January 2019

melNET: A Deep Learning Based Model For Melanoma Detection

Shudipto Sekhar Roy

Recommended Citation

Sekhar Roy, Shudipto, "melNET: A Deep Learning Based Model For Melanoma Detection" (2019). *Theses and Dissertations*. 2586.
https://commons.und.edu/theses/2586

melNET: A Deep Learning Based Model for Melanoma Detection

by

Shudipto Sekhar Roy
Bachelor of Science, Khulna University of Engineering and Technology, 2014

A Thesis
Submitted to the Graduate Faculty
of the
University of North Dakota
in partial fulfillment of the requirements

for the degree of
Master of Science

Grand Forks, North Dakota

August
2019

This thesis, submitted by Shudipto Sekhar Roy in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

Dr. Jeremiah Neubert

Dr. Ronald Marsh

Dr. Naima Kaabouch

This thesis is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies of the University of North Dakota and is hereby approved.

Dr. Chris Nelson,
Associate Dean, Graduate School

7/30/19
Date

PERMISSION

Title	melNET: A Deep Learning Based Model for Melanoma Detection
Department	Mechanical Engineering
Degree	Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in his absence, by the chairperson of the department or the dean of the Graduate School. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Shudipto Sekhar Roy
July 25th, 2019

Table of Contents

ACKNOWLEDGMENTS v

ABSTRACT vi

1. INTRODUCTION 1

1.1 Background 4

1.2 Related Works 4

2. DEEP LEARNING 11

2.1 Deep Learning Networks 13

2.1.1 Recursive Neural Network (RvNN) 13

2.1.2 Recurrent Neural Network (RNN) 15

2.1.3 Convolutional Neural Network (CNN) 16

2.2 Deep Learning Architecture Used 18

3. METHODOLOGY 22

3.1 Data Collection 22

3.1.1 Dermoscopic Data 22

3.1.2 Digital Data 24

3.2 Blurry Image Removal 25

3.3 Data Augmentation 27

3.3.1 Histogram Equalization 28

3.3.2 Dilation 29

3.3.3 Erosion 31

3.3.4 Median Filtering 32

3.3.5 Sharpening 32

3.3.6 Mirroring 34

3.3.7 Rotation 34

3.4 Model Training 36

3.5 Model Testing 41

3.6 Result Generation 42

4. melNET AS A TOOL 45

5. RESULT ANALYSIS 47

6. CONCLUSION 53

7. APPENDICES 54

7.1 Appendix A 54

REFERENCES 58

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor and mentor, Dr. Jeremiah Neubert for the support and guidance he has provided me throughout the course of my research and study. His guidance, boundless knowledge and motivation have pushed me to explore the breadth of my capabilities and I will forever be indebted to him for that.

I would like to extend my gratitude to my committee members, Dr. Ronald Marsh and Dr. Naima Kaabouch for their constant support and academic insight.

I am grateful to Akkas Uddin Haque, Ahmad Jarjis Hasan, and, Ramtin Kardan for lending me a hand in various stages throughout the research, including data acquisition and annotation, evaluating several deep learning models, and, reviewing this work. I would also like to thank Department of Dermatology of the University Medical Center Groningen (UMCG) and, Dermatology Service of Hospital Pedro Hispano, Portugal for providing datasets for this research.

Finally, I would like to thank all my friends and family for their unfailing support and encouragement throughout the years.

ABSTRACT

Melanoma is identified as the deadliest in the skin cancer category. However, early-stage detection may enhance the treatment result. In this research, a deep learning-based model, named “melNET”, has been developed to detect melanoma in both dermoscopic and digital images. melNET uses the Inception-v3 architecture to handle the deep learning part. To ensure quality optimization, the architectural aspects of Inception-v3 were designed using the Hebbian principle as well as taking the intuition of multi-scale processing. This architecture takes advantage of parallel computing across multiple GPUs to employ RMSprop as the optimizer. While going through the training phase, melNET uses the back-propagation method to retrain this Inception-v3 network by feeding the errors from each iteration, resulting in the fine-tuning of network weights. After the completion of the training step, melNET can be used to predict the diagnosis of a mole by taking the lesion image as an input to the system. With a dermoscopic dataset of 200 images, provided by PH2, melNET outperforms the work with YOLO-v2 network by improving the sensitivity value from 86.35% to 97.50%. Also, the specificity and accuracy values are found to be improved from 85.90% to 87.50%, and, from 86.00% to 89.50% respectively. melNET has also been evaluated on a digital dataset of 170 images, provided by UMCG, showing an accuracy of 84.71%, which outperforms the 81.00% accuracy of the MED-NODE model. In both cases, melNET got treated as a binary classifier and a five-fold cross validation method was applied for the evaluation. In addition, melNET has been found to perform the detections in real-time by leveraging the end-to-end Inception-v3 architecture.

1. INTRODUCTION

Though Melanoma covers only 1% of all skin cancers, it is responsible for the highest rate of death among all skin cancer related casualties [1]. A statistic from The Skin Cancer Foundation shows that melanoma accounts for approximately 10,130 deaths in the US annually [2]. For 2017, the American Cancer Society estimates 87,110 new incidences of melanoma resulting in 9,730 deaths [1]. The occurrence of both non-melanoma and melanoma skin malignant growths has been expanding over the previous decades. At the present time, each year somewhere in the range of 2 and 3 million non-melanoma skin growths and 132,000 melanoma skin cancers get diagnosed globally [3]. As indicated by Skin Cancer Foundation Statistics, one skin cancer carrier gets found in each three cancer diagnosed patients. In addition, the findings suggest that, one in each five Americans will produce skin cancer cells in their lifetime.

Because of depletion in the atmospheric ozone layers, the climate is losing its defensive capacity against the solar radiation. As a result, a large amount of ultraviolet ray from the sun hits the Earth's surface. Ultraviolet radiation can be expressed as a part of the electromagnetic spectrum which gets emitted from the sun and hits the earth’s surface [4]. As, the wavelengths are shorter than visible light, ultraviolet radiation is invisible to the naked eye. It penetrates our atmosphere and plays a negative role in our day to day life. UV radiation is highly responsible for several medical conditions such as melanoma and other skin cancers, premature skin aging, and, eye damage. It also attacks the immune system and weakens the ability of the body to fight off diseases naturally.

Practically all skin malignant growths (around 99% of non-melanoma skin diseases and 95% of melanoma) are brought about by an excessive amount of ultraviolet radiation from the sun or other sources, for example, solaria (solariums, sun lamps, and sunbeds) [5]. Ultraviolet radiation is comprised of UV-A and UV-B beams which can enter into the skin and cause permanent harm to the cells underneath. UV-A goes deeply into the skin and cause damages to cells in the genetic level. It is also responsible for photo-ageing such as wrinkling, blotchiness etc., and, suppresses the immunity level of the body. On the other hand, UV-B penetrates into the top skin layer and cause skin damage like sunburn which is a significant risk factor for melanoma skin cancer. Depending on body's present immunity level, these damages to skin cell caused by ultraviolet radiation may or may not get repaired. If the body is unable to repair these damages, there is a high chance that the damaged cells might start to partition and develop in an uncontrolled way. This unnatural growth can result a tumor in the long run. A study predicts that, an addition of 300,000 non-melanoma and 4,500 melanoma skin disease cases will get diagnosed only because of a 10 percent cut in recent ozone levels [3]. The worldwide rate of melanoma keeps on expanding, and, the primary factors that fuel this increase appear to be associated with the recreational activity under the sun and a previous record of sunburn.

Melanoma develops like other malignant growth types. Every cell in a body has DNA genes, which are responsible for cell division and reproduction of that cell [6]. Like other cancerous attacks, inherit behaviors of a cell get impaired in the case of melanoma and the damaged cell starts to grow without any control. That uncontrolled growth of the skin cells results a malignant tumor eventually. In this scenario, the harm to the DNA is generally brought about by the excessive exposure to ultraviolet radiation. Melanocytes is a type of cell, which produces the pigment melanin, gets affected in this process. Medical records show that, the first tumor gets usually developed in the skin [7]. If that first tumor stays undiagnosed and untreated, melanoma starts to expand along the epidermis. Eventually, it penetrates the deeper skin layers and infect the blood and lymph vessels.

At the point when a biopsy demonstrates that melanoma is present, determining the stage of the cancer is the first thing to concern about. Melanoma cancer has five designated stages (0 to IV) [8]. Treatment process and prognosis are fully depended on the stage of cancer. For a stage-0 melanoma, the cancer cells exist only the upper layer of the skin. Stage-0 melanoma indicates that the cancer has not spread to the lymph nodes. A stage-1 melanoma suggests that, the cancer cells have accumulated up to 2 mm in thickness, but it has not spread to the lymph nodes. At this stage, there is a chance that the cancer cells may have ulceration, which means that the epidermis, covering melanoma, may or may not in intact condition. At stage-2 the cancer cells do not spread to the lymph nodes but there is a high chance that the epidermis covering the melanoma cells might not be intact anymore. Stage-3 is also known as regional spread stage. In this level, the lymph node gets involved with the assurance of ulceration. This stage is further divided into four subclasses. Final stage is the stage-4, where the metastasis takes place beyond the regional lymph nodes. At this stage, the melanoma cancer cells spread to more distant areas of the body and more vital organs like lungs, brain, and bones get affected.

1.1 Background

Like other cancerous cells, melanoma can rapidly spread to other parts of the body causing severe damage. However, the probability of melanoma cells becoming metastasized remains low in early stages [9]. Physicians can take proper steps to keep this cancer under control, if diagnosed and treated in these early periods. They usually prescribe a treatment plan based on the stage of melanoma, type of melanoma, lesion location, age, and, the overall health condition of the patient. An early stage melanoma can be treated effectively with only surgical operation but, if the melanoma reaches to the advanced levels the treatment might include immunotherapy along with surgical steps [10]. To prevent these deaths, a feasible solution for rapid and early detection of melanoma has become a research topic. Some of contemporary works in this field are going to be discussed in the next section.

1.2 Related Works

In recent days, computer vision approaches for detecting skin cancer have been pursued by a big community of researchers. One common way of diagnosing skin cancer is through dermoscopic images. Dermoscopy is a technique of imaging an area of skin using a high-quality magnifying lens, and lighting system [11]. It can capture images with enough detail that skin structure and patterns can be examined through the image. Dermatologists use this type of images to apply one of the several melanoma diagnosis methods. One of these methods is ABCD (Asymmetrical Shape, Border, Color, Diameter) rule where a scoring process takes place based on the presence of those four different features [12]. If a combined score is higher than 5.45, the lesion is diagnosed as a melanoma. The ABCD rule also suggests that, among the four features of ABCD, asymmetrical border is the most prominent one. Quantifying the amount of asymmetry in lesion has been pursued by a number of researchers. In one study a geometrical measurement has been taken on the whole lesion area to calculate the symmetry features such as circularity and symmetric distance [13]. Another study proposes an index of circularity to quantify the amount of irregularity in lesion borders [14].

Menzies method for detecting melanoma is also practiced by dermatologists [15]. This is also a scoring method where dermoscopic features are divided into two types, namely, positive and negative features. Positive features consist of nine identifications such as, blue-white veil, multiple brown dots, radial streaming, etc. On the other hand, negative features include symmetry of pattern and presence of single color. A lesion is diagnosed as melanoma if it has at least one or more of the positive features and neither of both negative features.

Another method of diagnosing melanoma is the seven-point checklist method [16]. This widely used scoring method divides seven dermoscopic features, such as atypical pigment network, irregular streaks, atypical vascular pattern, etc., into two main groups, namely, major and minor. Major group features are responsible for a score of two and minor group features are bound for one. A combined score higher than three detects a lesion as melanoma.

In [17], an automated Global border-detection method has been proposed to detect melanoma in dermoscopic images. This method is done by analyzing color-space, and, thresholding global histogram which has been found to detect the borders of melanoma affected area with a high accuracy.

In another work, the authors have introduced a method where the input image gets divided into several regions which are clinically significant [18]. This method extracts color and texture dependent features from the input image using Euclidean distance transform. Though, dermoscopy is the most effective technique for detecting melanoma, the reliability of the detection also depends on the operating skill of the dermatologists. As the detection depends on human vision and previous experience, making it automatic is an encouraged research topic to pursue.

One work shows that, a bag-of-features classification method can be used on dermoscopic images for automatically detect melanoma [19]. Two methods were presented as global and local for the classification. The global method was performed by automatic segmentation, followed by an extraction of color and texture features for training the classifier. The local method was inclined towards image analysis and recognition.

Deep learning-based model approaches have shown promise in the detection of melanoma cancer. Deep learning algorithms, with the support of recent computational power and large datasets have been able to exceed human performance in many sectors, such as object recognition, playing Atari games or strategy themed board games like “Go” can be the examples [20].

One work used a convolutional neural network (CNN), support vector machine (SVM) and sparse coding to detect melanoma from images [21]. From the domain of natural photographs, this approach takes the benefit of feature transfer and reduces the necessity of feature extraction. The model has scored an accuracy 93.1% for the feature detection task and an accuracy of 73.9% for the classification task. For recognizing melanoma, another work proposes a framework using a fully convolutional residual network (FCRN) for extracting multi-scale features [22].

For acquiring more discriminative and richer features, this method uses a deeper network which has more than 50 layers. The residual network architecture handles the overfitting problem in deeper layers with the assurance of a gain in overall performance. On International Symposium on Biomedical Imaging (ISBI) 2016 Skin Lesion Analysis Towards Melanoma Detection Challenge dataset, this framework ranked first in classification task and second in segmentation. This work demonstrates that, deep residual networks can be a solution for classifying malignant moles from benign ones.

In many of the classification approaches, skin lesion segmentation is the most essential part. Accurate skin lesion segmentation can improve the performance of a classifier. Independent Histogram Pursuit (IHP) is an unsupervised algorithm proposed in a work [23], for the segmentation of skin lesion. This algorithm evaluates the existing image bands and estimates a set of linear combinations which enhance significant structures embedded in the input image. The algorithm achieved an accuracy of 97% by testing it on five different datasets of dermoscopic melanoma images.

A recent work [24] proposes a model where a multi-scale, fully-convolutional residual network (FCRN) simultaneously segments the lesion with an accuracy of 75.3% and predicts a coarse classification for the next step. For refining this coarse classification prediction, a lesion index calculation unit (LICU) has been developed by calculating the distance heat-map. Then another straight-forward CNN gets used for the extraction of dermoscopic features with an accuracy of 84.8%. The model has been found to classify a lesion with an accuracy of 91.2%. However, making the system fully automatic with a satisfactory accuracy rate, and run in real-time is an open research topic. Another work shows that; an automatic analysis of images is often impaired due to the presence of bodily hair. Using percolation algorithm, clusters of connected points can be analyzed in an image and a linear shape of the cluster, along with lower pixel intensity can indicate the presence of hair [25].

In our previous work, we proposed a method which can automatically detect the presence of melanoma characteristics in a mole from dermoscopic image and provides a real-time prediction percentage about possible condition of that mole, using a real-time object detection technique [26]. An available dataset with dermoscopic images of both malignant melanoma and benign moles was used for that purpose. Every image from that dataset was annotated according to the clinical classification of the mole. For making the system robust, the data was augmented by operating rotation and blurring methods. Dilation and erosion were also applied for augmentation but rejected because of the poor performance of the system. The system was trained using a state-of-the art object detection system YOLO-v2 (You Only Look Once: version 2) [27]. YOLO-v2 uses a single neural network to the full image, enabling real-time performance. It is capable of processing images at 40-60 fps using a Titan X GPU. Then, the trained model was tested using a five-fold cross validation technique and the results provided an accuracy of 86%. As the model got trained with the images where bodily hair is visible, while testing it showed invariance to detection of any bodily hair.

