



---

Reader: Dr. Yixin Chen

Copyright Akhil Reddy Alasandagutti 2021  
ALL RIGHTS RESERVED

## ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Dawn Wilkins for mentoring and guiding me throughout this project. I would also like to thank Dr. Ana Pavel for providing additional insight and guidance. I could not have gotten this far without them.

# ABSTRACT

**Akhil Reddy Alasandagutti**

Using Deep Learning To Automate The Diagnosis Of Skin Melanoma

(Under the direction of Dr. Dawn Wilkins)

Machine learning and image processing techniques have been widely implemented in the field of medicine to help accurately diagnose a multitude of medical conditions. The automated diagnosis of skin melanoma is one such instance. However, a majority of the successful machine learning models that have been implemented in the past have used deep learning approaches where only raw image data has been utilized to train machine learning models, such as neural networks. While they have been quite effective at predicting the condition of these lesions, they lack key information about the images, such as clinical data, and features that medical professionals consistently rely on for diagnosis. This research project will explore methods to enhance machine learning models with three drastically different skin melanoma datasets, each with their own set of unique challenges. Various preprocessing techniques, machine learning models, and feature extraction methods will be compared to determine the most optimal approach for each dataset. In addition, time and space complexities of the approaches will also be analyzed in order to minimize resource consumption without causing major performance degradation to the models

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	ii
LIST OF FIGURES . . . . .	iv
INTRODUCTION . . . . .	1
BACKGROUND . . . . .	2
ISIC ARCHIVE DATASET . . . . .	9
HAM10000 DATASET . . . . .	15
SIIM-ISIC MELANOMA 2020 DATASET . . . . .	22
DISCUSSION AND FUTURE WORK . . . . .	31
BIBLIOGRAPHY . . . . .	35

## LIST OF FIGURES

2.1	A Cross-Sectional Diagram of Skin Melanoma . . . . .	3
2.2	Example Decision Tree . . . . .	5
2.3	Example Deep Neural Network with 16 inputs and 1 output . . . . .	6
3.1	Sampled Benign Images . . . . .	9
3.2	Sampled Malignant Images . . . . .	10
3.3	Benign images vs Malignant images . . . . .	11
3.4	Masked benign images vs masked Malignant . . . . .	12
4.1	Sampled Images and Count . . . . .	15
4.2	ABCDE Rule for Early Melanoma Detection . . . . .	18
5.1	Sampled Benign Images . . . . .	23
5.2	Sampled Malignant Images . . . . .	23
5.3	Sample Reshaped Benign Images . . . . .	26
5.4	Sample Reshaped Malignant Images . . . . .	27
5.5	Sample Transformed Malignant Images . . . . .	28
5.6	Sample GAN output . . . . .	30

## CHAPTER 1

### INTRODUCTION

Machine learning and image processing techniques have been widely implemented in the field of medicine to help accurately diagnose a multitude of medical conditions. The automated diagnosis of skin melanoma is one such instance. However, a majority of the successful machine learning models that have been implemented in the past have used deep learning approaches where only raw image data has been utilized to train machine learning models, such as neural networks. While they have been quite effective at predicting the condition of these lesions, they lack key information about the images, such as clinical data, and features that medical professionals consistently rely on for diagnosis.

This research project will explore methods to enhance machine learning models with three drastically different skin melanoma datasets, each with their own set of unique challenges. Various preprocessing techniques, machine learning models, and feature extraction methods will be compared to determine the most optimal approach for each dataset. In addition, time and space complexities of the approaches will also be analyzed in order to minimize resource consumption without causing major performance degradation to the models.



## CHAPTER 2

### BACKGROUND

#### 2.1 SKIN MELANOMA

##### 2.1.1 GENERAL BACKGROUND AND DESCRIPTION

Melanoma is a type of skin cancer that causes pigment producing cells, called melanocytes to mutate and divide uncontrollably (Niederhuber JE, 2019). Melanoma can also manifest itself inside the eyes, and sometimes inside the nose and throat. While being much less common than other types of malignant skin conditions, melanoma still contributes to the most deaths caused by any skin condition. The American Cancer Society estimates that 106,110 new cases of melanoma will be diagnosed in the year 2021, and 7,180 people are expected to die of it (Howlader N, 2019).

##### 2.1.2 DETECTION TECHNIQUES

There are several techniques that can be utilized to diagnose skin melanoma. These techniques are generally divided into two categories - invasive and non-invasive. Invasive techniques are typically performed by extracting a sample of the tissue in question, while non-invasive techniques do not involve any alteration of tissue. Invasive techniques are generally avoided as they can often destroy the lesions and make it impossible to carry out further inspections on it. Furthermore, any error in sampling the tissue could cause the lesion to rupture and spread the melanoma to neighboring cells prematurely (Maarouf M, 2019). Non-invasive, or visual detection techniques are not only safer, but they're also relatively very inexpensive. The ABCDE criteria - which looks at asymmetry, irregular borders, variation

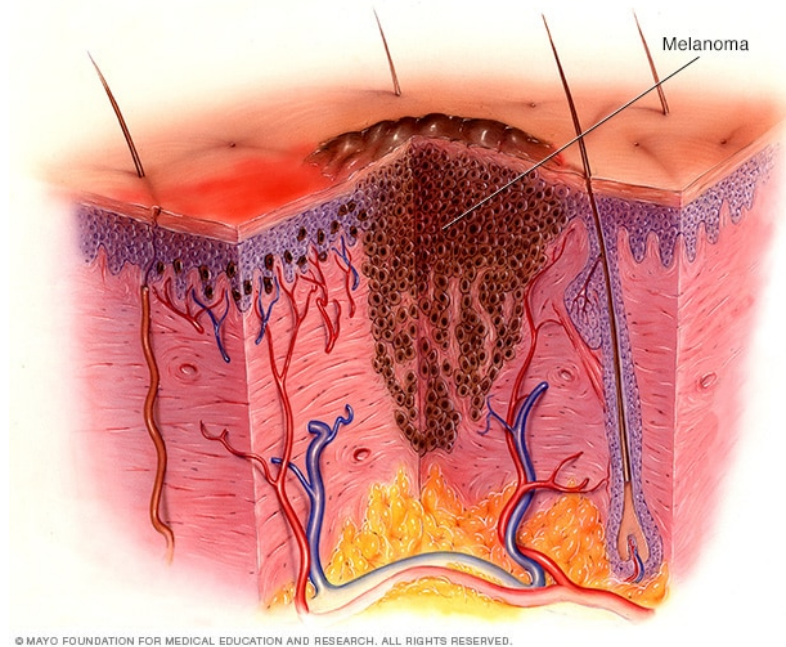


Figure 2.1: A Cross-Sectional Diagram of Skin Melanoma

in color, diameter greater than 6mm. and evolving size has been proven to be very effective at detecting skin melanoma early (Mitchell TC, 2020).

## 2.2 MACHINE LEARNING AND DEEP LEARNING

Machine Learning is a branch of Artificial Intelligence that tries to learn from data by detecting patterns in it (Hao, 2018). This data can be encompassed in any format - numbers, text, images, etc. Machine learning algorithms typically belong to one of three classes - supervised, unsupervised, and reinforcement. Supervised machine learning algorithms learn from past data that has definite outcomes, also known as labelled data, and attempts to learn from it and predict future events. This project utilizes supervised machine learning algorithms to classify skin lesions after being trained using labelled skin lesion data. Unsupervised machine learning models train using unlabelled data, and simply attempt to look for patterns that could help group subsets of this data. Reinforcement learning algorithms use a trial and error method to reach a specific goal. Actions that help the algorithm reach the

objective are rewarded, and the ones that hinder its progress are penalized. Reinforcement learning algorithms are often used to implement autonomous players in games like Chess, Go, and a many other video games. This project utilizes a Generative Adversarial Network, which is a Reinforcement Learning model. Deep Learning algorithms are highly complex machine learning algorithms that can amplify and identify minute details in data that shallow machine learning models cannot identify. These models are also often attributed with neural networks that consist of multiple deep hidden layers.

