CLASSIFICATION OF SKIN-LESION BASED ON CONVOLUTIONAL NEURAL NETWORKS

A Project Report submitted under the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by,

PRAVALLIKA. M - 121710307060

RAM CHARAN.S - 121710307045

KAVYA.D - 121710307011

SAI SUPRIYA.M - 121710307061

Under the esteemed guidance of

Mrs. **Naga Soujanya K**

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM

(Deemed to be University)

VISAKHAPATNAM

APRIL 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled **"CLASSIFICATION OF SKIN-LESION BASED ON CONVOLUTIONAL NEURAL NETWORKS"** is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

Registration No(s).	**Name(s)**
121710307060	**Pravallika.M**
121710307045	**Ram Charan.S**
121710307011	**Kavya.D**
121710307061	**Sai Supriya.M**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

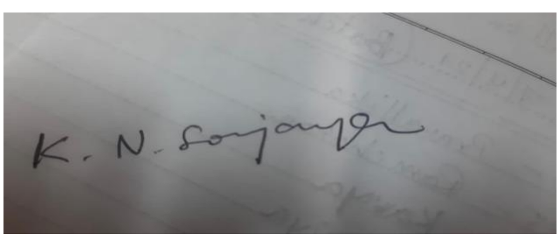
GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled **"CLASSIFICATION OF SKIN-LESION BASED ON CONVOLUTIONAL NEURAL NETWORKS"** is a bonafide record of work carried out by **Pravallika.M** (121710307060), **Ram Charan.S** (121710307045), **Kavya.D** (121710307011), **Sai Supriya.M** (121710307061) submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide **Head of the Department**



Mrs. NAGA SOUJANYA K

Assistant Professor

Dr. R.SIREESHA

Professor

ACKNOWLEDGEMENT

We would like to thank our project guide Mrs. Naga Soujanya K, Assistant professor, Dept of CSE for her stimulating guidance and profuse assistance. We shall always cherish our association with her for her guidance, encouragement and valuable suggestions throughout the progress of this work. We consider it a great privilege to work under her guidance and constant support.

We also express our thanks to the project reviewers Dr. Sireesha R, Professor, Head of the Department, CSE, GITAM(Deemed to be University) and Bhargav K, Assistant professor, Department of CSE, GITAM (Deemed to be University) for their valuable suggestions and guidance for doing our project.

We consider it is a privilege to express our deepest gratitude to Dr.R.Sireesha, Head of the Department, Computer Science Engineering for her valuable suggestions and constant motivation that greatly helped us to successfully complete this project.

Our sincere thanks to Dr. C. Dharma Raj, Principal, GITAM Institute of Technology, GITAM (Deemed to be University) for inspiring us to learn new technologies and tools.

We are thankful to Dr. C. Vijay Shekar, Dean of Engineering, GITAM (Deemed to be University) for constantly inspiring us and giving motivating us to complete this project.

Finally, we feel it as a great pleasure to thank one and all who helped us directly and indirectly throughout this project.

Pravallika.M-121710307060
Ram Charan.S-121710307045
Kavya.D-121710307011
Sai Supriya.M-121710307061

ABSTRACT

Skin lesions is one of the most dangerous diseases caused by, bacteria, allergy, viruses etc. The advancement in technologies like lasers and biopsy based medical technology is used in diagnosis of the skin lesions quickly and accurately. The medical equipment for such diagnosis are limited and most expensive. Skin cancer's early detection has very much importance to save the life of a victim but it is a challenging task for dermatologists .

So, the usage of Deep Learning techniques and convolutional neural networks architectures helps in detection of skin lesions in a tender stage. The feature extraction plays a key role in classification of skin lesion. The usage of Convolution Neural Networks reduced the need human power and lab equipment, and also reduced the usage of manual feature extraction and data reconstruction for classification purpose. In this project we are going to use transfer learning with mobilenet (A convolution neural network architecture) architecture for better speed and accuracy.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

The skin of the human body acts as the organ and also acts as the cover to the human body.In the integumentary system of the human, the skin is found as the biggest organ.The skin is treated as one of the humans five sense organs which covers and protects the internal systems of a human body.It has three layers: Epidermis,Dermis and the hypo dermis. Epidermis of the skin is the top layer which protects as waterproof shield and plays a major role in defining the color of the skin. The middle layer of the skin is known as dermis which contains hair follicles, tough connective tissues and sweat glands.The last layer of the skin is hypo dermis,this is the inner layer of fat and other connective tissues which is also known as Subcutaneous tissue.The skin functions in metabolic functions,sensation,protection and thermoregulation.

Skin always acts as a protection shield against the infections.It protects the body from pathogens and from more loss of water that causes dehydration by playing a immunity part of the system.It helps to the protection of vitamin B float and synthesis the vitamin D. A variety of factors from sun hydration to repetitive face movement,can cause skin damage.From sun burn to skin cancer,the skin can be damaged to a large extent.Keeping our skin moist helps to strengthen the barrier.The injured skin will self cure by creating tissue which is not pigmented beside discolored.

1.1 MOTIVATION

The recent emergence of deep learning and convolution neural networks methods for the analysis of medical images has increased the development of intelligent and fast imaging-based diagnosis systems that can assist the human better in making better decisions about a patients condition. As skin lesions has become one of the most disturbing problem these days which on neglected may cause severe problems. By the emerging of the different technologies on medical image analysis this is very inspiring to develop a project which is common and rather neglected problem that is skin lesion.

1.2 PROJECT OBJECTIVE

The main objective of this project is to create a fast and easy method that can tell doctors, lab workers and the patients to detect the seven highest probability diagnoses for a given skin lesion. This will help them quickly identify the patients who are more prone to danger and treat them in an early stage and speed up their treatment. The tool should produce a result in ten seconds. To protect the images from the external usage and to increase the efficiency of the tool the images are pre-processed and and to an external server because data is the key for every project.

1.3 OVERVIEW OF SKIN LESION

Skin lesion is any abnormality in skin that causes a infection or any malfunctioning in skin.The skin is responsible from protection of our inner organs from various germs and bacteria which are very cruel.The absence of skin's functionality may lead to several health problems such as Rosacea, Acne, Shingles etc.Skin has various cells namely mesodermal cell which protect human body from various radiations and rays.This skin also have pigmentation which consists of human skin DNA.This pigmentation also reasonable for skin having various skin tones like black,white,brown etc.Skin consists millions of skin pores responsible for sweating.

Biochemically the dermal collagen and elastic content material is comparable in Pig skin versus human body skin. Both these skin has pores which are having similar properties and responses.

Skin has various cells namely mesodermal cell which protect human body from various radiations and rays.This skin also have pigmentation which consists of human skin DNA.This pigmentation also reasonable for skin having various skin tones like black,white,brown etc.Skin consists millions of skin pores responsible for sweating.

Skin is the one of the largest organ of human body covering whole body.The surface vicinity of skin is one and half to two rectangular meters.The square inch of skin contains 650 sweat glands and around twenty blood vessels.

1.4 CAUSES OF SKIN LESION

Skin lesion is generally described as an abnormality that caused in skin affecting the shape,colour and functions of skin.There are many causes of skin lesion which leads to skin cancer.

BURNS

The major cause for skin lesion is burns,these burns are termed as any surface of skin that has been affected by heat or flames.These burns cause the skin layers to loose its normality and these burns also leads to several infections that are most precisely harmful and sometimes those leads to fatal infection in skin.

RADIATIONS

Radiations are the energy waves that are released from radioactive elements such as uranium,plutonium,ultraviolet etc. These radiations can cause burns such are radiation burns which is described as a damage to skin or other biological tissue in skin that is effected by radiations.The radiation types of greatest problems are thermal radiation,UV radiations,Radio Frequency radiation, Ionizing radiations.

1.5 SKIN LESIONS AND IT'S TYPES

Abnormal growth of skin is often defined as skin lesion.It is mostly developed on skin area which is mostly uncovered .But skin cancer can also occur on other areas of the skin which is not displayed to sun.The characterization of skin cancer type is done by observing the region where the cancer begins

Skin cancer is mainly characterized to three types:

i. Actinic Keratoses and intraepithelial carcinoma/Bowen's disease
ii. Basal cell carcinoma
iii. Benign keratosis-like lesions melanoma/seborrheic keratoses
iv. Melanocytic nevi
v. Vascular lesions
vi. Dermatofibroma
vii. Melanoma

1.5.1 ACTINIC KERATOSES AND INTRAPITHELIAL CARCINOMA / BOWEN'S DISEASE

These type of lesions which are detectable and have a varied appearance and can commonly occur in the head and neck and in hand regions. The majority of these lesions are congenital; even though some may be acquired and malignant which are of less dangerous. These lesser flow lesions appear at the birth to infant stage as flat pink areas of skin coloration which is unusual and usually darken over time to become a dark or deep purple color. These unusual color lesions can eventually develop changes in skin texture and type with a stone effect and can also be associated with overgrowth of the underlying bone and soft tissues and muscle issues.

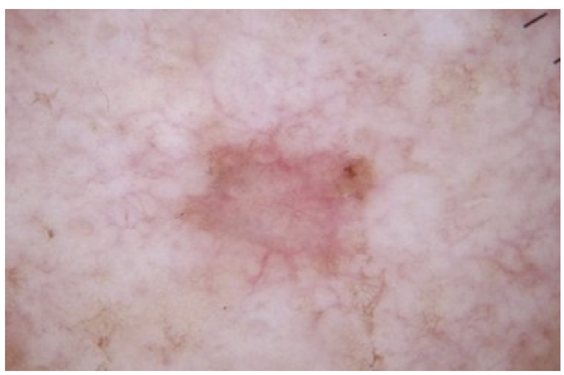


Figure 1.1:Actinic keratoses and intrapithelial carcinoma

1.5.2 BASAL CELL CARCINOMA

BCC is frequently observed in people having fair skin. Flesh-colored round growth or pinkish patch of skin is often treated as the symptoms of BCC.It mainly occurs due to indoor tanning or frequent sun exposure.BCC have high scopes to grow deep and can penetrate the nerves and bones causing them disfigurement and damage.

Figure 1.2:Basal cell carcinoma

1.5.3 BENIGN KERATOSES-LIKE LESIONS MELANOMA/ SEBORRHEIC KERATOSES

The asymmetric nature and the polychromic lesion, formed by the unarranged prominent pigmented network, multiple asymmetric pigmented globules and noises or dots on the skin and the periphery, and a conscious blue-whitish veil on the highlighted center of the lesion. Suspicious follicular openings can be identified by the relation of the most pigmented and raised area of the lesion.

Figure 1.3: Benign Keratoses

1.5.4 MELANOCYTIC NEVI

Melanocytic nevi represent the most common dermal melanocytic lesions in humans. They generally present as small (<0.6 cm), hyperpigmented macules located virtually anywhere on the skin surface. Three variants have been recognized, based on the location and distribution of the proliferating melanocites: junctional nivi(confined to the dermal-epidermal junction), dermal nevi (confined to the dermis), and compound nevi (dermis and epidermis).

Figure 1.4: Melanocytic Nevi

1.5.5 VASCULAR LESIONS

The vascular lesions type of lesions which are detectable and have a varied appearance and can commonly occur in the head and neck and in hand regions. The majority of these lesions are congenital; even though some may be acquired and malignant which are of less dangerous. These lesser flow lesions appear at the birth to infant stage as flat pink areas of skin coloration which is unusual and usually darken over time to become a dark or deep purple color. These unusual color lesions can eventually develop changes in skin texture and type with a stone effect and can also be associated with overgrowth of the underlying bone and soft tissues and muscle issues.

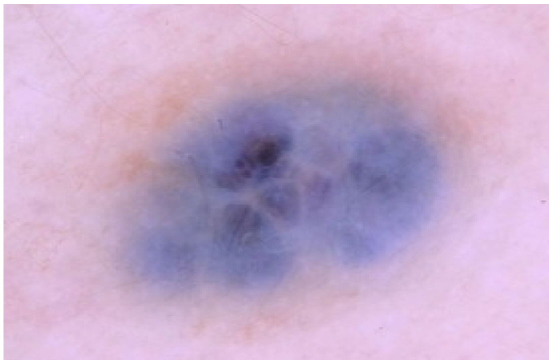


Figure 1.5 Vascular Lesions

1.5.6 DERMATOFIBROMA

The very firm bump that feels like a small rubbery button which is a kind of itchy lying just under the surface of the skin, often larger than .5 cm in radius. These can vary in color from purple to light pink and can sometimes appear in pale brown or grey. A dimple may appear over a dermatofibroma when pinched or rubbed. They can appear anywhere but are most common on the lower legs of the body for young to middle-aged adults, and on the upper arms and shoulder regions for females.

Figure 1.6 Dermatofibroma

1.5.7 MELANOMA

Melanoma is treated as “the most serious skin cancer” because of its high tendency to spread.Melanoma is developed within a mole which is already present on the skin or a sudden dark spot appears on the skin which seems different from the other.Diagnosing melanoma in the early stage helps to save the victim.Diagnosing last stages of melanoma is a challenging task to the dermatologists. ABCDE(Asymmetry, Border, Color, Diameter, Evolving) warning signs helps in early detection.