Recently, a new way for melanoma detection is being introduced in the form of web and mobile phone application to make this detection process more approachable for the general population. This new form of computer-assisted diagnosis system uses standard digital camera images rather than traditional dermoscopic images. According to the user reviews and press evaluation, one of the successful mobile applications is SpotMole Plus [28]. This application uses the ABCD diagnostic rules for analyzing the information extracted from imaging and pattern recognition methods. Another mentionable mobile application is MelApp [29]. Instead of providing mole classification, this application indicates the risk of melanoma in the level of low, medium or high. However, the lower accuracy of these applications makes it difficult to rely for making clinical decision [30].

In [31], the authors have presented a method with a preprocessing step which ensures noise removal from the digital images and a postprocessing step to localize the regions of interest for extracting features. These extracted features are then fed to a mono-layer perceptron classifier using ABCD rule as the governing identifier. This system results a sensitivity value of 75.1% and specificity of 83.1%. Another work shows a high sensitivity value of 94% with a low specificity of 68% in [32], where digital images are used along with some context information such as skin type, gender, affected part of the body, and, age.

In [33], a model named MED-NODE has been proposed, where k-means clustering method gets used on HSV color space of the input digital images for segmenting the lesion. From the segmented lesion, this model develops descriptors based on extracted color and texture pattern. With a requirement of visual attribute from the examining specialist, this model scored an accuracy of 81% on digital image data. One work with digital image uses a combination of Otsu and k-means clustering segmentation methods for detecting the affected area and extracting several linear and non-linear features from the lesion portion [34]. These features were then evaluated with a machine learning model consisting of five different classifiers. The result has provided an accuracy of 89.7%.

In this work, a model, named “melNET”, has been proposed which is based on deep learning and can be used to detect melanoma in both digital and dermoscopic images. melNET is designed to prompt the researcher throughout the whole development process which is divided into sections for data augmentation, training, testing, and, result evaluation. It uses Inception-v3 [35] network architecture to handle the deep learning part. On ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 classification challenge validation set, Inception-v3 network demonstrates substantial gains over the state of the art: 21.2% top-1 and 5.6% top-5 error for single frame evaluation. This end-to-end model employs RMSprop [36] across multiple GPUs to complete the training procedure. melNET retrains Inception-v3 network by back-propagating the errors to fine tune the network weights. In the event of two population sets (e.g.: melanoma, and, nevus), melNET acts like a binary classifier. The diagnostic ability gets evaluated by using ROC curve analysis on five-fold cross validation setup. Future work could include the development of a mobile application where the user can take images of a suspicious mole, and the mobile application may remotely run online and present the predicted result in real time.

2. DEEP LEARNING

Recent advances in deep learning are responsible for its widespread use. Deep learning is being used for a broad range of applications, including image classification, text mining, social network analysis, multimedia concept retrieval, video recommendation, and so forth [37]. Many other well-known applications like Natural Language Processing (NLP), speech and audio processing, and, visual data processing are also getting a boost up by the touch of deep learning [38]. The explosive growth and availability of data along with the outstanding advancement in hardware performance are fueling this trending field. Being rooted from convolutional neural network, deep learning outperforms its ancestor remarkably. For developing a many-layered model for learning, it takes advantage from graph technologies with neuron transformations.

In traditional machine learning, the performance of a model becomes highly dependent on the quality of the input data representation. A set of high-quality input data leads to a high performance compared to a poor-quality data. Because of this reason, feature engineering has dominated this field for a long duration of time. Feature engineering is the technique for building features from raw data requiring a significant amount of human effort. In the field of computer vision, different feature extraction methods have been proposed including, Scale Invariant Feature Transform (SIFT) [39], Bag of Words (BoW) [40], and Histogram of Oriented Gradients (HOG) [41].

On the other hand, in deep learning, this feature extraction is getting performed in an automated way without requiring a deep knowledge about the domain [42]. Researchers can also extract discriminative features with minimal effort. Deep learning algorithms use a multi-layered architecture to represent data, where the low-level features get extracted from the initial layers and last layers extract high-level features. This whole design is inspired by the sensorial areas in human brain, as our brain automatically extract information from different scenes by taking visual inputs using eyes. So, it can be said that, deep learning mimics the human brain in a way.

The modern era of deep learning had started in 1943, when a computer-based model was developed for mimicking the activity of neocortex region in human brain using neural networks [43]. That model, named as McCulloch-Pitts (MCP) model, became well-known as the prototype for artificial neural models [44]. The model uses a combination of algorithms and mathematics, known as Threshold Logic, for mimicking human thought process. But the drawback was, this model was not designed to learn. The next milestone of this arena was the Hebbian theory. This theory was basically used for the biological systems in nature [45]. The gate for modern neural networks got opened when Werbos introduced a technique called backpropagation [46]. It is a method of using the errors in each step as a feedback while training. In 1980, "Neocogitron" was introduced which inspired modern convolutional neural networks (CNN) [47].

The concept of recurrent neural networks (RNN) was proposed in 1986 [48]. Deep neural network (DNN) reached, when LeNet got introduced in 1990s [49]. But because of the hardware limitations, LeNet was unable to be applied with large datasets. Currently, modern graphics processing units (GPUs) are capable of working with large-scaled matrices, LeNet manages to take the higher spot as the pioneer of modern-day deep learning algorithms. In 2017, Google AplhaGo got the attention of the entire world as this deep learning-based application won 60 online games in a row by defeating professional Go players [50]. It seems like with modern deep learning algorithms, sufficient data for training, and, latest hardware resources any computer vision-based application worth a chance to get developed or improved.

2.1 Deep Learning Networks

As the deep learning community is expanding, in every few months many new network architectures get introduced. In this section, three of the most popular networks are going to be discussed. Those three networks are: Recursive Neural Network (RvNN), Recurrent Neural Network (RNN), and, Convolutional Neural Network (CNN).

2.1.1 Recursive Neural Network (RvNN)

Recursive neural network is a deep neural network which is designed to apply a set of weights over a structured input in a recursive manner. The goal of this network is to construct a structured or scaler prediction by traversing an input representation in topological order. While making predictions in a hierarchical structure, RvNN uses compositional vectors to classify outputs. RvNN has its root from Recursive Auto-associative Memory (RAAM). RAAM was developed to process specific objects which are structured in some arbitrary shape [51]. The concept was taking a recursive type data structure of varying sizes and output a representation with fixed-width distribution. The network gets trained by a scheme known as Backpropagation Through Structure (BTS) [51]. BTS is very similar to the standard backpropagation approach and additionally it supports any tree-like structure. The network gets trained in a way that it is able to reproduce an input layer pattern at the output. RvNN has been showing a remarkable success in NLP applications.

In [52], a model based on RvNN has been proposed which can handle different modality inputs. This work shows that, RvNN has been applied to classify both natural images and language sentences, where, the image gets segmented into different regions of interest and a sentence gets segmented into words. RvNN uses a scoring system to merge possible pairs and develop a syntactic tree. RvNN evaluates the merging plausibility score for each unit pair and the highest scoring pair gets combined in the form of a compositional vector. After getting done with the merging process, RvNN produces a big region containing multiple units. Then a compositional vector gets calculated to represent that region and at last a class gets labeled. Figure 2.1 demonstrates an example of RvNN tree [52].

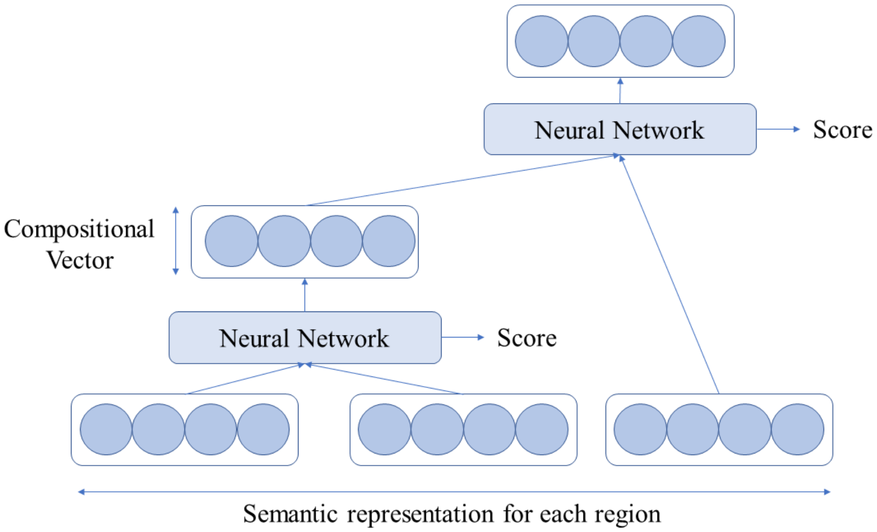


Figure 2.1: An example of RvNN architecture

2.1.2 Recurrent Neural Network (RNN)

For the applications related to natural language process and speech recognition, RNN is widely popular [53]. While traditional deep learning algorithms don't care for sequences in the input information, RNN utilizes this sequential pattern in input. This nature of RNN's design helps many applications to extract sequential pattern from data, for example, the context is essential to understand the role of a word in the sentence. Therefore, RNN architecture resembles a short-term memory unit consisting input, hidden and output layers. Figure 2.2 depicts an RNN architecture for an input sequence [53].

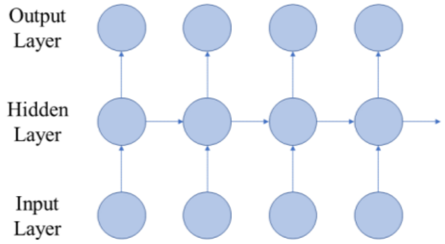


Figure 2.2: An example of RNN architecture

In a work [54], three different approaches targeting deep RNN have been introduced, naming "input-to-hidden", "hidden-to-hidden", and, "hidden-to-output". The proposed method using these three approaches help to develop a deeper RNN efficiently. But, RNN designs has always faced an issue with the exploding and vanishing nature of the gradients due to the multiplications of very small or large values of derivatives while training [55]. This issue introduces a sensitivity problem where the network discards the actual input information. One solution to this problem uses the Long Short-Term Memory (LSTM) technique where some blocks get placed on the recurrent connection for storing memory. Each of these memory blocks has cells inside which stores the temporal state of the network.

2.1.3 Convolutional Neural Network (CNN)

In the field of computer vision, CNN is a very popular term this-day. It also gets used for the applications related to natural language processing and speech recognition. Similar to the other traditional neural networks, the structure of CNN is also inspired by the neurons in a human brain. This most popular deep learning algorithm simulates the visual cortex activity in a cat's brain [56]. CNN is well-known for three main concepts it offers, namely, sparse interactions, parameter sharing, and, equivalent representation [57]. For an image data, CNN is designed to process multi-dimensional structure by utilizing local connections and shared weight. Where, other traditional networks process multi-dimensional data in a fully connected structure, CNN architecture can handle it more efficiently with fewer parameters. This design makes CNN faster than other networks and easier to train. As mentioned earlier, CNN design resembles the visual cortex in the brain, where the brain cells respond for a small section of a scene instead of the whole scene.

In general, CNN architectures contain a bunch of convolutional layers which get followed by pooling layers for subsampling, and, fully connected layers in the final stage which is similar to the design of Multilayer Perceptron (MLP) design [58]. For image classification, the input data (x) gets represented in a matrix form of dimension m x n x r, where m, n and r represent the height, width and depth of the input data respectively.

In each convolution layer, several filters get deployed. These filters contain weight and bias parameters, and, carry out the convolution operation with the input data for generating a feature map. The convolution operation results dot product of the weights and the input from a specific region of the whole volume. Then a nonlinearity gets added by an activation function. Output of a convolutional layer can be defined as

$$h = f(W * x + b), \tag{1}$$

where W, b and x represent the weight, bias and input matrices respectively [59]. In equation 1, function f denotes to the nonlinearity function.

In the pooling layer, the overall volume gets shrunk by down-sampling the representation resulted from convolutional layer. As, the feature map gets down-sampled, the number of parameters gets minimized, hence the training process gets speeded up. This also decreases the risk of overfitting. Two widely used pooling techniques are max pooling and average pooling. On feature maps, this pooling operation is done over a contiguous region.

Finally, the last stage layers are fully connected by design where the low-level and mid-level features obtained from previous layers get utilized to generate the high-level abstraction of input classes. The output layer usually uses the SoftMax or support vector machine methods to generate probability scores of certain classes for a given instance. Figure 2.3 demonstrates the architecture of a convolutional neural network [60] used to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes.

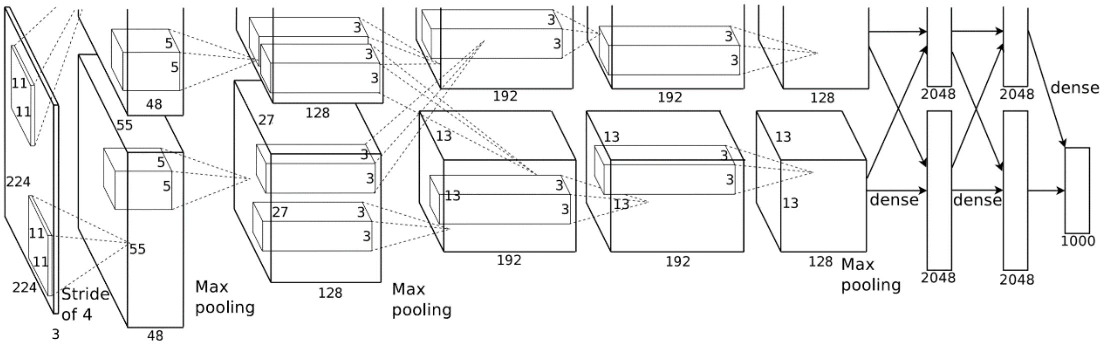


Figure 2.3: An example of CNN architecture

2.2 Deep Learning Architecture Used

In this work, a deep convolutional neural network has been used to execute the deep learning part. melNET uses Inception-v3 [35] network architecture to train and test any dataset. Inception architecture surfaced as the state of the art for image classification and detection task in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14) [61]. The main target of this architecture is to improve the performance by increasing the depth and width of network without any abrupt burden on computation [62]. Increasing the depth of a network refers to an increment on the number of levels, while an increased width means an addition of units at a level. Considering a large amount of labeled data is available for training, those two simple modifications have two considerable drawbacks. Firstly, an increment in size of the network will result an increment in the number of parameters, which will push the enlarged network towards overfitting. Another drawback is the requirement of additional computational power. For example, in the case of a network with two fully connected convolutional layers, with the uniform increase of any number of filters will result a requirement of quadratic increase of computation.

This Inception architecture is designed to utilize the computing resources in an improved way, which has allowed it to increase the depth and width of the network. It solves those two major drawbacks by turning from fully connected to sparsely connected design. Fundamentally, the Inception architecture was designed on the basis of Hebbian principle, which says that, neurons that fire together, wire together. The underlying idea for using this principle refers that, when a deep neural network with a large and very sparse architecture represents the probability distribution of a dataset, the construction of the optimal network topology can be made by doing an analysis on correlation statistics of last layer activations and highly correlated clustering neurons.

Inception network architecture uses the filters of sizes 1 x 1, 3 x 3, and, 5 x 5. This network concatenates the resulting volumes after convoling an input through those filters and forms a single output vector which will act as an input for the next step. Since, modern convolutional neural networks have found pooling operations beneficial, Inception architecture suggests adding a parallel pooling in each stage. A single block of this setup is termed as the "Inception module". Figure 2.4 demonstrates a naive version of Inception module [35].