### 2.2.1 DECISION TREE CLASSIFIERS

A Decision Tree Classifier is a supervised machine learning algorithm that generates a hierarchical structure of questions and their possible answers from a dataset, called a decision tree. This decision tree can then be used to predict the labels from data. Decision Tree Classifiers are typically quick learners and are also readable in most cases (Hao, 2018). Humans can understand its predictions by simply looking at the generated decision tree. However, in cases where the number of attributes in a dataset is large, very complex decision trees are generated which makes it extremely difficult for humans to visualize and understand. Image data is one such example.

Below is an example of a decision tree that classifies data using X and Y coordinates into two possible classes - C1 and C2.

### 2.2.2 NEURAL NETWORKS

#### 2.2.2.1 ARTIFICIAL NEURAL NETWORKS

Neural Networks have been inspired by the human brain. An Artificial Neural Network consists of multiple processing units, called nodes, that are connected to each other through weighted direct links. These nodes correspond to the neurons in a human brain, the links correspond to the connections between the neurons, and the weights of the links correspond to the strength of the connection between neurons. A large number of artificial

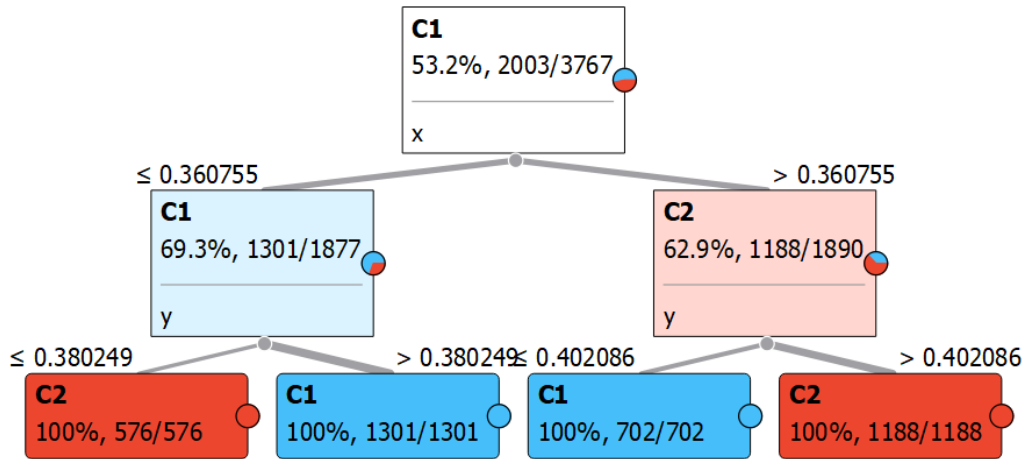


Figure 2.2: Example Decision Tree

neurons connected together as pools of multiple layers form deep neural networks. While deep neural networks perform significantly better than other shallow machine learning models, it is very expensive to train them as they require a significant amount of compute time and memory (Hao, 2018).

### 2.2.2.2 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks, or CNNs are a type of Artificial Neural Networks that contain a convolutional layer. A convolutional layer can take a multidimensional layer as an input, filter it based on the set hyperparameters, and pass it on to the neural network layer. It replicates the response of the visual cortex in the human eye to a stimulus (Hao, 2018). CNNs are very effective in image classification tasks as spatial relations between separate features are registered in convolutional layers, unlike in traditional artificial neural networks.

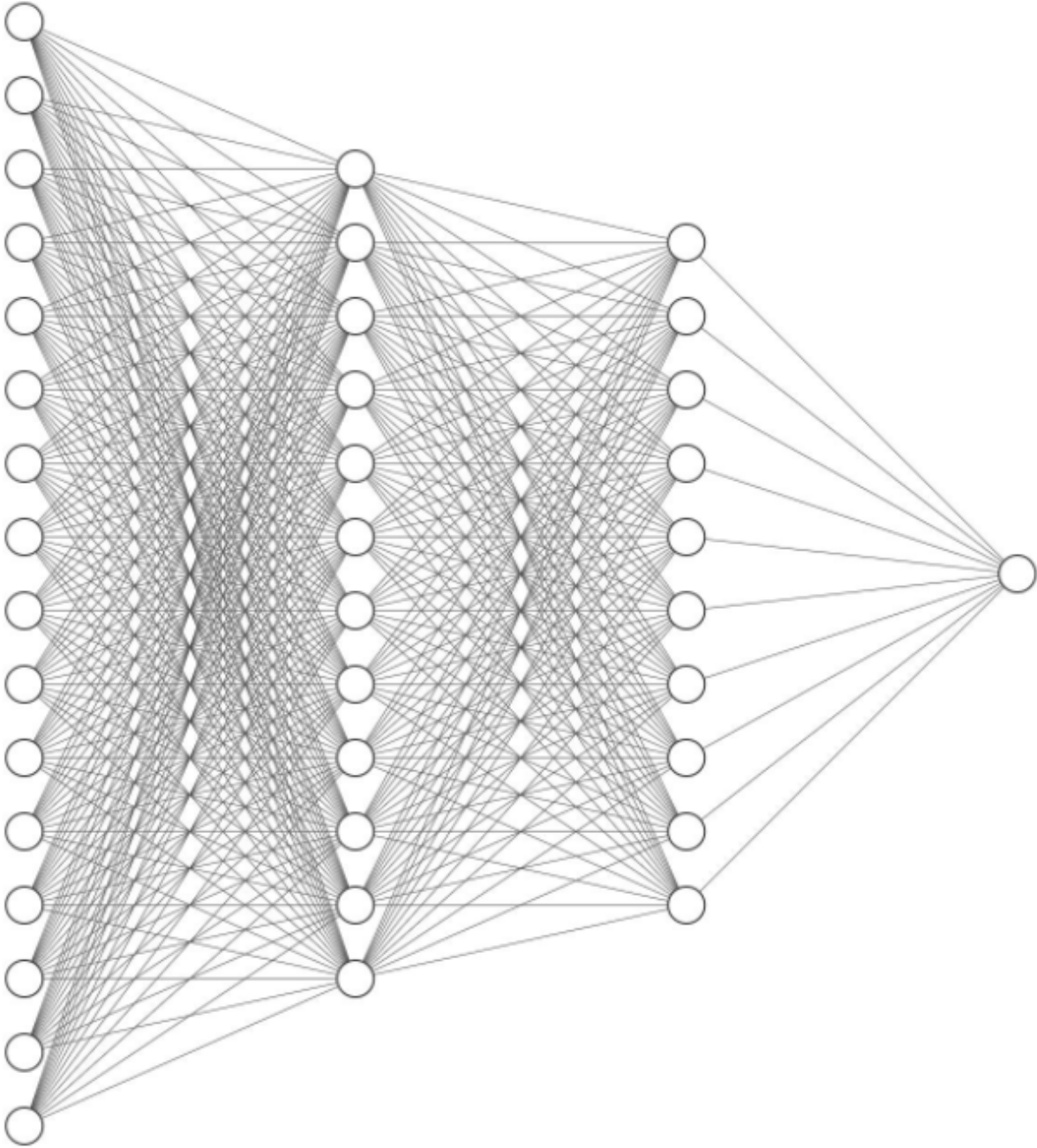


Figure 2.3: Example Deep Neural Network with 16 inputs and 1 output

### 2.2.3 GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks, or GANs are a type of Reinforcement Learning models that can synthetically generate images. A GAN typically consists of two components -

a generator, and a discriminator. The generator attempts to generate an image that is similar to an image in the provided dataset, and the discriminator attempts to distinguish between the original images and the images generated by the generator. The generator is rewarded if it is able to successfully trick the discriminator, and is penalized if it is unsuccessful. The discriminator is rewarded if it is able to successfully distinguish between the synthetic images and original images, and is penalized if it is unsuccessful. These adversarial networks play against each other for multiple generations of training until the generator learns to produce images that are indistinguishable from the original images (Goodfellow, 2014). GANs can be very expensive to train, especially with higher resolution images.

