Project Synopsis

Skin Lesion Classification using Deep Learning

Submitted by

Abhishek C. Salian (Roll No. 32)
Gulam Nasir Shaikh (Roll No. 39)
Pragya Singh (Roll No. 49)
Shalaka Vaze (Roll No.60)

in partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

in
Electronics & Telecommunication Engineering

Under the Guidance of

Mr. Santosh Chapaneri

St. Francis Institute of Technology, Mumbai
University of Mumbai
2019-2020

ABSTRACT

Skin cancer is one of the major types of cancers. They can arise from various dermatologic disorders and can be classified into various types according to their texture, structure, color and other morphological features. Identifying lesions from skin images can be an important step in pre-diagnosis to aid the doctors and infer the medical condition of the patient. By far many of the projects have focussed on classifying only melanoma from a given set of skin lesion images. However, some types of skin ailments have a similar structure and morphological features to that of melanoma and such ailments have a considerable chance of getting wrongly diagnosed as melanoma. Our project is an attempt to classify different types of skin lesions(basal cell carcinoma, benign keratosis, dermatofibroma, vascular lesions, melanoma, and melanocytic nevi) with good accuracy. This might be able to give the patient a good idea of whether or not is there a need for medical attention and can avoid unnecessary panic/false alarms. We are using different deep learning architectures to achieve good accuracy. Considering the size of the dataset, we have opted for data augmentation for ease in training and to have good accuracy

CERTIFICATE

This is to certify that Shalaka Vaze, Pragya Singh, Gulam Nasir Shaikh and Abhishek C. Salian are the bonafide students of St. Francis Institute of Technology, Mumbai. They have successfully carried out the project(Stage-I) titled "Skin Lesion Classificaton using Deep learning" in partial fulfilment of the requirement of B. E. Degree in Electronics and Telecommunication Engineering of Mumbai University during the academic year 2019-2020. The work has not been presented elsewhere for the award of any other degree or diploma prior to this.

(Mr. Santosh Chapneri)
Internal Guide

Internal Examiner

External Examiner

(Dr. Gautam Shah)
EXTC HOD

(Dr. Sincy George)
Principal

ACKNOWLEDGEMENT

We are thankful to a number of individuals who have contributed towards our final year project and without their help; it would not have been possible. Firstly, we offer our sincere thanks to our project guide, Mr. Santosh Chapaneri for his constant and timely help and guidance throughout our preparation.

We are grateful to all project co-ordinators for their valuable inputs to our project. We are also grateful to the college authorities and the entire faculty for their support in providing us with the facilities required throughout this semester.

We are also highly grateful to Dr. Gautam A. Shah, Head of Department (EXTC), Principal, Dr. Sincy George, and Director Bro. Jose Thuruthiyil for providing the facilities, a conducive environment and encouragement.

Signatures of all the students in the group

(Abhishek C. Salian)

(Gulam Nasir Shaikh)

(Pragya Singh)

(Shalaka Vaze)

Contents

List of Figures vi

List of Tables vii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Scope of Project . . . 1
- 1.3 Organization of Project . . . 2

2 Literature Survey 3
- 2.1 Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images (2015) [1]: . . . 3
- 2.2 Image Classification of Melanoma, Nevus and Seborrhoeic Keratosis by Deep Neural Network Ensemble (2017) [2]: . . . 4
- 2.3 Incorporating the knowledge dermatologist to Deep convolution Neural Network (2017) [3]: . . . 6
- 2.4 Knowledge transfer for melanoma screening with deep learning overview (2017) [4]: . . . 8

3 Preliminaries 10
- 3.1 Artificial Neural Network . . . 10
- 3.2 Fully Connected Network . . . 11
- 3.3 Training . . . 11
- 3.4 Backpropagation . . . 12
- 3.5 Loss Functions . . . 12
- 3.6 Gradient Descent . . . 14
- 3.7 Learning rate . . . 15
- 3.8 Activation Functions . . . 15
- 3.9 Optimization . . . 16
- 3.10 Convolutional Neural Network . . . 19
- 3.11 Batch Normalization . . . 22
- 3.12 Dropout [6] . . . 23

4 Dataset Analysis 24
- 4.1 PH^2 Dataset: . . . 24
  - 4.1.1 Dermoscopic criteria [14] . . . 26
  - 4.1.2 Analysis of PH^2 Dataset . . . 27
- 4.2 HAM10000 Dataset: . . . 28

5 Architecture **31**
5.1 VGG-16 31
5.2 VGG-19 33
5.3 MobileNet 34

6 Conclusion **36**

Bibliography **37**

List of Figures

2.1 Classification system. 5
2.2 Basic Structure of model. 6
2.3 Diagnosis network. 8

3.1 Artificial Neural Network. 10
3.2 Fully connected neural network. 11
3.3 Neural network as a black box. 12
3.4 Backpropagation. 12
3.5 Hard decision from softmax probailities. 13
3.6 Schematic of Gradient Descent. 14
3.7 Consequence of high and low learning rate. 15
3.8 Sigmoid function. 16
3.9 ReLU function. 16
3.10 SGD convergence. 17
3.11 A example of CNN architecture. 19
3.12 Image matrix multiplies with kernel. 20
3.13 Convolution calculation on grayscale image. 20
3.14 Convolution operation on a RGB image matrix with a 3x3x3 Kernel. . . . 21
3.15 Pooling. 22
3.16 Dropout (Source: [8]). 23

4.1 An illustrative collection of images from PH^2 database, including common nevi (1st row), atypical nevi (2nd row) and melanomas (3rd row). 25
4.2 Manual segmentation of three melanocytic lesions: common nevus (left), atypical nevus (middle) and melanoma (right). 25
4.3 Original image (left), blue-gray (middle) and darkbrown region (right). . . 26
4.4 Dermoscopic features identification. 27
4.5 PH^2 Dataset Analysis. 27
4.6 Manual correction of a scanned diapositive. Original scanned image(left)with remaining black border on the lower left, lesion off center, yellow hue and reduced luminance. On the right, the final image after manual quality review. 28
4.7 Disease classification within dermoscopic images. 29

5.1 VGG-16 architecture 31
5.2 Comparison between VGG-16 and VGG-19 architecture 33
5.3 Modified model(sequential-3) with VGG16 for PH^2 training 34
5.4 Modified model(sequential-2) with VGG19 for PH^2 training 34
5.5 Modified model of MobileNet for PH^2 training 35

List of Tables

2.1 Task (1) Melanoma vs Atypical and Benign lesions. 4
2.2 Task (2) Melanoma vs Atypical lesions. 4
2.3 Results of MM and SK classifiers. 5
2.4 ISIC with VGG-16 with transfer learning. 9
2.5 Comparison between VGG-M and VGG-16 and classes. 9
2.6 Comparison between malignant and benign. 9

4.1 Summary of publicly available dermatoscopic image datasets of PH^2. . . . 25
4.2 Summary of publicly available dermatoscopic image datasets of HAM10000 manually augmented. 28
4.3 Summary of publicly available dermatoscopic image datasets of HAM10000 without augmentation. 29

List of Abbreviations

MM	Malignant Melanoma
SK	Seborrohoeic Keratosis
SE	Sensitivity
SP	Specificity
Mel	Melanoma
ACC	Accuracy
FC	Fully Connected
SVM	Support Vector Machine
CE	Cross Entropy
SGD	Stochastic Gradient Descent
ADAM	Adaptive moment estimation
CNN	Convolutional Neural Network
HAM	Human Against Machine
FCN	Fully Convolutional Network
ISBI	International Symposium on Biomedical Imaging
AUC	Area Under Curve
ISIC	International Skin Imaging Collaboration
ANN	Artificial Neural Network
$J(w)$	Cost function
w	Parameter
ReLU	Rectified Linear Unit
C0	Non melanoma
C1	Melanoma
AT	Atypical
T	Typical
A	Absent
P	Present

Chapter 1

Introduction

Skin cancer is one of the major types of cancers and its incidence has been increasing over the past decades. They can arise from various dermatological disorders and can be classified into various types according to their texture, structure, color and other morphological features. There are 2 types of skin cancer called melanoma and non-melanoma. Melanoma is by far the most aggressive and deadly, perhaps the most universally known and most likely to spread to other parts of the body. There are higher chances of curing, if the cancer is detected in its early stages and removed, the cure rate can be about over 90% [9].

Skin cancer diagnosis is conducted using visual examination of the lesion and then the clinical analysis is conducted if there is a suspicion. Image-based machine learning and deep learning, in particular, have recently shown considerable accuracy in medical image classification.

1.1 Motivation

We want to build a robust and accurate deep learning model that will assist dermatologists in detecting skin cancer and will help to take necessary actions without much delay. Mostly the skin lesion classification is of binary type i.e Melanoma and Non-Melanoma, we are extending this to multiclass classification, since some classes of Non-melanoma may turn up into melanoma (eg: Basal cell carcinoma) which could be fatal for the patients. By feeding the trained deep learning models with appropriate labeled data, the doctor can know the type of lesion and decide whether it holds the potential to metastasize in the future or not. The intention of the outcome of this project is not to replace dermatologists but to assist dermatologists for more productive results, thereby saving more skin cancer patients.

1.2 Scope of Project

Identifying lesions from skin images can be an important step in pre-diagnosis to aid the doctors and infer the medical condition of the patient. The features can be extracted using CNN. This deep learning architecture can help us to avoid the step of manual feature extraction. Depending on the dataset, augmentation is performed to increase the data set size and accuracy.

1.3 Organization of Project

- Collecting datasets for skin lesion
- Literature survey
- Analysis of PH^2 dataset
- Pre-processing the dataset images and applying various standard deep learning architecture
- Applying deep learning architectures on an augmented dataset
- HAM10000 dataset analysis

Chapter 2

Literature Survey

2.1 Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images (2015) [1]:

Given melanoma recognition in the clinical setting has trended toward the use of pattern descriptions with analogies and expert experience, the work in the paper explores whether these same underlying principles could be used to improve the performance of automated approaches. Prior work toward automated melanoma recognition has followed classical computer vision approaches that extract hand coded low-level visual features, combined with some form of classifier training Application of deep learning strategies, which have been successful for the task of recognition in natural photographs, have been limited by the relatively small size of the datasets. The work in the mentioned paper combines the use of deep convolutional networks trained in the domain of natural photographs, in addition to specialized features learned via an efficient sparse coding algorithm to eliminate the need of large collections of annotated data to learn good features, and allowing the system to draw analogies.

Dataset Description:

The International Skin Imaging Collaboration (ISIC) dataset contains one of the largest collections of contact non-polarized dermoscopy images, complete with manual bounding boxes placed around the lesions for analysis, involving 334 images of melanoma and 144 images of atypical nevi, as well as 2146 clearly benign lesions (2624 total). Atypical nevi represent borderline cases: lesions that are not melanoma, but are visually similar to melanoma (as determined by expert analysis). Experiments of 2-fold cross-validation are performed 20 times (40 experiments total) for evaluation on this dataset. Two variants of the task are also performed: One task discriminating melanoma from both atypical and benign lesions (Table 2.1) (easier task), and one task discriminating melanoma from only atypical lesions (Table 2.2) (harder task).

The presented deep learning approach in the paper uses two parallel paths: 1) transfer of convolutional neural network features learned from the domain of natural photographs and 2) unsupervised feature learning, using sparse coding, within the domain of dermoscopy images. Classifiers are then subsequently trained for each using non-linear SVMs, and the models are then combined in late fusion (score averaging).

The Caffe convolutional neural network (CNN) was used for transfer learning. This pre-trained model includes 5 convolutional layers, 2 fully connected layers, and a final 1000 dimensional concept detector layer. In this work, the concept detector layer of this model (1000 dimensions, referred to as "FC8"), as well as the first fully connected layer (4096 dimensions, referred to as "FC6"), are used as visual descriptors for dermoscopy images.