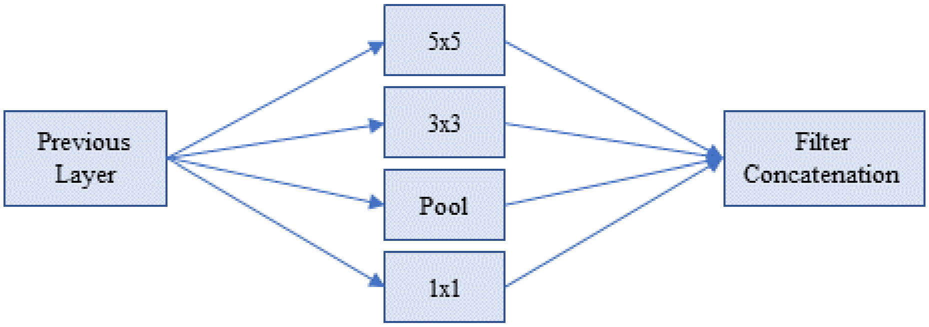


Figure 2.4: Naive version of Inception module

As the Inception modules are designed to be stacked on top of each other, the higher abstraction can be achieved from the outer layers. As a result, the spatial concentration is expected to get decreased in the outer layers, which requires an increase in the ratio of 3 x 3 and 5 x 5 convolutions. But with the naive form of this Inception module, it gets highly expensive to have 5 x 5 convolutions with a large number of filters in the outer layers. Traditional Inception module uses a max-pooling of stride 2 to shrink down the volume in halves. But, adding pooling layer in this combination makes is even worse. The proposed solution of this huge issue is a dimension reduction technique. A 1 x 1 convolution is proposed to be done before the completion of 3 x 3 and 5 x 5 convolutions. This technique not only reduces the expenses of computation but also acts as a rectified linear action. This dual-purpose act of this 1 x 1 convolution allows this network to have a deep and wide architecture. Figure 2.5 shows an Inception module with dimension reduction design [35].

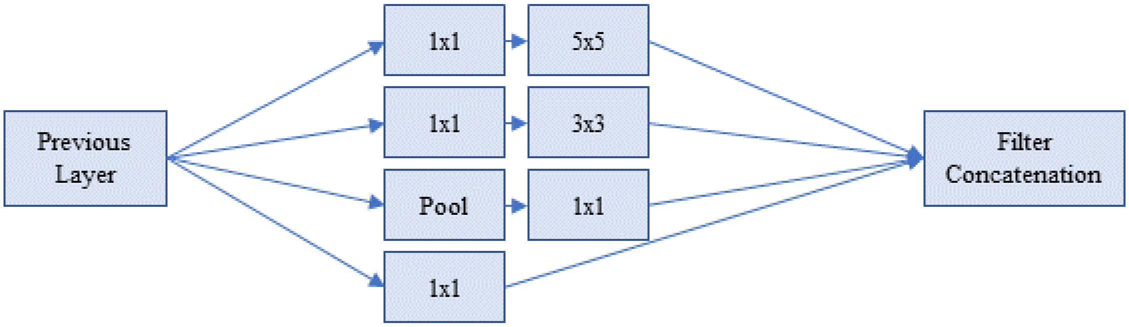


Figure 2.5: Inception module with dimension reductions

The main benefit of this technique is that, it allows the increment in the number of units at each stage without making the computation highly expensive. It shields the input filters of one stage to the next by reducing the dimension before the start of convolution operation. It also allows the network to resemble the visual feature extraction process of the brain by processing the information at several scales followed by an aggregation which allows the next stage to abstract features from multiple scales at a time. This architecture was used in GoogLeNet in the ILSVRC14 competition. The network was 27 layers deep after the inclusion of pooling ones.

In [35], a modified architecture for the Inception network has been proposed. This third version of Inception architecture has been found to result 21.2% top-1 and 5.6% top-5 error for single frame evaluation. It achieves that score with an expense of 5 billion multiply-adds computation per inference while using not more than 25 million parameters. This version mainly factorizes the bigger convolutions into smaller ones. In that paper, it is proposed that any n x n convolution can be replaced by a 1 x n followed by n x 1 convolution. As the value of n gets higher, the computation cost goes down dramatically. Figure 2.6 demonstrates a schematic of Inception-v3 module [35].

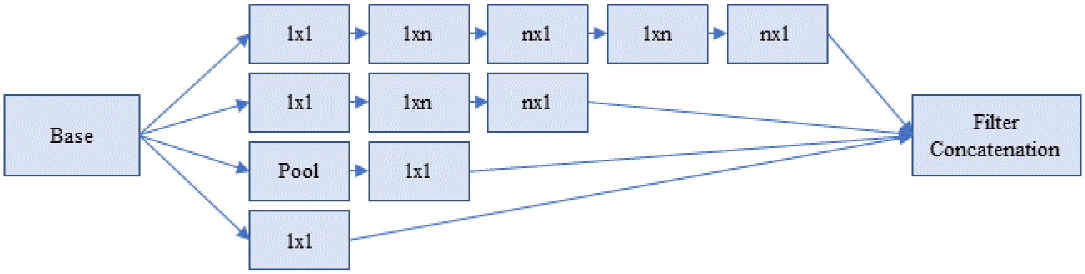


Figure 2.6: Inception-v3 module

3. METHODOLOGY

Our work can be divided into the following main steps where last four steps including Data Augmentation, Model Training, Model Testing, and Result Evaluation gets prompted by the proposed model: “melNET” for taking researcher inputs. First two steps including Data Collection, and Blur Detection are essential for running the model efficiently. Figure 3.1 presents a flow chart demonstrating the methodology used in this work.

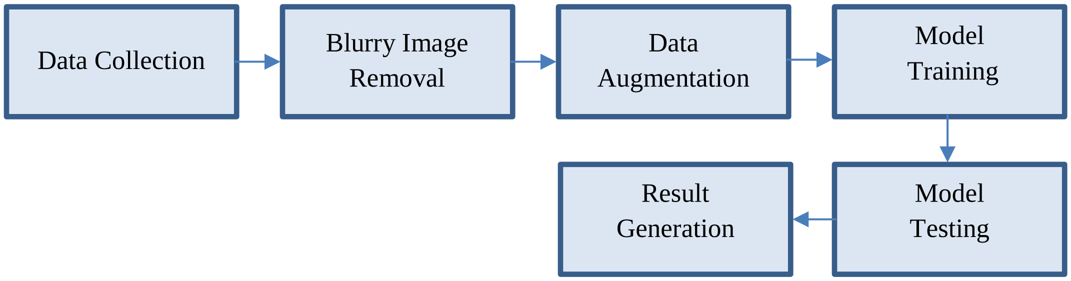


Figure 3.1: melNET methodology

3.1 Data Collection

For facilitating the computer-aided diagnosis related work on melanoma many datasets can be found online. Depending on the image capturing setup and equipment, these datasets can be divided into two main classes.

3.1.1 Dermoscopic Data

In dermoscopic image dataset images are captured using a high-quality magnifying lens in a well-illuminated environment setup. One work shows that, while performing in a clinical setup, dermoscopy results a more accurate detection of melanoma than examining only with the naked eye [63]. The sensitivity value for melanoma detection can be improved by 10%-27% by applying dermoscopy [64]. Typically, the instrument uses a magnifier of x10 capacity. Along with that magnifier, the instrument fashions with a non-polarized light source, a plate with high transparency, and a liquid medium. The design of this instrument allows it to cancel out any skin surface reflections while inspecting skin lesions. In some modern dermoscopic instruments polarized light get used for cancelling out the skin reflection more effortlessly. In dermoscopic dataset, high resolution images of skin lesions get found where skin pattern and structure are clearly visible.

In this work, a dermoscopic image dataset for melanoma detection has been used from PH2 database [65]. The samples used in this dataset are acquired from the Dermatology Service of Hospital Pedro Hispano, Matosinhos, Portugal. This dataset consists of a total of 200 dermoscopic 8-bit RGB color images of melanocytic lesions, together with 80 common nevi, 80 atypical nevi, and 40 melanomas. These images are captured using a magnification of 20x with a resolution of 768 x 560 pixels. For convenience of data extraction, a binary mask of the segmented lesions is also given, along with the original dermoscopic images. The classification of all images can be found within the dataset package. In figure 3.2 some samples of dermoscopic data from this dataset can be observed.

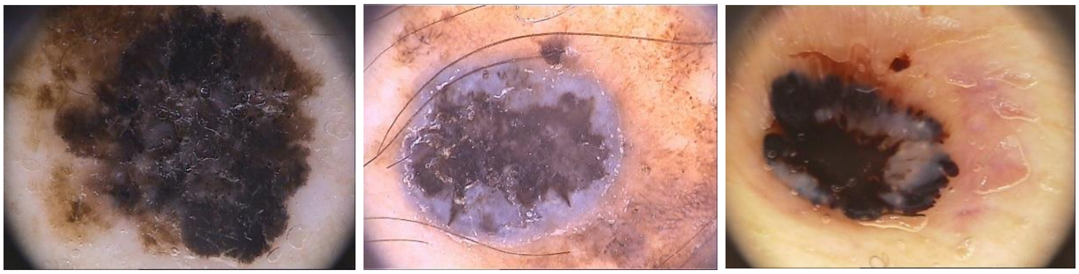


Figure 3.2: Three samples of dermoscopic data

3.1.2 Digital Data

Digital images are quite different from the dermoscopic ones. These are also images of skin lesions but the image capturing equipment and environmental setup are not the same. In this case, images are taken using mobile devices and lighting condition may vary in a wide range of possible setups. Skin pattern and structure are not be able to observe clearly in this type of data. As a result, dermatologists generally do not use this data for the inspection purpose. But as this type of data can be easily gathered, detection of melanoma using digital data has become a new research topic in this field of study. In this work, a dataset consists of 70 digital images of melanoma and 100 digital images of nevus has been used to evaluate the performance of melNET. This dataset has been taken from the digital image archive of the Department of Dermatology of the University Medical Center Groningen (UMCG) [66]. This dataset had already been used for the development and evaluation of MED-NODE system for skin cancer detection [33]. Figure 3.3 presents some samples of digital data from this dataset.

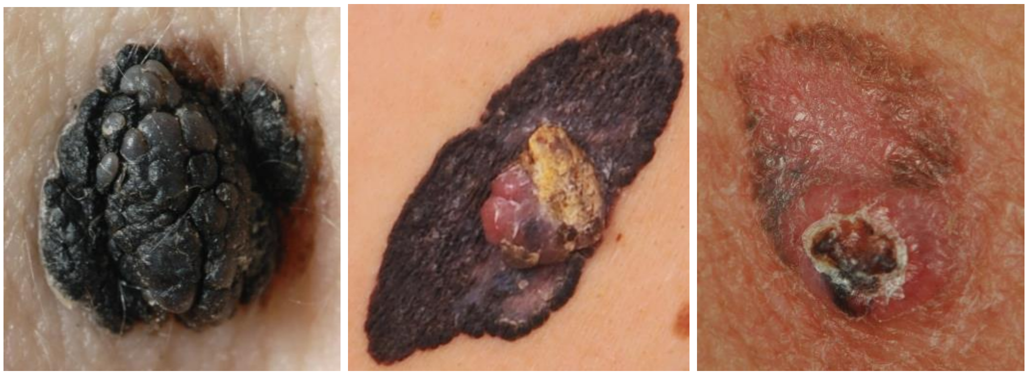


Figure 3.3: Three samples of digital data

3.2 Blurry Image Removal

While dealing with digital images, blurriness is a very popular term. Depending on the lighting condition and movement of camera while capturing a frame, blur might appear on the image. Blurriness changes the distribution of pixel intensity in a way that the image gets treated as an outlier from the entire dataset. This negative impact of blurriness hurts the performance of a deep learning model if it gets trained with blurry images. In this work, as a digital dataset has been used to evaluate the proposed model, dealing with the blurry images was on demand for making the model robust.

One survey work reviewed nearly 36 different methods of estimating the focus measure of an image [67]. One popular way of measuring the blurriness is by computing the Fast Fourier Transform (FFT) of an image [68]. This FFT calculation will help to examine the high and low frequency distribution of that image. If the image has a lower number of high frequency than the applied threshold, it gets considered as blurry. But setting that threshold has been found problematic in some scenarios. A number of methods computes “blurriness metric” for the detection of blur in an image. One method calculates this metric by using a simple and straightforward approach where intensity statistics of grayscale pixels is only needed [69]. On the other hand, another method applies a more advanced approach where an image gets evaluated by its local binary patterns. Another simple and easy to implement method for blur detection is mentioned in this work [70]. Here, the variation of the Laplacian has been analyzed to evaluate the blur. melNET uses this method to detect blur in an image.

The Laplacian operator is used to measure the 2nd derivative of an image. If an image contains rapid changes in pixel intensities, this Laplacian operator helps to highlight those areas. Laplacian operator is usually used for detecting edges in an image. This 3 x 3 kernel: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ convolves with a channel of an image to complete the Laplacian operation. After getting done with the Laplacian operation, the variance of that response needs to be calculated for blur evaluation. According to the probability theory, variance represents the squared deviation of a variable from its average value or mean. It helps to understand the spread of different values of that variable from the mean.

In this work, the underlying assumption is that, if an image gets found with high variance after going through Laplacian operation, it will be considered as an in-focus image. The reason behind this assumption is that, when an image goes through Laplacian operation and the responses have a wide spread, it means the image has a good number of edge pixels present all over in it. And, as in the cases of blurry images the edges yield weak responses, an optimal variance threshold can easily detect the blur. But finding an optimal variance threshold takes some analysis throughout the dataset. In this work, all images from the dataset have gone through the Laplacian operation followed by a variance calculation for estimating the optimal threshold. An optimal threshold value of 7.0 has been found to take the blurry images out of the dataset and improve the overall performance by cutting down false negative detections. Figure 3.4 shows some blur detection examples from this study.

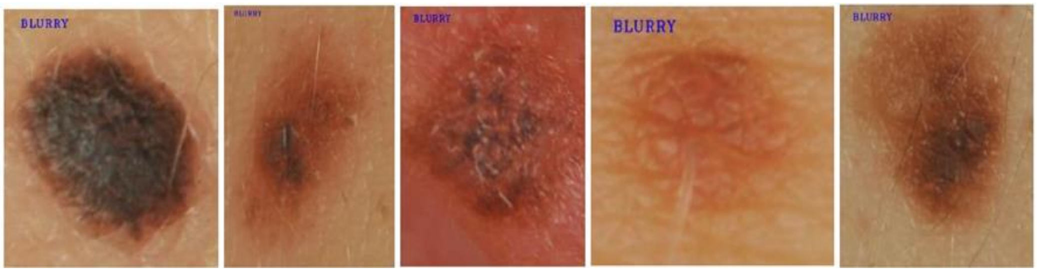


Figure 3.4: Sample images of blur detection process

3.3 Data Augmentation

While working with deep learning, getting a relevant and large enough dataset to train the network has always been a concern. Most of the time, a dataset contains few hundreds of images which is not good enough to make the network robust. Training a deep learning model refers to the tuning of that model parameters. The tuning gets done in a way that it can map a specific input to an output label. The objective of the optimization step is to find a spot where the loss stays at its minimum, which occurs when the tuning gets done in the right way.

As the modern neural networks contain millions of parameters, it takes a proportional amount of data to result considerable performance [71]. So, it is understandable that deep learning models require a large dataset of images to provide desired performance. But it is a wild goose chase to look for novel images in a huge quantity. Instead, minor alterations to the existing data will pull the trick. A poorly trained neural network considers these slightly modified images as distinct ones. Some common examples of minor alteration operations might be rotation, translation, and, flipping. A convolutional neural network is considered to have a property called “invariance”, when it correctly classifies an object without being dependent on its orientation. Modern CNN architectures are expected to be invariant to translation, rotation, size, or, illumination which is the premise of data augmentation. One dataset might have images taken in some limited set of conditions, but the target application might appear in different orientation, scale, lighting, or location. As these situations must be considered while training, the training dataset needs to be enriched with some synthetically modified data.

Data augmentation helps to increase more relevant data in the dataset without suffering the hassle of searching novel data. It also prevents the neural network from learning unnecessary and irrelevant patterns which helps to boost the overall performance of the network. As the dermoscopic dataset utilized only around 200 images and the digital one contains only 170 images, augmentation of the image data was necessary.