## 2.3 SUPER COMPUTING

Training deep learning models can be time and resource intensive, and it is often not practical to train these models on a personal computer. The Mississippi Center for Supercomputing Research (MCSR) currently has 3 supercomputers, all three of which have been utilized for this project.

### 2.3.1 SEQUOIA

The Sequoia cluster consists of a total of 1304 CPU cores, and has a memory capacity of 35GB per node. This cluster has been utilized to run non-memory intensive tasks, like feature extraction, and training smaller neural networks and other machine learning models.

### 2.3.2 MAPLE

The Maple cluster is a Cray cluster, and it comprises of 3,726 CPU cores along with 29 Nvidia Kepler K20 GPUs for regular GPU intensive tasks, and 4 Nvidia Tesla P100 GPUs for large memory GPU intensive tasks. This cluster has been utilized to train most of the CNN models on a regular GPU queue as neural networks train much more efficiently on GPUs than on CPUs.

### 2.3.3 CATALPA

Catalpa is a single-image shared-memory system, and is only reserved for tasks that require a very large amount of memory. Catalpa has been utilized to train the GANs as neither Maple, nor Sequoia could handle its memory requirements.

## CHAPTER 3

### ISIC ARCHIVE DATASET

#### 3.1 DATASET DESCRIPTION

This dataset contains a total of 3,297 224x224pi images that were extracted from the International Skin Imaging Collaboration Archive (Fanconi, 2019). These images have been labelled as either "benign" or "malignant", and are well balanced - 1,800 images belong to the former class, and 1,497 belong to the latter. These images have further been split into training and testing subsets - with 1,440 benign and 1,197 malignant belonging to the training set, and 360 benign and 300 malignant images belonging to the testing set.

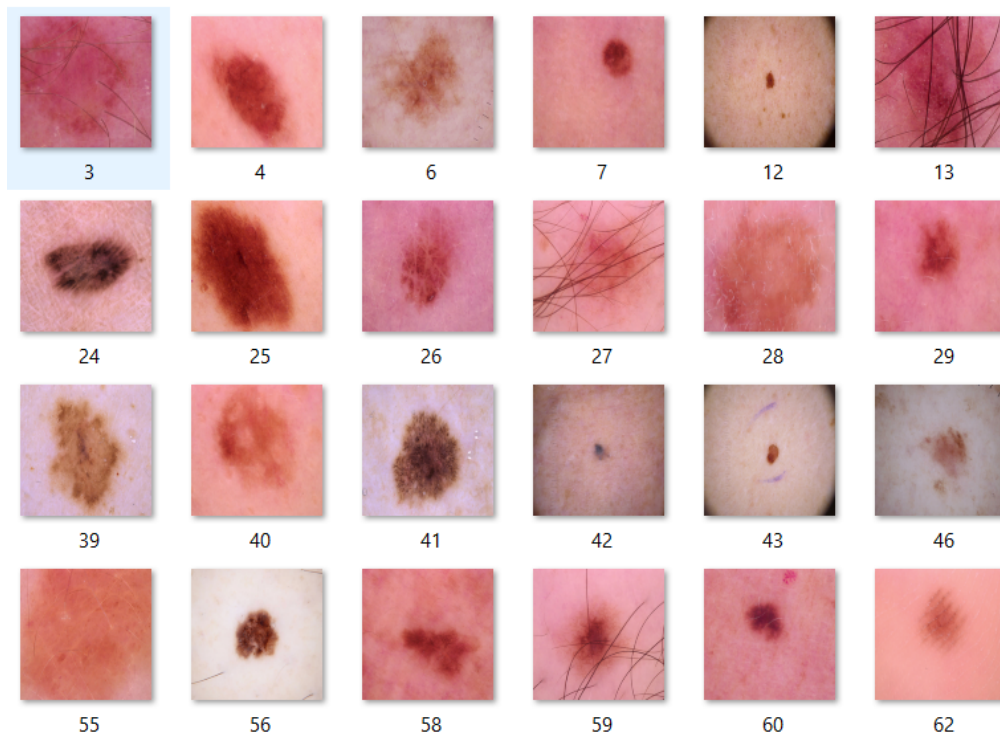


Figure 3.1: Sampled Benign Images



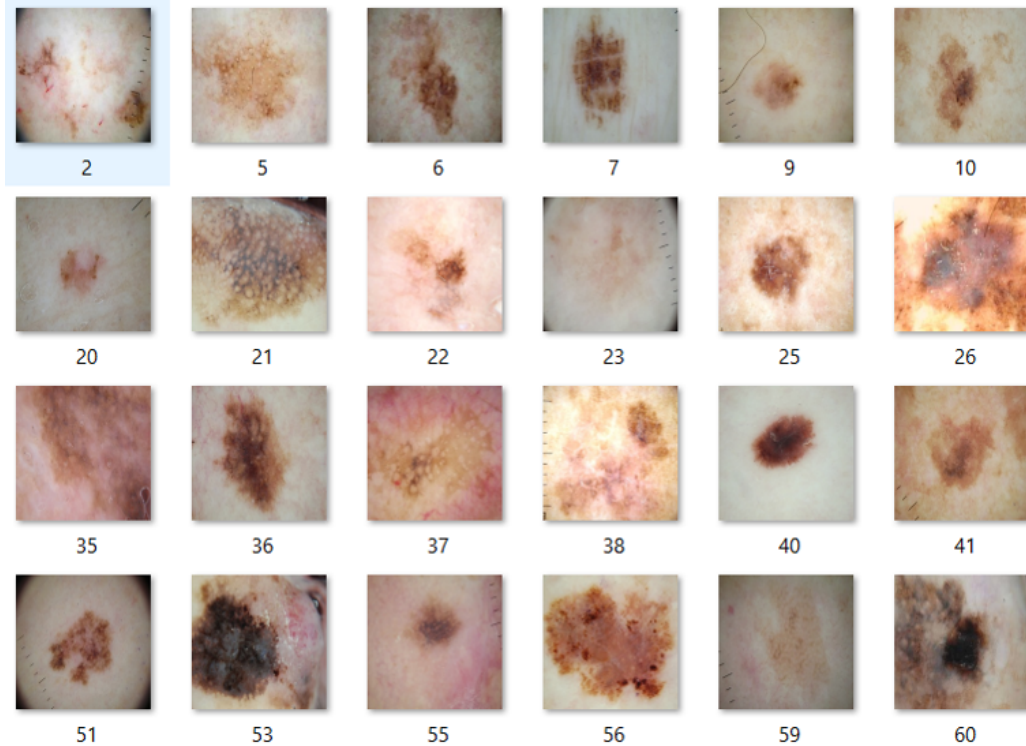


Figure 3.2: Sampled Malignant Images

### 3.2 PREPROCESSING

All the images in the training subset were read using Python Imaging Library abbreviated as PIL, which is an open source image processing library. The pixel RGB values of each image were extracted, and were linearly appended to form a row of RGB values. Each row of this file consists of a total of 150,528 values, that represent 50,176 RGB triplets, and these values were later written to a CSV (Comma Separated Value) file.

### 3.3 RAW IMAGE TRAINING

Scikit Learn's Decision Tree Classifier was utilized as the machine learning model for this dataset. Gini impurity was the criterion used to measure the quality of the splits, and the max depth of the tree was set to 4. After training, the model was tested on the test dataset, and it yielded an accuracy of 77.8%.

### 3.4 SUSPECTED BIAS

The Decision Trees are not ideal for pattern recognition and complex feature recognition extraction within images, yet the classifier had performed relatively very well. This could be due to an apparent bias in the dataset. A majority of the benign images have a pinkish undertone on the skin, while the malignant images are much paler. This could've led the Decision Tree to simply classify all the images with a pink undertone as benign, and as malignant otherwise. To combat this issue, several techniques were experimented with in order to eliminate the bias by replacing all the non-lesion skin with white space.