Two dictionaries are constructed in colour (RGB) and grayscale colour spaces. Images are rescaled to 128×128 pixel dimensions before extraction of 8×8 patches, to learn dictionaries of 1024 elements. λ values of 0.15 and 1000 iterations (recommended defaults in the SPAMS implementation) were used for minimization of the objective function. To train melanoma classifiers from various deep features under study, a non-linear SVM using a histogram intersection kernel and sigmoid feature normalization was employed. SVM scores were mapped to probabilities using logistic regression on training data. A probability of 50% is used as the binary classification threshold. Fusion is done by un-weighted SVM score averaging (late fusion).

Table 2.1: Task (1) Melanoma vs Atypical and Benign lesions.

	Hand Coded		Caffe CNN		Sparse Coding			Fusions	
	Ensemble	4K FC6	1K FC8	Fusion	GRAY	RGB	Fusion	Deep	All
ACC	0.912	0.919	0.853	0.910	0.825	0.903	0.907	0.923	0.931
SEN	0.930	0.903	0.805	0.893	0.823	0.885	0.905	0.925	0.949
SPE	0.910	0.921	0.860	0.912	0.825	0.906	0.907	0.923	0.928

Table 2.2: Task (2) Melanoma vs Atypical lesions.

	Hand Coded		Caffe CNN		Sparse Coding			Fusions	
	Ensemble	4K FC6	1K FC8	Fusion	GRAY	RGB	Fusion	Deep	All
ACC	0.715	0.723	0.654	0.725	0.651	0.681	0.695	0.728	0.739
SEN	0.727	0.724	0.664	0.725	0.643	0.685	0.691	0.728	0.738
SPE	0.689	0.722	0.632	0.723	0.670	0.673	0.706	0.729	0.743

2.2 Image Classification of Melanoma, Nevus and Seborrhoeic Keratosis by Deep Neural Network Ensemble (2017) [2]:

Task: To classify skin lesion images into three classes – melanoma (MM malignant melanoma), nevus (NCN; nevocellular nevus) and seborrheic keratosis (SK) through two binary classifiers – MM vs. rest (MM classifier) and SK vs. rest (SK classifier). The mentioned paper makes use of external training data and the use of the age/sex information tagged with a number of the provided samples.

The luminance and colour balance of input images are normalized exploiting colour constancy. Normalized images are input to a base classifier trained for SK vs. rest as well as to a base classifier trained for MM vs. rest. Both base classifiers have identical

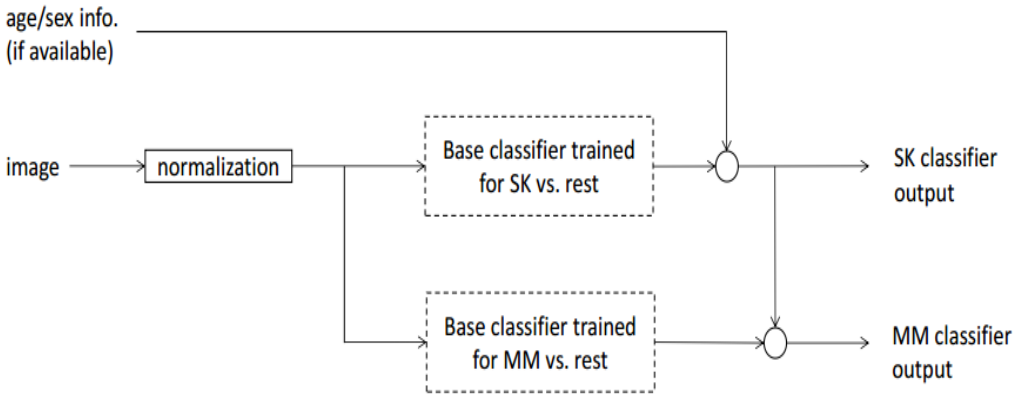


Figure 2.1: Classification system.

composition. Geometrically transformed images (combinations of rotation, translation, scaling and flipping) are input in parallel to an ensemble of convolutional neural networks (CNNs) and a prediction value in [0.0, 1.0] is output. Adoption of a 50-layer ResNet implemented in Keras with small modifications was used as the architecture. A straightforward thresholding was adopted by age/sex information only for SK classification. For MM classification, it was observed no significant increase by cross-validation. In addition, it was noticed that SK classifier was far more reliable than MM classifier. Ad-hoc linear approximation was applied. In addition to the provided training data (374 MM, 254 SK, 1372 NCN samples), usage of external training data (409 MM, 66 SK, 969 NCN samples) from the subset of the ISIC Archive was made. CNNs were fine-tuned with the training samples from the initial pre-trained model for generic object recognition in Keras. They applied different types of optimization and selected the best combination of fine-tuned CNNs through cross-validations. The optimization methods used were RMSProp and AdaGrad.

Table 2.3: Results of MM and SK classifiers.

	Proposed
MM classifier AUC	0.924
SK classifier AUC	0.993
Mean	0.958

2.3 Incorporating the knowledge dermatologist to Deep convolution Neural Network (2017) [3]:

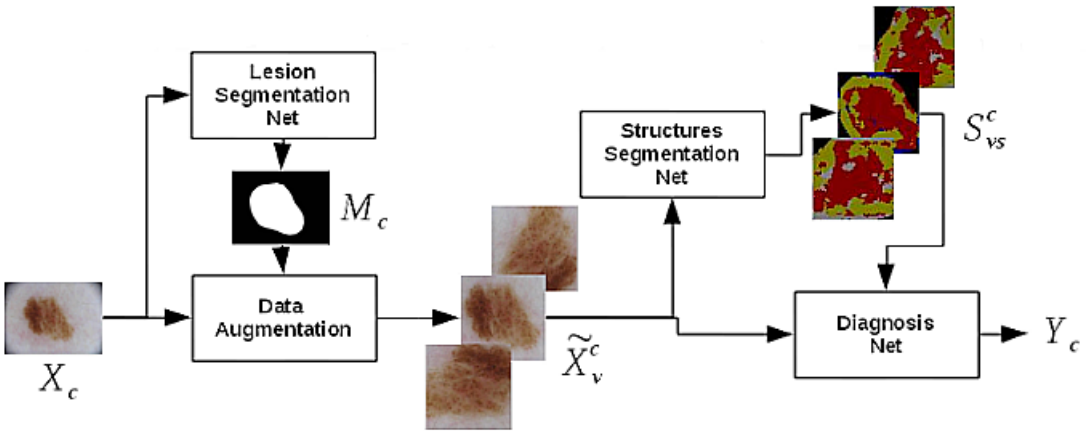


Figure 2.2: Basic Structure of model.

Overview:

This paper consists of segmentation network which includes lesion segmentation network, data augmentation, structure segmentation network and diagnosis network each has their specific function. The Lesion Segmentation Network has been developed by learning a Fully Convolutional Network (FCN). In order to train a network for our particular task of lesion/skin segmentation, they have used the training set for the lesion segmentation task in the 2017 ISBI challenge the idea is to broadly identify the area of the image that corresponds to the lesion, giving place to a binary map M_c for each clinical case.

It is well known that data augmentation notably boosts the performance of deep neural networks, and which will be given to the structure segmentation network. It aims to segment each view of the lesion into eight sets of global and local structure. Finally, the image map is given to diagnosis network where it provides final diagnosis.

- Lesion segmentation network: The lesion segmentation network consists of fully connected network which helped to achieved semantic image segmentation to generate accurate segmentation map of the lesion and to identify the lesion that corresponds in the image.

- Data augmentation and normalized polar coordinated: To boost the performance of deep neural network data augmentation is used when the amount of training data is limited. Among all the potential image variations and artifacts, invariance to orientation is the main requirement in this method. The steps involved are as follows:

  i. First, from the pair X_c, M_c there is generation of a set of rotations.

  ii. As rotating an image without losing any visual information which consist of new areas, in the new image. We need to crop the augmented image in order to retain the original image size.

iii. The next step is structure segmentation which involves cropping of the image to the size of 256×256 blocks which is the required input to the subsequent CNNs.

iv. The extracted blocks are then resized to the original image dimension.

The need for normalized polar coordinate is to support subsequent processing block by providing invariance against shift, rotation, change in size and even irregular shapes of the lesion. There is a need to transform Cartesian coordinates of the pixel $(X_i,\ Y_i)$ into normalized polar coordinates (ρ_i, θ_i) where $\rho_i \in [0, 1]$ and $\theta_i \in [0, 2\pi]$. The mask of the lesion is approximated by an ellipse with 2^{nd} order moments and then the affine matrix that transforms the ellipse into a normalized circle centered at (0, 0) forms the new rotated and cropped view of the lesion.

- Structure segmentation network: The goal of this module is, given an input view of the lesion X, to provide a corresponding segmentation into a pre-defined set of textural patterns and local structures that are of special interest for dermatologists in their diagnosis. In particular, there is a set of eight structures: (i) Dots, globules and cobblestone pattern (ii) Reticular patterns and pigmented Network (iii) Homogenous areas (iv) Regression areas (v) Blue-whitish veil (vi) Streaks (vii) Vascular structure (viii) Unspecified pattern

  The labels are assigned for each structure: 0 - if the structure is absent, 1 - if locally present, 2 - if it is present large enough to be considered a global pattern. The output of this network is a reduced version of the input image (64×64) where, for each pixel location X_i, softmax is used to transform the net outputs into probabilities as follows:

$$P_i(X_i|\theta) = \frac{1}{Z_i} \times \exp(f_i(X_i|\theta)) \tag{2.1}$$

  where θ represents the parameters of the CNN.

$$Z_i = \sum_{s=1}^{8} \exp(f_i(s|\theta)) \tag{2.2}$$

  Eq. (2.2) is partition function at the location i. The presence or absence of a class, as well as, an estimate of its size in the image, lead to particular constraints over the probability accumulated over all pixel locations in the segmentation map given by Eq. (2.3)

$$P_s = \sum_i P_i(s|\theta) \tag{2.3}$$

  Interpretation: (i) If a structure s is not present in an image, the constraint acts as an upper bound over the accumulated probability P_s, which has to be nearly zero. (ii) If a structure s is local in an image, we impose a lower and upper bound on the accumulated probability P_s in the image to control the total area of the structure in the lesion. (iii) If a structure s is global in an image, we impose a lower bound on the accumulated probability P, in the image to ensure a minimum area corresponding to the structure.

- Diagnosis network:

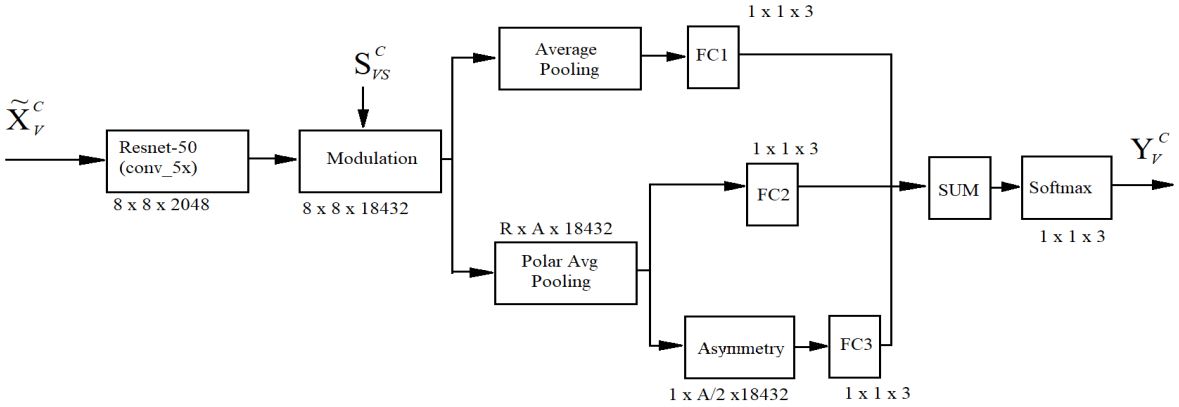


Figure 2.3: Diagnosis network.