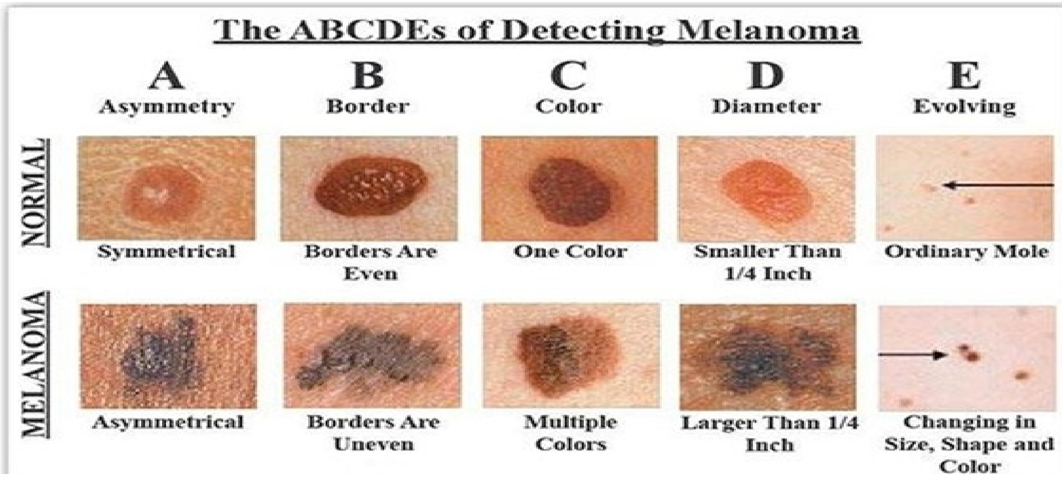


Figure 1.7 Melanoma

1.6 CONVOLUTION NEURAL NETWORKS

A Convolutional Neural Network (ConvNet/CNN)which is a Deep Learning calculation that retains an info picture, relegate significance (learnable loads and inclinations) to changed perspectives/objects inside the picture and be prepared to separate one from the inverse . The pre-handling needed during a ConvNet is route lower when contrasted with other grouping calculations. While in crude techniques channels are hand-designed, with enough preparing, ConvNets have the office to search out these channels/qualities.

The design of a ConvNet is undifferentiated from thereto of the availability example of Neurons inside the Human Brain and was enlivened by the association of the visual cortex . Singular neurons answer upgrades just during a limited locale of the area of vision referenced on the grounds that the Receptive Field. a bunch of such fields cover to cover the whole visual territory .

Layers of CNN

i. Input layer
ii. Convo layer(convolution+Relu)
iii. Pooling layer
iv. Fully connected(FC) layer
v. Softmax/logistic layer
vi. Output layer

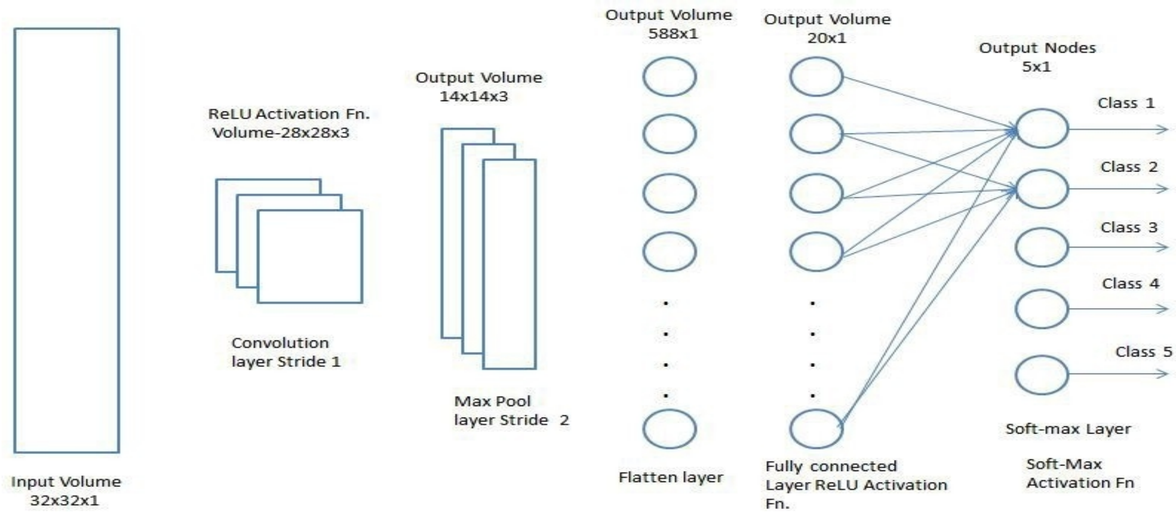


Figure 1.8 Architecture of Convolution Neural Networks

1.6.1 INPUT LAYER

Info layer in CNN ought to contain picture information. Picture information is addressed by a three dimensional framework as . You need to reshape it from a network to a solitary segment. On the off chance that you have a picture of measurement 28 x 28 =784, you need to change over it into 784 x 1 preceding taking care of it into input. Assuming you have "n" preparing models measurement of information will be (784, n).

1.6.2 CONVOLUTION LAYER

It is the initial phase during the time spent removing important highlights from a picture. The convolution layer has a few channels that play out the convolution activity. Each and every picture is considered as a lattice of pixel esteems.

Considering a image matrix of 5x5 picture whose pixel esteems are either 0 or 1. There is likewise a channel lattice with a component of 3x3. Slide the channel framework over the picture and afterward register the speck item to get the convolved include grid.

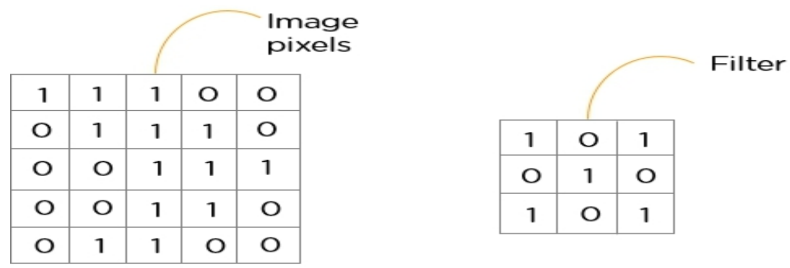


Figure 1.9 Convolution Layer

1.6.3 RELU LAYER

ReLU also known as Rectified Linear represents the amended direct unit. When all the component maps are removed, them the subsequent stage is to move them to a ReLU actuation layer.

ReLU plays out a component astute activity and afterward sets all the negative(- ve) pixels to 0. It acquaints non-linearity with the organization, and afterward the produced yield is a redressed include map. The following is the diagram of a ReLU initiation work:

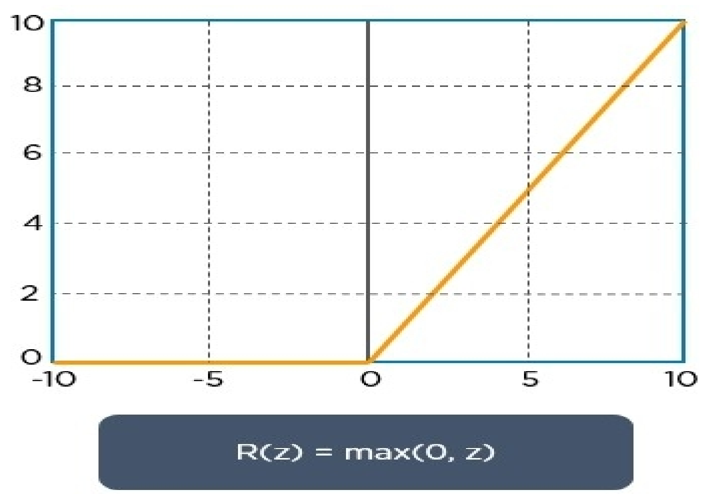


Figure 1.10 Relu Layer

1.6.4 POOLING LAYER

Pooling is one of the down-examining activities that decreases the dimensionality of the component map. The amended included guide presently goes through one pooling layer to create a pooled highlight map that it down examples and gives a pooled include map. The pooling layer utilizes different channels and bits to recognize various pieces of the picture like edges, corners, body and so on as per our prerequisite.

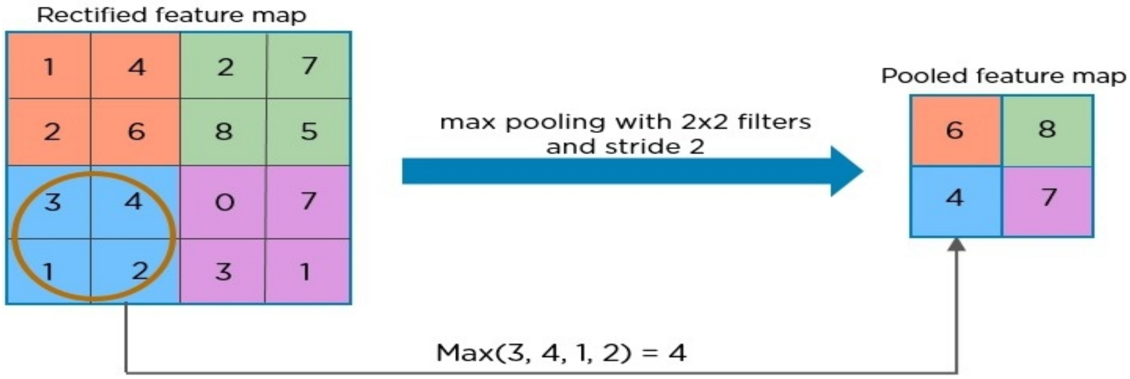


Figure 1.11 Pooling Layer

1.6.5 FLATTEN LAYER

The progression in the wake of pooling layer in the process is called smoothing. Straightening is utilized to change over every one of the resultant 2-Dimensional clusters which we got from pooled include maps into single long ceaseless direct vector which is the yield of smooth layer. The straightened framework is taken care of as contribution to the completely associated layer to arrange the picture in like manner.

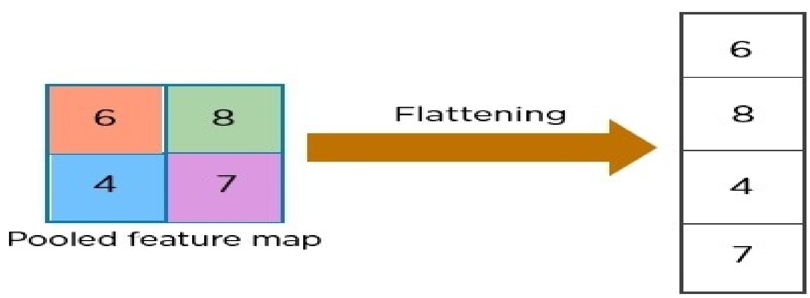


Figure 1.12 Flatten layer

1.6.6 FULLY CONNECTED LAYER

The Fully Connected (FC) layer comprises of the loads and predispositions alongside the neurons and is utilized to interface the neurons between two distinct layers of the convolution neural organization. Every one of these layers are generally positioned before the yield layer and structure the last couple of layers of a CNN Architecture.In the completely associated layer the information picture from the past layers are smoothed and taken care of to the FC layer. The smoothed vector at that point goes through barely any more Fully associated layers where the numerical capacities activities as a rule happen. In this specific stage, the grouping interaction starts to occur.

1.6.7 SOFTMAX/LOGISTIC LAYER

The softmax work, which is otherwise called softargmax or standardized outstanding capacity is a speculation of the strategic capacity to tasteful multi measurements. This softmax work is utilized in multinomial strategic regression(alike is the AI calculation) and is frequently utilized as the last enactment capacity of a neural organization to standardize the yield of an organization to a likelihood dissemination over anticipated yield classes, in light of Luce's decision adage.

1.6.8 OUTPUT LAYER

The output layer is the yield layer contains the mark which is as one-hot encoded which is the last yield required.

CHAPTER 2

LITERATURE SURVEY

2.1 A METHOD FOR MELANOMA SKIN CANCER DETECTION USING DERMOSCOPY IMAGES.

2.1.1 DESCRIPTION

The authors work significantly deals with the computer aided diagnosis for skin cancer detection.The proposed system preprocess the skin images and then segmentation is performed followed by feature extraction.Classification techniques are applied on the extracted features.In the pre-processing a skin image was taken as the input and this was converted to gray color. Pre-processing in this system also includes removal of hair follicles which was performed by Gaussian Filter.The Region of interest ROI was differentiated from the remaining in the image segmentation.Only the lesion part was treated as ROI which was separated from the healthy part using Otsu Threshold method. Otsu Threshold method will convert the image and causes irregular edges.Morphological filter was used for smoothing the edges. Feature extraction is a method to calculate the unique features of an image.It extracted features like color,perimeter area,irregularities and texture.A Deep Learning method Support Vector Mechanism(SVM) was used.The proposed system by the author was modeled using three Machine Learning functions:SVM, Bayes net Classifier, Radial Basis function.Out of which SVM Linear function observed the best accuracy of 92.30% whereas Bayes net observed an accuracy of 82% for 80% training and 20% testing of the data.