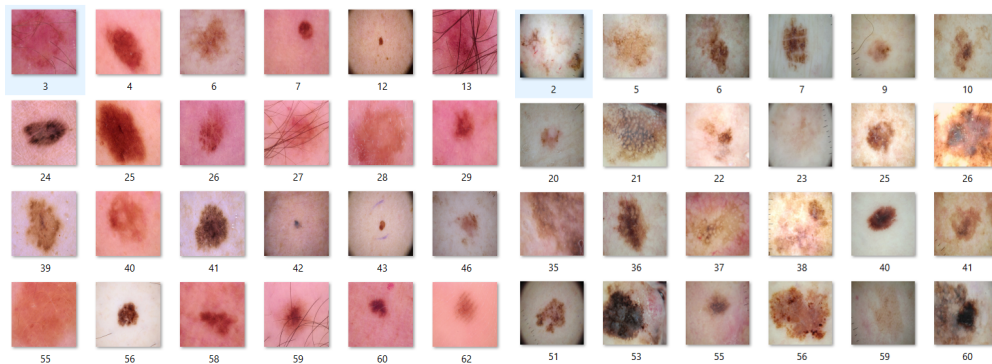


Figure 3.3: Benign images vs Malignant images

### 3.5 ELIMINATING THE BIAS

Multiple approaches were utilized to try and eliminate the dataset's color bias. The most successful approach entailed detecting contours in images and determining if they formed blobs, and then replacing all the non-blob area with whitespace. This process resulted in the following set of images.

The most significant downside of this process was that it was very computationally expensive. It initially took a little over 2 weeks to finish masking all the images when run on MCSR's Sequoia cluster. However, after parallelizing the tasks by utilizing Python's multiprocessing module, the total run time was cut down to 3 days. While the run time for the masking process was reasonably low on this dataset, it would not however be feasible

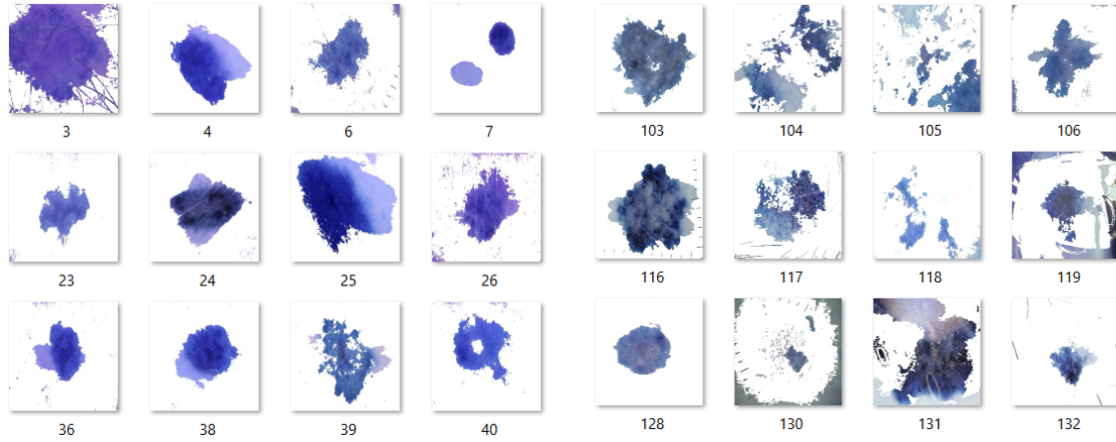


Figure 3.4: Masked benign images vs masked Malignant

to run it on datasets that have a much higher number of images due to the high computing requirements.

### 3.6 TRAINING USING MASKED IMAGES

Scikit Learn's Decision Tree Classifier was utilized as the machine learning model for this dataset. Gini impurity was the criterion used to measure the quality of the splits, and the max depth of the tree was set to 4. After training, the model was tested on the test dataset, and it yielded an accuracy of 72.99%, which was an accuracy drop of 4.81% compared to the model trained using non-masked images.

### 3.7 FEATURE EXTRACTION

Manual feature extraction can be used as a much more efficient alternative to training using only the raw image data. By reducing the number of attributes being passed on to the machine learning model, the time consumed to train it can be reduced significantly (Brownlee, 2020). For this particular dataset, these were the features that were extracted from the images:

- Mean of RGB values.

- Median of RGB values.
- Standard Deviation of RGB values.
- Circularity of the lesion.
- Number of blobs.
- Mean circularity of blobs.
- Edge data using Canny edge detector.
- Roughness of lesion calculated using fractal dimensional analysis.

Similar to masking the images, extracting features was a very computationally expensive task. Even after parallelizing the tasks and utilizing multiple processing nodes, it took 68 hours to extract all the features. After analyzing the algorithms later, it was discovered that the box counting algorithm that was used to compute the fractal dimension score had a time complexity of  $O(n^3)$ .

### 3.8 TRAINING USING ONLY FEATURES

Scikit Learn's Decision Tree Classifier was used as the machine learning model for this dataset. Gini impurity was used as the splitting criterion, and the max\_depth was set to 4. The best accuracy achieved was 69.1%, which was an accuracy drop of 3.89% when compared to the model that was trained using masked images only.

### 3.9 TRAINING FEATURES + MASKED IMAGE DATA

For this trial, the extracted features were appended as columns to the end of the files containing the RGB images. The previously used model was then trained using this data, and yielded an accuracy of 67.7%, which was a 1.4% drop from the previous result.

## 3.10 ISSUES WITH METHODOLOGIES

### 3.10.1 CONFUSION MATRIX

A confusion matrix is a table that is used to determine the performance of a classifier. When the model is tested on a validation dataset, a confusion matrix generates an ordered table of true values and predicted values. These values can further be processed to calculate useful metrics like Recall, Precision, and F-Value (Mishra, 2018). A confusion matrix was never used during training, and it could have helped provide more information about the performance of the model.

### 3.10.2 NORMALIZATION

The image pixel values were not normalized. Normalization helps prevent the overshadowing of certain features (Shalabi, 2006). For example, the range of the RGB values 0 to 255, but the range of some of the extracted features is 0 to 1, and some have an infinite range. This causes the smaller values like circularity score and fractal dimension to be not weighed enough during training.

## CHAPTER 4

### HAM10000 DATASET

#### 4.1 DATASET DESCRIPTION

This dataset contains a total of 10,015 650pi x 400pi images. These images belong to 7 distinct classes - Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv) and vascular lesions (vasc) (Tschandl, 2018). All of the images had been consolidated into a single folder, and a comma separated value file that contained diagnosis and patient information pertaining to these images had been provided. In addition, two comma separated files that contained labelled RGB triplets pertaining to downsized 28x28 and 8x8 images had been provided as well.

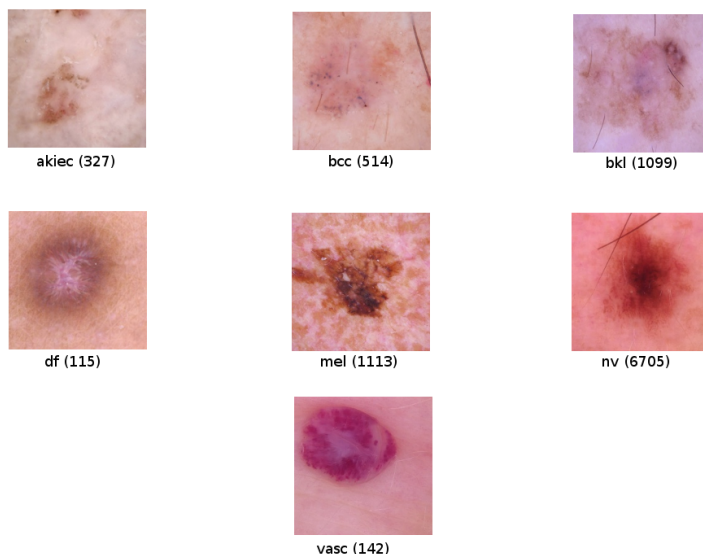


Figure 4.1: Sampled Images and Count

## 4.2 ISSUES WITH DATASET

The high resolution of the images from the original dataset would make it very difficult to train. The data set contains 10,015 images, with 810,000 parameters in every image. It is not practical to use this unprocessed dataset for training as that would require several hundred gigabytes of memory, and it would also take several days, if not weeks to finish training.