This “Data Augmentation” step of melNET takes care of the overall data generation and processing for the following steps in the entire process. Before running the augmentation operations, it detects and deletes the blurry images from the dataset for preventing the network from getting confused due to bad data. As the used deep learning network has been tested to work better with 400 x 400 x 3 size images, melNET also provides an option to resize the entire dataset. In this work, the result got evaluated using a five-fold cross validation method [72]. For executing that evaluation step the whole dataset needs to get divided into five sections. melNET provides an option to organize the augmented dataset compatible for five-fold evaluation.

The seven augmentation operations have been chosen due to their familiarity among augmentation methods for modern deep learning models [73]. Augmentation methods used in this work are going to be discussed in the following paragraphs.

3.3.1 Histogram Equalization

Histogram equalization can be defined as a method for adjusting image intensities in a way to enhance the overall contrast of the image. It reassigns the intensity values of pixels in input images to result an output image of uniformly distributed intensity values.

Let’s consider f as a given image which is represented as a m_r x m_c matrix of pixel intensities. Let these intensity values be integer and ranging from 0 to $L-1$, where L is the total number of possible intensity values. Therefore, the normalized histogram of f, denoted by p, with a bin for each possible intensity becomes:

$$p_n = \frac{number\ of\ pixels\ with\ intensity\ n}{total\ number\ of\ pixels}, \tag{2}$$

where, $n = 0,1,2,3,...,L-1$. Thus, the image after histogram equalization, g can be defined by,

$$g_{i,j} = floor((L-1)\sum_{n=0}^{f_{i,j}} p_n), \tag{3}$$

where, $floor()$ operator rounds down to the nearest integer. Figure 3.5 illustrates an example of histogram equalization operation applied in this work.

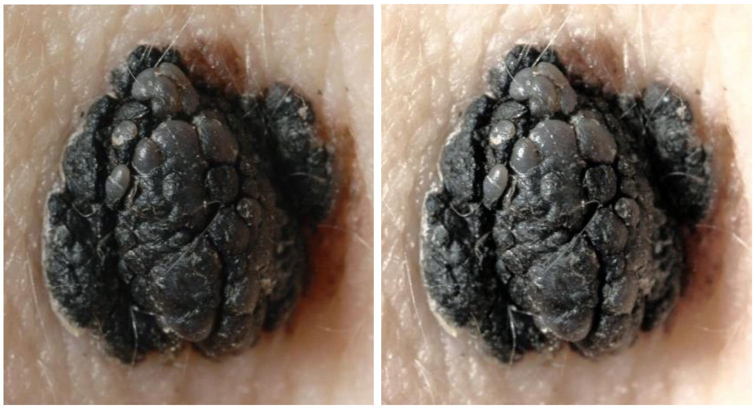


Figure 3.5: An example of histogram equalization operation: original version on left side and hist. equalized version on right

3.3.2 Dilation

Dilation is one of the morphological operations which gets commonly used to enhance the features of an image. Dilation operation requires a set of two inputs including an image to be dilated, and a two-dimensional matrix called structuring element. Dilation operation gets used for many computer vision and image processing applications for mostly exaggerating features in an image. On a binary image, where white pixels refer to the foreground typically, this operator gradually enlarges the boundary size of the foreground while minimizing the size of the holes within the image.

The dilation operation is completed by convolving an image matrix by a structuring element in a Euclidean space. If k is the structuring element of a defined anchor point, and f_b is the translation of an image matrix, f by b, the dilation operation can be given by the equation [74]:

$$f \oplus k = \cup_{f \in b} f_b. \tag{4}$$

For this morphological operation, the structuring element is also termed as a kernel. Usually, the center of the kernel acts as the anchor point. In this work, a 3x3 kernel of a rectangular shape has been used as the structuring element. That structuring element convolves over the whole image while computing the maximal pixel value overlapped by it and then, the pixel in the anchor point position gets replaced by that maximal value. The resulting effect of this operation appears to grow the white regions in an image. Figure 3.6 presents an example of the dilation operation applied in this work.

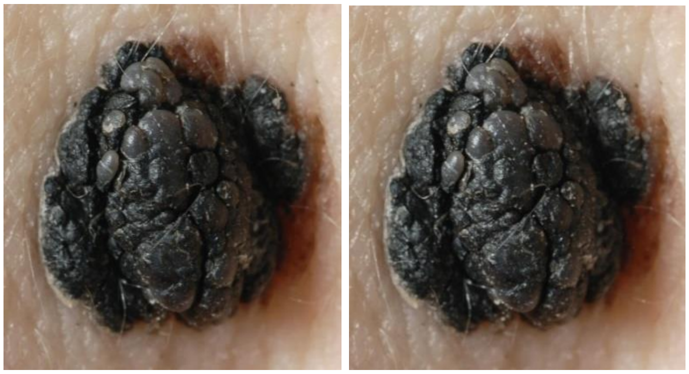


Figure 3.6: An example of dilation operation: original version on left side and dilated version on right

3.3.3 Erosion

Erosion is the other one of morphological operations which gets commonly used for feature enhancement of an image. This operation also requires a set of two inputs including an image to be eroded, and a two-dimensional matrix called structuring element. Many computer vision and image processing applications use erosion for mostly exaggerating features in an image. On a binary image, where dark pixels refer to the background typically, this operator gradually enlarges holes while casting a thinning effect on the white foreground.

The erosion operation can be given by the equation:

$$\mathrm{f} \ominus \mathrm{k} = \{\mathrm{z} \in \mathrm{E} \mid \mathrm{k}_{\mathrm{z}} \subseteq f\}, \quad (5)$$

where an image matrix (f) gets convolved using a structuring element (k) in the Euclidean space (E) and k_z represents the translation of k by the vector z [74].

In this work, a 3x3 kernel of a rectangular shape has been used as the structuring element. The center of the kernel acts as the anchor point. That structuring element convolves over the whole image while computing the minimal pixel value overlapped by it and then, the pixel in the anchor point position gets replaced by that minimal value. The resulting effect of this operation appears to grow the dark regions in an image. Figure 3.7 presents an example of the erosion operation applied in this work.

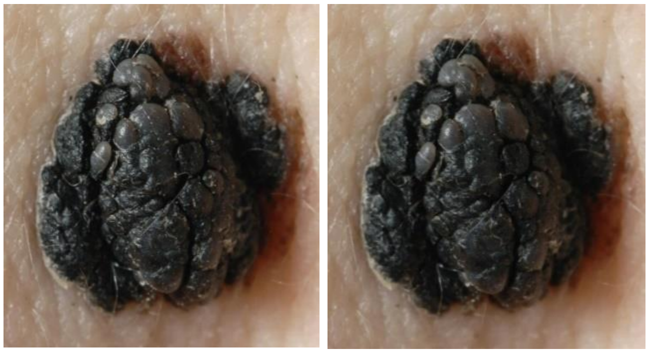


Figure 3.7: An example of erosion operation: original version on left side and eroded version on right

3.3.4 Median Filtering

Median filtering is a nonlinear filtering technique that is most commonly used to remove salt-and-pepper noise from images. As the name suggests, salt-and-pepper noise shows up as randomly occurring white and black pixels that are sharply different from the surrounding. In color images, salt-and-pepper noise may appear as small random color spots.

As the dataset used in this research contained images with salt-and-pepper noise, a median filtering was also needed to be applied. This operation computes the median of all the pixels under a kernel and the central pixel of that kernel gets replaced with this median value. Figure 3.8 illustrates an example of the blurring operation applied in this work.

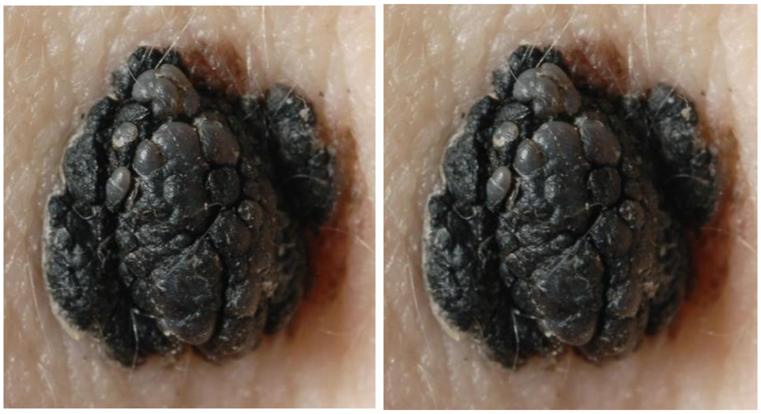


Figure 3.8: An example of median filtering operation: original version on left side and median-filtered version on right

3.3.5 Sharpening

Sharpening is another powerful method while augmenting data for training modern deep learning models. It enhances the components with higher frequencies in an image. As a result, the edge pixels in an image get highlighted. This fine detailing technique is not only used for data augmentation, but also widely used for photography and printing industries for increasing the contrast locally.

Basically, this sharpening technique consists of two main steps. At first, it filters the original image with a high-pass filter which extracts the high-frequency information from that image. Then, that output from the previous step gets scaled and added to the original image resulting a sharpened version of the image. This operation can be represented by,

$$S_{i,j} = x_{i,j} + \lambda F(x_{i,j}), \tag{6}$$

where $x_{i,j}$ refers to the pixel value of the original images, Function F represents a function for high-pass filtering, λ is a parameter used for tuning, and, $S_{i,j}$ is the calculated pixel value of the sharpened image. Figure 3.9 shows an example of sharpening operation applied in this work.

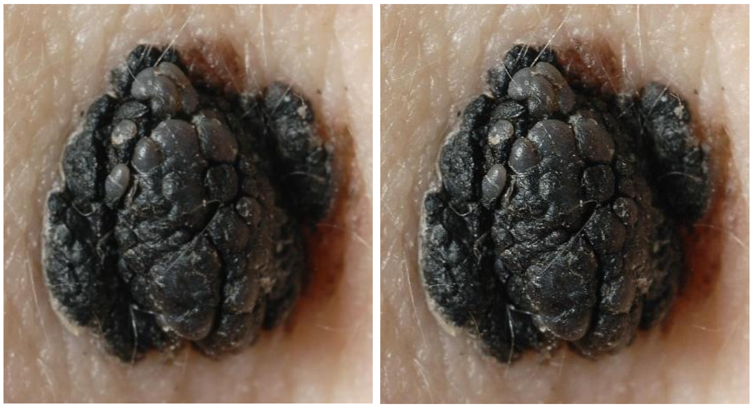


Figure 3.9: An example of sharpening operation: original version on left side and sharpened version on right

3.3.6 Mirroring

Mirroring is one of the common occurrences in real world. Polished surfaces cause specular reflection resulting this mirroring effect in our day to day life. But this simple phenomenon can also be used to train a deep learning model robust. As we know, an image is nothing but a matrix of numbers, simply flipping that matrix does this mirror effect effortlessly. As, flipping can be done with respect to both x and y direction, in this work it has only be done about y-direction. Let's assume a matrix: $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ denoted by A. Then flipping A with respect to y-direction will result the matrix: $\begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{bmatrix}$.

As, the augmentation stage includes a step for rotation, flipping the images about both directions is quite redundant. Figure 3.10 presents an example of mirroring operation applied in this work.

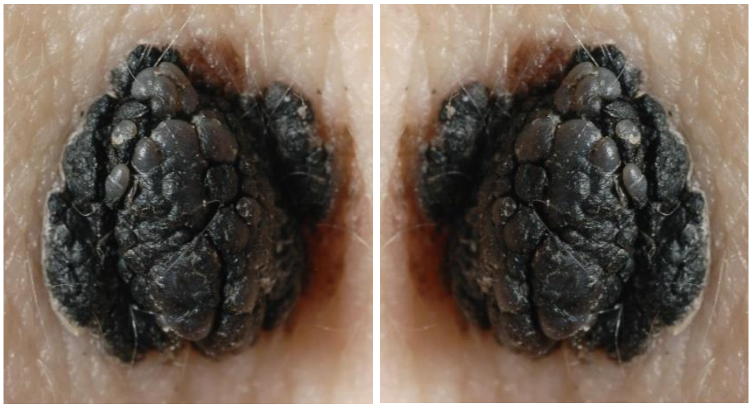


Figure 3.10: An example of mirroring operation: original version on left side and mirrored version on right

3.3.7 Rotation

Final method of augmentation stage was rotation. Rotation of the images was necessary as the system could be trained with the same image features as viewed from several different angles. To rotate the image, a 2 x 3 transformation matrix was calculated. Given the center of the image, angle of rotation and, scale value, the transformation matrix becomes:

$$\begin{bmatrix} \alpha & \beta & (1-\alpha)\cdot center\cdot x-\beta\cdot center\cdot y \\ -\beta & \alpha & \beta\cdot center\cdot x+(1-\alpha)\cdot center\cdot y \end{bmatrix}. \tag{7}$$

Here, equation (7) represents the transformation matrix, where, $\alpha = scale\cdot\cos(angle)$, $\beta = scale\cdot\sin(angle)$. Scale value was kept as one because the images were not gone through any zooming process. Applying this transformation matrix in the following equation provided the desired rotation in the output image:

$$dst(x,y) = src(M_{11}x+M_{12}y+M_{13}, M_{21}x+M_{22}y+M_{23})\,. \tag{8}$$

Equation (8) is used for getting the desired rotation, where, M refers to the transformation matrix, $src(x,y)$ is the input image, and, $dst(x,y)$ is the output image. Figure 3.11 illustrates an example of sharpening operation applied in this work.

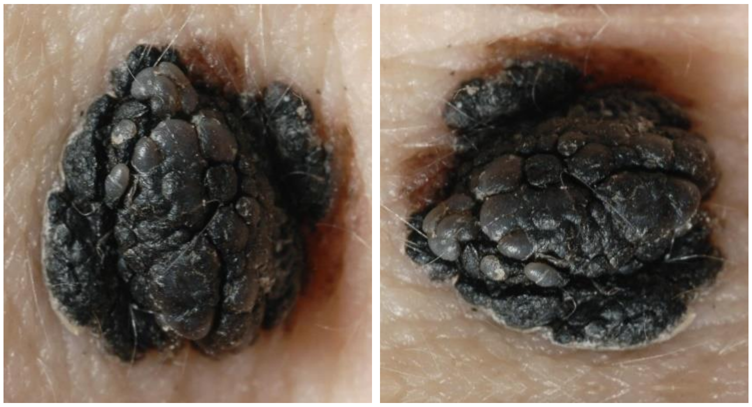


Figure 3.11: An example of rotation operation: original version on left side and a 90-degree rotated version on right

3.4 Model Training

As the modern deep neural networks require millions of images to train the entire model, it is inconvenient to collect that amount of data for a particular task. Therefore, a method called transfer learning has been getting attention for training a deep neural network with a limited amount of data. Transfer learning is the method of taking a pre-trained model and fine-tune the model with a smaller dataset. In this work, a pre-trained Inception-v3 model has been utilized which was already trained with ImageNet 2012 Challenge validation dataset. This ImageNet dataset contains 14 million images with over 1,000 classes. Last five layers of that pre-trained model have been retrained using the augmented data kept for training. All other layers were kept frozen as the weights of those layers remain same throughout the whole period. In the concept of transfer learning these frozen layers act as the feature extractors for retraining the model. The output layer has been defined with only two nodes, as we have two classes for this task. This layer applies the SoftMax function over the nodes to generate class probabilities for a given image. SoftMax function maps the non-normalized output of a network to a probability distribution over predicted output classes. SoftMax equation is defined as,

$$S(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{n} e^{y_i}} \tag{9}$$

where n is the number of classes, and y is the activation value for a node [57].

melNET also uses batch normalization for improving the stability of this network. This method normalizes the output of an activation layer by a two-step operation. At first it subtracts the batch mean and then divide the activation by the batch standard deviation. This normalization method minimizes the amount of covariance shift in the hidden layers which helps to make the model robust to small changes in distribution of the input data. Batch normalization operation adds some noise to the activations of the hidden layers, resulting a slight regularization effect which minimizes the risk of overfitting. Also, this normalization method allows a higher value for the learning rate by minimizing the risk of exploding and vanishing problems of the gradients.