2.1.2 DRAWBACK

There was a high computational complexity.

2.2 DETECTION OF SKIN CANCER "MELANOMA" THROUGH COMPUTER VISION

2.2.1 DESCRIPTION

The author has developed image processing model to find asymmetry,border,color,diameters(ABCD) of the melanoma skin cancer.The author has generated an algorithm after analyzing a wide range of 200 different images.Moles on the skin plays a crucial role in melanoma. So the moles are analyzed and classified using Neural Networks.The features are extracted and classified the mole as common injury,not common or melanoma.Based on Asymmetry,a mole region area was determined by using Mumford-Shah which allows to find the diameter,center and area of the binary image.Harris-Stephens algorithm is used to determine the edge of the skin picture based on the intensity of the pixels. This determines the border aspect of the melanoma.The lesion was obtained by using two different methods. The first method converts an RGB image to HSV.This will complete a nonlinear transform of the RGB color space.And from this prevailing value of a mole to get injured is determined. In the second method the image was converted to grayscale which was a predominate in the image.But this method has a disadvantage as it does not focus on the light colors which are not highlighted.The color is analyzed based on the difference range of colors of melanoma and normal skin.Mumford-Shah was used to determine the value of the diameter in pixels.The extracted features are classified using Feedforward Neural Network(FNN).

If a mole was characterized as not common then it has uniform color,irregular borders,symmetry and diameter is less than 6mm.The algorithm resulted in a high accuracy of 97.51% which was obtained after analyzing 201 images.

2.2.2 DRAWBACK

The system showed a slow convergence rate.

2.3 A COLOR-BASED APPROACH FOR MELANOMA SKIN CANCER DETECTION

2.3.1 DESCRIPTION

The Writer implemented a new way of classification of skin lesion based on image processing.Generally skin lesions are possible to detect at early stages and treatement is done but for melanoma skin lesion it varies from other so the author developed a new method to overcome this problem.In this the author developed a system based on machine learning where datasets are collected , pre-processed,feature extraction,segmentation and classification is done using several algorithms.This system is developed by using a MED-NODE datasets of digital images.These data sets are preprocessed such as removing artifacts like hair.Then the segmentation takes place where Active Contour segmentation is used to identify the area of interest and to extract features like color,shape.After segmentation the segmented image is used for classification which is carried out by using three classifiers namely Naive Bayes,Decision Tree,KNN by utilizing these three classifiers the System is used for skin cancer detection.Consider this system to detect a skin cancer from given sample Basically First of all it collects MED-NODE datasets and Then it proceeds to pre-processing where the unnecessary artifacts present on skin are removed ex hair is removed using Dull-Razor Technique. After pre-processing segmentation is carried out by Active Contour segmentation after the conversion of original image into gray scale.After this segmentation the area of interest is obtained,such area is used for feature extraction by converting segmented RGB color into HSV and YCbCr color space.Then it proceeds for classification by using three classifiers Naive Bayes ,Decision Tree,KNN.Naive Bayes is an classification algorithm that works on Bayes therom i.e, it uses probability.Decision Tree is used for splitting of data into homogeneous tree.Whereas KNN is K-Nearest Neighbour which works on principle of centroid by storing in vectors.Thus this system achieves an accuracy of 82.35% on decision tree which is greater than any other algorithms.

2.3.2 DRAWBACKS

This System is effective only if there is a high contrast between lesion area.

2.4 DEEP LEARNING FOR TWO STEP CLASSIFICATION OF MALIGNANT PIGMENTED SKIN LESIONS

2.4.1 DESCRIPTION

The authors proposed method is mainly focused on the automatic detection of malignant pigmented skin lesions.A Deep Learning model is used first to classify the skins as melanocytic or non melanocytic.Which was then followed by deep learning methods to detect the malignant types.Convolution Neural Network (CNN) are used for detection,classification and segmentation of melanoma.A Fully convolutional network (FCN) has some drawbacks when compared to CNN, like a notable decrease in the number of parameters. A hybrid method is used which consists of CNN and Support Vector Machine(SVM) algorithm with a sparse coding method.This hybrid method was used for melanoma identification and feature extraction is performed using AlexNet based CNN.A large data set containing high resolution skin images has been investigated by deep Convolution based Neural Network.To solve the over fitting problem a fully connected layer network was introduced.A fully convolutional deep learning algorithm with 19 layers was presented for segmentation of melanoma skin lesions.Wavelet and Curvelet transformation transformation techniques are used for feature extraction. a tumor area extraction algorithm was used to differentiate Melanocytic and Non- Melanocytic skin lesions. Around 20000 clinical images are analyzed and classified using Deep learning algorithms and classified them into 12 skin diseases.SVM was used to randomly cross validate the features extracted.Using AlexNet, the classification of Melanocytic and Non Melanocytic skin lesions obtained an accuracy of 78%. Classification of Melanoma and Melanocytic Nevus Skin Lesions got an accuracy of 84%. Classification performance of Non Melanocytic malignant and Benign skin lesions have an accuracy of 58%.

2.4.2 DRAWBACKS

There was a high computational complexity observed in the system

2.5 SKIN DIEASE DETECTION BASED ON DIFFERENTSEGMENTATION TECHINUQUES

2.5 1 DESCRIPTION

In this the authors clearly states that the image can be processed using several segmentation techniques that gives an accurate outcome.The author's works clearly depicts of how an image is processed , segmented and Classification based on few classifiers. In this the author states the reasons for skin diseases to be treated early and he mentioned the people need to have effective solutions for skin diseases by comparing with tuberculosis and AIDS.First the image is carried out by pre-processing where the image is pre-processed by using individual component analysis of the data sets.The other are implemented a new system to detect skin tumors,psoriasis,vascular dermatosis using computer tomography techniques. The other authors implemented systems for edge prevention, noise removal, and unwanted hair removal.The algorithm is used for improvement in recognition of skin by pixels known like RGB,HSV,chrominance and luminance.BThe system implemented in this project is mainly focused on how to detect skin lesion like chicken pox,eczema,psoriasis mainly.This system utilizes the most important segmentation techniques in order to achieve better outcome,Thus it uses four algorithms namely Adaptive thresholding ,Edge Detection,K-means,Morphology based segmentation.The methodology behind this system is first the image is preprocessed i.e preserving of color contrasts,brightness other features of that image,Then image segmentation takes place here author implemented edge detection to detect the edges of the lesion area and to apply Gaussian filter and detects the edges of the lesion region by this we can detect ringworm.By Adaptive threshold we can detect chicken pox this works by identifying the clear region of lesion by identify the pixels.Similarly K-means is used for finding the cluster points by using centroids by this eczema is identified.Morphological is method used by implementing points to input image and data sets and compare them by this psoriasis is identified accurately.It utilizes signal to noise ratio efficiently.This System is most accurate and is gives exact identified region and also precise outcome whether disease is present or not.

2.5.2DRAWBACK

SNR values are not accurate and This System is slow due to four segmentation techniques.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 EXISTING SYSTEM DRAWBACKS

The following drawbacks are observed in the existing system

i. Large Training sets Required.

ii. Slow Convergence rate.

iii. High Computational Complexity.

iv. .High Contrast in lesion region is needed.

3.1.1 USE OF CLASSIFIED ALGORITHMS

The use of classified algorithms are not able to help in predicting the skin lesion accurately. This will lead to low accuracy in the results which make wrong impression on patient and doctors perception of doing medication. There are many algorithms that are in use and had many drawbacks.

-Principle component analysis

-local binary patterns and shape features

-KNN classifier

3.1.2 USAGE OF PRINCIPLE COMPONENT ANALYSIS

After data set is implemented with PCA, Principal Components will be the result and return for our original components.Our original features will be linear components with principle components. Principal Components are not as readable as original features.

3.1.3 USAGE OF LBP AND SHAPE FEATURES

In computer vision the classification takes in patterns area they are known as LBP. visual descriptors are have a part known as local binary patterns. To convert the RGB image into gray scale we need to constructing the LBP a type of texture descriptor. By taking the center pixel from the gray scale image for each pixel.

3.2 PROPOSED SYSTEM

In the proposed framework we will utilize a design of convolution neural organizations known as Mobilenet, where we will prepare our dataset and afterward we will perform picture increase procedures and afterward the picture preprocessing and the subsequent stage is to prepare the model, as mobilenet is a pre-prepared model, we can do move realizing which means accepting a pre-prepared model as a base and altering the pre-prepared model as per the prerequisite of the client, this will upgrade the nature of the model and afterward we perform testing.

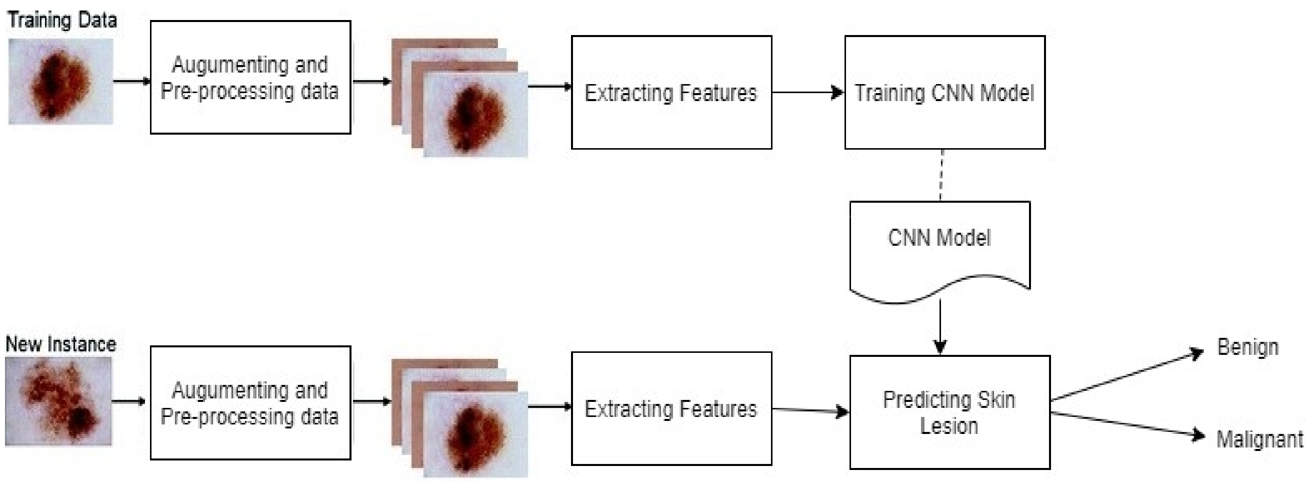


Figure 3.1 Proposed system

3.2.1 ADVANTAGES:

- The performance will result with high accuracy.
- Achieving best results with less train set.
- Effective features extractions.

CHAPTER 4
SYSTEM ANALYSIS

4.1 SOFTWARE REQUIREMENTS SPECIFICATION

4.1.1 SOFTWARE REQUIREMENTS

- Operating system : Windows 10
- Programming Language: Python 3.8
- IDE : IDLE
- Dependencies : numpy, pandas, cv2, tensorflow, keras, sklearn, matplotlib, Flask.

4.1.2 HARDWARE REQUIREMENTS

- Processor : Intel processor
- RAM :8GB
- Disk space : 1TB
- CPU :4GB

4.1.3 NON-FUNCTIONAL REQUIREMENTS:

- The limitations on the administrations or capacities offered by the framework like planning requirements, usability,, imperatives on the improvement interaction, principles, and so on Apply to the framework as entirety.
- Non-functional requirements in this system are:

  1. Reliability: Reliability based on this system defines the evaluation result of the system
  2. Ease of Use: The system is simple, user friendly, so any can use this system without any difficulties.

4.2 SOFTWARE DESCRIPTION

4.2.1 IDE

IDLE-PYTHON 3.8

IDLE is a Python's Integrated Development and Learning

Environment. IDLE has the accompanying highlights::

- IDLE is coded in 100% unadulterated Python, utilizing the tkinter which ia a GUI tool stash.
- cross-stage: works generally something very similar in all Windows, Unix, and macOS
- Python shell window with colorizing of code information, yield, and blunder messages will be shown
- multi-window content manager with various fix, Python colorizing, brilliant indent, call tips, auto finishing, and different highlights are comprised in python IDLE
- search inside any window, supplant inside manager windows, and search through numerous records (grep) are uncommon highlights of this IDE.
- debugger with tireless breakpoints, venturing, and review of worldwide and neighborhood namespaces are likewise comprised in it.
- configuration, programs, and different discoursed are unique highlights of python IDLE

4.2.2 LIBRARIES

4.2.3 NUMPY:

NumPy is a Python library utilized for working with varieties of any dimensions.Numpy moreover capacities for working in the area of direct polynomial math and matrices.NumPy was made in 2005 by Travis Oliphant. It is an open source undertaking and you can utilize it unreservedly and numerous mathematical capacities should be possible easily.NumPy represents Numerical Python.In Python we have records that fill the need of exhibits, however they are delayed in measure and less dependable so we use numpy clusters to tackle these problems.NumPy expects to give an array(one or two measurements) object that is up to 50x quicker than conventional Python lists.The cluster object in NumPy is called ndarray, which gives a great deal of supporting capacities that make working with ndarray exceptionally simple and speedy.