The diagnosis network gathers the information from previous modules for diagnosis. This network consists of resnet-50 which uses residual layers to avoid the degradation problem when more layers are stacked to the network. Subdividing the top fully-connected layer providing the lesion diagnosis into three arms: (a) The original image with an average pooling followed by a fully connected layer (FC1), (b) A second step that performs a normalized polar pooling (3×6) by a fully connected layer (FC2), (c) A third step aims, that estimates the asymmetry of lesion based on the previous polar pooling and applies then a Fully Connected layer (FC3). The results of the three arms are then linearly combined using a sum block.

2.4 Knowledge transfer for melanoma screening with deep learning overview (2017) [4]:

The datasets used are Atlas, ISIC, Retinopathy, ImageNet on which image classification, augmentation and transfer learning is implemented. The deep learning architecture used here are ResNet, DRN-101, AlexNet and VGG-16. With the help of fine tuning they reached accuracy (AUC) 80.7% and 84.05% for two different datasets.

Dataset desciption:

The dataset employed to train and test the models is taken from Interactive Atlas of dermoscopy and ISIC challenge 2016. The Atlas is a multimedia guide designed for training medical personnel to diagnose skin lesion. This consists of 1000+ clinical data and most images are 768×512. The dataset consists of Melanoma, Basal cell carcinoma, Blue nevus, Clark's nevus, combined nevus, Congenital nevus, Dermal nevus, Dermatofibroma, Lentigo, Melanosis, Recurrent nevus, Reed nevus, Seborrheic keratosis, and Vascular lesion. Some images were cropped automatically with ImageMagick. The ISIC challenge 2016 consists of 1279 dermoscopy image. The

challenge dataset consists of 900 images for training (273 melanoma) and 379 for testing (115 for melanoma).

Implementation:

The dataset was augmented as it helped to balance the classes. To balance the classes they had only augmented the minority classes i.e. melanoma, malignant and basal cell carcinoma. Initially the weights of the network were assigned as an orthogonal matrix while their biases were initialized as 0.05. The input size of image was resized to 224×224 pixels which was required for VGG. Re-center was required by VGG architecture which was accomplished by subtracting mean of training dataset. Training was done with the deep CNN model which was fine-tuned along with freezing the last few layers. Additional softmax layer was added as the last layer to make decision on the skin lesion.

Table 2.4: ISIC with VGG-16 with transfer learning.

	AUC	mAP	ACC	SE	SP
1^{st} place	80.4	63.7	85.5	50.7	94.1
2^{nd} place	80.2	61.9	83.1	57.3	87.2
3^{rd} place	82.6	59.8	83.4	32.0	96.1
This paper	80.7	54.9	79.2	47.6	88.1

The learning rate was 10^{-3} with 60 epochs there are all tighter 19 layers with 4096-dimension vector the output of this 19^{th} layer is feed to SVM classifier Table 2.5.

Table 2.5: Comparison between VGG-M and VGG-16 and classes.

	AUC(%)		
Architecture	Mal×Ben	Mela×Ben	Mela×Carc×Ben
VGG-M	82.5	80.9	83.6
VGG-16	83.8	83.5	84.5

Table 2.6: Comparison between malignant and benign.

	AUC(%)			
Experimental Design	Low	Medium	High	All
Malignant *vs.* Benign	93.7	82.5	58.8	82.5
Melanoma *vs.* Benign	93.0	79.6	56.6	80.9

Low, Medium, High represents difficulties of test images images as All represents performance of whole dataset. Low, Medium, High – difficulty level 38.1%, 36.3%, 25.6% of whole dataset.

Chapter 3

Preliminaries

3.1 Artificial Neural Network

Artificial neural networks (ANN) are systems that are used to replicate human brains. A neural network consists of input, hidden and output layers, where the hidden layer transforms the inputs to obtain a particular output that we want. It can be considered as one of the tools which can be helpful to human to extract complex patterns and make the machine to make decisions on their own. The decision making capability can be improved by the self-learning process. Neural networks are also called Multilayer perceptrons which is one of the important parts of artificial intelligence. It is due to backpropagation which allows the networks to adjust their hidden layers in situations where output is not matching as per the expected results. For example, if a network is designed to identify a boy but it has recognized as a girl. Here multilayer network extracts different features until it can recognize as per the expected results.

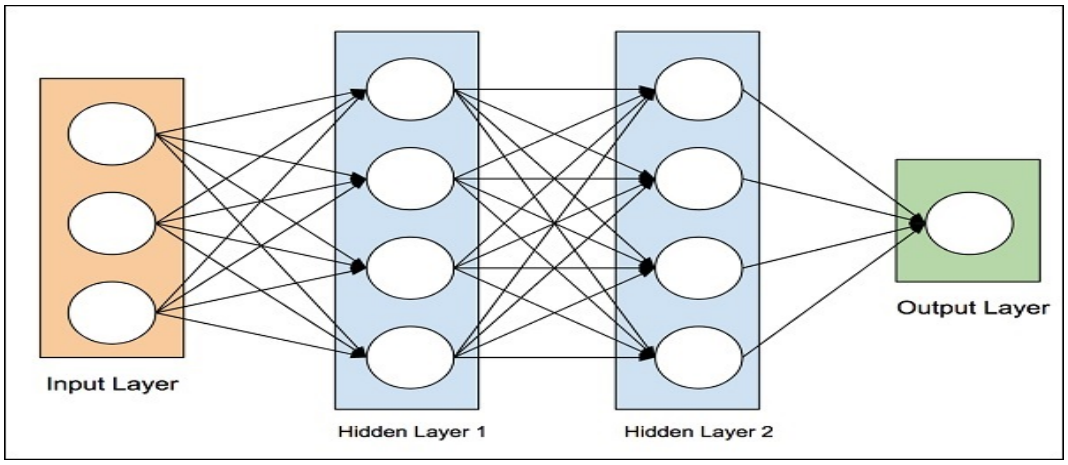


Figure 3.1: Artificial Neural Network.

Input Layer:

The input layer can be in the form of text, images, numbers, etc. Every input neuron represents a variable that has some influence over the output of the neural network.

Hidden Layer:

The hidden layer process the inputs which are obtained by the previous layer. It extracts the required feature from the input which has activation function applied to it. There can be multiple hidden layers in a Neural Network for complex feature extraction.

Output Layer:

The output layer of the neural network collects the output from the hidden layer and if the outcomes are not matched by the desired outputs then by using backpropagation and changing the hyperparameters the required outputs can be obtained.

3.2 Fully Connected Network

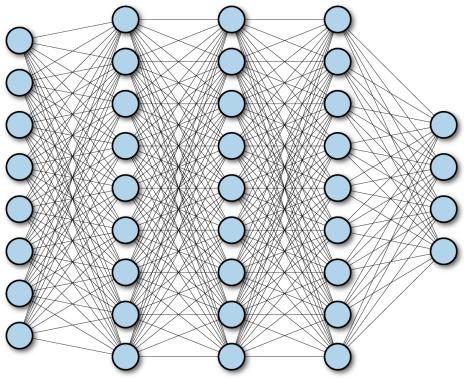


Figure 3.2: Fully connected neural network.

A fully connected neural network consists of a series of fully connected layers. A fully connected layer is a function from R_m to R_n . Each output dimension depends on each input dimension. Let $x \in R_m$ represent the input to a fully connected layer. Let y_i *in* R be the i_{th} output from the fully connected layer. Then y_i *in* R is computed as follows:

$$y_i = \sigma(x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots + x_m \cdot w_m) \tag{3.1}$$

Here, σ is a nonlinear function (σ as the sigmoid function), and the w_i are learnable parameters in the network. The full output y is given by:

$$y_i = \sigma(x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots + x_m \cdot w_m) + b \tag{3.2}$$

where b is the bias. The nodes in fully connected networks are commonly referred to as "neurons." Where every node is connected to every other nodes in the next stage.

3.3 Training

A supervised neural network can be presented as a black box with two methods such as learn and predict as following:

The learning process takes the input and as per the desired outputs it updates the parameter accordingly, so the output obtained is as close as possible from the desired

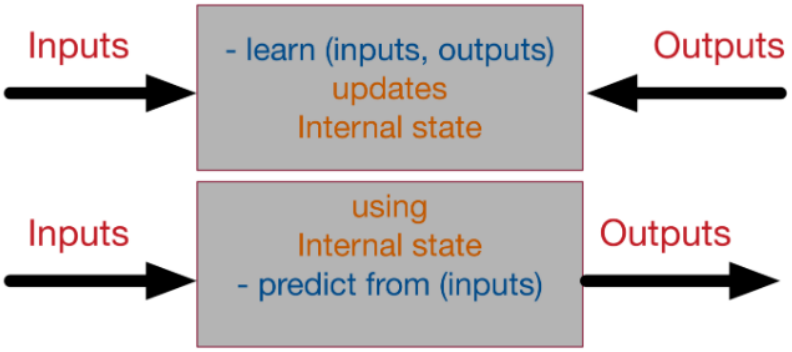


Figure 3.3: Neural network as a black box.

output. After the learning process is done the model is ready to predict the output as per its past training experience. In order to achieve training it is divided into several processes.

3.4 Backpropagation

Back-propagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e. iteration). Proper tuning of the weights allows to reduce error rates and to make the model reliable by increasing its generalization.

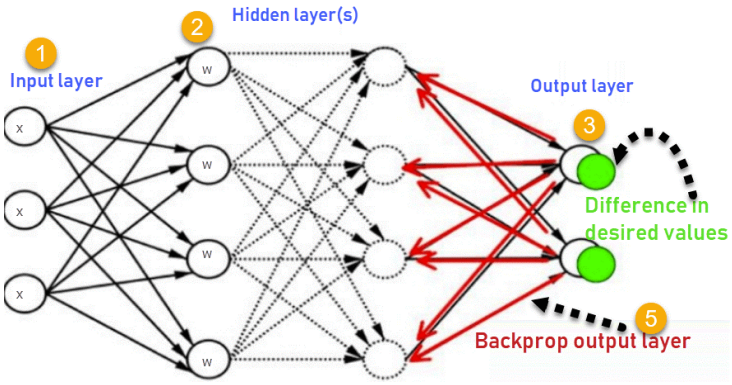


Figure 3.4: Backpropagation.

3.5 Loss Functions

Loss functions are used to evaluate how accurately our model has performed. If the prediction deviates more from the actual value, then the loss function will give high value. To produce a good prediction our model should produce low loss i.e. low deviation from an actual value. There are some optimization techniques to reduce loss such as gradient descent. We can't use loss function randomly because some loss function is sensitive which may produce some additional error. Hence it is necessary to know about loss function before using it in our model for calculating the loss in our prediction.

Difference between a Loss Function and a Cost Function:

The loss function is used for loss of single training example whereas cost function is used as the average loss of the entire training example. Hence optimization techniques are used to reduce the cost function.