In the realm of deep learning, one iteration is termed to indicate the completion of one backpropagation step. Backpropagation is a technique to tune the weights on each node. Basically, this method consists of four distinct sections naming the forward pass, the cost function, the backward pass, and, the weight update.

During forward pass, the training images get passed through the whole network. Batch size is one of the hyperparameters plays an important role for improving the speed and performance of the model. It refers to the number of images get passed through the network at a time. In this work, a batch size of 100 has been tried to execute the training step. A smaller batch size results a fast update of the parameters, whereas a larger batch size takes a very long time to get the model working correctly. The initial layers, which are responsible for extracting lower level features works good with these new images, but as the last layers cannot correspond these images with the pre-established weights, the output layer cannot deliver the desired result during the initial iterations.

As the output prediction fails to match the ground-truth, the backpropagation method reaches to the second step called the cost function. As every image used for training has a label, this section is used to calculate the overall error because of the current weights on each node. A cost function was developed which represents the overall error of the model. melNET utilizes the mean squared error (MSE) method for cost function which basically calculates the average of the squared differences between the actual and predicted values. If a batch of n images get passed through the network resulting n number of predictions, then the MSE is defined as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overset{\wedge}{Y_i})^2 \ , \tag{10}$$

where Y is the vector of input labels and $\hat{Y}$ is the vector of output predictions [75].

After developing the cost function, the backpropagation method steps into the third section naming backward pass. This section uses an optimizer to minimize the cost function from previous section. The objective of the optimizer is to determine the parameters in correspondence to the global minima of the cost function. There are many well-known optimizers which are getting used for different deep learning applications.

Gradient descent optimizer is the most popular among all because of its simplicity and effectiveness. It applies an iterative movement towards the steepest descent of the cost function which gets guided by the negative value of the gradient. This gradient descent optimizer can further be classified into two different approaches. Batch gradient descent is a technique where the entire dataset passes through the network for a single optimization. On the other hand, stochastic gradient descent which is used by the previous versions of Inception networks, optimizes the network by passing a single image at a time. This method is fast for approximating the global minima of the cost function.

Inception-v3 network uses the RMSprop optimizer which is very similar to the gradient descent with momentum. The role of momentum is to blend the current values of update with the ones from previous step. As, a result a bigger value for learning rate can be used which results a faster convergence to the global minima. Learning rate is one of the hyperparameters which has a vital role for the overall performance of the optimizer. It represents the step size the optimizer will take for each iteration. If a smaller learning rate is chosen for the application, the smaller steps might take a very longer time to reach the global minima. On the other hand, a larger learning rate might take bigger steps with a risk of missing the global minima and bouncing around it. The operation of this optimizer can be represented by the following equations:

$$v_{dw} = \beta \cdot v_{dw} + (1-\beta) \cdot dw^2, \tag{11}$$

and $$v_{db} = \beta \cdot v_{db} + (1-\beta) \cdot db^2 \tag{12}$$

where β represents the momentum and conventionally the value of it was set to 0.9. dw and db represent the differentiation of the cost with respect to a weight and bias variable respectively. v_{dw} and v_{db} represent the amount of change needed to be updated [76].

The last section of the backpropagation is termed as weight update. If the learning rate is denoted by α, the weight update operation for RMSprop optimizer calculates the new weight and bias by using:

$$W = W - \alpha \cdot \frac{dw}{\sqrt{v_{dw}} + \varepsilon}, \tag{13}$$

and $$b = b - \alpha \cdot \frac{db}{\sqrt{v_{db}} + \varepsilon} \tag{14}$$

where W and b represent the weight and bias respectively. The values v_{dw} and v_{db} can get very close to zero. In that case, the gradients stay under the risk of blowing up. In equation 13 and 14, that ε parameter in the denominator which contains a very small value saves the gradients from blowing up in the deeper layers. In this work that ε value is set to 10^{-8} [76]. The completion of these four steps are defined to be one step in the backpropagation algorithm.

Two types of scenarios have been observed to hammer the performance of a deep learning model. First one of them is termed as high bias problem which can be simplified by naming it under fitting. This occurs when the model is not well-tuned with the training set images. An underfit model seems to neither perform well with the validation dataset nor with the training dataset. In this work, training the model for a longer time has been used as the remedy for this problem. On the other hand, an overfit model is over-tuned with the training set images. The model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This scenario is also known as the high variance problem. Using a regularization method can minimize the risk of overfitting. In this work, a method called dropout has been implemented to overcome this problem. Dropout is a method of deactivating a random set of activations by setting them to zero. By doing this, the model gets restricted to put a higher amount of importance to a specific feature.

This work uses five-fold cross validation method to evaluate the performance of the model. The entire dataset gets divided into five subsets with a similar combination of images from each class. For reducing the risk of overfitting, one image along with its all augmented versions are kept in a single subset. Every subset then further gets divided into several groups naming after different classes. While training, this group name gets used as the label for the images it includes. For each training fold, one subset gets separated as a validation set. This validation set then gets used for testing the resulted final weight.

3.5 Model Testing

This step utilizes the weights achieved from training step to predict the classes for the images in the test-set. As the training has been done using a five-fold cross validation approach, melNET needs to execute the testing in the similar manner. melNET also provides an option for the researcher to visualize the detection on test data. In this work, this monitoring approach has been proved to be useful for finding the difficulties the model is facing while going through this test-phase. For example, rotation technique for augmentation had been found to generate some artifacts in the augmented data which was hurting the performance of the model. Figure 3.12 demonstrates the artifacts in the augmented images after rotating an image by 30 degrees. Border interpolation methods are also mentioned in the figure.

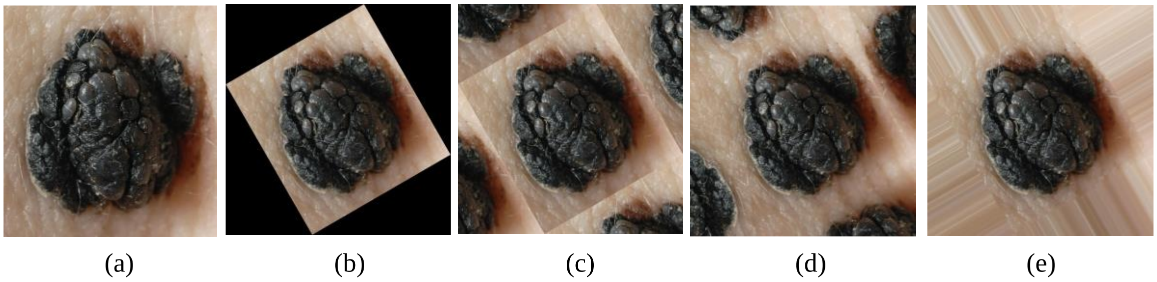


Figure 3.12: Rotating a sample image by 30 degrees using different border interpolation methods: (a) original image, (b) 30 degrees rotation using “constant” border interpolation, (c) 30 degrees rotation using “wrap” border interpolation, (d) 30 degrees rotation using “reflect” border interpolation, (e) 30 degrees rotation using “replicate” border interpolation

Later, these artifacts had been removed by limiting the rotation angles to the multiples of 90 from 0 to 270. This model also saves the performance report for further analysis. It is designed to generate a confusion matrix to plot a ROC curve which ultimately determines the capability of this model.

3.6 Result Generation

In this work, as melanoma and nevus are the only two classes getting dealt with, melNET treats this model as a binary classifier. Binary classification refers to assigning elements in a dataset to one of two groups. It is widely used in medical testing when determining whether a patient has a certain disease or not.

For evaluating the performance of a binary classifier, a confusion matrix has been generated in this work. This confusion matrix is a table that describes the performance of melNET on a given dataset. The detection of melanoma moles is the prime purpose of this work, melanoma class has been set to be the positive class.

Depending on a threshold value, this confusion matrix identifies the detection of a test image as one of the four possible scenarios. If a mole gets detected as melanoma, melNET considers this detection as a positive detection. Now, depending on the ground truth this positive detection can either be a true or a false. If the ground-truth labels this image as a melanoma, this detection will be considered as a true positive (TP), otherwise it is a false positive (FP). Similar definition goes for negative scenarios. If melNET does not detect a mole as a melanoma class, that detection gets considered as a negative. Then, depending on the ground-truth label, that detection could either be a true negative (TN) or, a false negative (FN).

Threshold for detection is an important parameter to determine while testing the dataset. A specific value of threshold allows a certain range of predicted probability to be claimed as detected. Here, we ran the test by varying the threshold from 0.0 to 1.0 for each 0.05 interval to generate a Receiver Operating Characteristic (ROC) curve. ROC curve is widely used for measuring the performance of a deep learning-based classifier [77]. It is a probability curve which indicates the capability of a model to distinguish between classes accurately. The area under this curve (AUC) indicates the ability of a model to classify positive data as positive and negative data as negative. In this work, a ROC curve was used for each of the five folds to evaluate the response of melNET as either melanoma or non-melanoma (benign). The ROC curve was generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. The TPR is also known as sensitivity which can be formulated by,

$$Sensitivity, or, TPR = \frac{TP}{TP+FN}. \tag{15}$$

Sensitivity can be described as the possibility of a model to detect the positive class as a positive detection. On the other hand, specificity is the possibility of a model to classify a negative class as a negative. Specificity is

$$Specificity = 1 - FPR, \tag{16}$$

whereas, the FPR, or, fall-out is defined as

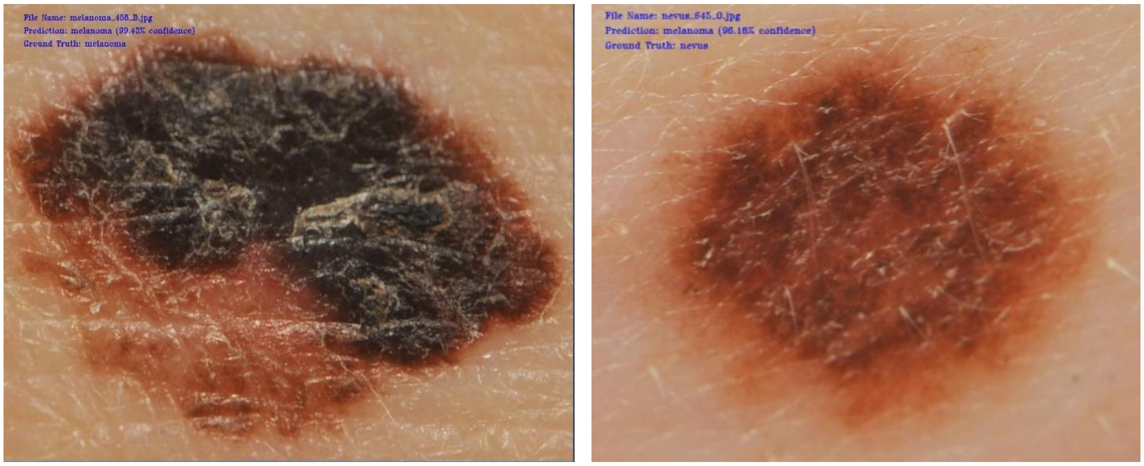
$$FPR = \frac{FP}{FP+TN}. \tag{17}$$

Accuracy (ACC) yields the correct detection ability of a model in both positive and negative cases. Accuracy of a classifier for a certain threshold is

$$ACC = \frac{TP+TN}{TP+TN+FP+FN}. \tag{18}$$

The area under the curve (AUC) is a measurement of how well the classifier can distinguish between two groups. An acceptable threshold value was found through analyzing these curves.

The five final weights obtained from five training folds were used for testing the system. Each final weight was applied for testing the subset which was not included while training that fold. The output of the testing provided the detected class type, along with the class prediction probability for each tested image. Figure 3.13 demonstrates two sample detections from testing.



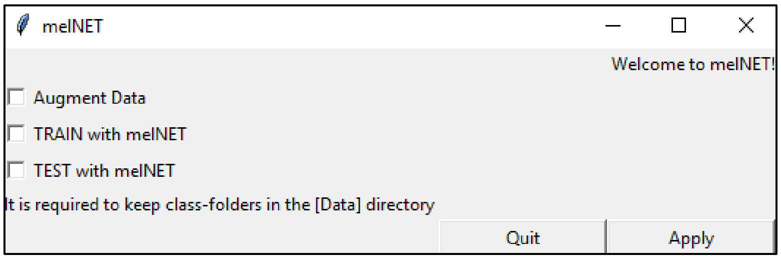
```
File Name: melanoma_456_0.jpg
Prediction: melanoma (99.43% confidence)
Ground Truth: melanoma
```

```
File Name: nevus_645_0.jpg
Prediction: melanoma (96.16% confidence)
Ground Truth: nevus
```

Figure 3.13: Sample detections from testing melNET: a correct detection sample on left side, and, a wrong detection sample on the right

4. melNET AS A TOOL

melNET is designed to prompt the researcher throughout the whole development process. The user-interface makes the use of this model easy for trying different sets of augmentation methods and hyper-parameters. melNET starts with a window providing basic three choices for data augmentation, model training and test the model for performance evaluation. Figure 4.1 illustrates the prompt window for basic operations.



melNET

Welcome to melNET!

□ Augment Data

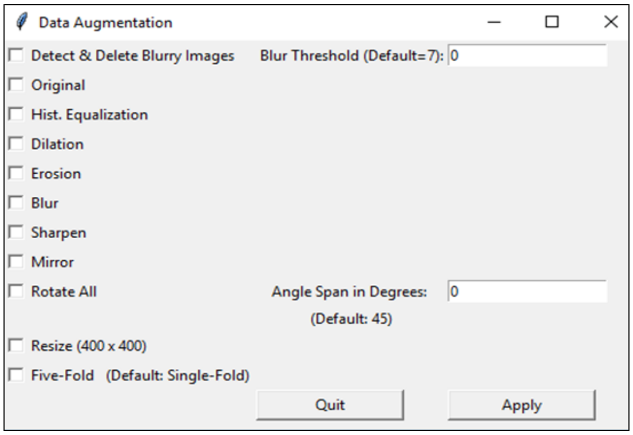
□ TRAIN with melNET

□ TEST with melNET

It is required to keep class-folders in the [Data] directory

Figure 4.1: Basic operation window of melNET

melNET is designed in a way that it makes the augmentation process effortless. To initiate the augmentation, it provides seven options to the researcher including histogram equalization, dilation, erosion, blurring, sharpening, mirroring, and rotation. The process for removing the blurry images from the dataset and resizing the input data in a specific size are also included in this prompt. Figure 4.2 displays the prompt window for data augmentation of melNET.



Data Augmentation

□ Detect & Delete Blurry Images Blur Threshold (Default=7): 0

□ Original

□ Hist. Equalization

□ Dilation

□ Erosion

□ Blur

□ Sharpen

□ Mirror

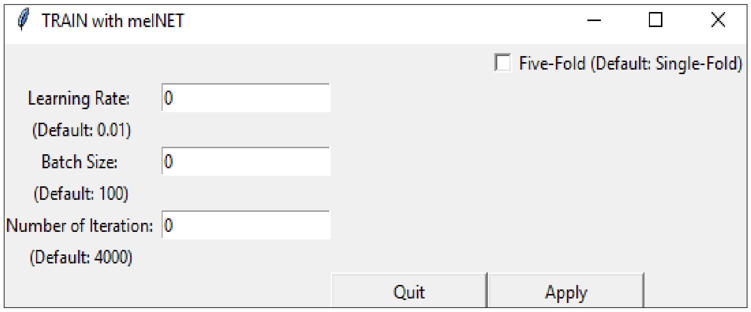
□ Rotate All Angle Span in Degrees: 0
(Default: 45)

□ Resize (400 x 400)

□ Five-Fold (Default: Single-Fold)

Figure 4.2: Data augmentation window of melNET

After completion of the augmentation part, the model needs to be trained with the train-set data. Like the previous step, melNET prompts the researcher to input essential hyperparameters including learning rate, batch size and number of iterations to initialize the training. Figure 4.3 presents melNET's training prompt window.