PANDAS:

Pandas is one of the Python library utilized for working with informational collections. It has capacities for breaking down, cleaning, investigating, controlling and numerous different capacities that manages information. The name "Pandas" which has reference to both "Board Data", and "Python Data Analysis" and was made by Wes McKinney in mid 2008. Pandas permits the clients to examine enormous information and make ends dependent on factual speculations which make them simple to consider and perform required tasks. Pandas can clean untidy informational indexes, and make them meaningful and applicable that significantly utilizes pandas as pertinent information is vital to an information researcher.

CV2:

OpenCV is one of the gigantic open-source libraries for PC vision, AI, and picture handling which have a wide scope of utilizations. It upholds many programming dialects like Python, C++, Java, and so forth OpenCV can handle pictures and recordings to recognize items, faces, or even the penmanship of a human as per the prerequisite of the client. At the point when it is incorporated with different libraries, for example, Numpy or whatever other libraries which is a profoundly improved library for mathematical tasks, at that point the quantity of weapons expansions in your Arsenal i.e whatever activities one can do in Numpy can be joined with OpenCV and that is the motivation behind why OpenCV is one of the amazing libraries

TENSORFLOW:

Tensorflow is the center open source library to assist clients with creating and train Machine Learning models. It helps in beginning rapidly by running Colab journals straightforwardly in your program. TensorFlow gives an assortment of work processes to create and prepare models utilizing Python or JavaScript, and to effectively convey in the cloud in the program, or on-gadget regardless of what language you use or on the nearby host. The tensorflow information API empowers you to construct complex information pipelines from straightforward, reusable pieces which is the fundamental target of the library

KERAS:

Keras is one of the moderate Python libraries for profound discovering that can run on top of on TensorFlow. It is created to carry out profound learning and neural organizations models as quick and as simple as feasible for innovative work. It manages enormous datasets in a quick and dependable manner.Keras is an API intended for people for better and simple utilization, not machines. Keras follows the best and substantial practices for diminishing burden and it offers steady and basic APIs, it limits the quantity of client activities needed for basic use cases, and it gives clear and noteworthy criticism upon client blunder and manages high dataset as per the prerequisite of the user.This makes Keras simple to learn and simple to utilize and straightforward. The convenience of keras doesn't come at the expense of decreased adaptability: since Keras coordinates profoundly with low-level TensorFlow usefulness, it truly empowers you to grow exceptionally got work processes where any piece of usefulness can be tweaked.

SKLEARN:

Scikit-learn (Sklearn) is the most valuable and strong libraries for AI and profound learning in Python. It gives a wide reach and determination of proficient devices for AI, profound learning and measurable demonstrating including arrangement, relapse, bunching and dimensionality decrease by means of a constant interface in Python. Tensorflow can likewise be an option for certain measurements for scikit-learn in profound learning.

MATPLOTLIB.PYPLOT:

Pyplot is an assortment of capacities in the famous perception bundle called Matplotlib used to plot various kinds of diagrams like histogram and so on Its capacities control components like making a figure, making charts, making a plotting territory, plotting lines, adding plot marks, and so on

FLASK:

Flask (source code) is a web structure worked with a little center and simple to-broaden reasoning in python. Flagon is a lightweight WSGI web application structure. It is intended to make beginning speedy and direct , with the ability to extent to complex applications. It started as a basic covering around and has gotten one among the first famous Python web application systems. Carafe proposes thoughts, however never uphold any conditions or undertaking format. It is dependent upon the engineer to choose the devices and libraries they need to utilize. There are a large number given by the designers that make adding new usefulness simple.

CHAPTER 5
SYSTEM DESIGN

5.1 INPUT DESIGN

Input Design is the manner by which an info picture is continued in the whole framework i.e from Collection of informational index then training,algorithm implementation,testing then at last grouped yield is given

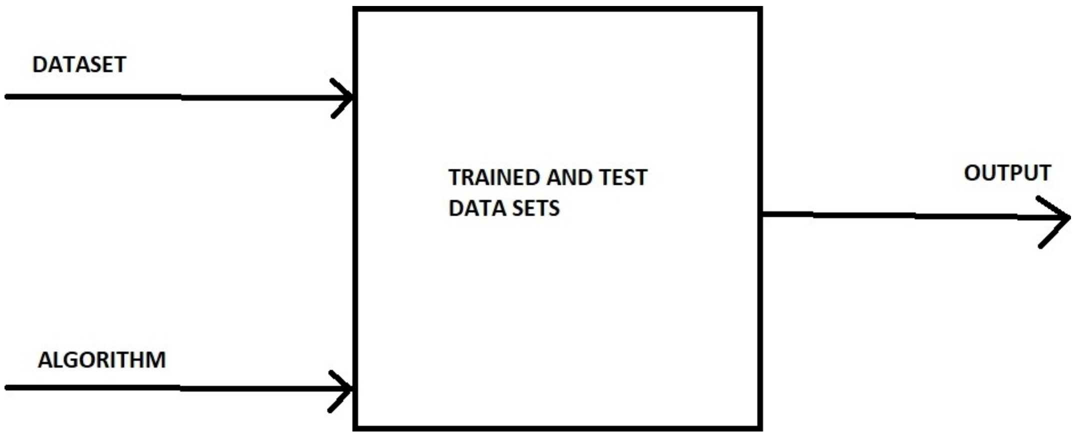


Figure 5.1 Input design

5.2 UML DIAGRAMS

5.2.1 ACTIVITY DIAGRAM

An activity diagram is a pictorial portrayal used to address the conduct of system.This chart used to portray stream of framework from start point and to end point. This graph shows the different ways how a control stream must be show in contains different ways followed by choice conditions.

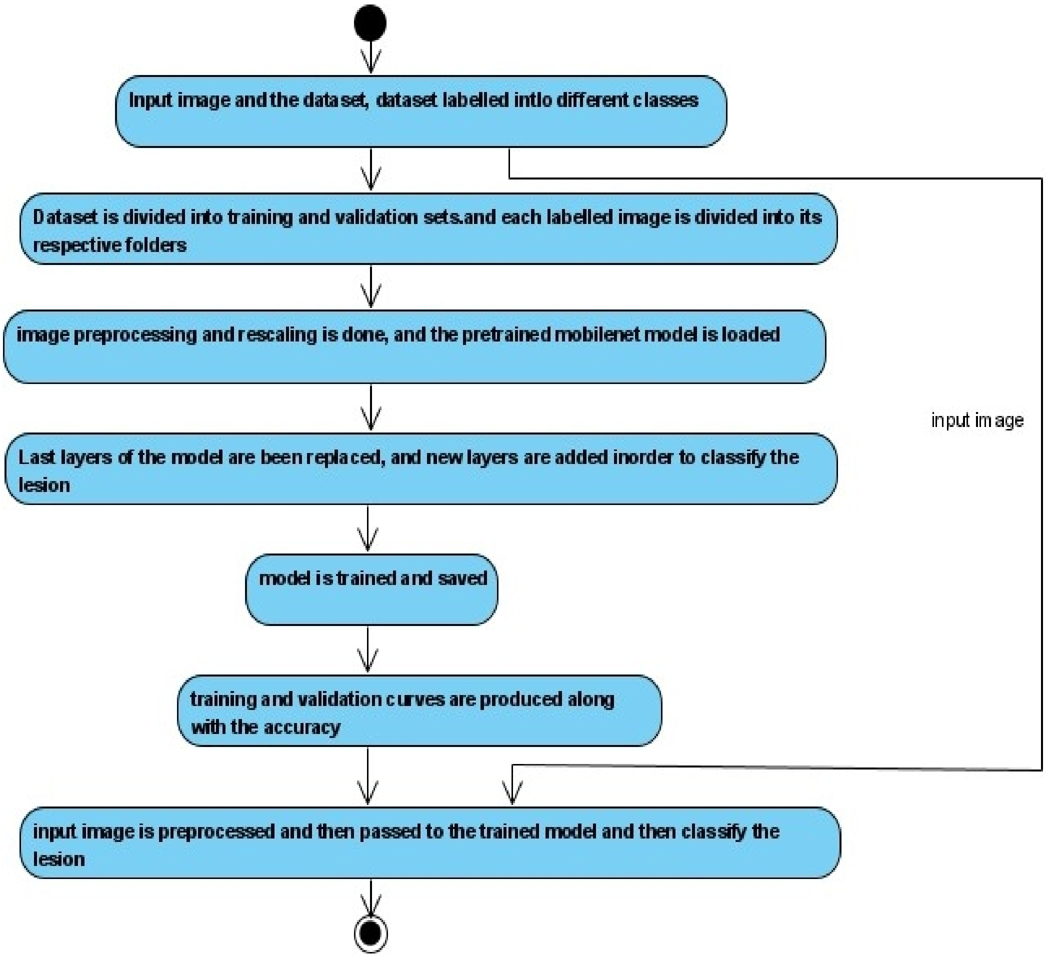


Figure 5.2 Activity diagram

5.2.2 USECASE DIAGRAM

Use Case Diagram is the simpler method of addressing the client association with the system. This is utilized as connection among framework and use cases.Use cases are individuals or entertainers using that framework and usecase is very beneficial and useful in UML.

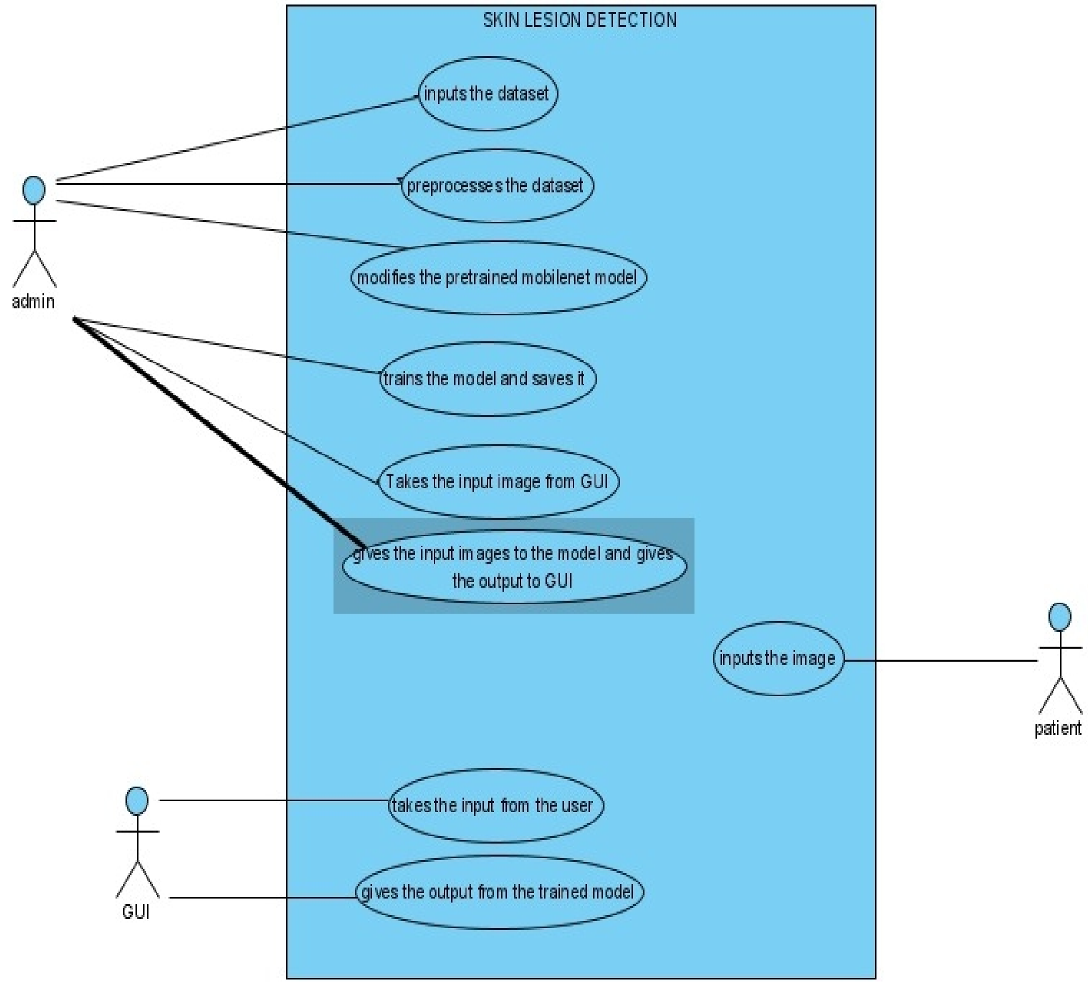


Figure 5.3 Use case diagram

Here the actors are the administrator, the GUI and the patient, where the administrator does the model preparing and saves the model and the patient who inputs the picture and the GUI who go about as the connector between the patient and the administrator in order to take the info and to give the yield.