## 4.3 PREPROCESSING

This dataset has been randomly split into a training set and a testing set - with 8,015 images belonging to the training set, and 2,000 images belonging to the testing set. The patient diagnosis information had been utilized to separate both, the training data, and testing data into 7 different folders, each representing the images' diagnosis. The images were also cropped and downsized into 224x224pi to reduce training time and memory load.

## 4.4 TRAINING

### 4.4.1 28X28 IMAGES

#### 4.4.1.1 DECISION TREE CLASSIFIER

The 28x28 RGB dataset was utilized to train Scikit Learn's Decision Tree Classifier. The training and testing subsets comprised of 8,015, and 2000 images respectively. The best accuracy achieved with this model was 61.35%, when the max depth was set to 8, and Gini was used as the splitting criterion.

#### 4.4.1.2 ARTIFICIAL NEURAL NETWORK

Keras was used as the framework along with Tensorflow as the back-end engine to construct the following neural networks. An Artificial Neural Network with 6 layers - 2,352 neuron input layer, 8x16x16x8 hidden layers, and a 7 neuron output layer, was trained with the same data for 20 epochs. It achieved a testing accuracy of 66.95%.

#### 4.4.1.3 CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network with 2 convolutional layers, 2 max pooling layers, and 128x50 hidden layers, was trained using the same data for a varying number of epochs. The best testing accuracy that was achieved was 72.3% after the model was trained for 20 epochs.

#### 4.4.2 224X224 IMAGES

##### 4.4.2.1 VGG16

VGG16, a pre-trained competitive model was trained using the cropped and downsized 224x224 dataset. After training for 20 epochs, it achieved a testing accuracy of 66.92%. This task took 15 hours and 34 minutes of wall time to train on a GPU on MCSR's Maple cluster.

##### 4.4.2.2 InceptionV3

InceptionV3, a pre-trained competitive model that is mainly used for computer vision in medicine was further trained using the cropped and downsized 224x224 images (Szegedy et al., 2016). After training for 20 epochs, it achieved a testing accuracy of 73.7%, which was a 1.4% increase when compared to the highest accuracy achieved by a CNN on the 28x28 dataset. This task took 19 hours of wall time to finish training.

#### 4.5 FEATURE EXTRACTION

The ABCDE rule for early melanoma detection (Weigert et al., 2012) was used to extract relevant features from the images. The ABCDE rule states that asymmetry of the lesions, uneven and jagged borders, variation in lesion color, diameter and darkness, and the evolving of any of the above parameters could indicate skin melanoma.

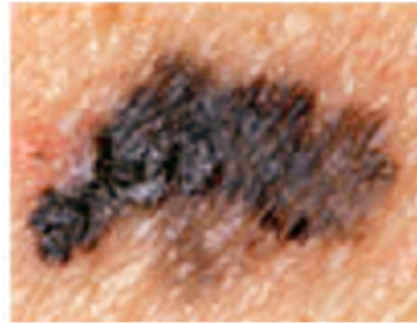
Based on this rule, the following features have been extracted from the image:

- Mean of RGB values.





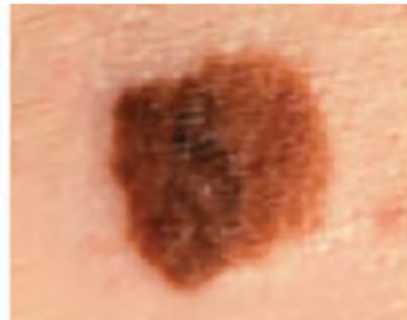
A is for Asymmetry



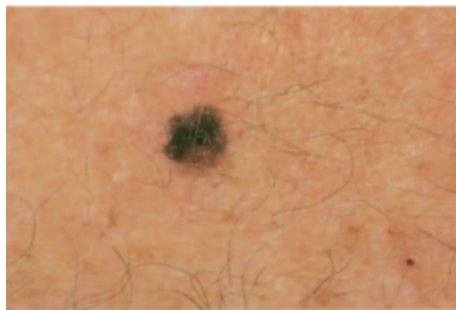
B is for Border



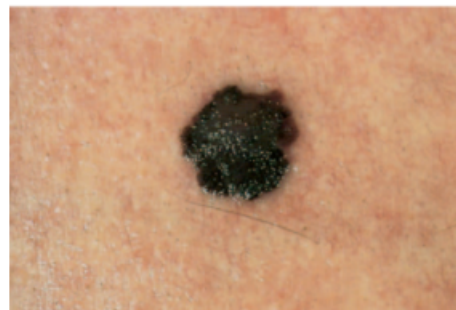
C is for Color



D is for Diameter or Dark



E is for Evolving (Before)



E is for Evolving (After)

Figure 4.2: ABCDE Rule for Early Melanoma Detection

- Median of RGB values.
- Standard Deviation of RGB values.
- Symmetry of the lesion
- Circularity of the lesion

- Roughness of the edges and color variation within the lesion calculated using fractal dimensional analysis.

#### 4.5.1 ALGORITHM OPTIMIZATION

Algorithm optimization was a very crucial aspect during feature extraction using this dataset. Without necessary revisions to the original feature extraction code that was used for the ISIC archive dataset, this could have taken multiple weeks, if not months to finish executing due to the sheer size of this dataset. Below is a comprehensive list of revisions made to the original code:

1. **Ignoring edge data:**

Removing the edge data while training and testing using the ISIC archive features made no difference to the testing accuracy. The number of features that comprised of the edge data were equal to the total number of pixels in the image - 50,176. Including this feature did not just add up to the feature extraction time, but it also significantly increased the training time.

2. **Using OpenCV functions:**

An image masking experiment performed in lieu of the "Caravana Image Masking Challenge" on Kaggle demonstrated that OpenCV's edge detection function was 3.6 times faster than the Python Imaging Library counterpart (vfdev, 2017). As the features "Symmetry", "Circularity", and "Fractal Dimension" rely on edge detection, all PIL functions were replaced with that of OpenCV.

3. **Storing repeating attributes:**

Contours, that were extracted from the edge data were used to calculate symmetry, circularity, and fractal dimension of the images. When extracting features from the ISIC archive images, contours were redundantly extracted for each of the above features. Contours have now been extracted only once and reused for all the features that

depend on it.

#### 4.5.1.1 SPEEDUP

Dataset	Image Resolution	Number of images	Extraction Time (hours)
ISIC Archive	224x224	3,297	68
HAM10000	224x224	10,015	0.467

Even though there were 203.761% more images in the HAM10000 dataset, the extraction time went dropped by 99.31%.

## 4.6 TRAINING USING FEATURES

Artificial Neural Networks, Decision Tree Classifiers, and Random Forest Classifiers were trained using the extracted features, and the highest accuracy obtained by each of them has been listed below.

Model	Hyperparameters	Testing Accuracy
ANN	layers=8x16x32x64x32x16x7, 50 epochs	69.55%
Random Forest	gini, max depth=4	69.72%
<b>Decision Tree</b>	<b>gini, max depth=None</b>	<b>70.1%</b>

## 4.7 TRAINING USING FEATURES + IMAGES

A Hybrid Neural Networks that comprised of a CNN and an ANN, Decision Tree Classifiers, and Random Forest Classifiers were trained using the extracted features concatenated with the 28x28 image pixel data. Image data and extracted features were passed in as separate streams of input layers in the Hybrid Neural Network. The accuracies obtained by them have been listed below.

Model	Hyperparameters	Testing Accuracy
Hybrid Neural Net	CNN=Input(28,28,3)xMP2d(2x2)x Conv2d(15,3,3)xMP2d(2x2)x128x50x7, ANN=12x8x16x32x64x64x32x16x7, 50 epochs	66.67%
Decision Tree	gini, max depth=None	72.72%
<b>Random Forest</b>	<b>gini, max depth=None</b>	<b>73.9%</b>

## 4.8 ISSUES WITH METHODOLOGIES

### 4.8.1 UNINFORMATIVE PERFORMANCE METRICS

Accuracy was not a very informative metric for this dataset. Given that it contained 7 distinct classes with an uneven distribution of images across the classes, accuracy was not helpful in determining the actual performance of the model. Other metrics like precision and recall would have been more robust in evaluating the models.