Categorical Cross Entropy:

Categorical crossentropy will compare the distribution of the predictions with the true distribution, where the probability of the true class is set to 1 and 0 for the other classes. the true class is represented as a one-hot encoded vector, the closer the model's outputs are to that of the original vector, the lower the loss. Refer Eq. 3.3.

$$\mathcal{L}(y, \hat{y}) = -\sum_{j=1}^{M}\sum_{i=1}^{N}(y_{ij}\log(\hat{y}_{ij})) \tag{3.3}$$

where $\hat{y}$ is the predicted value. It is a Softmax activation and Cross-Entropy loss. using this loss, we can train a CNN to output a probability. It is used for multi-class classification. The CE Loss with Softmax activations is:

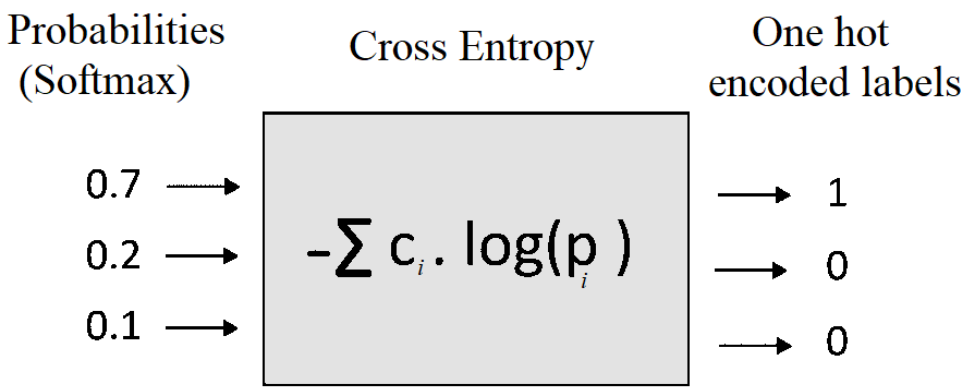


Figure 3.5: Hard decision from softmax probailities.

$$CategoricalLoss = \left(\frac{-1}{M}\right)\sum_{p=1}^{M}\log\left(\frac{e^{S_p}}{\sum_{j=1}^{C}e^{S_j}}\right) \tag{3.4}$$

Where each S_p in M is the CNN score for each positive class.

Binary Cross Entropy:

Binary crossentropy is a loss function used on problems involving yes/no (binary) decisions. For instance, in multi-label problems, where an example can belong to multiple classes at the same time, the model tries to decide for each class whether the example belongs to that class or not. Refer Eq. 3.5

$$\mathcal{L}(y, \hat{y}) = -\sum_{j=1}^{M}\sum_{i=1}^{N}(y_j \log(P(y_j)) - (1 - y_j)\log(1 - P(y_j))) \tag{3.5}$$

3.6 Gradient Descent

Gradient descent is an optimization algorithm that is used to minimize some function by continuously moving towards the minimal value of cost function. We use gradient descent to update the parameters of our machine learning model.

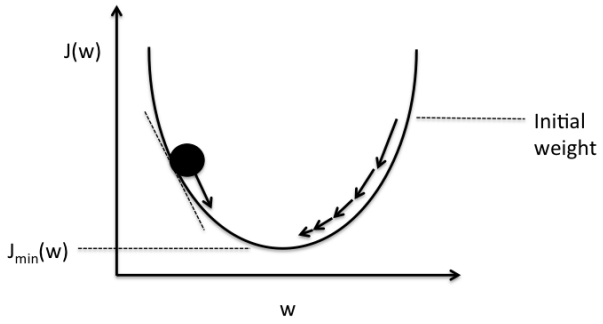


Figure 3.6: Schematic of Gradient Descent.

As shown in Fig. 3.6, error *J(w)* is a function of internal parameters of the model i.e. weights and bias. The current error is typically propagated backward to a previous layer, where it is used to modify the weights and bias in such a way that the error is minimized. The arrow is shown in Fig. 3.6 refers to the size of these steps is termed as the learning rate. Gradient descent is an iterative updating process continues until convergence, and the variable vector w achieved at convergence will be outputted as the (globally or locally) optimal variablefor the deep learning models. The pseudo-code of the vanilla gradient descent algorithm is available in Algorithm 1.

Algorithm 1 Gradient descent algorithm [5]

Require: Training set $\mathcal{T}$; Learning rate α; Normal distribution std: σ.
Ensure: Model parameter w
  Initialize parameter with Normal distribution $w \sim \mathcal{N}(0, \sigma^2)$
  Initialize convergence tag = *False*
  while tag == *False* **do**
    Compute gradient $\nabla_w J(w; \mathcal{T})$ on the training set $\mathcal{T}$
    Update variable $w = w$ - $\alpha \cdot \nabla_w\ J(w; \mathcal{T})$)
    if convergence condition holds **then**
      tag = *True*
    end if
  end while
  return model variable w

3.7 Learning rate

With a high learning rate shown in Fig. 3.7 (big learning rate), we can move towards minima faster in each step, but the risk here overshooting the minimal point because the slope changes continuously. With a very low learning rate (small learning rate), we can move in the direction of the negative gradient since we are changing the learning rate frequently. A low learning rate is more precise compared to a high learning rate, but the only disadvantage is it is time-consuming.

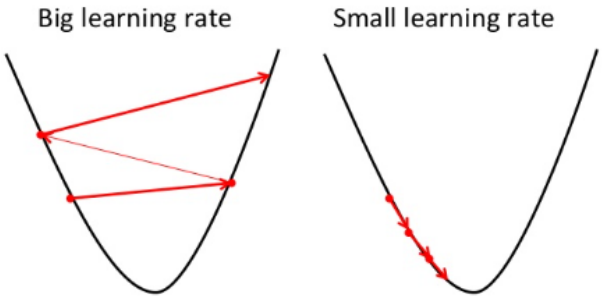


Figure 3.7: Consequence of high and low learning rate.

3.8 Activation Functions

Activation functions acts as a gate between input feeding and the output going to the next layer. It decides whether a neuron should be activated or not for the next layer. It does the non-linear task to the input so it can learn and perform more complex operations. As Activation function is a differentiable non linear function back propagation is possible as the gradient along with errors are passed to previous layers to update the weights and bias It is used to obtain the output in a range using different activation functions such as -1 to 1 or 0 to 1 depending upon the output needed.

Sigmoid:

Sigmoid function is a smooth function and can be continuously differentiable bounds the output in the range of 0 to 1. It is used for the models where the output is in the form of probabilities as probability lies in the range 0 to 1. The small change in input would bring a large change in output, so sigmoid pushes the output towards its extreme which is the desirable quality. Refer Fig. 3.8 and Eq. 3.6

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{3.6}$$

ReLU:

Rectified Linear Unit (ReLU) is computationally efficient, it allows the network to converge quickly. It looks like a linear function but it has derivative function which is useful for the back propagation. *R(z)* is zero when z is less than zero and *R(z)* is equal to z when z is above or equal to zero. All negative values become zero which in turn affects the result by not mapping negative values accurately. Refer Fig. 3.9 and Eq. 3.7

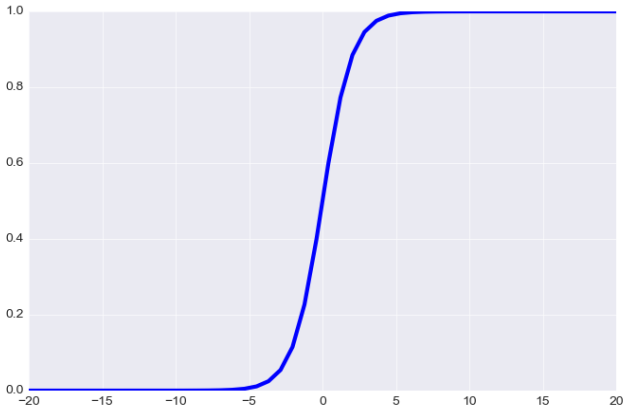


Figure 3.8: Sigmoid function.

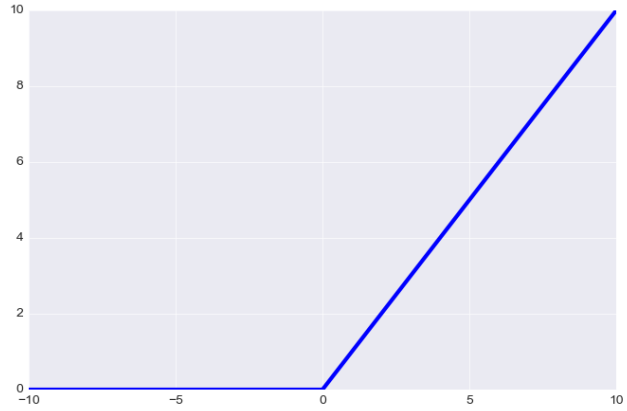


Figure 3.9: ReLU function.

$$R(z) = max(0, z) \tag{3.7}$$

Softmax:

Softmax is used to handle classification problem where there are more than 2 classes. It normalizes the output for each class in the range 0 to 1, and divides by their sum from which a probability is obtained which is used for the classification. For example consider [2.0, 1.0, 0.1], when we apply the softmax function we would get [0.7, 0.2, 0.2]. So now we can use these as probabilities for the value to be in each class. Refer Eq. 3.8

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \qquad for \;\; i, j = 1, 2, \ldots, K \tag{3.8}$$

3.9 Optimization

Optimization algorithms help to minimize the error function. The error function is a function that is dependent on the model's learning parameters from which the model

predicts the output. The learnable parameters of a model are responsible for effective training and to produce accurate results. To update these parameters different optimizers are used such as Adam, SGD, etc.

SGD:

Finding the gradient of the cost function at each iteration instead of the sum of the gradient of the cost function of all the examples.In SGD, since only one sample from the dataset is chosen at random for each iteration, the path taken by the algorithm to reach the minima is usually noisier than your typical Gradient Descent algorithm. One thing to be

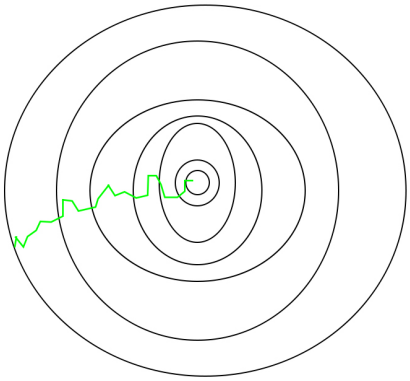


Figure 3.10: SGD convergence.

noted is that, as SGD is generally noisier than typical Gradient Descent, it usually took a higher number of iterations to reach the minima, because of its randomness in its descent. Even though it requires a higher number of iterations to reach the minima than typical Gradient Descent, it is still computationally much less expensive than typical Gradient Descent. Hence, in most scenarios, SGD is preferred over Batch Gradient Descent for optimizing a learning algorithm.

Stochastic Gradient Descent update and computes the gradient of the parameters using only a single or a few training examples

$$w := w - \alpha \cdot \nabla J(w; x^{(i)}, y^{(i)}) \tag{3.9}$$

with a pair ($x^{(i)}$,$y^{(i)}$) from the training set. Generally each parameter update in SGD is computed w.r.t a few training examples or a minibatch as opposed to a single example. The reason for this is : first this reduces the variance in the parameter update and can lead to more stable convergence, second this allows the computation to take advantage of highly optimized matrix operations that should be used in a well vectorized computation of the cost and gradient. A typical minibatch size is 256, although the optimal size of the minibatch can vary for different applications and architectures.

In SGD the learning rate α is typically much smaller than a corresponding learning rate in batch gradient descent because there is much more variance in the update.This tends to give good convergence to a local optima.

SGD is the order in which we present the data to the algorithm. If the data is given in some meaningful order, this can bias the gradient and lead to poor convergence. Generally a good method to avoid this is to randomly shuffle the data prior to each epoch of training.