5.2.3 SEQUENCE DIAGRAM

UML Sequence Diagrams are the communication charts that detail how activities and the cooperations between them are completed. They catch the connection between objects in the unique circumstance and their correspondence . Succession Diagrams are time situated and they shows the game plan and the request and the arrangement of how they cycle of the connection outwardly by utilizing the upward hub called as life saver of the chart to address time what messages are sent and when they are sent the sequence is created for the image diagram.

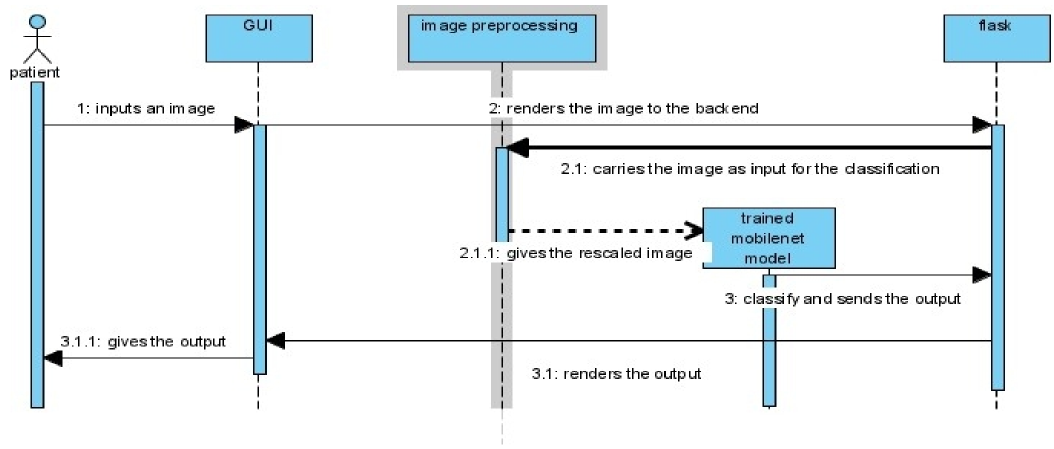


Figure 5.4 sequence diagram

The sequence diagram above shows the sequence how the project works from the patient input the image and then he gets the output, and the sequences in between show how the actual collaboration and communications between the actors have taken place.

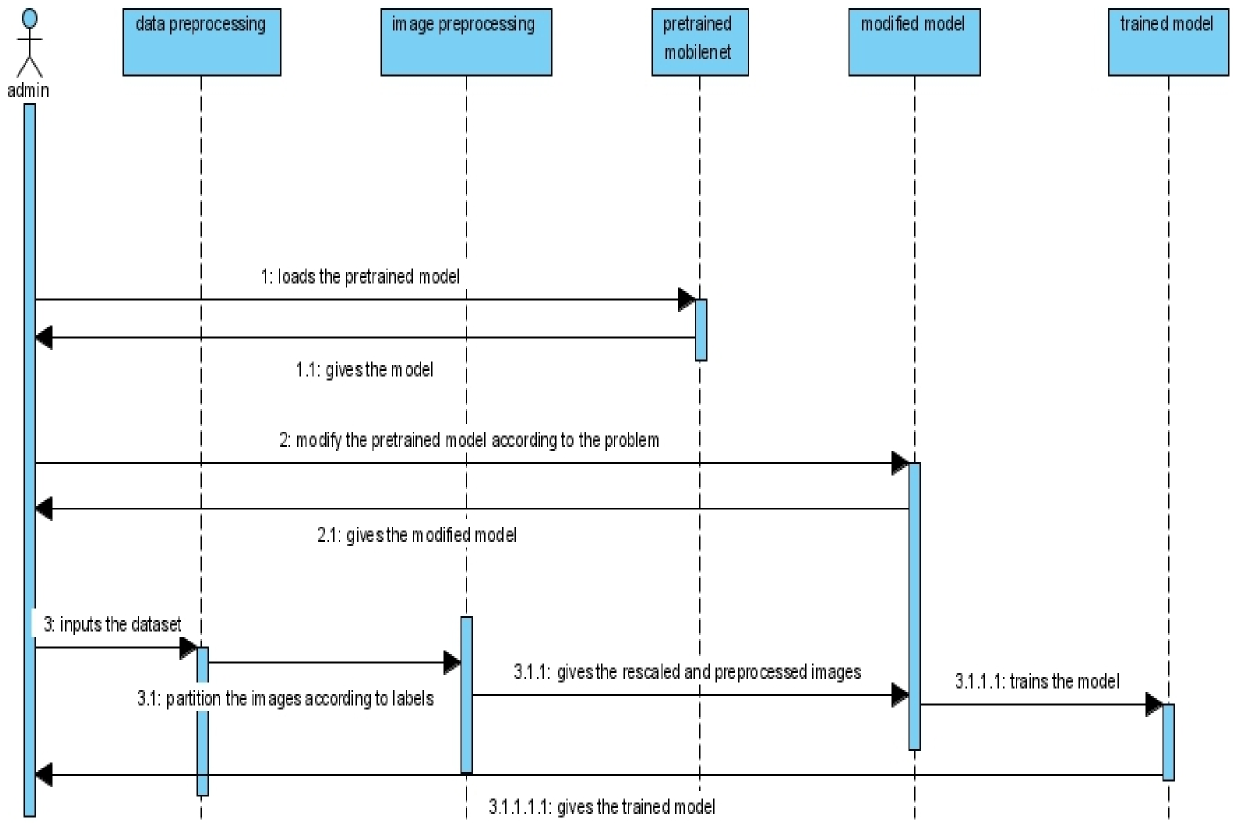


Figure 5.5 sequence diagram for the model

The above sequence diagram shows how the model is created and saved

5.2.4 DATA FLOW DIAGRAM

The data flow diagram describes the flow of the process in different levels, where the first level is the level 0 and then the sequential level numbers comes to place

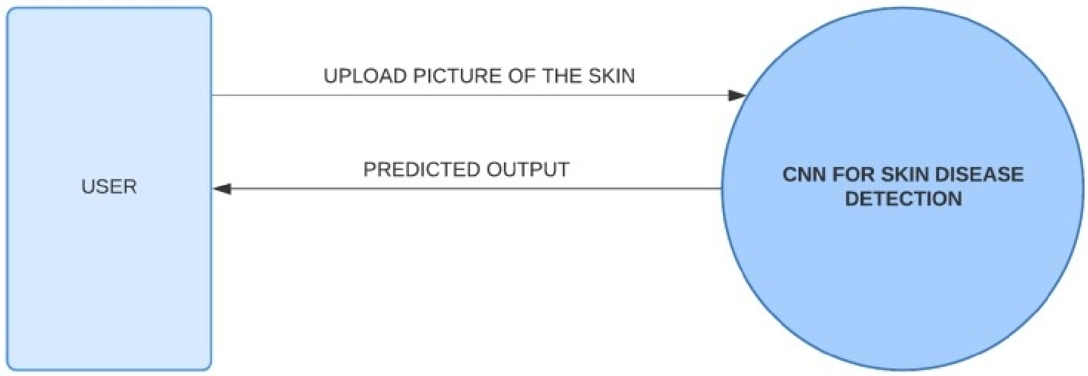


Figure 5.6 data flow diagram level-0

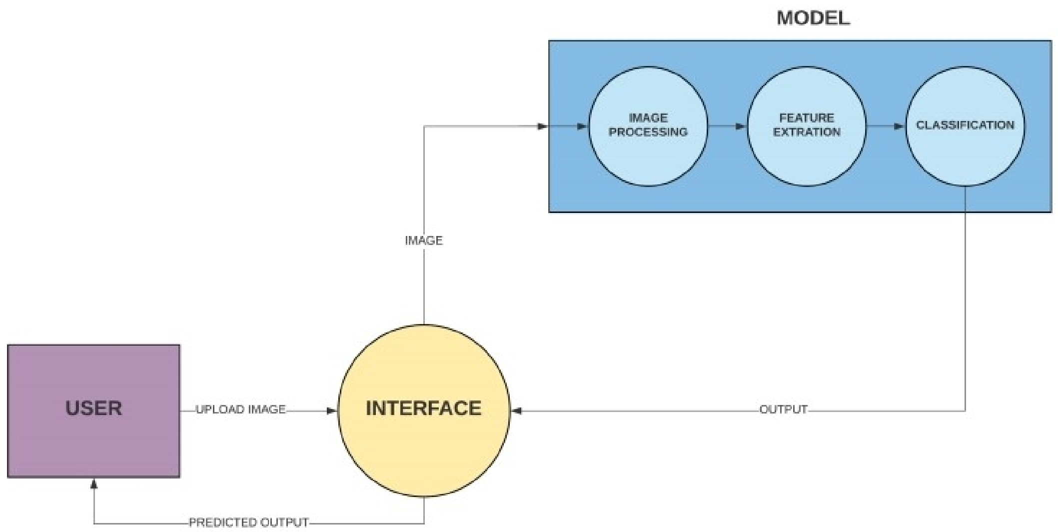


Figure 5.7 data flow diagram level-1

CHAPTER 6
MODULAR ANALYSIS

6.1 DATA COLLECTION

Data Collection is a method which we get all the information that is needed for project. The gathered information is put away in an information base which is known as informational collection. Informational collection that is put away is utilized for preparing and testing the counterfeit neural organizations and calculations. Information is gathered from various areas and sources contains crude and high commotion content. So,the information should be pre prepared.

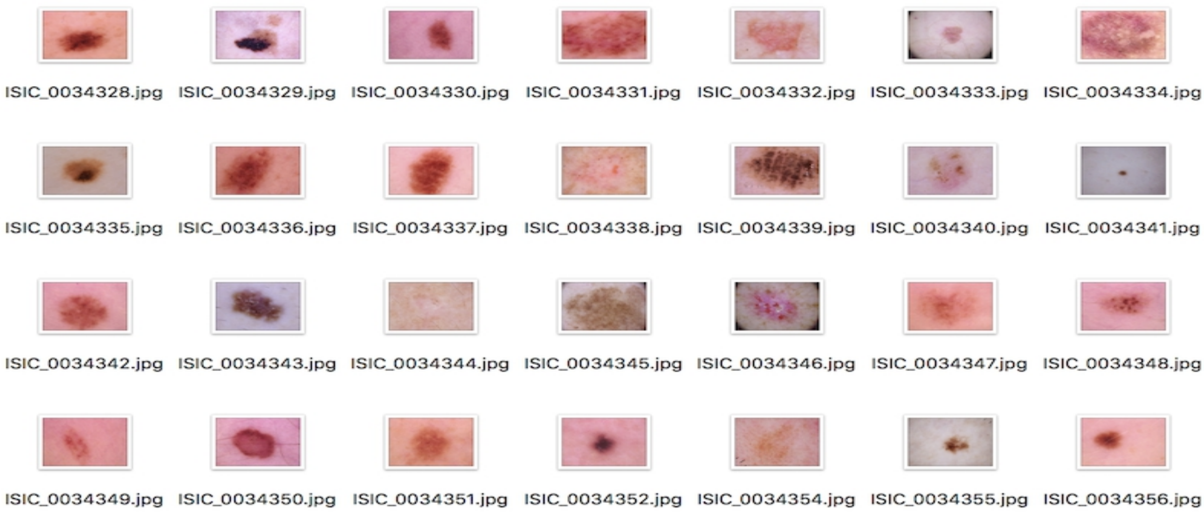


Figure 6.1 Data collection

Cases include a representative collection of all important diagnostic categories in the realm of pigmented lesions:

1.Actinic Keratoses and intraepithelial carcinoma(akiec)
2.basal cell carcinoma (bcc)
3.benign keratosis-like lesions melanoma (bkl)
4.melanocytic nevi (nv)
5.vascular lesions(vasc)
6.dermatofibroma(df)
7.melanoma(mel)

	A	B	C	D	E	F	G
1	lesion_id	image_id	dx	dx_type	age	sex	localization
2	HAM_0000118	ISIC_0027419	bkl	histo	80	male	scalp
3	HAM_0000118	ISIC_0025030	bkl	histo	80	male	scalp
4	HAM_0002730	ISIC_0026769	bkl	histo	80	male	scalp
5	HAM_0002730	ISIC_0025661	bkl	histo	80	male	scalp
6	HAM_0001466	ISIC_0031633	bkl	histo	75	male	ear
7	HAM_0001466	ISIC_0027850	bkl	histo	75	male	ear
8	HAM_0002761	ISIC_0029176	bkl	histo	60	male	face
9	HAM_0002761	ISIC_0029068	bkl	histo	60	male	face
10	HAM_0005132	ISIC_0025837	bkl	histo	70	female	back
11	HAM_0005132	ISIC_0025209	bkl	histo	70	female	back
12	HAM_0001396	ISIC_0025276	bkl	histo	55	female	trunk
13	HAM_0004234	ISIC_0029396	bkl	histo	85	female	chest
14	HAM_0004234	ISIC_0025984	bkl	histo	85	female	chest
15	HAM_0001949	ISIC_0025767	bkl	histo	70	male	trunk
16	HAM_0001949	ISIC_0032417	bkl	histo	70	male	trunk
17	HAM_0007207	ISIC_0031326	bkl	histo	65	male	back
18	HAM_0001601	ISIC_0025915	bkl	histo	75	male	upper extremity
19	HAM_0001601	ISIC_0031029	bkl	histo	75	male	upper extremity
20	HAM_0007571	ISIC_0029836	bkl	histo	70	male	chest
21	HAM_0007571	ISIC_0032129	bkl	histo	70	male	chest
22	HAM_0006071	ISIC_0032343	bkl	histo	70	female	face
23	HAM_0003301	ISIC_0025033	bkl	histo	60	male	back
24	HAM_0003301	ISIC_0027310	bkl	histo	60	male	back

Figure 6.2 Dataset

The final dataset consists of 10015 dermatoscopic images which can serve as a training set for academic machine learning purposes. lesion_id is the id of the lesion given by HAM and the image_id is the id for each image, More than 50% of lesions are confirmed through histopathology (histo), the ground truth for the rest of the cases is either follow-up examination (follow_up), expert consensus (consenses), or confirmation by in-vivo confocal microscopy (confocal).