### 4.8.2 NORMALIZATION AND STANDARDIZATION

The concatenated image and feature data was neither normalized nor standardized before training. This might have caused the models to overlook some of the features that were exclusively floating point numbers between 0 and 1 - like circularity and symmetry.

### 4.8.3 SAVING TRAINED MODELS

Training neural networks using this dataset took multiple days. If the models had been saved, they could have been used to generate more performance metrics like precision and recall, and could have also been trained further with more data.

## CHAPTER 5

### SIIM-ISIC MELANOMA 2020 DATASET

#### 5.1 DATASET DESCRIPTION

This dataset contains a total of 33,126 6000x4000pi images that were extracted from Society for Imaging Informatics in Medicine’s 2020 classification challenge (SIIM-ISIC, 2020). This dataset is heavily imbalanced, with 32,542 images belonging to the benign class, and only 584 images belonging to the malignant class. The test labels hadn’t been made public yet due to the competition still being active. The training data has been split into a training subset and a testing subset as shown below.

Subset	Benign Count	Malignant Count	Total
Training	27,635	491	28,126
Testing	4,907	93	5,000

#### 5.2 PREPROCESSING

It would have been impossible to train any model with the unprocessed images due to their very high resolution and count. To tackle this issue, they were downsized to the following resolutions, and their 3:2 aspect ratio was maintained.

- 30x20
- 75x50
- 120x80

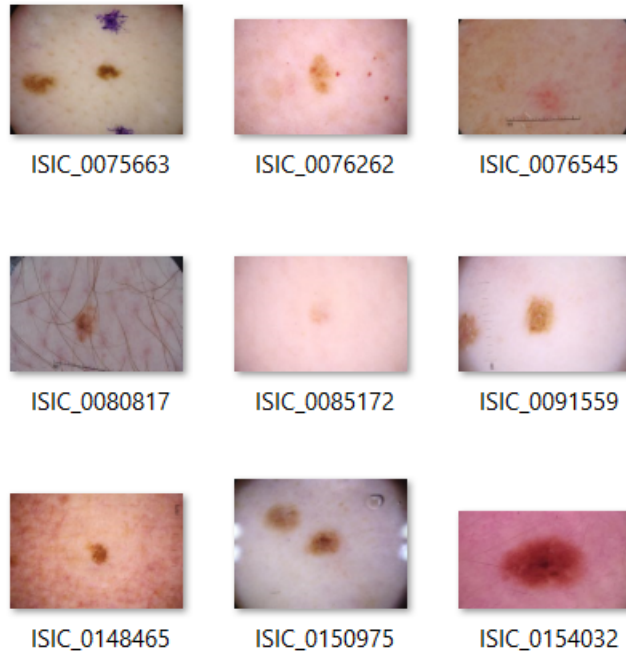


Figure 5.1: Sampled Benign Images

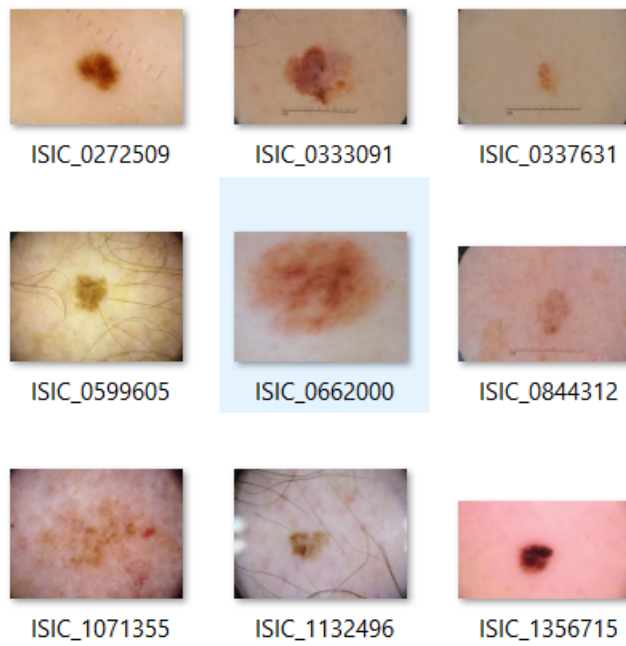


Figure 5.2: Sampled Malignant Images

- 180x120
- 240x160
- 510x340

### 5.3 TRAINING

#### 5.3.1 ISSUES WITH TRAINING

Due to the heavily imbalanced nature of the dataset, all the CNNs that were trained with any resolution of this data defaulted to predicting every image as "benign". To tackle this issue, several techniques were experimented with in an attempt to enhance the minority class.

#### 5.3.2 UNDERSAMPLING THE MAJORITY CLASS

Undersampling the majority class was used as a strategy to reduce the bias in the dataset. The benign class in the training and testing subsets were undersampled to match the size of the malignant class.

Subset	Benign Count	Malignant Count	Total
Training	2,700	491	3,191
Testing	93	93	186

Subset Resolution	Hyperparameters	Train Accuracy	Test Accuracy
30x20	Input(30,20,3)xMP2d(2,2) xConv2d(15,3,3)x Mp2d(2,2)x4x8x16x8x2, 40 epochs	85.11%	61.83%
75x50	Input(30,20,3)xMP2d(2,2) xConv2d(15,3,3)x Mp2d(2,2)x8x16x32x32x16 x8x2, 40 epochs	74.16%	61.22%
120x80	Input(30,20,3)xMP2d(2,2) xConv2d(15,3,3)x Mp2d(2,2)x32x64x128x128 x64x32x2, 40 epochs	71.12%	58.19%
180x120	Input(30,20,3)xMP2d(2,2) xConv2d(15,3,3)x Mp2d(2,2)x32x64x128x128 x64x32x2, 40 epochs	70.33%	57.76%

As the resolution of the images kept increasing, the performance of the models kept worsening.

### 5.3.3 TRANSFORMING IMAGES

To implement this and all the subsequent strategies, the original image dataset has been reshaped on a square grid of resolution 120x120, and all the empty pixels have been filled with RGB(0, 0, 0).



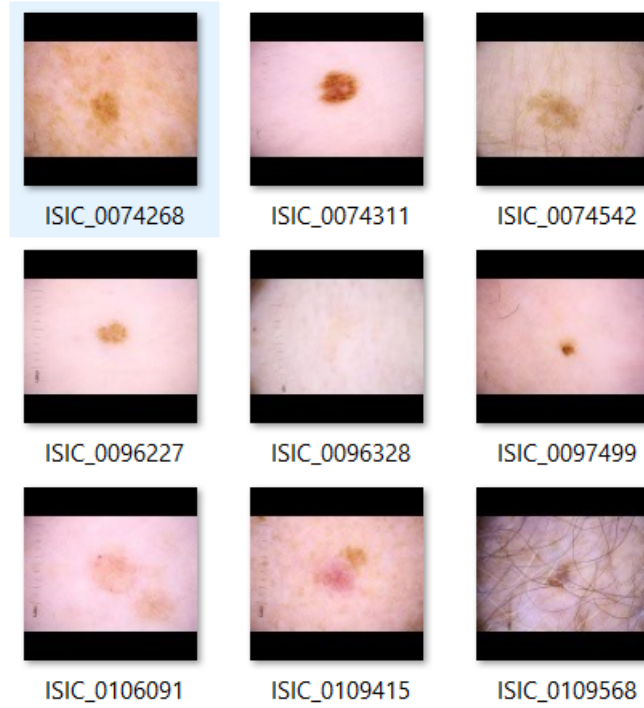


Figure 5.3: Sample Reshaped Benign Images

Every reshaped image in the malignant class was then flipped once, randomly rotated 6 times, and randomly scaled 3 times.