Algorithm 2 SGD algorithm [5]

Require: Training set τ; Learning rate α; Normal distribution std: σ.
Ensure: Model parameter w

Initialize parameter with Normal distribution $\theta \sim \mathcal{N}(0, \sigma^2)$
Initialize convergence tag = *False*
while tag == *False* **do**
  Shuffle the training set τ
  for each data instance $(x_i, y_i) \in \tau$ **do**
    Compute gradient $\nabla_w J(w; (x_i, y_i))$ on the training instance (x_i, y_i)
    Update variable $w = w$ - $\alpha \cdot \nabla_w J(w; (x_i, y_i))$
  end for
  if convergence condition holds **then**
    tag = *True*
  end if
end while
return model variable w

ADAM:

Adam optimizer is an adaptive learning rate optimizer. Adam is composed of RMSprop and Stochastic Gradient Descent with momentum. It uses squared gradients to scale the learning rate like RMSprop and also it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. Adam is an adaptive learning rate method, it computes individual learning rates for different parameters. Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network. The first moment is mean, and the second moment is uncentered variance i.e. the mean is not subtracted while taking gradient. Adam is evaluated by using exponential moving averages, computed on the gradient

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.10}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) {g_t}^2 \tag{3.11}$$

Moving averages of gradient and squared gradient. Where m and v are moving averages, g is gradient ,and beta is new introduced hyper-parameters of the algorithm. They have default values of 0.9 and 0.999 respectively. for different values of t the gradient changes we can generalise this as

$$m_t = (1 - \beta_1) \sum_{i=0}^{t} {\beta_1}^{t-i} g_i \tag{3.12}$$

Following are the steps for Adam optimization algorithm:

Algorithm 3 Adam algorithm. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise [7]

Require:
α: Stepsize
β_1, $\beta_2 \in$ [0,1): Exponential decay rates for the moment estimates
f(w): Stochastic objective function with parameters w
w_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1^{st} moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t_0 \leftarrow 0$ (Initialize timestep)
while w_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_w f_t(w_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\hat{v}_t \leftarrow m_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$w_t \leftarrow w_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end while
return w_t (resulting parameters)

3.10 Convolutional Neural Network

Convolutional Neural Network also known as CNN are like neural networks, made up of neurons with learnable weights and biases. CNN preserves the spatial structure by learning internal feature which is divided into small squares of input data. It were developed for object recognition tasks such as handwritten digit recognition, image classification, etc. The hidden layers of a CNN typically consist of convolutional layers, pooling layers and fully connected layers.

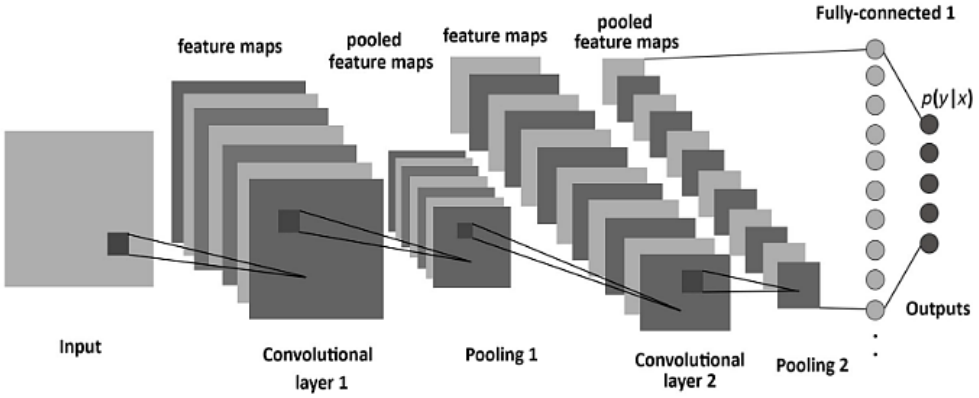


Figure 3.11: A example of CNN architecture.

Convolutional Layer:

Convolution is the first layer to extract features from an input image. The relationship between pixels is preserved by learning image features using small squares of input data. Convolution layer takes image matrix and filter or kernel as input and performs mathematical operation on it.

- An image matrix of dimension $(h \times w \times d)$
- A kernel or filter $(f_h \times f_w \times d)$
- Outputs a volume dimension $(h - f_h + 1) \times (w - f_w + 1)$

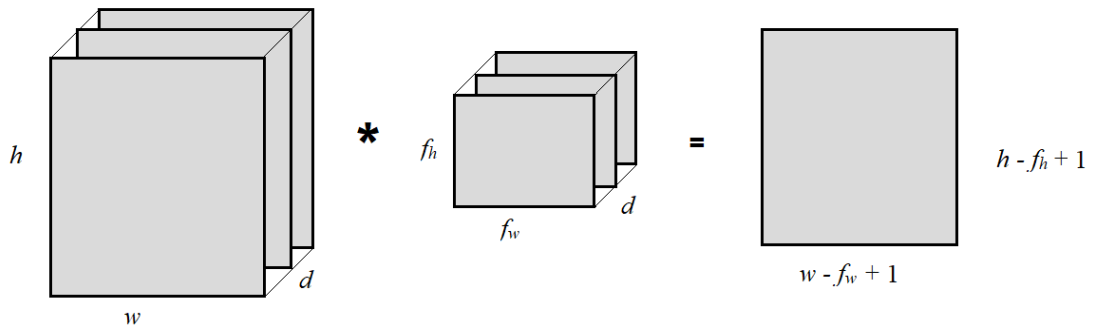


Figure 3.12: Image matrix multiplies with kernel.

To perform convolution on an image, following steps should be take into consideration

- Place the center of the mask at each element of an image
- Slide the kernel onto the image
- Multiply the corresponding elements and then add them
- Paste the result onto the element of the image on which you place the center of mask
- Repeat this procedure until all values of the image has been calculated

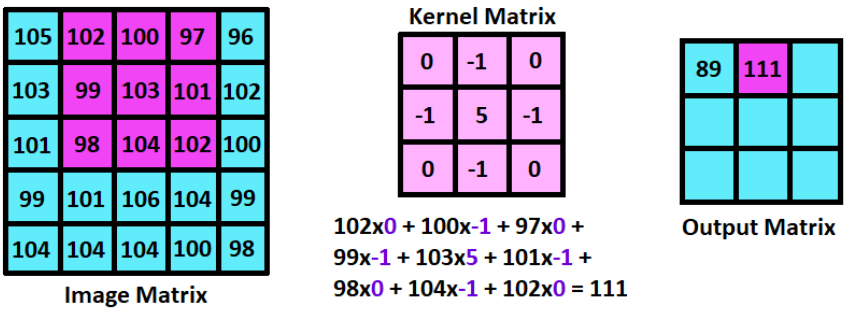


Figure 3.13: Convolution calculation on grayscale image.

The objective of the Convolution operation is to extract the high-level features such as edges, from the input image. Convolutional layer need not be limited to only one Convolutional Layer. The first convolutional layer captures the low-level features such as edges, colour, etc. With more convolution layers ahead, the architecture captures complex features termed high level features. This leds the model to understand the dataset in similar manner as humans do.

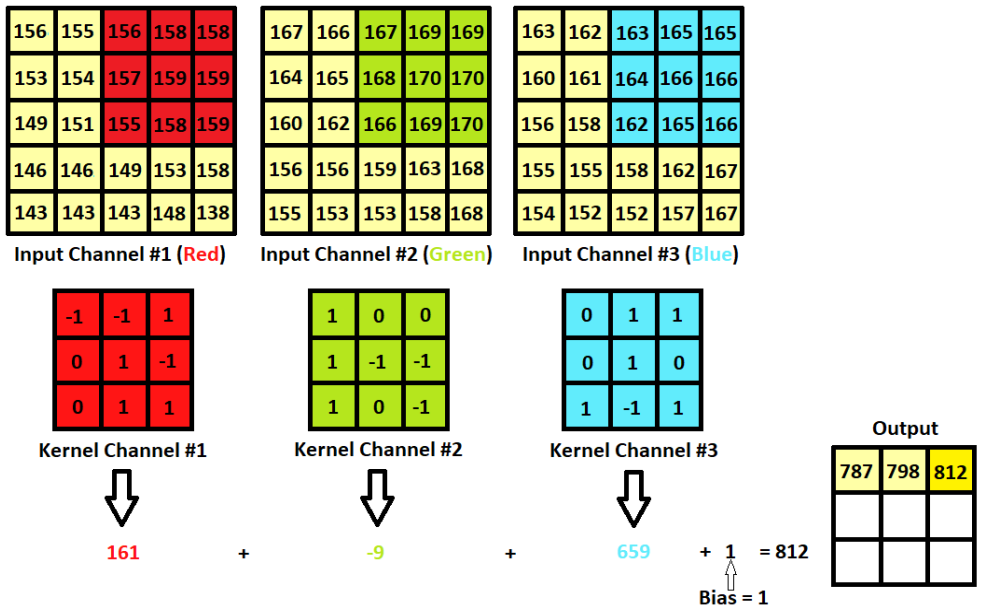


Figure 3.14: Convolution operation on a RGB image matrix with a 3x3x3 Kernel.

Pooling Layer:

Pooling layer is used to reduce the spatial size of the convolved feature obtained from the previous layer. It decreases the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are 2 types of pooling

- Max Pooling: It returns the maximum value from the portion of the image covered by the Kernel.
- Average Pooling: It returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling acts as a noise suppressant. It discards the noisy activations and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

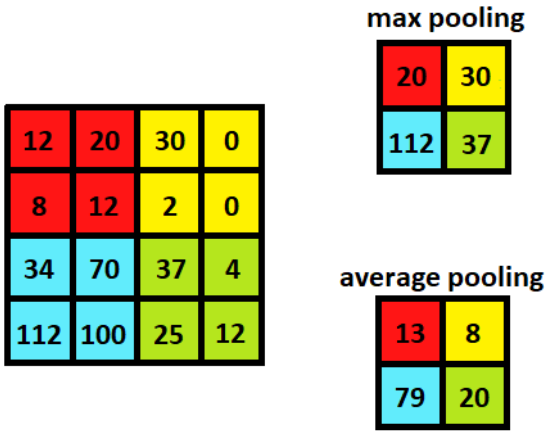


Figure 3.15: Pooling.

3.11 Batch Normalization

Batch normalization is achieved by adjusting and scaling the activations. For example, when we have features from 0 to 1 and some from 1 to 1000, we should normalize them to speed up learning, and get 10 times or more improvement in the training speed. Batch normalization reduces the amount by what the hidden unit values shift around (covariance shift). Also, batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers. We can use higher learning rates because batch normalization makes sure that there's no activation that's gone really high or really low. It reduces overfitting because it has a slight regularization effects. Similar to dropout, it adds some noise to each hidden layer's activations. Therefore, if we use batch normalization, we will use less dropout, which is a good thing because we are not going to lose a lot of information. However, we should not depend only on batch normalization for regularization; we should better use it together with dropout.

To increase the stability of a neural network, batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. However, after this shift/scale of activation outputs by some randomly initialized parameters, the weights in the next layer are no longer optimal. SGD (Stochastic gradient descent) undoes this normalization if it's a way for it to minimize the loss function. Consequently, batch normalization adds two trainable parameters to each layer, so the normalized output is multiplied by a"standard deviation" parameter (γ) and add a "mean" parameter (beta). In other words, batch normalization lets the SGD to do the denormalization by changing only these two weights for each activation, instead of losing the stability of the network by changing all the weights.