6.2 DATA PREPROCESSING:

In data preprocessing we will be considering the .csv file and checking and handling the missing or duplicate values, and then we will be splitting the data set to 2 sub directories namely train_dir and the val_dir in which each contains the images of all the 7 classes these splitting is done by using the csv file.

6.2.1 IMAGE PREPROCESSING:

In the picture preprocessing we will rescale the picture, as portable net will be taking an info which is of size(224,224) which signifies the pixels and the other preprocessing strategies which are available in the most required and significant library keras.applications.mobilenet.preprocess_input, this library alters the picture as per the necessity of the mobilenet design, and some different procedures are turning a picture flipping a picture and so forth Furthermore, a famous method known as picture information argumentation which is a procedure that can be utilized to misleadingly extend the size of a preparation dataset by making altered adaptations of the pictures which are available in the dataset . Picture information expansion is utilized to grow the preparation dataset to improve the presentation and capacity of the model to sum up as the more no.of information more right the model will be.

6.3 MODEL BUILDING

6.3.1 TRANSFER LEARNING

Transfer learning is a methodology where a model executed for a specific undertaking is reused as the underlying point and to carry out different errands by changing it which increments is the effectiveness.Transfer learning is one of the famous methodologies in profound learning and AI where an all around prepared models are utilized as the beginning stage on PC vision and normal language handling assignments or any picture order undertakings.

As order assignments given the immense figure and time assets needed to create neural organization models on these issues inorder to decrease every one of the intricacies move learning is a decent arrangement.In this model I took a pre-prepared organization called MobileNet(a Convolution Neural Network design) which is prepared with ImageNet, a dataset of more than 14 million pictures inorder to build the quantity of pictures, and added some more layers and eliminate some layer and supplant it with other so it can characterize skin injuries.

The principle thought behind this is that we take a model that was prepared with a colossal measure of information and utilize that for another assignment where we don't have as much information or when we have less information this comes helpful. By utilizing this exchange learning as the model is pretrained with numerous pictures it works great with the client gave informational index as well as to the continuous data sources.

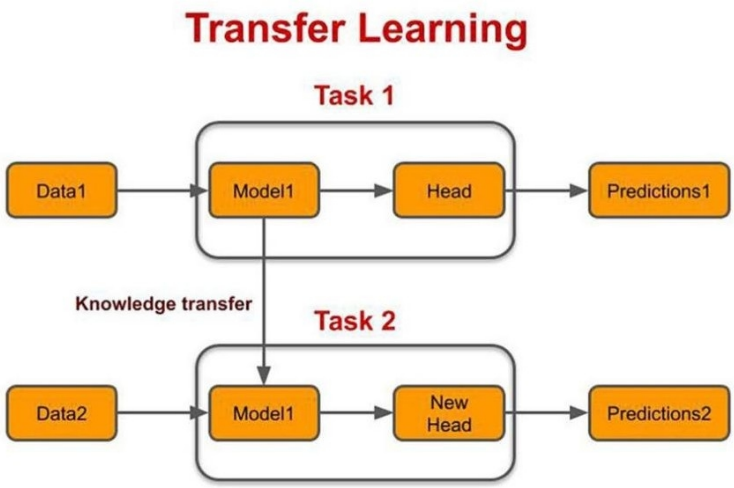


Figure 6.3 Transfer Learning

6.3.2 FEATURE EXTRACTION AND MODEL TRAINING

6.3.2.1 MOBILENET ARCHITECTURE

Highlight Extraction Module: The module comprises of performing tasks like convolution, max pooling and ReLu. In view of the prerequisites, this layer may broaden. The target of the convolution activity is to extricate the significant level highlights like shape and surface, from the information picture. Max Pooling returns the most extreme worth from the picture covered by the Kernel. Max Pooling additionally proceeds as a Noise Suppressant. It eliminates the clamors and furthermore performs de-noising alongside dimensionality decrease. ReLU is an actuation work that has solid natural and numerical activities.

Classifier Module: The module comprises of thick layer, dropout layer and softmaxlayer. Dropout layer is a method used to improve over-fitting on neural networks. During forecast, the dropout layer is deactivated. Thick layer is trailed by a non-direct activation. The framework can be comprehensively arranged into following significant stages: Pre-preparing the pictures, testing andpreparing.

Coming to the Mobilenet architecture,

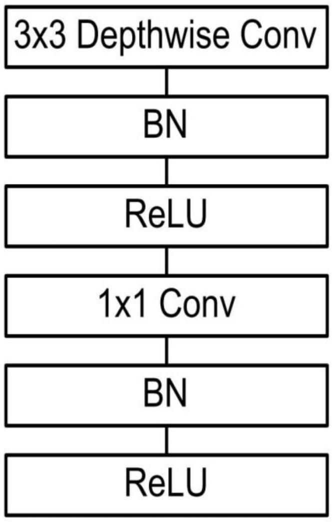


Figure 6.4 view of mobilenet

By survey the design, it is seen that that each convolution layer followed by one cluster standardization and a ReLU actuation work. Additionally, in the last a normal pooling is been presented before the completely associated layer to diminish the spatial measurement to 1.

Clump standardization might be a strategy for preparing extremely profound neural organizations that normalizes the contributions to a layer for every scaled down group. This settles the preparation cycle and drastically lessening the measure of instructing ages needed to mentor profound organizations.

MobileNet is a Convolution Neural Network engineering model for Image Classification, text acknowledgment and numerous different errands in Mobile Vision. Contrasting and numerous different structures Mobilenet is more dependable and quick since it utilizes profundity savvy distinct neural organizations. MobileNets which hails from the group of portable first PC vision models for TensorFlow, intended for working proficiently, expanding the exactness while thinking about the assets and the information.. MobileNets are little, low-dormancy, low-power models that is the motivation behind why they are very utilizing in utilizing. Mobilenet can be based upon for characterization, identification, embeddings and division like how other mainstream enormous.

MobileNet utilizes depthwise divisible convolutions. As mobilenet choosily diminishes the quantity of boundaries when contrasted with the other organization with customary convolutions with similar profundity in the nets. This outcomes in lightweight profound neural organizations which is an additional benefit .

A depthwise separable convolution is made up from two operations.

Depthwise convolution.

Pointwise convolution.

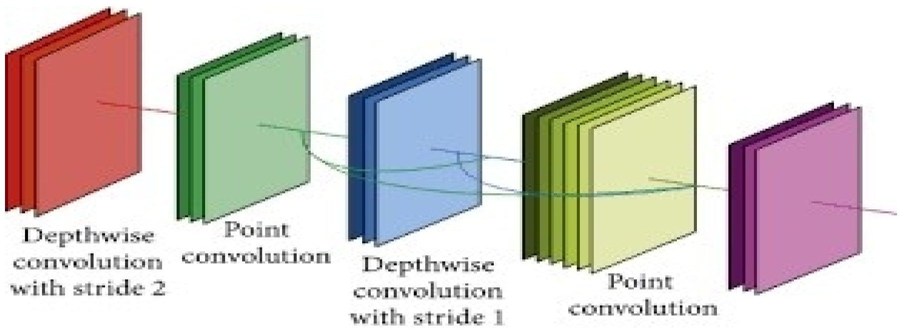


Figure 6.5 working flow of mobilenet

The actual flow goes like this, firstly the layer is depth wise separated and then they are combined but the point wise convolutions.

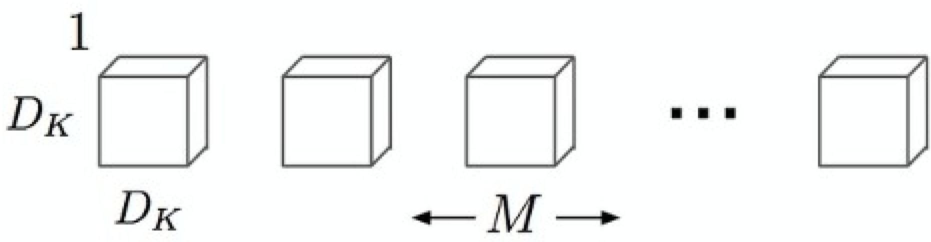


Figure 6.6 Depth wise separable

This is how the depth wise convolution looks like with cost Df^2*M*Dk^2.. After this a point wise convolution is done inorder to combine.

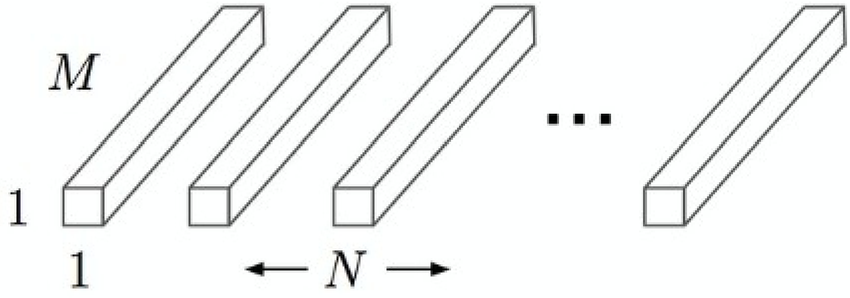


Figure 6.7 Point wise separable

This is how point wise convolution looks like witha a cost of M*N*Df^2.

In this undertaking, We use MobileNet,which has 28 layers by default. We determined Dense what's more, dropout.The class sensitivities are additionally specified.The model is instated with pre-prepared weights.This project is a 4 class grouping profound learning convolutional model which acknowledges picture in 224 x 224 measurements and 3 channels (RGB).The yield expectation of this model is finished by the assistance of worldwide normal pooling(GAP) layers to limit overfitting by diminishing the absolute number of boundaries in the model and thick layer actuated with softmax, hense if any instance of halfway coordinating to another class,we can recognize it by assessing the qualities given by every hub in layer. As of now we are taking the most biggest of the qualities and file it to get the illness class.

6.4 TESTING

The advancement of an undertaking incorporates consecutive strides of beneficial exercises and testing is a significant action of them. The testing stage is basic stage and component of programming quality affirmation and addresses a definitive survey of particular, coding and testing.

The primary goals of testing are as per the following:

1) Testing is a cycle of executing a program with the plan of discovering a blunder or will figure out how to manage that mistake.

2)A great experiment is one that has a high likelihood of tracking down an unseen mistake that is the sole motivation behind testing.

3) Successful test is one which have the capacity of revealing an unseen mistake.

In testing we will stack the recently saved model, and afterward we will include a picture and in the testing interaction the picture will again go through all the picture preparing procedures and afterward the picture is given to the saved model and after we will get the ordered yield by the model to the client.

6.5 DEPLOYMENT

We are utilizing the Flask system for making a web application. Flagon is a web system that gives a few devices, libraries, and advancements that permit us to assemble a web application. Carafe gives us a formats envelope, a static organizer where we can incorporate our HTML record and CSS document required by the web application. By utilizing these organizers, instruments, libraries given by the Flask system, we can make a web API.

We are utilizing werkzeug.utils.secure_filename module in back-end which helps in returning a safe rendition of a record when a specific filename is passed to this module. This safe form of the record can then securely be put away on an ordinary document framework and passed to the way utilizing os.path.join(). For accomplishing most extreme conveyability, this filename will be returned uniquely in an ASCII just string. The module likewise guarantees that the record isn't named after one of the specific gadget documents on windows frameworks.