TRAIN with melNET

□ Five-Fold (Default: Single-Fold)

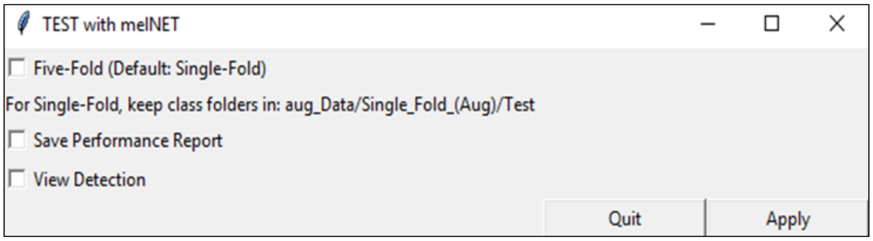
Learning Rate: 0
(Default: 0.01)

Batch Size: 0
(Default: 100)

Number of Iteration: 0
(Default: 4000)

Figure 4.3: Training window of melNET

When the model gets trained with the training images, melNET prompts the researcher to evaluate the model using the test-set images. Figure 4.4 illustrates that prompt window used by melNET for initializing the testing step.


TEST with melNET

□ Five-Fold (Default: Single-Fold)

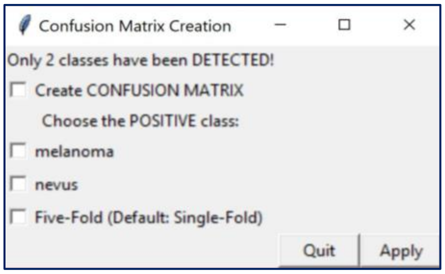
For Single-Fold, keep class folders in: aug_Data/Single_Fold_(Aug)/Test

□ Save Performance Report

□ View Detection

Figure 4.4 Testing window of melNET

melNET treats the model as a binary classifier if only two classes get detected. Figure 4.5 shows the prompt window melNET generates on the detection of two classes.


Confusion Matrix Creation

Only 2 classes have been DETECTED!

□ Create CONFUSION MATRIX

Choose the POSITIVE class:

□ melanoma

□ nevus

□ Five-Fold (Default: Single-Fold)

Figure 4.5 Confusion Matrix Creation window of melNET

5. RESULT ANALYSIS

In this work, an analysis of the generated ROC curves has been carried out to evaluate the performance of melNET as a binary classier. melNET has been trained and tested with images from both digital and dermoscopic datasets for several conditions. For each observation, two population sets, one with melanoma and the other without melanoma, have been categorized. The result for this work has been analyzed by evaluating the sensitivity value of an observation. A higher sensitivity value indicates a lower number of false negatives. For every set of observations, the observation with the highest sensitivity value while keeping the specificity and accuracy above 80.00%, has been chosen as the best one.

For analyzing an observation, a ROC curve has been generated using the average result of all five folds from that observation. So, the overall performance of an observation has been represented by the corresponding ROC curve. ROC curves have been plotted for all the observations in this work. Those ROC curves can be found in appendix A of chapter seven. From those ROC curves, it has been observed that the density of the plotted points becomes higher near the region of threshold value 0.75. As a result, it has been approximated to use 0.75 as an optimal threshold for result evaluation.

Table 1 demonstrates the performance of melNET on the digital dataset by incrementing the number of iterations by 1000 in a range of 1000 to 10000. The learning rate and batch size for this set of observations have been kept as 0.01 and 100 respectively.

TABLE 1. Average of five-fold cross validation data for first ten observations on digital dataset while keeping the batch size and learning rate as 100 and 0.01 respectively.

Dataset: UMCG, Batch Size: 100, Learning Rate: 0.01, Threshold: 0.75					
Obs. Num.	*Iterations*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
01	1000	83.06%	82.61%	82.79%	90.84%
02	2000	84.34%	81.07%	82.42%	90.86%
03	3000	83.52%	82.00%	82.63%	90.91%
04	4000	82.91%	82.64%	82.75%	90.86%
05	5000	84.13%	80.89%	82.23%	90.55%
06	6000	85.00%	80.00%	82.06%	90.35%
07	7000	83.11%	81.86%	82.37%	90.40%
08	8000	83.67%	80.75%	81.95%	90.08%
09	9000	81.58%	80.64%	81.03%	89.21%
10	10000	79.29%	82.61%	81.24%	89.12%

The first ten observations were taken place using the digital dataset. After analyzing these observations from table 1, it can be stated that, the model showed the best result on the sixth observation by executing 6000 iterations while keeping the learning rate and batch size as 0.01 and 100 respectively. This statement has been made by judging the value of sensitivity for this set of observations. For this observation, the sensitivity value reaches to 85.00%, which has been found as the highest while keeping the specificity and accuracy as 80.00% and 82.06% respectively.

Table 2 demonstrates the performance of melNET on the same digital dataset by incrementing the number of iterations by 2000 in a range of 2000 to 10000. The learning rate and batch size for this set of observations have been kept as 0.005 and 100 respectively.

TABLE 2. Average of five-fold cross validation data for observations on digital dataset while keeping the batch size and learning rate as 100 and 0.005 respectively.

Dataset: UMCG, Batch Size: 100, Learning Rate: 0.005, Threshold: 0.75					
Obs. Num.	*Iterations*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
11	2000	78.78%	85.68%	82.84%	90.95%
12	4000	82.35%	82.43%	82.40%	90.90%
13	6000	83.83%	82.82%	83.24%	91.08%
14	8000	83.27%	82.61%	82.88%	90.80%
15	10000	83.83%	83.25%	83.49%	91.68%

This next batch of five observations has been made for observing the influence of a lower learning rate on the performance of the model while operating on the digital dataset. A lower learning rate value of 0.005 has been applied for undertaking these observations. The batch size has been kept as 100 like before. Table 2 demonstrates that, the fifteenth observation with 10000 iterations shows the best performance with a sensitivity of 83.83%, specificity of 83.25% and an accuracy of 83.49%.

Next set of observations has been done on the dermoscopic dataset. The model has been evaluated by incrementing the number of iterations by 2000 in the range of 2000 to 10000. For this set of five observations, the learning rate and batch size have been kept as 0.01 and 100. Table 3 demonstrates the performance of melNET for this set of observations.

TABLE 3. Average of five-fold cross validation data for observations on dermoscopic dataset while keeping the batch size and learning rate as 100 and 0.01 respectively.

Dataset: PH2, Batch Size: 100, Learning Rate: 0.01, Threshold: 0.75					
Obs. Num.	*Iterations*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
16	2000	93.75%	89.82%	90.61%	96.77%
17	4000	94.46%	88.75%	89.89%	96.82%
18	6000	92.50%	82.43%	84.45%	94.12%
19	8000	95.00%	86.94%	88.55%	96.72%
20	10000	94.82%	79.26%	82.38%	93.45%

Table 3 shows the nineteenth observation with 8000 iterations as the best one among this entire set. This observation results a sensitivity of 95.00%, specificity of 86.94% and an accuracy of 88.55%.

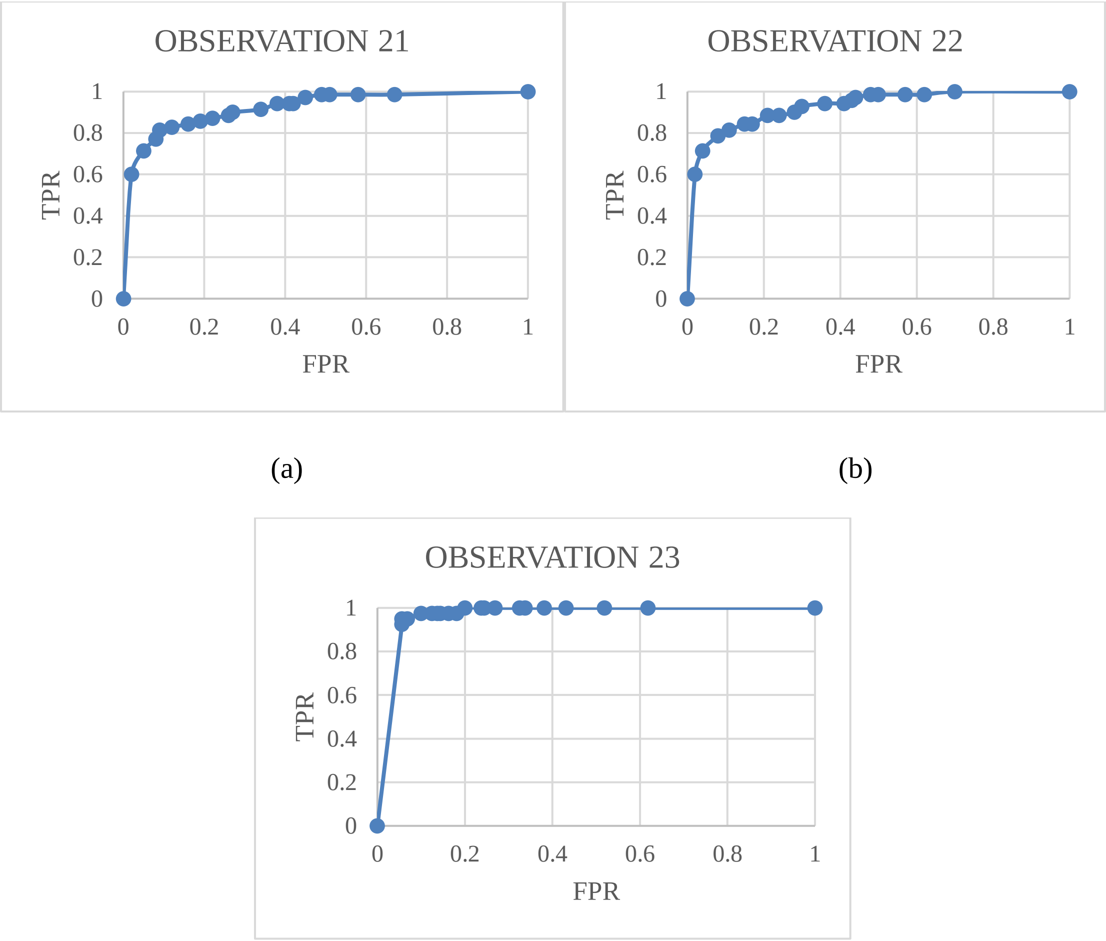
After completing the three sets of observations, melNET has been compared against the contemporary models of melanoma detection. For this comparison, the training has been done on the augmented five-fold dataset utilizing the hyperparameter values found best on the above three sets of observations. As the comparison needs to be legit, the testing has been done only on the un-augmented dataset without removing the images previously found as blurry. Table 4 demonstrates the performance of melNET utilizing the three sets of hyperparameters, which are observed as the best ones, in order to compare it with the MED-NODE model [33] and YOLO-v2 model [26].

TABLE 4. Average of five-fold cross validation data, keeping the threshold 0.75, for comparing melNET with MED-NODE and YOLO-v2 models using the corresponding datasets.

Obs. Num.	*Dataset*	*Iterations*	*Batch Size*	*Learning Rate*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
21	Digital (UMCG)	6000	100	0.01	82.86%	88.00%	85.88%	92.08%
22	Digital (UMCG)	10000	100	0.005	84.29%	85.00%	84.71%	92.68%
23	Dermoscopic (PH2)	8000	100	0.01	97.50%	87.50%	89.50%	96.57%

From the 23rd observation of table 4, it can be stated that, melNET outperformed the work in [26], which utilizes the same dermoscopic dataset, by improving the sensitivity value from 86.35% to 97.50%. This can be considered as a development on minimizing the number of false negative detections. Also, the specificity and accuracy values are found to be improved from 85.90% to 87.50% and from 86.00% to 89.50% respectively. ROC curve is illustrated in figure 5.1 (c).

With the same digital dataset, melNET outperformed the work of the MED-NODE model [33] by improving the accuracy from 81.00% to 84.71%, as shown in the 22nd observation in table 4. ROC curves for 21st and 22nd observations are shown in figure 5.1(a) and 5.1(b) respectively.



(a)

(b)

(c)

Figure 5.1 Average ROC curves generated by using: (a) digital dataset, learning rate of 0.01, batch size of 100, and, 6000 iterations; (b) digital dataset, learning rate of 0.005, batch size of 100, and, 10000 iterations; (c) dermoscopic dataset, learning rate of 0.01, batch size of 100, and, 8000 iterations

In addition, melNET has been found to perform the detections in real time by using the end-to-end architecture of Inception-v3 model.

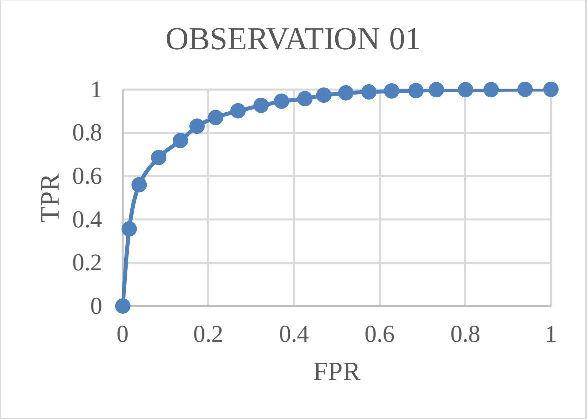
6. CONCLUSION

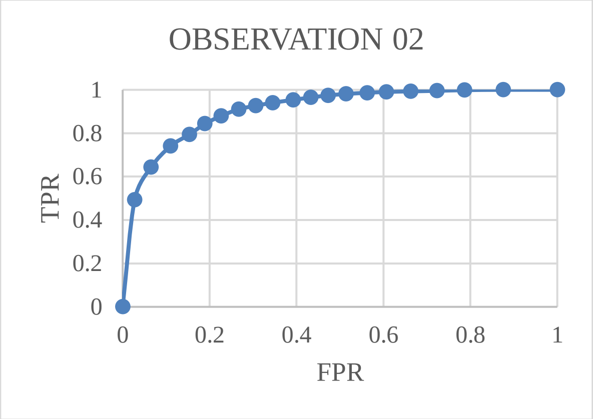
The proposed work was aimed to develop a model for detecting melanoma in both digital and dermoscopic images by using a deep learning model. By the end of this work, a model, named “melNET”, has been developed. melNET uses the Inception-v3 network as the architecture for the deep learning model. An available dataset of dermoscopic images, provided by PH2, has been utilized to train and evaluate the performance of melNET on the dermoscopic data. A digital dataset, provided by UMCG, has also been used to train and analyze melNET’s performance on the digital data. Every image from the datasets was annotated according to the clinical classification of the mole. For making the system robust, the data was augmented by operating several augmentation methods. During the training phase, melNET retrained the Inception-v3 network by feeding the errors from each iteration, resulting in the fine tuning of the network weights. Then, the trained system was evaluated by using a five-fold cross validation technique.