Steps of Operation

We can use to normalize the inputs of each layer, in order to fight the internal covariate shift problem. During training time, a batch normalization layer does the following:

i. Calculate the mean and variance of the layers input.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3.13}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{3.14}$$

where, Eq. 3.13 is batch mean and Eq. 3.14 is batch variance.

ii. Normalize the layer inputs using the previously calculated batch statistics.

$$\bar{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.15}$$

iii. Scale and shift in order to obtain the output of the layer.

$$y_i = \gamma \cdot \bar{x}_i + \beta \tag{3.16}$$

γ and β are learned during training along with the original parameters of the network. So, if each batch had m samples and there where j batches:

$$E_x = \frac{1}{m}\sum_{i=1}^{j}\mu_B^{(i)} \tag{3.17}$$

$$Var_x = (\frac{1}{m-1})\sum_{i=1}^{j}\sigma_B^{(i)} \tag{3.18}$$

$$y = \frac{\gamma}{\sqrt{Var_x + \epsilon}} \cdot x + \left(\beta + \frac{\gamma E_x}{\sqrt{Var_x + \epsilon}}\right) \tag{3.19}$$

where Eq. 3.17 is inference mean and Eq. 3.18 is inference variance and Eq. 3.19 is inference scaling/shifting.

3.12 Dropout [6]

In this regularization technique we randomly drop connection in a neural network and train it. The most common dropout implementd is "Inverted Dropout". In its most simple form, during training, at each example presentation, feature detectors are deleted with probability q and the remaining weights are trained by backpropagation. All weights are shared across all example presentations. The main motivation behind the algorithm is to prevent the co-adaptation of feature detectors, or overfitting, by forcing neurons to be robust and rely on population behavior, rather than on the activity of other specific units.

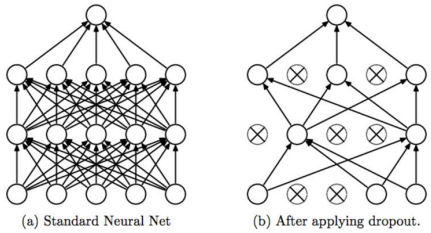


Figure 3.16: Dropout (Source: [8]).

Chapter 4

Dataset Analysis

4.1 PH^2 Dataset:

PH^2 is a dermoscopic [10] image database. Dermoscopic images provide us a more detailed view of patterns and structures present on skin as compared to the normal magnified images. PH^2 dataset is used as ground truth for testing and validating the segmentation and classify the types using different algorithms. The PH^2 database contains a total number of 200 lesions images, including 80 common nevi, 80 atypical nevi, and 40 melanomas images. Here common nevi and atypical nevi can be combined to be termed as 160 non-melanoma images. The total size of the PH^2 database i.e. 200 images only might seem small for validating and relying on the results obtained by any machine learning algorithm.

The PH^2 database was built up by a joint research collaboration between the Universidade do Porto, Tecnico Lisboa, and the Service provided by Hospital Pedro Hispano in Matosinhos, Portugal. The dermoscopic images were obtained through the Tuebinger Mole Analyzer [11] system all under the same condition by using a magnification of 20x. The database consists of RGB color images with a maximum resolution of 768×560 pixels and there is some variation in resolution in images. Every image was evaluated by a dermoscopic [12] with respect to the following parameters:

- Manual segmentation of the skin lesion
- Clinical and histological (when available) diagnosis
- Dermoscopic criteria (Asymmetry, Colors, Pigment network, Dots/Globules, Streaks, Regression areas, Blue-whitish veil)

The lesions can be divided into two main groups considering their nature: benign lesions (which include common and atypical nevus) and malignant lesions (melanoma). Therefore, each image of the database is classified into non-melanoma (common nevus and atypical nevus) or melanoma Fig. 4.1.

The manual segmentation of each image of the database is available in a binary format, more precisely as the binary mask of the image which of the same size as that of the original image. The pixels of the skin lesion have intensity value 1 whereas the background region has values of 0. This output can be used to extract the boundary position of the skin lesion. An example of a dermoscopic image and the corresponding manual segmentation is presented in Fig. 4.2.

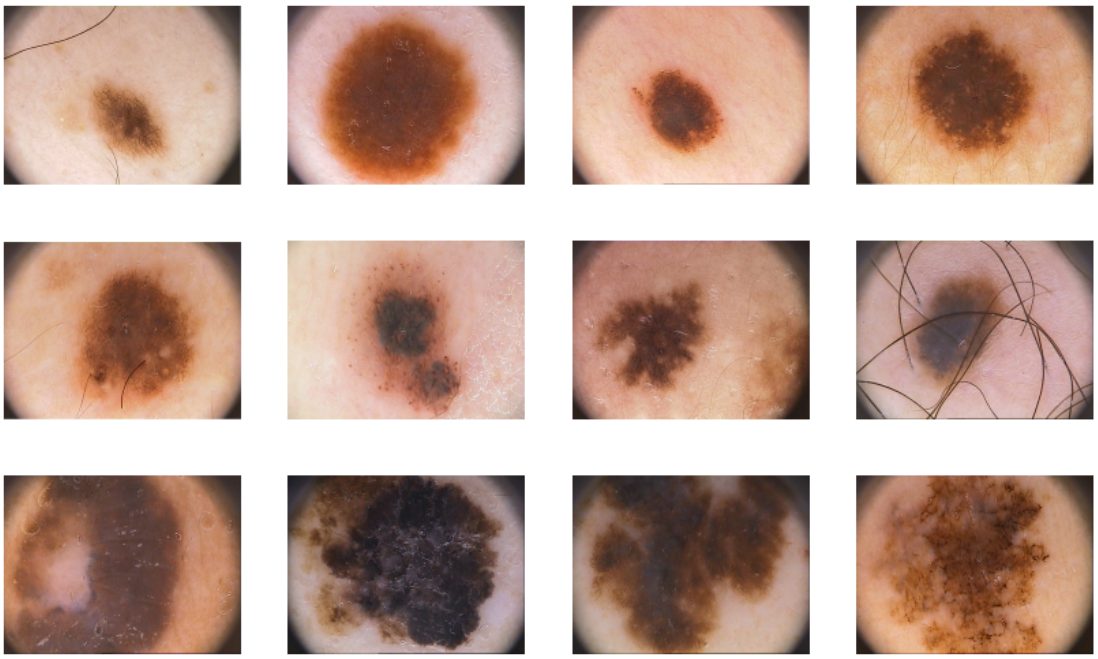


Figure 4.1: An illustrative collection of images from PH^2 database, including common nevi (1st row), atypical nevi (2nd row) and melanomas (3rd row).

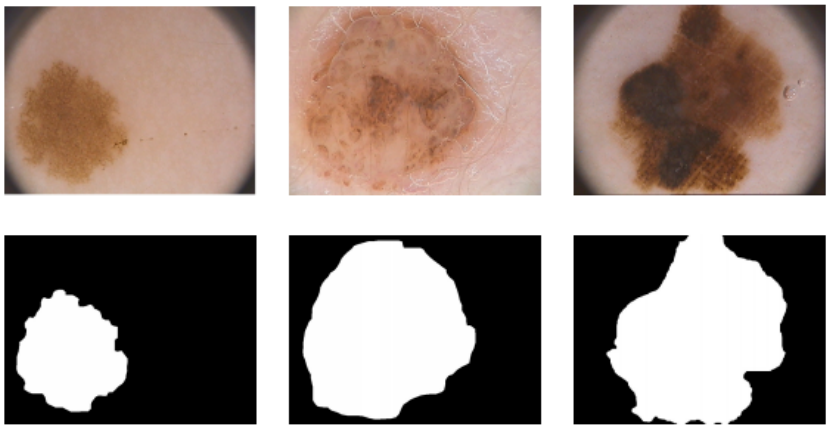


Figure 4.2: Manual segmentation of three melanocytic lesions: common nevus (left), atypical nevus (middle) and melanoma (right).

Table 4.1: Summary of publicly available dermatoscopic image datasets of PH^2.

Dataset	Total images	mel	nv
PH^2	200	40	160

The PH^2 dataset consists of 200 dermoscopic images where 40 images are of melanoma and 160 images were of non-melanoma (which includes 80 common nevus images and 80 atypical nevus images). This dataset was pathologically verified by the dermatologists of Hospital Pedro Hispano in Matosinhos, Portugal.

4.1.1 Dermoscopic criteria [14]

The PH^2 database corresponds to features that can be termed as more relevant for clinical diagnosis. All of these features, as well as their evaluation process, are described below.

- Asymmetricity: One of the most important features for diagnosing a melanocytic lesion is asymmetricity. According to the ABCD rule [13] of dermoscopy, asymmetry is the largest weight factor. The lesion asymmetry was evaluated according to the ABCD rule it is assessed regarding its contour, colors, and structures distribution simultaneously.

- Colors: Overall, six different colors are taken into account during the diagnosis of a melanocytic lesion. The set of color classes comprises the white, red, light-brown, dark-brown, blue-gray, and black. Each image of the database was evaluated by a dermatologist in order to identify the presence, as well as the location, of the six color classes. An example is presented in Fig.4.3, where two color classes were identified.

Figure 4.3: Original image (left), blue-gray (middle) and darkbrown region (right).

- Pigment Network: The pigment network is a grid-like network consisting of pigmented lines (brown or black) and hypopigmented holes. This structure plays an important rule in the distinction between melanoma and non-melanoma lessons. It is classified as typical or atypical. Refer Fig. 4.4.

- Dots/Globules: As illustrated in Fig. 4.4, dots/globules are spherical or oval, variously sized, black, brown or gray structures (dots are usually smaller than globules). The presence of these dermoscopic structures is also particularly useful for the distinction between melanocytic and non-melanocytic lesions. When dots/globules are present in a given lesion, these structures are further classified as regular or irregular concerning their distribution in the lesion.

- Streaks: Streaks are finger-like projections of the pigment network from the periphery of the lesion. Instead of both pigment network and dots/globules, the presence of streaks in a skin lesion is by itself a sign of malignancy. Therefore, these structures are just classified as present or absent in each image of the database. Fig. 4.4 illustrates the presence of a streak in a skin lesion (identified on the upper area).

- Regression areas: Regression areas are defined as white, scar-like depigmentation often combined with pepperlike regions (speckled blue-gray granules). In the PH^2 database, this parameter is classified in two main groups (present or absent) concerning its presence in the skin lesion.

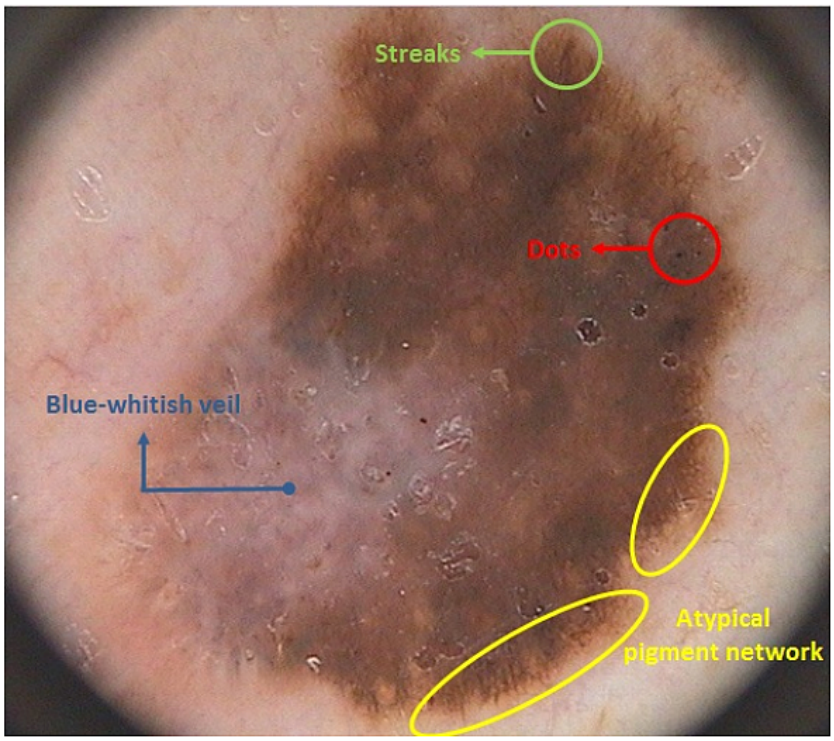


Figure 4.4: Dermoscopic features identification.