CHAPTER 7

IMPLEMENTATION

```
import pandas as pd
import numpy as np
import keras
from keras import backend as K
from keras.layers.core import Dense, Dropout
from keras.optimizers import Adam
from keras.metrics import categorical_crossentropy
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Model
from keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint

import os

from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
import itertools
import shutil
import matplotlib.pyplot as plt
```

These are the libraries that are required for our project code

7.1 HANDLING THE DATASET

7.1.1 HANDLING THE .csv FILE

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

Figure 7.1 head of the .csv file

```
# here we identify lesion_id's that have duplicate images and those that have only
# one image.

def identify_duplicates(x):

    unique_list = list(df['lesion_id'])

    if x in unique_list:
        return 'no_duplicates'
    else:
        return 'has_duplicates'

# create a new colum that is a copy of the lesion_id column
df_data['duplicates'] = df_data['lesion_id']
# apply the function to this new column
df_data['duplicates'] = df_data['duplicates'].apply(identify_duplicates)
```

And now we will find the lesion_id s which have duplicate images with only one image, so after that we will split the train test images inorder to split the images for testing without having duplicates.

```
# now we create a val set using df because we are sure that none of these images
# have augmented duplicates in the train set
y = df['dx']

_, df_val = train_test_split(df, test_size=0.17, random_state=101, stratify=y)

df_val.shape
df_val['dx'].value_counts()
```

This is the test split

```
# This set will be df_data excluding all rows that are in the val set

# This function identifies if an image is part of the train
# or val set.
def identify_val_rows(x):
    # create a list of all the lesion_id's in the val set
    val_list = list(df_val['image_id'])

    if str(x) in val_list:
        return 'val'
    else:
        return 'train'
```

Identifying the parts of train test split

7.1.2 HANDLING THE IMAGE FOLDERS

```
for image in train_list:
    fname = image + '.jpg'
    label = df_data.loc[image,'dx']
    if fname in folder_1:
        # source path to image
        src = os.path.join(r'C:\Users\Dell\OneDrive\Desktop\project\New folder\HAM10000_images_part_1', fname)
        # destination path to image
        dst = os.path.join(train_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src, dst)

    if fname in folder_2:
        # source path to image
        src = os.path.join(r'C:\Users\Dell\OneDrive\Desktop\project\New folder\HAM10000_images_part_2', fname)
        # destination path to image
        dst = os.path.join(train_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src, dst)
```

Now we are setting the images to their respective folders and transfering the images accordingly.

```
for image in val_list:
    fname = image + '.jpg'
    label = df_data.loc[image,'dx']
    if fname in folder_1:
        # source path to image
        src = os.path.join(r'C:\Users\Dell\OneDrive\Desktop\project\New folder\HAM10000_images_part_1', fname)
        # destination path to image
        dst = os.path.join(val_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src, dst)
    if fname in folder_2:
        # source path to image
        src = os.path.join(r'C:\Users\Dell\OneDrive\Desktop\project\New folder\HAM10000_images_part_2', fname)
        # destination path to image
        dst = os.path.join(val_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src,dst)
```

Same as the handling of the train images we does the same to the test images.

7.2 IMAGE AUGMENTATION

For the training dataset image, they have to undergo some process to get high accuracy, and good model, so inorder to do that and to increase the dataset images we use image augmentation, which have the techniques like flipping the image rotating etc, that is the advantage of image augmentation.

```
# Create a data generator
datagen = ImageDataGenerator(
    rotation_range=180,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    #brightness_range=(0.9,1.1),
    fill_mode='nearest')batch_size = 50 aug_datagen = datagen.flow_from_directory(path,
                                   save_to_dir=save_path,
                                   save_format='jpg',
                                            target_size=(224,224),
                                            batch_size=batch_size)
```

This is the code for image augmentation, where we use ImageDataGenerator, is the method keras preprocessing library which generates the augmented image and push them randomly, so that the model does not know whether the image is augmented

7.3 IMAGE PREPROCESSING

As mobilenet can only train models if the target size is 224 pixles, and a method called mobilenet.preprocess will take care of the preprocessing of the image, and we take an RGB image as an input, and this mobilenet.preprocess is an library from keras.

```
train_batches = ImageDataGenerator(
    preprocessing_function= \
    keras.applications.mobilenet.preprocess_input).flow_from_directory(
                                                    train_path,
                                                    target_size=(image_size,image_size),
                                                    batch_size=train_batch_size)
valid_batches = ImageDataGenerator(
    preprocessing_function= \
    keras.applications.mobilenet.preprocess_input).flow_from_directory(
                                                    valid_path,
                                                    target_size=(image_size,image_size),
                                                    batch_size=val_batch_size)
```

We will do the preprocessing for the both train and the validation data, and to generate those preprocessed images we will use a function called ImageDataGenerator.

7.4 MODEL BUILDING

```
mobile = keras.applications.mobilenet.MobileNet()
```

The above code creates a copy of the mobilenet architecture

```
_________________________________________________________________
conv_dw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_pw_13 (Conv2D)          (None, 7, 7, 1024)        1048576
_________________________________________________________________
conv_pw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_pw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1024)              0
_________________________________________________________________
reshape_1 (Reshape)          (None, 1, 1, 1024)        0
_________________________________________________________________
dropout (Dropout)            (None, 1, 1, 1024)        0
_________________________________________________________________
conv_preds (Conv2D)          (None, 1, 1, 1000)        1025000
_________________________________________________________________
reshape_2 (Reshape)          (None, 1000)              0
_________________________________________________________________
predictions (Activation)     (None, 1000)              0
=================================================================
Total params: 4,253,864
Trainable params: 4,231,976
Non-trainable params: 21,888
_________________________________________________________________
```

Figure 7.2 model summary of mobile net

```
# CREATE THE MODEL ARCHITECTURE

# Exclude the last 5 layers of the above model.
# This will include all layers up to and including global_average_pooling2d_1
x = mobile.layers[-6].output

# Create a new dense layer for predictions
# 7 corresponds to the number of classes
x = Dropout(0.25)(x)
predictions = Dense(7, activation='softmax')(x)

# inputs=mobile.input selects the input layer, outputs=predictions refers to the
# dense layer we created above.

model = Model(inputs=mobile.input, outputs=predictions)
```

The above code is the modification for the model, where we will exclude the last 5 layers and add our own customized labels inorder to built our model, and we use a droupout layer inorder to avoid overfitting the model, so droupout function randomly drops every 1 in 5 neurons.

```
# We need to choose how many layers we actually want to be trained.

# Here we are freezing the weights of all layers except the
# last 23 layers in the new model.
# The last 23 layers of the model will be trained.

for layer in model.layers[:-23]:
    layer.trainable = False
```

We will remove all the top layers except the last 23 layers, inorder to predict our output

7.5 TRAINING THE MODEL

7.5.1 TRAINING THE MODEL WITH TOP2 AND TOP3 ACCURACY METRICS

```
# Define Top2 and Top3 Accuracy

from tensorflow.keras.metrics import categorical_accuracy, top_k_categorical_accuracy

def top_3_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=3)

def top_2_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=2)
```

```
model.compile(Adam(lr=0.01), loss='categorical_crossentropy',
              metrics=[categorical_accuracy, top_2_accuracy, top_3_accuracy])
```

```
# Add weights to try to make the model more sensitive to melanoma

class_weights={
    0: 1.0, # akiec
    1: 1.0, # bcc
    2: 1.0, # bkl
    3: 1.0, # df
    4: 3.0, # mel # Try to make the model more sensitive to Melanoma.
    5: 1.0, # nv
    6: 1.0, # vasc
}
```

```python
filepath = "model.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_top_3_accuracy', verbose=1,
                             save_best_only=True, mode='max')

reduce_lr = ReduceLROnPlateau(monitor='val_top_3_accuracy', factor=0.5, patience=2,
                                   verbose=1, mode='max', min_lr=0.00001)


callbacks_list = [checkpoint, reduce_lr]

history = model.fit_generator(train_batches, steps_per_epoch=train_steps,
                              class_weight=class_weights,
                    validation_data=valid_batches,
                    validation_steps=val_steps,
                    epochs=30, verbose=1,
                   callbacks=callbacks_list)
```

The metrics we used for training is top_3_accuracy and top_2_accuracy from top_k_categorical_accuracy and categorical_accuracy are the accuracy metrics and categorical_crossentropy for loss function

```
Epoch 29/30
908/908 [==============================] - ETA: 0s - loss: 0.2518 - categorical_accuracy: 0.9209 - top_2_accuracy: 0.9877 - top
_3_accuracy: 0.9985
Epoch 00029: val_top_3_accuracy did not improve from 0.98934
908/908 [==============================] - 712s 785ms/step - loss: 0.2518 - categorical_accuracy: 0.9209 - top_2_accuracy: 0.98
77 - top_3_accuracy: 0.9985 - val_loss: 0.3349 - val_categorical_accuracy: 0.8838 - val_top_2_accuracy: 0.9670 - val_top_3_accu
racy: 0.9861
Epoch 30/30
908/908 [==============================] - ETA: 0s - loss: 0.2453 - categorical_accuracy: 0.9248 - top_2_accuracy: 0.9855 - top
_3_accuracy: 0.9972
Epoch 00030: val_top_3_accuracy did not improve from 0.98934
908/908 [==============================] - 713s 785ms/step - loss: 0.2453 - categorical_accuracy: 0.9248 - top_2_accuracy: 0.98
55 - top_3_accuracy: 0.9972 - val_loss: 0.3377 - val_categorical_accuracy: 0.8849 - val_top_2_accuracy: 0.9648 - val_top_3_accu
racy: 0.9861
```

Figure 7.3 After 30 epoch with top_k_categorical_accuracy metrics function

7.5.2 TRAINING THE MODEL WITH ACCURACY AND CATEGORICAL_ACCURACY METRICS

```python
from tensorflow.keras.metrics import categorical_accuracy
```

```python
model.compile(Adam(lr=0.01), loss='categorical_crossentropy',
              metrics=[categorical_accuracy, 'accuracy'])
```

```python
# Add weights to try to make the model more sensitive to melanoma

class_weights={
    0: 1.0, # akiec
    1: 1.0, # bcc
    2: 1.0, # bkl
    3: 1.0, # df
    4: 3.0, # mel # Try to make the model more sensitive to Melanoma.
    5: 1.0, # nv
    6: 1.0, # vasc
}
```

```python
filepath = "model.h5"
checkpoint = ModelCheckpoint(filepath, monitor='accuracy', verbose=1,
                             save_best_only=True, mode='max')

reduce_lr = ReduceLROnPlateau(monitor='accuracy', factor=0.5, patience=2,
                                   verbose=1, mode='max', min_lr=0.00001)


callbacks_list = [checkpoint, reduce_lr]

history = model.fit(train_batches, steps_per_epoch=train_steps,
                              class_weight=class_weights,
                    validation_data=valid_batches,
                    validation_steps=val_steps,
                    epochs=30, verbose=1,
                   callbacks=callbacks_list)
```

The metrics we used for training is accuracy and categorical_accuracy are the accuracy metrics and categorical_crossentropy for loss function

```
Epoch 29/30
908/908 [==============================] - 336s 370ms/step - loss: 0.4317 - categorical_accuracy: 0.8681 - accuracy: 0.8681 - v
al_loss: 0.4461 - val_categorical_accuracy: 0.8721 - val_accuracy: 0.8721

Epoch 00029: accuracy did not improve from 0.87234
Epoch 30/30
908/908 [==============================] - 344s 379ms/step - loss: 0.4209 - categorical_accuracy: 0.8750 - accuracy: 0.8750 - v
al_loss: 0.4099 - val_categorical_accuracy: 0.8817 - val_accuracy: 0.8817

Epoch 00030: accuracy improved from 0.87234 to 0.87500, saving model to model.h5
```

Figure 7.4 After 30 epoch with accuracy metrics function

7.6 COMPARISION BETWEEN THE METRICS

```
val_loss: 0.2965870797634125
val_cat_acc: 0.890191912651062
val_top_2_acc: 0.9690831303596497
val_top_3_acc: 0.9893389940261841
```

Figure 7.5 val_loss, val_cat_acc,val_top2_acc, val_top3_acc

Output after evaluating with validation sets for categorical_crossentropy, categorical_accuracy, top_2_accuracy and top_3_accuracy

```
val_loss: 0.40985801815986633
val_cat_acc: 0.8816630840301514
val_acc: 0.8816630840301514
```

Figure 7.6 val_loss, val_cat_acc,val_acc

Output after evaluating with validation sets for categorical_crossentropy, categorical_accuracy, accuracy metrics