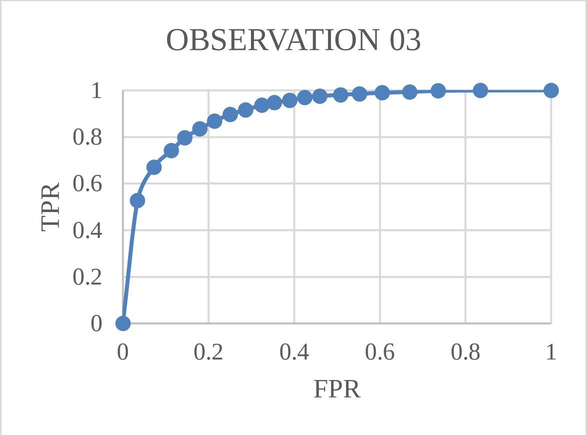
With the dermoscopic dataset, melNET has outperformed the work [26] with the YOLO-v2 network by improving the sensitivity value from 86.35% to 97.50%. This can be considered as a development on minimizing the number of false negative detections. Also, the specificity and accuracy values are found to be improved from 85.90% to 87.50%, and, from 86.00% to 89.50% respectively. With the digital dataset, melNET has outperformed the work of the MED-NODE model [33] by improving the accuracy from 81.00% to 84.71%. It has also been found that, as melNET uses an end-to-end Inception-v3 architecture, it can perform the detections in real-time.