- Blue-whitish veil: The blue-whitish veil can be defined as a confluent, opaque, irregular blue pigmentation with an overlying, white, ground-glass haze. Its presence is a strong malignancy indicator Refer Fig. 4.4.

4.1.2 Analysis of PH^2 Dataset

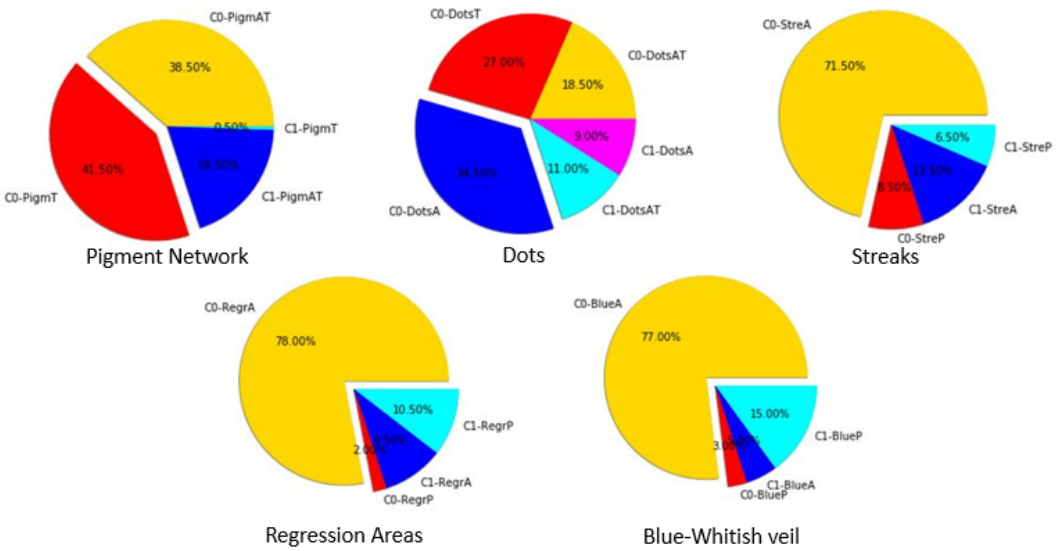


Figure 4.5: PH^2 Dataset Analysis.

PH^2 database is a dermoscopic image acquired at Pedro Hispano Hospital. It in-

cludes medical analysis of all the images for skin lesion detection and dermoscopic criteria (asymmetry, colors, pigment network, dots/streaks, regression, blue-whitish veil and the presence of typical and atypical differential structures). Asymmetricity is one of the most important features for diagnosing a melanoma lesion (dots are usually smaller than globules). The presence of dots/globules is useful for the distinction between melanoma and non-melanoma lesions. The presence of streaks and blue-whitish veil in a skin lesion is a strong melanoma indicator. Melanoma is mostly symmetrical in shape whereas non-melanoma is asymmetrical in shape.

4.2 HAM10000 Dataset:

The problem of small size and lack of available dataset of dermatoscopic images is tackled using HAM10000 (with 10000 training images) dataset. The 10015 dermatoscopic images of the HAM10000 training set were collected over a period of 20 years from two different sites, the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia.

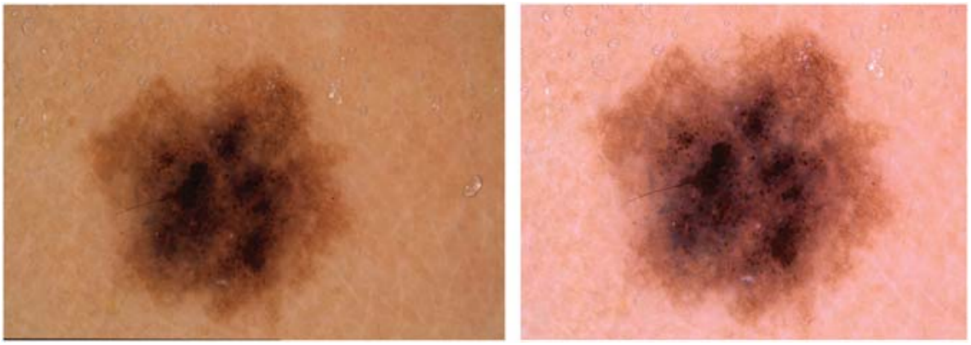


Figure 4.6: Manual correction of a scanned diapositive. Original scanned image(left)with remaining black border on the lower left, lesion off center, yellow hue and reduced luminance. On the right, the final image after manual quality review.

The Australian site stored images and metadata in PowerPoint files and Excel databases. The Austrian site started to collect images before the era of digital cameras and stored images and metadata in different formats during different time periods. Before the introduction of digital cameras, dermatoscopic images at the Department of Dermatology in Vienna, Austria were stored as diapositives. We digitized the diapositives with a Nikon Coolscan 5000 ED scanner with a two-fold scan with Digital ICE and stored files as JPEG Images (8-bit color depth) in highest quality (300DPI; 15×10 cm). We manually cropped the scanned images with the lesion centered to 800×600px at 72DPI, and applied manual histogram corrections to enhance visual contrast and color reproduction (Fig. 4.6).

Table 4.2: Summary of publicly available dermatoscopic image datasets of HAM10000 manually augmented.

Dataset	Total Images	akiec	bcc	bkl	df	mel	nv	vasc
HAM 10000	10015	327	514	1099	115	1113	6705	142

The HAM10000 dataset consists of 10015 dermoscopic images which consist of manually augmented images. Here the manual augmentation was done by cropping the images

Table 4.3: Summary of publicly available dermatoscopic image datasets of HAM10000 without augmentation.

Dataset	Total Images	akiec	bcc	bkl	df	mel	nv	vasc
HAM 10000	5515	151	175	440	39	230	4415	64

with lesions in the centre, changing the magnification size, changing the histogram, etc. This dataset is divided into seven categories of images such as 327 images of akiec, 514 images of bcc, 1099 images of bkl, 115 images of df, 1113 images of mel, 6705 images of nv and 142 images of vasc. This dataset was pathologically verified by the dermatologists. Refer Table 4.2.

The HAM10000 dataset consists of 5515 dermoscopic images which consist of images without augmentation. This dataset are divided into seven categories of images such as 151 images of akiec, 175 images of bcc, 440 images of bkl, 39 images of df, 230 images of mel, 4415 images of nv and 64 images of vasc. This dataset was pathologically verified by the dermatologists. Table 4.3.

The following are the description of diagnostic categories:

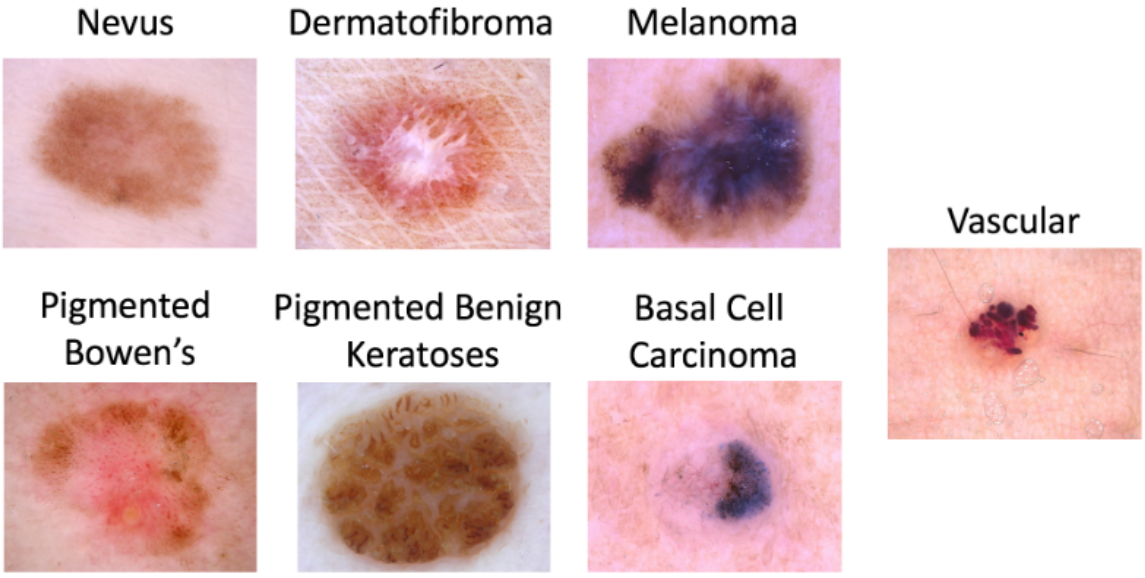


Figure 4.7: Disease classification within dermoscopic images.

- Actinic Keratoses/Pigmented Bowen's (akiec): Actinic Keratoses (Solar Keratoses) and Intraepithelial Carcinoma (Bowen's disease) are common noninvasive, variants of squamous cell carcinoma that can be treated locally without surgery. There is, however, agreement that these lesions may progress to invasive squamous cell carcinoma – which is usually not pigmented. Both neoplasms commonly show surface scaling and commonly are devoid of pigment. Actinic keratoses are more common on the face and Bowen's disease is more common on other body sites. Because both types are induced by UV-light the surrounding skin is usually typified by severe sun damaged except in cases of Bowen's disease that are caused by human papilloma virus infection and not by UV.

- Basal cell carcinoma (bcc): Basal cell carcinoma is a common variant of epithelial skin cancer that rarely metastasizes but grows destructively if untreated. It appears in different morphologic variants (flat, nodular, pigmented, cystic).

- Pigmeneted Benign keratosis (bkl): "Benign keratosis" is a generic class that includes seborrheic keratoses ("senile wart"), solar lentigo - which can be regarded a flat variant of seborrheic keratosis - and lichen-planus like keratoses (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression. The three subgroups may look different dermatoscopically, but we grouped them together because they are similar biologically and often reported under the same generic term histopathologically.

- Dermatofibroma (df): Dermatofibroma is a benign skin lesion regarded as either a benign proliferation or an inflammatory reaction to minimal trauma. The most common dermatoscopic presentation is reticular lines at the periphery with a central white patch denoting fibrosis.

- Nevus (nv): Melanocytic nevi are benign neoplasms of melanocytes and appear in a myriad of variants, which all are included in our series. The variants may differ significantly from a dermatoscopic point of view. In contrast to melanoma they are usually symmetric with regard to the distribution of color and structure.

- Melanoma (mel): Melanoma is a malignant neoplasm derived from melanocytes that may appear in different variants. If excised in an early stage it can be cured by simple surgical excision. Melanomas can be invasive or noninvasive (in situ). We included all variants of melanoma including melanoma in situ, but did exclude non-pigmented, subungual, ocular or mucosal melanoma.

- Vascular (vasc): Vascular skin lesions in the dataset range from cherry angiomas to angiokeratomas and pyogenic granulomas. Hemorrhage is also included in this category. Angiomas are dermatoscopically characterized by red or purple color and solid, well circumscribed structures known as red clods or lacunes.

Chapter 5

Architecture

Following architectures were used for training on PH^2 dataset (short explaination of original architecture followed by the modified one that we used).