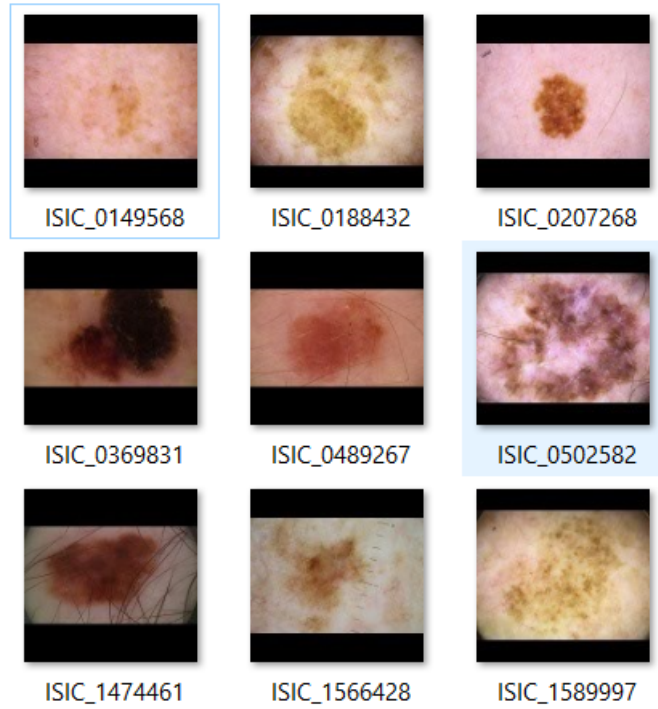


Figure 5.4: Sample Reshaped Malignant Images

Transformed	Hyperparameters	Accuracy	Precision
No	Input(120,120,3)xMP2d(2x2)xConv2d(15,3,3) xMP2d(2x2)x16x32x64x32x16x2, 40 epochs	0.9825	0
No	Input(120,120,3)xMP2d(2x2)xConv2d(15,3,3) xMP2d(2x2)x32x64x128x64x32x2, 40 epochs	0.9825	0
Yes	Input(120,120,3)xMP2d(2x2)xConv2d(15,3,3) xMP2d(2x2)x16x32x64x32x16x2, 40 epochs	0.97	0.075
Yes	Input(120,120,3)xMP2d(2x2)xConv2d(15,3,3) xMP2d(2x2)x32x64x128x64x32x2, 40 epochs	0.97	0.075

Transforming the minority class did not make a significant impact on the performance of the models, but it did however prevent them from simply guessing all the images to be benign.

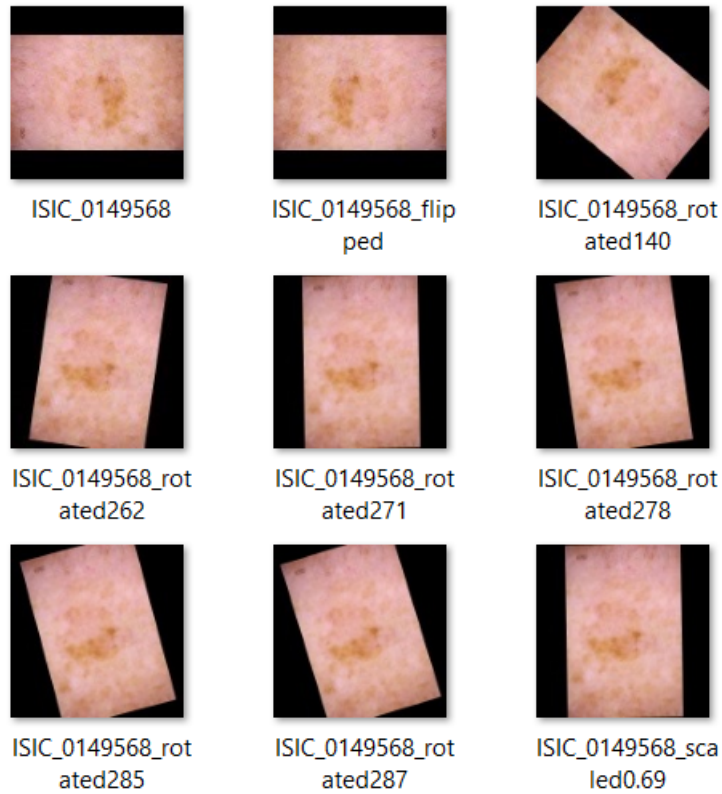


Figure 5.5: Sample Transformed Malignant Images

#### 5.3.4 GENERATIVE ADVERSARIAL NETWORKS AND TRANSFER MODELS

To further enhance the dataset, a generative adversarial network (GAN) has been used to create more replicates of the transformed malignant images. Below is the architecture of the GAN.

Type	Hyperparameters
Generator	Input(120,120,3)xReshape(4,4,256)xUpSampling2d()Conv2d(256)xBatchNorm(0.8)xActiv(relu)xConv2d(256)xBatchNorm(0.8)xActiv(relu)xUpsampling2d()xConv3d(128)xBatchNorm(0.8)xActiv(relu)xUpSampling2d(2,2)xConv2d(128)xBatchNorm(0.8)xActiv(relu)xConv2d(3)xActiv(tanh), Output = RGB image
Discriminator	Input(120,120,3)xConv2d(32)xLeakyReLU(0.2)xDropout(0.25)xConv2d(64)xZeroPadding2d((0,1),(0,1))xBatchNorm(0.8)xLeakyReLU(0.2)xDropout(0.25)xConv2d(128)xBatchNorm(0.8)xLeakyReLU(0.2)xDropout(0.25)xConv2d(512)xBatchNorm(0.8)xLeakyReLU(0.2)xDropout(0.25)xFlatten()xDense(1)xActiv(sigmoid), Output = Boolean value

The GAN was trained for 50 epochs on MCSR’s Catalpa cluster. Catalpa is a cluster reserved for very large memory jobs. Maple’s GPUs could not be utilized due to memory restrictions. The total training time was approximately 11 days and 17 hours.

Below is the new composition of the enhanced dataset.

Subset	Benign Count	Malignant Count	Total
Training	27,635	25,850	53,485
Testing	4,907	93	5000

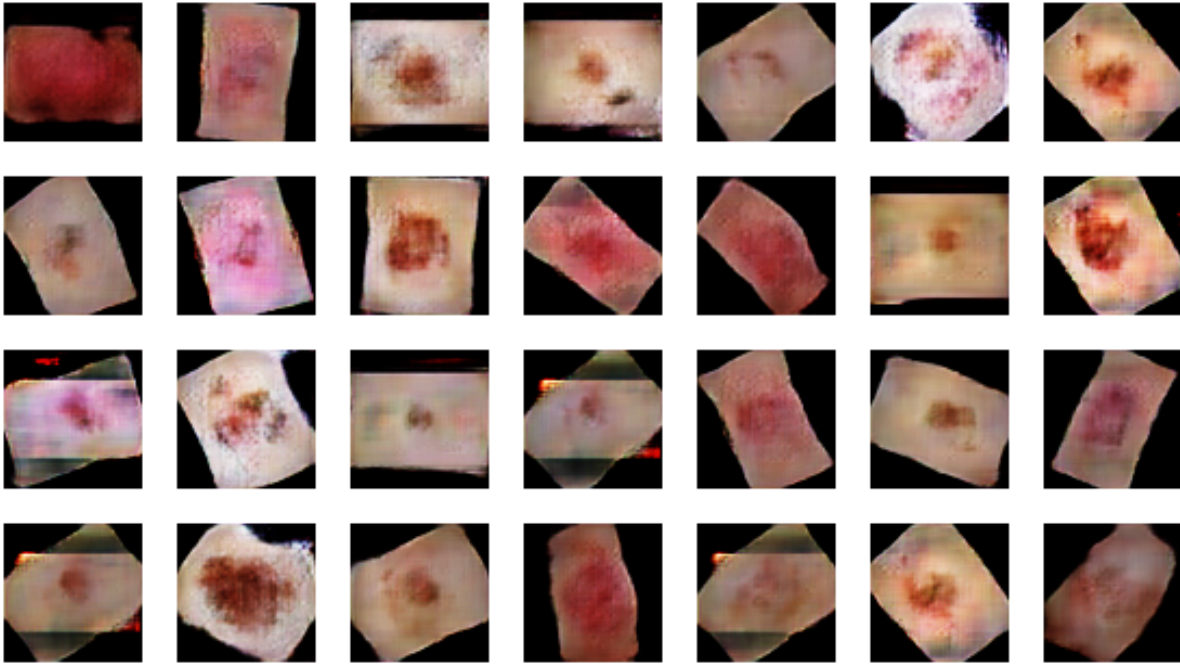


Figure 5.6: Sample GAN output

Hyperparameters	Accuracy	Precision
Input(120,120,3)xMP2d(2x2)xConv2d(15,3,3) xMP2d(2x2)x8x16x32x16x8x2, 20 epochs	0.2285	0.5286
Input(120,120,3)xMP2d(2x2)xConv2d(15,3,3) xMP2d(2x2)x32x64x128x64x32x2, 20 epochs	0.0186	1.0
EfficientNet Transfer Model, 20 epochs	0.1932	0.7312
EfficientNet Transfer Model, 40 epochs	0.2718	0.6667

While the images generated by the GAN reduced the bias of the earlier models, they did not make any significant improvement to the performance of the models.