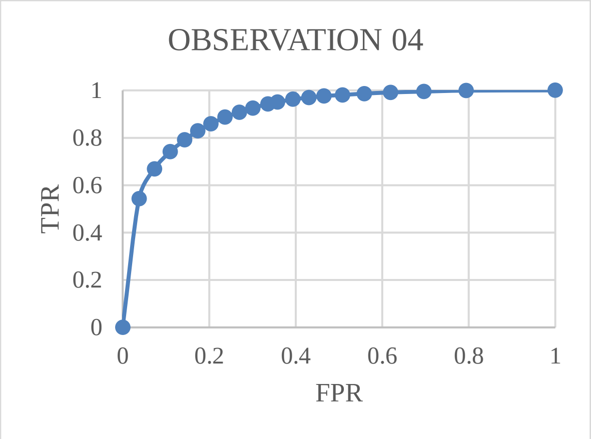
7. APPENDICES

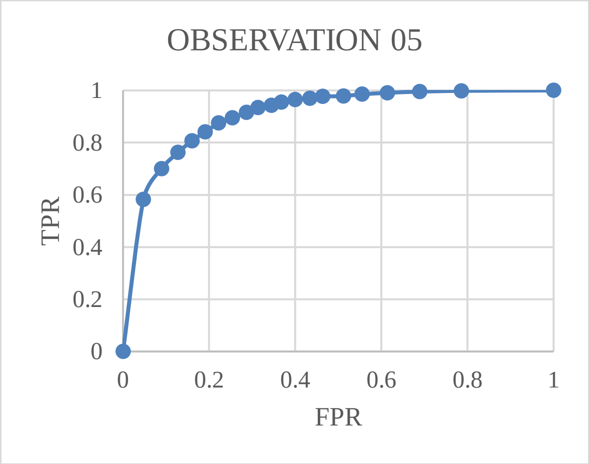
7.1 Appendix A

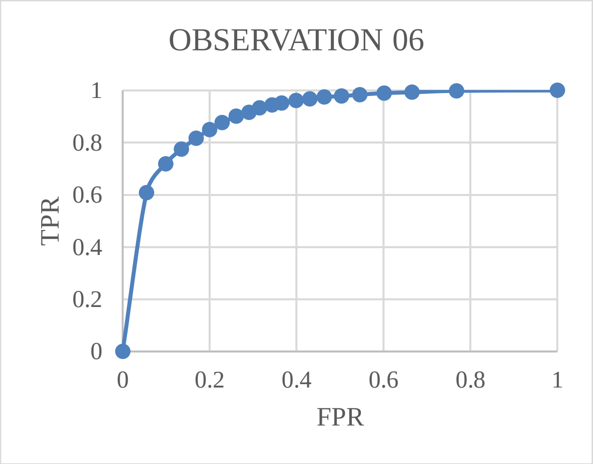


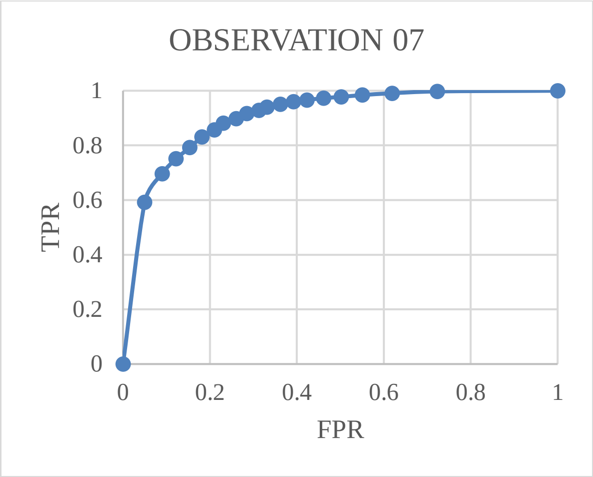


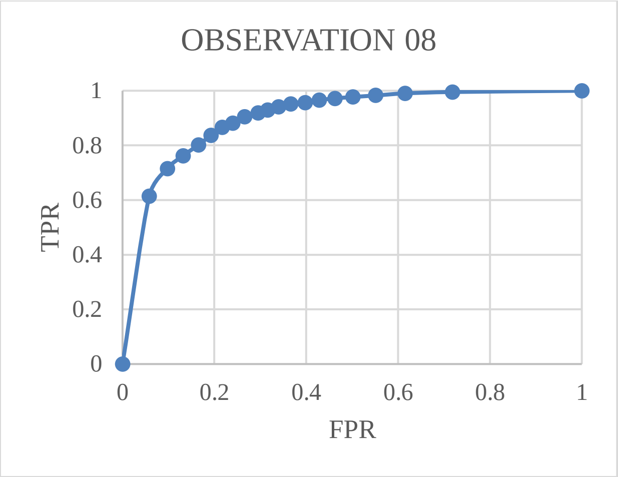


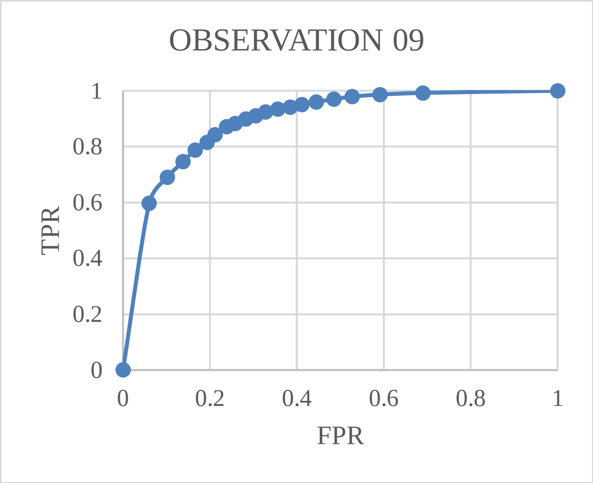


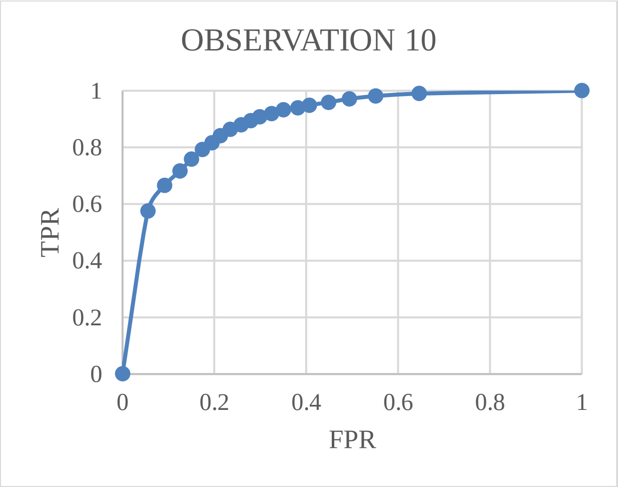


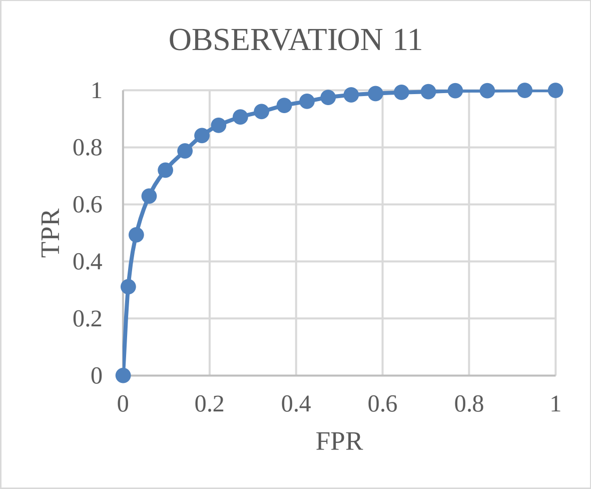


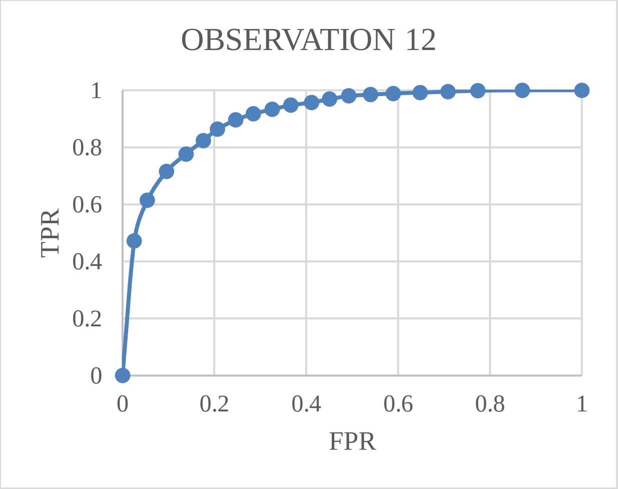

OBSERVATION 07
TPR
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
FPR

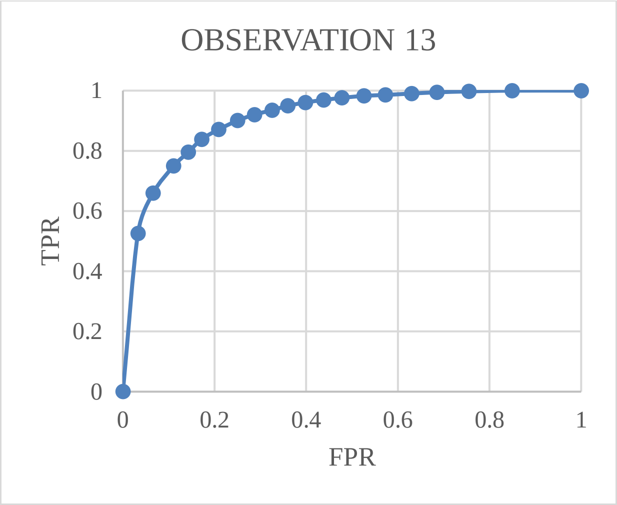

OBSERVATION 08
TPR
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
FPR

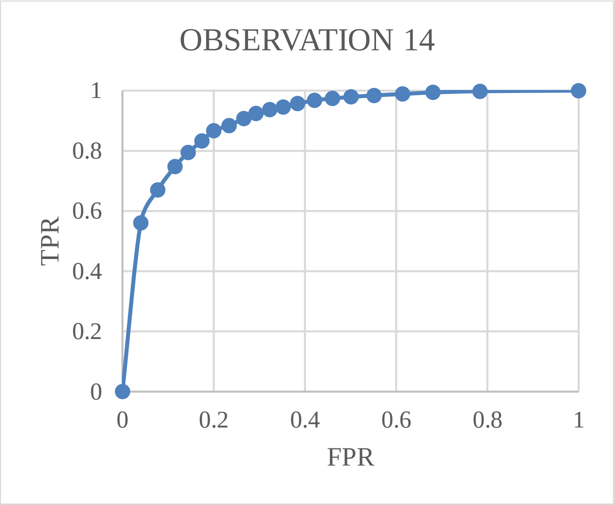

OBSERVATION 09
TPR
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
FPR

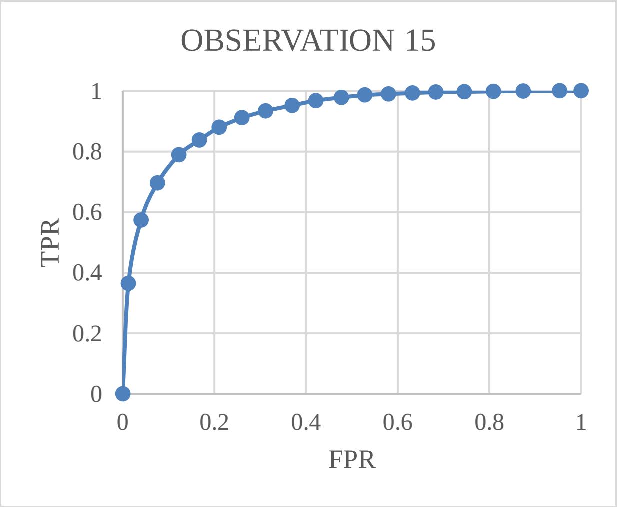

OBSERVATION 10
TPR
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
FPR

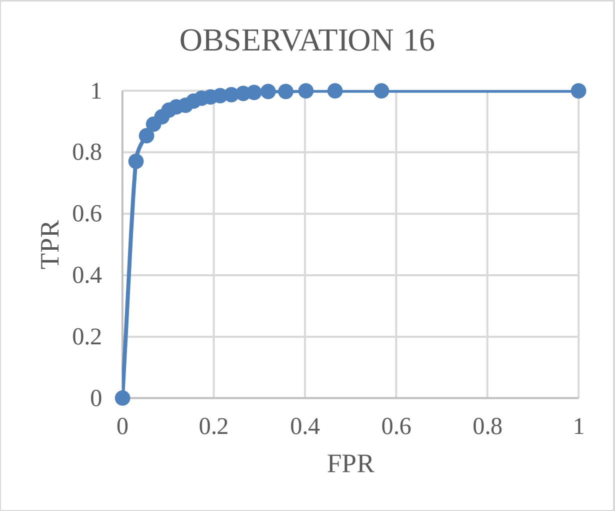

OBSERVATION 11
TPR
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
FPR

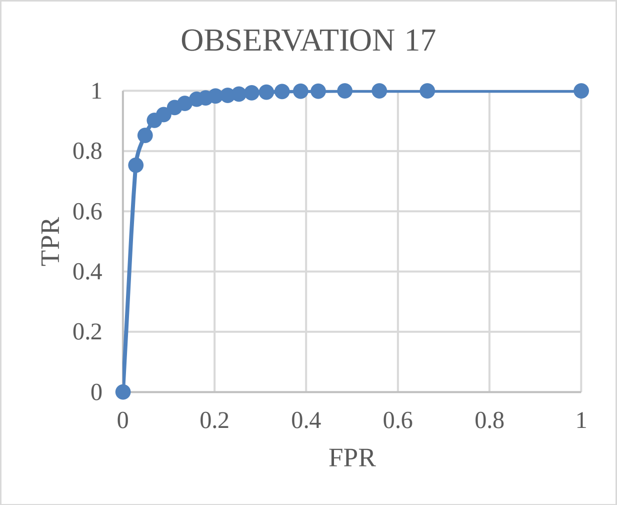

OBSERVATION 12
TPR
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
FPR

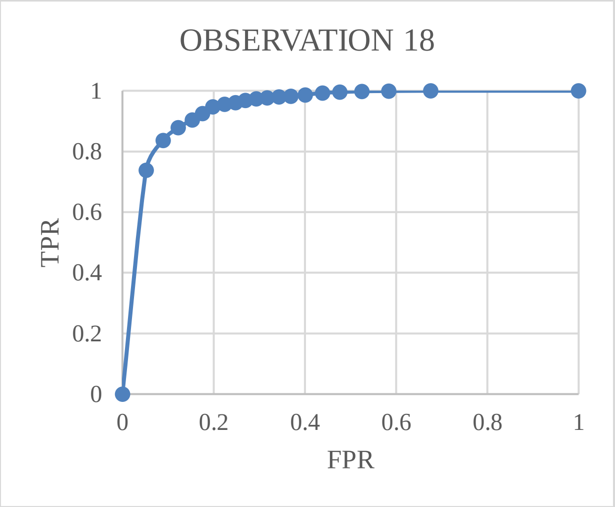

OBSERVATION 13
TPR
FPR
0
0.2
0.4
0.6
0.8
1


OBSERVATION 14
TPR
FPR
0
0.2
0.4
0.6
0.8
1


OBSERVATION 15
TPR
FPR
0
0.2
0.4
0.6
0.8
1


OBSERVATION 16
TPR
FPR
0
0.2
0.4
0.6
0.8
1


OBSERVATION 17
TPR
FPR
0
0.2
0.4
0.6
0.8
1


OBSERVATION 18
TPR
FPR
0
0.2
0.4
0.6
0.8
1

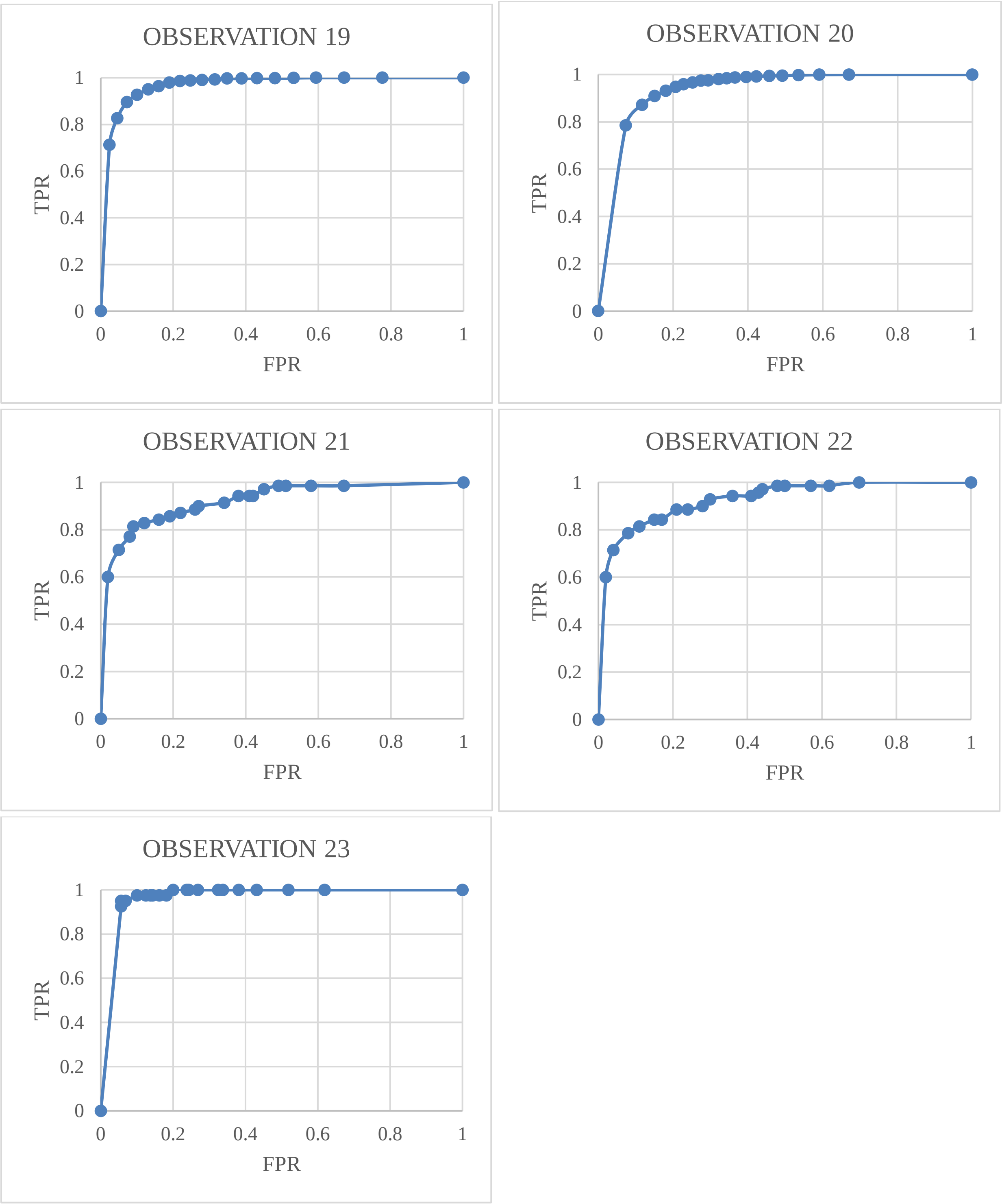


Figure 7.1 Average-ROC curves for all the observations

REFERENCES

[1] “Key Statistics for Melanoma Skin Cancer.” [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html. [Accessed: 01-Sep-2017].

[2] “Melanoma - SkinCancer.org.” [Online]. Available: http://www.skincancer.org/skin-cancer-information/melanoma. [Accessed: 01-Sep-2017].

[3] “WHO | Skin cancers,” *WHO*, 2017.

[4] S. Ranjan, R. Wordsworth, and D. D. Sasselov, “The Surface UV Environment on Planets Orbiting M Dwarfs: Implications for Prebiotic Chemistry and the Need for Experimental Follow-up,” *Astrophys. J.*, vol. 843, no. 2, p. 110, Jul. 2017.

[5] “How ultraviolet (UV) radiation causes skin cancer - Cancer Council NSW.” [Online]. Available: https://www.cancercouncil.com.au/63295/cancer-prevention/sun-protection/sun-protection-sport-and-recreation/sun-protection-information-for-sporting-groups/how-ultraviolet-uv-radiation-causes-skin-cancer/. [Accessed: 24-May-2019].

[6] “How do genes control the growth and division of cells? - Genetics Home Reference - NIH.” [Online]. Available: https://ghr.nlm.nih.gov/primer/howgeneswork/genesanddivision. [Accessed: 24-May-2019].

[7] Y. Yamaguchi and V. J. Hearing, “Melanocytes and their diseases.,” *Cold Spring Harb. Perspect. Med.*, vol. 4, no. 5, May 2014.

[8] “Stages of Melanoma - AIM at Melanoma Foundation.” [Online]. Available: https://www.aimatmelanoma.org/stages-of-melanoma/. [Accessed: 24-May-2019].

[9] “Treatment of Melanoma Skin Cancer, by Stage.” [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer/treating/by-stage.html. [Accessed: 24-May-2019].

[10] “Surgery for Melanoma Skin Cancer.” [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer/treating/surgery.html. [Accessed: 24-May-2019].

[11] N. K. Mishra and M. E. Celebi, “An Overview of Melanoma Detection in Dermoscopy Images Using Image Processing and Machine Learning,” 2016.

[12] O. B.-F. Stolz, W, A. Riemann, Armand B. Cognetta, “ABCD rule of dermatoscopy: a new practical method for early recognition of malignant melanoma,” *EJD. Eur. J. dermatology*, vol. 4, pp. 521–527, 1994.

[13] R. Garnavi, “Computer-aided Diagnosis of Melanoma Produced on archival quality paper,” 2011.

[14] A. G. Manousaki *et al.*, “A simple digital image processing system to aid in melanoma diagnosis in an everyday melanocytic skin lesion unit. A preliminary report,” *Int. J. Dermatol.*, vol. 45, no. 4, pp. 402–410, Apr. 2006.

[15] S. W. Menzies, C. Ingvar, K. A. Crotty, and W. H. McCarthy, “Frequency and morphologic characteristics of invasive melanomas lacking specific surface microscopic features.,” *Arch. Dermatol.*, vol. 132, no. 10, pp. 1178–82, 1996.

[16] G. Argenziano, G. Fabbrocini, P. Carli, V. De Giorgi, E. Sammarco, and M. Delfino, “Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions. Comparison of the ABCD rule of

dermatoscopy and a new 7-point checklist based on pattern analysis.," *Arch. Dermatol.*, vol. 134, no. 12, pp. 1563–1570, 1998.

[17] Y. Faziloglu, R. J. Stanley, R. H. Moss, W. Van Stoecker, and R. P. McLean, "Colour histogram analysis for melanoma discrimination in clinical images," *Ski. Res. Technol.*, 2003.

[18] M. Emre Celebi *et al.*, "Border detection in dermoscopy images using statistical region merging," *Ski. Res. Technol.*, vol. 14, no. 3, pp. 347–353, Aug. 2008.

[19] C. Barata, M. Ruela, M. Francisco, T. Mendonca, and J. S. Marques, "Two Systems for the Detection of Melanomas in Dermoscopy Images Using Texture and Color Features," *IEEE Syst. J.*, vol. 8, no. 3, pp. 965–979, Sep. 2014.

[20] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning."

[21] N. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, and J. R. Smith, "Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images," Springer, Cham, 2015, pp. 118–126.

[22] L. Yu, H. Chen, Q. Dou, J. Qin, and P.-A. Heng, "Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks," *IEEE Trans. Med. Imaging*, vol. 36, no. 4, pp. 994–1004, Apr. 2017.

[23] D. D. Gomez, C. Butakoff, B. K. Ersboll, and W. Stoecker, "Independent Histogram Pursuit for Segmentation of Skin Lesions," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 157–161, Jan. 2008.

[24] Y. Li and L. Shen, "Skin Lesion Analysis towards Melanoma Detection Using Deep Learning Network.," *Sensors (Basel).*, vol. 18, no. 2, Feb. 2018.

[25] A. Afonso and M. Silveira, "Hair detection in dermoscopic images using Percolation," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2012, pp. 4378–4381.

[26] S. S. Roy, A. U. Haque, and J. Neubert, "Automatic diagnosis of melanoma from dermoscopic image using real-time object detection," in *2018 52nd Annual Conference on Information Sciences and Systems, CISS 2018*, 2018.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015.

[28] C. M. and S. L. Cooclea, "'SpotMole Plus' (v3.9)." [Online]. Available: http://www.spotmole.com/.

[29] H. D. Corporation, "'MelApp' (vA1.0)," 2012. [Online]. Available: http://www.melapp.net/.

[30] J. A. Wolf *et al.*, "Diagnostic Inaccuracy of Smartphone Applications for Melanoma Detection," *JAMA Dermatology*, vol. 149, no. 4, p. 422, 2013.

[31] E. Zagrouba and W. Barhoumi, "A PRELIMARY APPROACH FOR THE AUTOMATED RECOGNITION OF MALIGNANT MELANOMA," *Image Anal. Stereol.*, 2011.

[32] J. Fernández Alcón *et al.*, "Automatic imaging system with decision support for inspection of pigmented skin lesions and melanoma diagnosis," *IEEE J. Sel. Top. Signal Process.*, 2009.

[33] I. Giotis, N. Molders, S. Land, M. Biehl, M. F. Jonkman, and N. Petkov, "MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images," *Expert Syst. Appl.*, vol. 42, no. 19, pp.

6578–6585, Nov. 2015.

[34] T. T. K. Munia, M. N. Alam, J. Neubert, and R. Fazel-Rezai, “Automatic diagnosis of melanoma using linear and nonlinear features from digital image,” in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 4281–4284.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Dec. 2015.

[36] S. Ruder, “An overview of gradient descent optimization algorithms *.”

[37] L. Deng, “A tutorial survey of architectures, algorithms, and applications for deep learning,” *APSIPA Trans. Signal Inf. Process.*, vol. 3, p. e2, Jan. 2014.

[38] Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen, “Deep Learning for Imbalanced Multimedia Data Classification,” in *2015 IEEE International Symposium on Multimedia (ISM)*, 2015, pp. 483–488.

[39] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features,” 1999.

[40] Y. Zhang, R. Jin, and Z.-H. Zhou, “Understanding Bag-of-Words Model: A Statistical Framework.”

[41] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection.”

[42] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *J. Big Data*, vol. 2, no. 1, p. 1, Dec. 2015.

[43] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[44] W. S. Mcculloch and W. Pitts, “A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY* n,” 1990.

[45] F. Rosenblatt, “THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN 1.”

[46] P. J. Werbos, “Backpropagation Through Time: What It Does and How to Do It.”

[47] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[48] M. I. Jordan, “Serial Order: A Parallel Distributed Processing Approach,” *Adv. Psychol.*, vol. 121, pp. 471–495, Jan. 1997.

[49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[50] D. Silver *et al.*, “Mastering the game of Go with deep neural networks and tree search,” 2016.

[51] C. Goller and A. Kuchler, “Learning task-dependent distributed representations by backpropagation through structure,” in *Proceedings of International Conference on Neural Networks (ICNN’96)*, vol. 1, pp. 347–352.

[52] R. Socher, C. Chiung, Y. Lin, A. Y. Ng, and C. D. Manning, “Parsing Natural Scenes and Natural Language with Recursive Neural Networks,” 2011.

[53] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” Jun. 2014.

[54] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to Construct Deep Recurrent Neural Networks,”

Dec. 2013.

[55] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks.”

[56] D. H. HUBEL and T. N. WIESEL, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex.,” *J. Physiol.*, vol. 160, no. 1, pp. 106–54, Jan. 1962.

[57] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[58] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” Springer, Cham, 2014, pp. 818–833.

[59] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning - whole book,” *Nature*, 2016.

[60] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “Imagenet,” *Adv. Neural Inf. Process. Syst. 25*, 2012.

[61] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[62] C. Szegedy *et al.*, “Going Deeper with Convolutions.”

[63] M. E. Vestergaard, P. Macaskill, P. E. Holt, and S. W. Menzies, “Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting,” *Br. J. Dermatol.*, p. ???-???, Jun. 2008.

[64] J. Mayer, “Systematic review of the diagnostic accuracy of dermatoscopy in detecting malignant melanoma,” *Medical Journal of Australia*. 1997.

[65] “ADDI - Automatic computer-based Diagnosis system for Dermoscopy Images.” [Online]. Available: https://www.fc.up.pt/addi/ph2 database.html. [Accessed: 01-Sep-2017].

[66] “Dermatology database used in MED-NODE.” [Online]. Available: http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/. [Accessed: 01-Sep-2017].

[67] S. Pertuz, D. Puig, and M. A. Garcia, “Analysis of focus measure operators for shape-from-focus,” *Pattern Recognit.*, vol. 46, no. 5, pp. 1415–1432, May 2013.

[68] A. Chakrabarti, T. Zickler, and W. T. Freeman, “Analyzing spatially-varying blur,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2512–2519.

[69] R. Ferzli and L. J. Karam, “A no-reference objective image sharpness metric based on the notion of Just Noticeable Blur (JNB),” *IEEE Trans. Image Process.*, 2009.

[70] J. L. Pech-Pacheco, G. Crist, J. Chamorro-Mart Nez, and & J. Fern Andez-Valdivia, “Diatom autofocusing in brightteld microscopy: a comparative study.”

[71] L. Perez and J. Wang, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” Dec. 2017.

[72] T.-T. Wong, “Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation,” *Pattern Recognit.*, vol. 48, no. 9, pp. 2839–2846, Sep. 2015.

[73] “Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2.” [Online]. Available: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced. [Accessed: 24-May-2019].

[74] P. J. Diggle and J. Serra, “Image Analysis and Mathematical Morphology.,” *Biometrics*, 2006.

[75] J. Fürnkranz *et al.*, “Mean Squared Error,” in *Encyclopedia of Machine Learning*, 2010.

[76] Y. N. Dauphin, H. De Vries, J. Chung, and Y. Bengio, “RMSProp and equilibrated adaptive learning rates for non-convex optimization,” *Cornell Univ. Libr.*, 2014.

[77] “ROC curve analysis with MedCalc.” [Online]. Available: https://www.medcalc.org/manual/roc-curves.php. [Accessed: 24-May-2019].