5.1 VGG-16

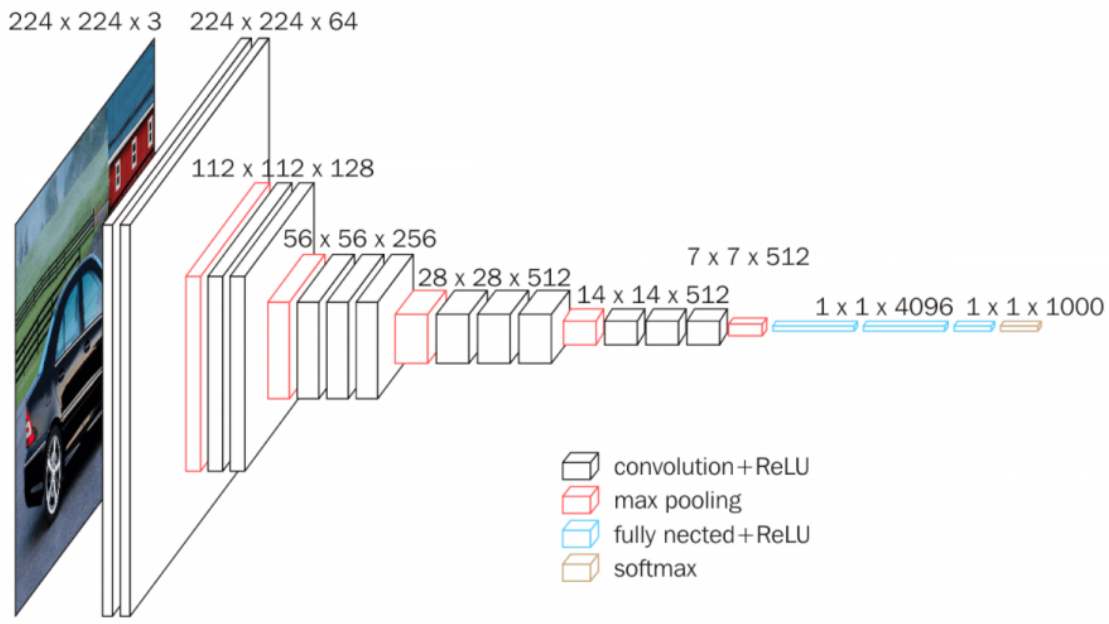


Figure 5.1: VGG-16 architecture

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3× 3 kernel-sized filters one after another.The activation function used throughout the architechture is ReLU.

CONV 1:

The input for Vgg16 is a 224×224× 3 RGB image which passes through first and second convolutional layers with 64 feature maps or filters having size 3×3 and same pooling with a stride of 14. The image dimensions changes to 224×224×64.

Then the VGG16 applies maximum pooling layer or sub-sampling layer with a filter size 3× 3 and a stride of two. The resulting image dimensions will be reduced to 112×112×64.

CONV 2:

Next,there are two convolutional layer with 128 feature maps having size 3×3 and a stride of 1. Then there is again a maximum pooling layer with filter size 3×3 and a stride of 2. This layer is same as previous pooling layer except it has 128 feature maps so the output will be reduced to 56×56×128.

CONV 3:

The 3 covolutional layers have filter size of 3×3 and a stride of one. Both use 256 feature maps.The two convolutional layers are followed by a maximum pooling layer with filter size 3×3, a stride of 2 and have 256 feature maps. The output changes to 28× 28×256.

CONV 4:

Next are the two sets of 3 convolutional layers followed by a maximum pooling layer. All convolutional layers have 512 filters of size 3×3 and a stride of one. The final size will be reduced to 7×7×512.

CONV 5:

The convolutional layer output is flatten through a fully connected layer with 25088 feature maps each of size 1×1.

FC classifier:

Next is again two fully connected layers with 4096 units. Finally, there is a softmax output layer with 1000 possible values.

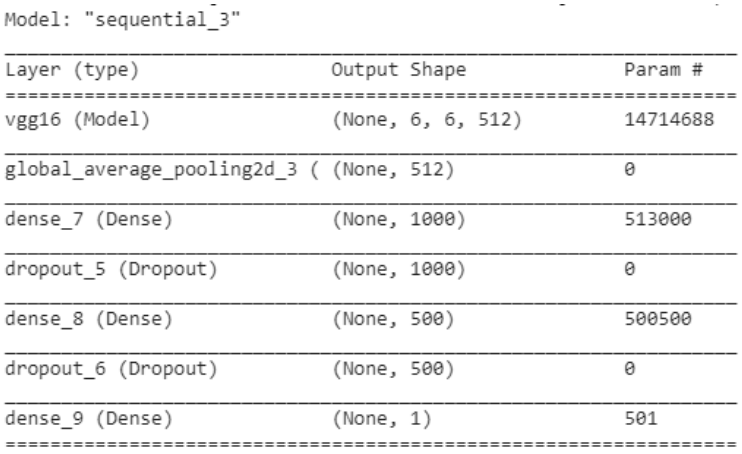
5.2 VGG-19

The difference between VGG-16 and VGG-19 is described in the diagram below:

Figure 5.2: Comparison between VGG-16 and VGG-19 architecture

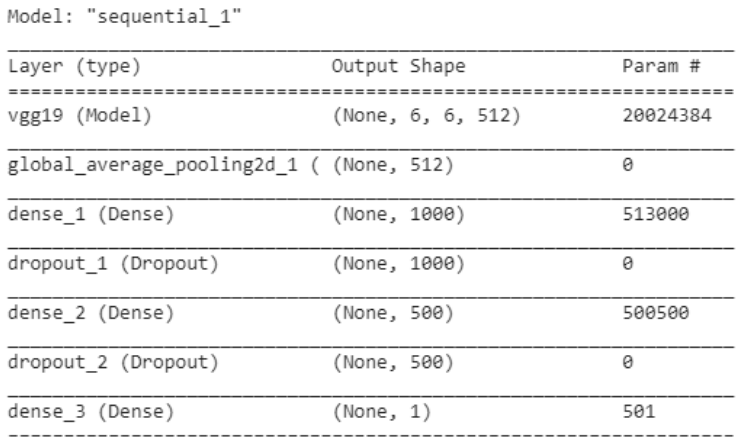
The roles of each block and their functioning in VGG-19 are the same as that of VGG-16. The CONV 3 block in VGG-16 contains 3 convolutional layers whereas the same block in VGG-19 contains 4 convolutional layers. The same is true for CONV 4 and CONV5 blocks. Since these three blocks have 1 extra convolutional layer in them, a total of 19 layers exist in the modified architecture. Thus this modified architecture is called VGG-19. However, the performance of both VGG-16 and VGG-19 is more or less similar. Hence VGG-16 is widely used compared to VGG-19.

We made use of VGG16 and VGG19 Architecture along with some of the other added layers and used the below-modified model for training on the PH^2 dataset. Note: The input image size is of 200·200 to the original VGG16 and VGG19 model.



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 6, 6, 512)         14714688
_________________________________________________________________
global_average_pooling2d_3 ( (None, 512)               0
_________________________________________________________________
dense_7 (Dense)              (None, 1000)              513000
_________________________________________________________________
dropout_5 (Dropout)          (None, 1000)              0
_________________________________________________________________
dense_8 (Dense)              (None, 500)               500500
_________________________________________________________________
dropout_6 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 501
=================================================================
```

Figure 5.3: Modified model(sequential-3) with VGG16 for PH^2 training



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Model)                (None, 6, 6, 512)         20024384
_________________________________________________________________
global_average_pooling2d_1 ( (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1000)              513000
_________________________________________________________________
dropout_1 (Dropout)          (None, 1000)              0
_________________________________________________________________
dense_2 (Dense)              (None, 500)               500500
_________________________________________________________________
dropout_2 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 501
_________________________________________________________________
```

Figure 5.4: Modified model(sequential-2) with VGG19 for PH^2 training

5.3 MobileNet

MobileNet is an efficient network architecture and with a set of two hyper-parameters to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications. Depthwise Separable Convolution is used to reduce the model size and complexity.Two parameters are introduced so that MobileNet can be tuned easily: Width Multiplier α and Resolution Multiplier ρ .Width Multiplier α is introduced to control the input width of a layer.

Layer (type)	Output Shape	Param #
mobilenet_1.00_224 (Model)	(None, 17, 23, 1024)	3228864
global_average_pooling2d_1 (	(None, 1024)	0
dense_1 (Dense)	(None, 1000)	1025000
dropout_1 (Dropout)	(None, 1000)	0
dense_2 (Dense)	(None, 500)	500500
dropout_2 (Dropout)	(None, 500)	0
dense_3 (Dense)	(None, 1)	501

Total params: 4,754,865
Trainable params: 1,526,001
Non-trainable params: 3,228,864

Figure 5.5: Modified model of MobileNet for PH^2 training

We will be fine tuning these pretrained models to get good accuracy for PH^2 and HAM10000 datasets.

Chapter 6

Conclusion

From the literature, the importance of data augmentation was evident. By applying data augmentation to PH^2 dataset we were able to achieve good accuracy(93%training). We also experimented with different architectures (VGG-16, VGG-19, MobileNet) on our PH^2 dataset. We are planning to use the concept of transfer learning on the HAM10000 dataset. The trained model (PH^2) can classify melanoma and non-melanoma with good accuracy. We intend to achieve the same for the classification of 7 different skin lesion classes using the HAM10000 dataset. The work done on PH^2 dataset has served the foundation for our future analysis and processing on the HAM10000 dataset.

Bibliography

[1] 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"*. Machine Learning in Medical Imaging Lecture Notes in Computer Science, pp. 118-126, 2015. Link:
`https://link.springer.com/chapter/10.1007/978-3-319-24888-2_15`

[2] Matsunaga, Kazuhisa, Hamada, Akira, Minagawa, Akane, Koga, and Hiroshi, *"Image Classification of Melanoma, Nevus and Seborrheic Keratosis by Deep Neural Network Ensemble"*, Published on Jan 1, 2017in arXiv: Computer Vision and Pattern Recognition. Link:
`https://arxiv.org/abs/1703.03108`

[3] Iván González Díaz. *"Incorporating the Knowledge of Dermatologists to Convolutional Neural Networks for the Diagnosis of Skin Lesions"*, IEEE Journal of Biomedical and Health Informatics , 16 February 2018. Link:
`https://ieeexplore.ieee.org/document/8293766`

[4] Afonso, Michel, Pires, Ramon, Bittencourt, F. Vasques, Avila, Sandra, and Eduardo, href*"Knowledge Transfer for Melanoma Screening with Deep Learning"*, Published in: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017). Link:
`https://ieeexplore.ieee.org/abstract/document/7950523`

[5] Jiawei Zhang, *"Gradient Descent based Optimization Algorithms for Deep Learning Models Training"*, Published in ArXiv 2019. Link:
`https://arxiv.org/abs/1903.03614`

[6] Pierre Baldi,Peter Sadowski, *"Understanding Dropout"*, published in NIPS 2013. Link:
`https://papers.nips.cc/paper/4878-understanding-dropout`

[7] Diederik P. Kingma, Jimmy Ba, *"Adam: A Method for Stochastic Optimization"*, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. Link:
`https://arxiv.org/abs/1412.6980`

[8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky , Ilya Sutskever, Ruslan Salakhutdinov, *"Dropout: A Simple Way to Prevent Neural Networks from Overfitting"*, Journal of Machine Learning Research, published 6/14. Link:
`https://www.cs.toronto.edu/ hinton/absps/JMLRdropout.pdf`

[9] "Skin Cancer Facts & Statistics," The Skin Cancer Foundation. [Online]. Available https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/

[10] "Dermoscopic", https://www.dermnetnz.org/topics/dermoscopy/

[11] "Tuebinger Mole Analyzer", http://www.moleanalyzer.com/engl/frameset.htm

[12] "Dermatologist", https://www.abderm.org/public/what-is-a-dermatologist.aspx

[13] "ABCD rule", http://www.dermoscopy.org/atlas/4step/abcd_d.htm

[14] "Dermoscopic Criteria", https://dermoscopedia.org/Dermoscopic_structures