## CHAPTER 6

### DISCUSSION AND FUTURE WORK

#### 6.1 OVERVIEW

The goal of this project has been to use deep learning to detect skin melanoma in three vastly different datasets - the ISIC Archive dataset, HAM1000, and SIIM-ISIC 2020. Each dataset entailed its own unique challenges, and required a tremendous amount of preprocessing in order to work with.

#### 6.2 PREPROCESSING AND COMMON CHALLENGES

One common challenge among all three datasets was optimizing the code to minimize compute time. This challenge was successfully overcome in all 3 cases by parallelizing the tasks, implementing dimensionality reduction by downsizing the images, and carefully tweaking the algorithms to minimize time complexity. While the resolution and size of the ISIC Archive dataset was relatively low when compared to the other two, this dataset consisted of a bias in the skin color of the images that needed to be eliminated. The masking of these images required a large amount of compute time, and took a little over 2 weeks during the first run. Parallelizing the tasks by employing multiple nodes on the Sequoia cluster to work asynchronously reduced the total run time to a approximately 3 days. The feature extraction process of the ISIC Archive dataset was also very computationally expensive, and it took a total of 68 hours to finish even after the job was parallelized. Eliminating redundancies in code, switching to OpenCV functions, ignoring unnecessary edge data, and employing more number of nodes dropped the runtime of this section from 68 hours to 0.1556 hours. Using this revised feature extraction algorithm on the much larger HAM10000 dataset resulted in

a runtime of 0.467 hours. Image resolutions were drastically downsized in the HAM10000 and SIIM-ISIC 2020 datasets as the original data was too large to train a model with in a practical amount of time. In addition, the Maple GPU cluster was utilized to train the images in these two datasets in order to increase efficiency.

### 6.3 ISIC ARCHIVE DATASET

Scikit Learn’s Decision Tree Classifier was the only model used for this dataset. The performance of this model when trained using the unmasked images was higher than its performance when trained using masked images. This was probably the case because the classifier simply picked up the pink undertones in the benign images and classified the images appropriately. Training the model with only the extracted features negatively effected the performance, and training the model with the image data and appended feature data made it even worse. This might have been the case because ”edge data” was one of the features that was extracted from the images, and it represented a 224x224 image filled with a black (0, 0, 0) background, and with the edges being outlined with white pixels (255, 255, 255) in the foreground. This feature defeated the purpose of the feature extraction process - dimensionality reduction. It instead ended up adding more attributes to the data, which led to a decrease in performance.

### 6.4 HAM10000 DATASET

The Decision Tree Classifier performed the worst on this dataset - with a peak accuracy of 61.35% on the 28x28 RGB downsized dataset. Artificial Neural Networks performed slightly better on the same dataset with a peak accuracy of 66.95%. A Convolutional Neural Network with hidden layers of dimensions 128x50 achieved the highest accuracy on the 28x28 dataset - 72.3%. The 224x224 versions of the datasets were trained using transfer models VGG16 and InceptionV3. InceptionV3 achieved the highest accuracy on this version of the dataset - 73.7%. A Decision Tree Classifier with max depth=None and gini as the splitting

criterion achieved 70.1% accuracy and performed slightly better than the ANN and Random Forest models. When trained using features and image data, a Random Forest Classifier with gini as the splitting criterion and max depth=None achieved the highest accuracy of 73.9% on the 28x28 dataset.

## 6.5 SIIM-ISIC 2020 DATASET

Due to the imbalanced nature of the dataset, all the CNNs that were trained ended up defaulting to predicting "benign". Undersampling the benign class to approximately 10% of its original size resulted in a maximum accuracy of 61.3% on the 30x20 downsized dataset. Interestingly, the accuracies of the models went down consistently as the the resolutions of the images were increased. Transforming the malignant class and creating additional replicates of them did not make a significant impact on the performance of the models, but it did prevent the model from solely predicting the benign class. The GAN was trained for 50 epochs for approximately 12 days. While most of the images that were generated aren't very representative for the actual malignant class, very few of them did look very convincing. With more time and more computing power, the GAN could have generated sharper and more realistic images. While the GAN images did completely eliminate the older bias of the models, they did little to improve their performance, and counterproductively caused some of the models to default their prediction to the malignant class.

## 6.6 FUTURE WORK

One major recurring issue with this project has been the lack of detailed performance reports for the models. Additional metrics like precision and recall would have helped better understand the performance of the models, but rerunning the training computations would take several hundreds of hours. This issue could have been avoided by simply saving the trained models to non-volatile storage for later access or evaluation. The SIIM-ISIC dataset can be enhanced by appending malignant images to it from the previous years' competitions.



Transforming the dataset after adding the previous years' images will result in a much larger number of malignant images, which could help the models perform better. While the GAN did not perform very well on the existing data, it can be trained again using this enhanced dataset to try and produce higher quality images. Time and resources permitting, the models could also try to be trained with the original full resolution datasets, as downsizing the images could have contributed to a loss of features.

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- (2020), Siim-isic melanoma classification. <https://www.kaggle.com/c/siim-isic-melanoma-classification>.
- Brownlee, J. (2020), Introduction to dimensionality reduction for machine learning.
- Fanconi, C. (2019), Skin cancer: Benign vs malignant.
- Goodfellow, I. Pouget-Abadie, J. Mirza, M. Xu, Bing. Warde-Farley, D. Ozair, S. Courville, A. Bengio, Y. (2014), Generative adversarial networks. arXiv:1406.2661
- Hao, K. (2018), What is machine learning?  
<https://www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/#>:
- Howlander N, Noone AM, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis DR, Chen HS, Feuer EJ, Cronin KA (eds). SEER Cancer Statistics Review, 1975-2018, National Cancer Institute. Bethesda, MD, [https://seer.cancer.gov/csr/1975\\_2018/](https://seer.cancer.gov/csr/1975_2018/), based on November 2020 SEER data submission, posted to the SEER web site, April 2021.
- Maarouf M, Costello. CM. Gonzalez, S. (2019), In vivo reflectance confocal microscopy: emerging role in noninvasive diagnosis and monitoring of eczematous der-matoses.
- Mishra, A. (2018), Metrics to evaluate your machine learning algorithm.
- Mitchell TC, Karakousis G (2020), Chapter 66: Melanoma. in: Abeloff's clinical oncology. 6th ed. Philadelphia, pa: Elsevier.
- Niederhuber JE, e. a. (2019), Melanoma: descripción general.
- Shalabi, L. (2006), Coding and normalization: The effect of accuracy, simplicity, and training time.
- Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016), Rethinking the inception architecture for computer vision, doi:10.1109/CVPR.2016.308.
- Tschandl, P. (2018), The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, doi:10.7910/DVN/DBW86T.
- vfdev (2017), Caravana masking challenge. <https://www.kaggle.com/c/carvana-image-masking-challenge/code>
- Weigert, U., W. Burgdorf, and W. Stolz (2012), ABCD rule, pp. 113–117, doi: 10.3109/9781841847627-13.