7.6.1 PLOTTING OF THE TRAINING GRAPHS

FOR TOP_K_CATEGORICAL ACCURACY METRICS

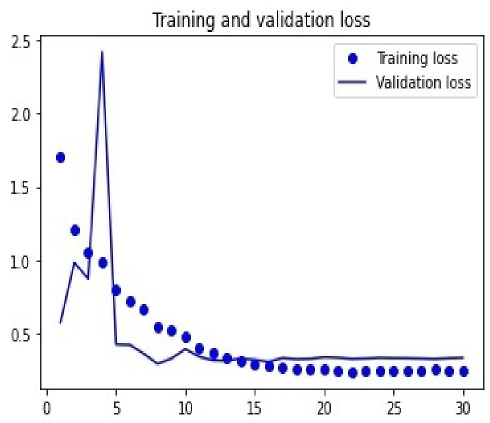


Figure 7.7 Training and validation loss

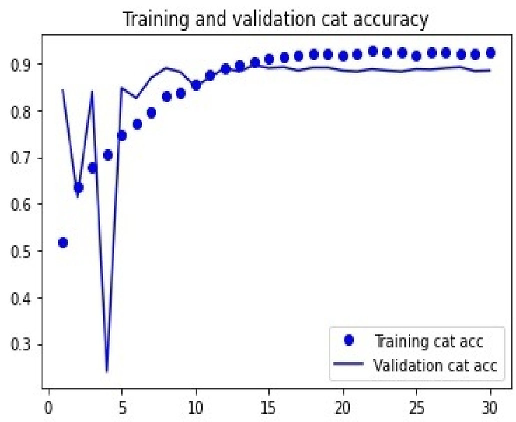


Figure 7.8 Training and validation cat accuracy

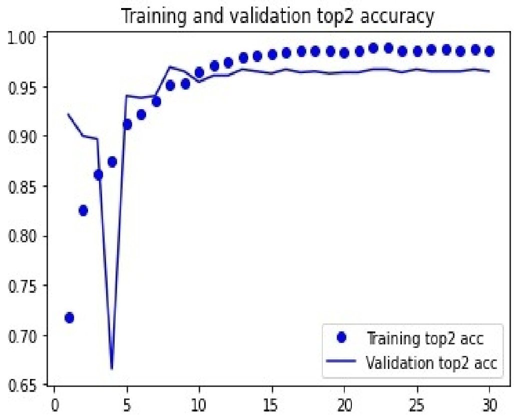


Figure 7.9 Training and validation Top2 accuracy

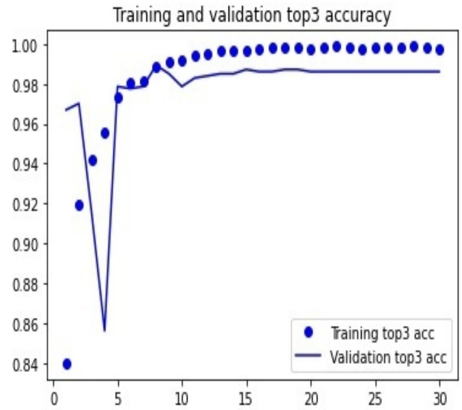


Figure 7.10 Training and validation Top3 accuracy

FOR ACCURACY METRICS

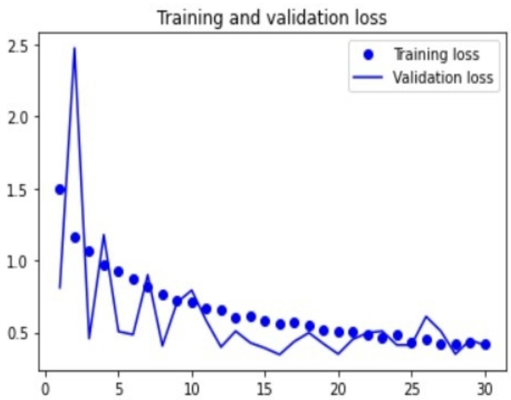


Figure 7.11 Training and validation loss

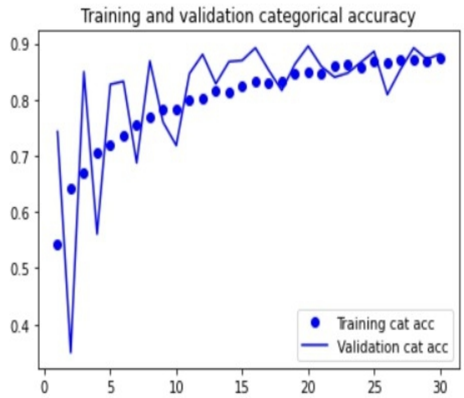


Figure 7.12 Training and validation categorical accuracy

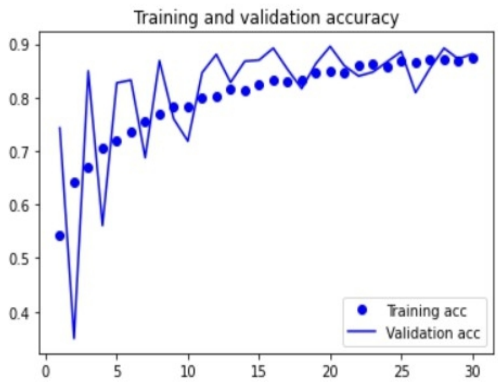
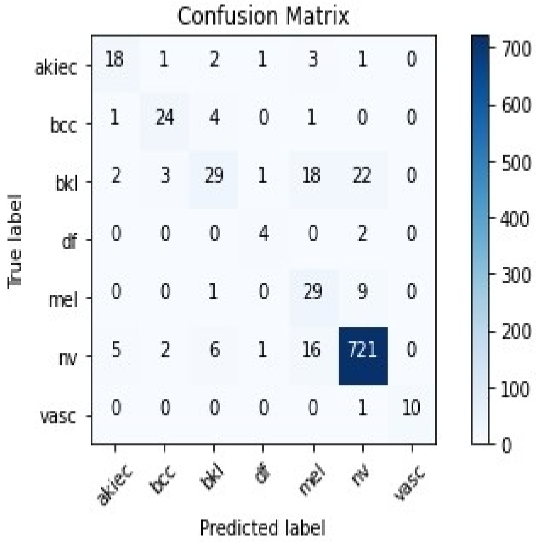


Figure 7.13 Training and validation accuracy

7.6.2 CONFUSION MATRIX



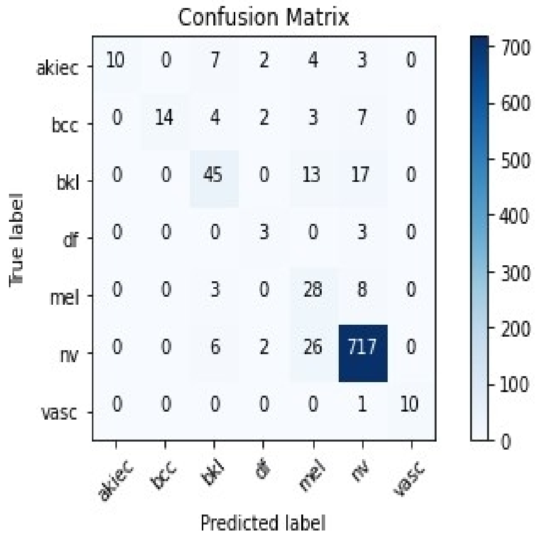


Figure 7.14 confusion matrix for top_k_categorical_accuracy

Figure 7.15 Confusion matrix for accuracy metrics

7.7 THE USER INTERFACE MODEL

7.7.1 DEVELOPING UI

To make the model useful, interpretable and interactable there should be a UI through which the pathologists can query the model and get results . So, a UI with flask framework and python backend makes it much easier in making web apps**.**

```
import keras
from keras.models import Model
import pandas as pd
import numpy as np
import itertools
from flask_ngrok import run_with_ngrok
from flask import Flask, render_template, request, redirect, url_for, send_from_directory
from werkzeug.utils import secure_filename
import os
from keras.models import load_model
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
```

Importing the required libraries

```
base = r"C:\Users\Dell\OneDrive\Desktop\project\New folder"

model_path = os.path.join(base, 'model.h5')
model = keras.models.load_model(model_path)


def show_output(image1):

   img1=keras.preprocessing.image.load_img(image1,target_size=(224,224))
   img1_data=keras.preprocessing.image.img_to_array(img1)
   img1_data=np.expand_dims(img1_data,axis=0)
   img1_data=keras.applications.mobilenet.preprocess_input(img1_data)
   features=model.predict(img1_data)
   out=np.argmax(features)
   skin_type_dict = {
   'Actinic Keratoses (Solar Keratoses) or intraepithelial Carcinoma (Bowen's disease)':0,
   'Basal Cell Carcinoma':1,
   'Benign Keratosis':2,
   'Dermatofibroma':3,
   'Melanoma':4,
   'Melanocytic Nevi':5,
   'Vascular skin lesion':6}
   for key,value in skin_type_dict.items():
        if np.array_equal(out,value):
            return key
```

We will load the saved model, and themn in show_output we will do the preprocessing for the input image, and then check in which label they fall

```
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

app = Flask(__name__, template_folder = os.path.join(base,"templates"),static_folder= os.path.join(base,'static'))
UPLOAD_FOLDER = os.path.join(base,'upload')
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

run_with_ngrok(app)   #starts ngrok when the app is run
@app.route("/",methods=['GET', 'POST'])
def home():
    out = ''
    if request.method == 'POST':
        image1 = request.files['main_img']
        if image1 and allowed_file(image1.filename):
            filename = secure_filename(image1.filename)
            image1.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            return redirect(url_for('uploaded_file',filename=filename))

    return render_template('base.html',out=out)

@app.route('/show/<filename>')
def uploaded_file(filename):
    image1 = os.path.join(UPLOAD_FOLDER,filename)
    out = show_output(image1)
    return render_template('base.html', out=out,filename=filename)


@app.route('/uploads/<filename>')
def send_file(filename):
    return send_from_directory(UPLOAD_FOLDER, filename)


app.run()
```

Routes are defined in flask to route the user's request by using the function defined in the route

7.7.2 RUNNING THE UI MODEL

```
======= RESTART: C:\Users\Dell\OneDrive\Desktop\project\New folder\web.py ======
 * Serving Flask app "web" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 7.16 running the UI model

7.7.3 VISUAL INTEGRATION AND DEPLOYMENT

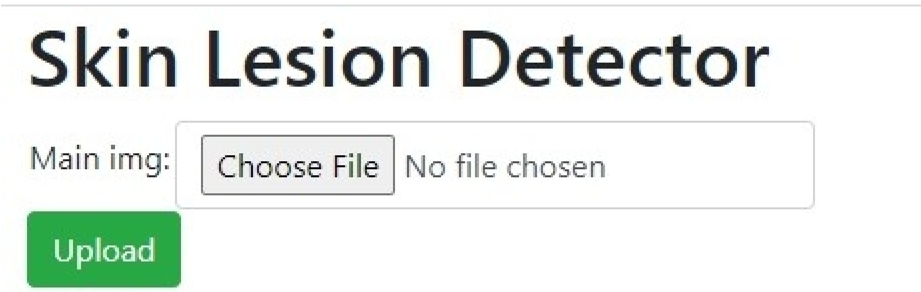


Figure 7.17 visual integration

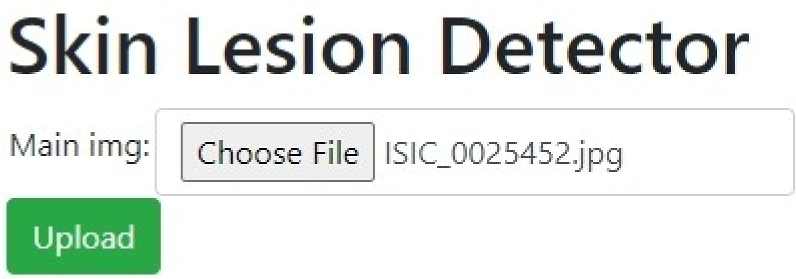


Figure 7.18 after selecting an image

Figure 7.19 Final output after classification

CHAPTER 8
CONCLUSIONS

Skin lesion (skin cancer) has become a serious problem and there are many methods and techniques that helps in predicting and classifying the skin lesion. The overall scope of project is to develop a method to predict and classify the skin cancer with more accuracy and less complexity when compared to existing techniques and that is achieved.

The project is also developed in a effective way that ensure better understanding of doctors and patients. In order to achieve this, we used transfer learning with mobilenet model, which is a pre-trained Convolution Neural Networks model, and various python packages. At present, the model is still in development stage and we got 85% accuracy and there is a way and path to be continue to do research in this filed of Neural Networks. To conclude this project we got results in best possible way to predict the skin lesions.

CHAPTER 9

FUTURE ENHANCEMENTS

As for now, we achieved an accuracy of 85% which can be increased in the future with new Convolution Neural Network architectures, or a method known as ensemble learning where we can combine different architectures and predict the output. The result displaying portal can also be improved with the new techniques emerging in the world. This project is limited to skin lesion prediction only, but we can develop the project so that it can also be used for prediction of other diseases also with accuracy and efficiently.

CHAPTER 10
REFERENCES

[1] Soniya Mane, Dr. Swati Shinde, "A Method for Melanoma Skin Cancer Detection Using Dermoscopy Images",2018.

[2] Wilson F. Cueva, F. Muñoz, G. Vásquez., G. Delgado, "Detection of skin cancer Melanoma through Computer Vision",2017.

[3] Shalu, Aman Kamboj, "A Color-Based Approach for Melanoma Skin Cancer Detection",2018.

[4] Sertan Kaymak, Parvaneh Esmaili, and Ali Serener, "Deep Learning for Two-Step Classification of Malignant Pigmented Skin Lesions,2018.

[5] Kyamelia Roy, Sheli Sinha Chaudhuri, Sanjana Ghosh, Swarna Kamal Dutta "Skin Disease detection based on different Segmentation Techniques,2019.

[6] Omar Abuzaghleh, Buket D. Barkana and Miad Faezipour, "Automated Skin Lesion Analysis Based on Color and Shape Geometry Feature Set for Melanoma Early Detection and Prevention.2014.

[7] Reinaldi Nur Pranata, Andrew Brian Osmond, Casi Setianingsih, "Potential Level Detection of Skin Cancer with Expert System Using Forward Chaining and Certainty Factor Method,2018.

[8] Giulia Mansutti, Ahmed Toaha Mobashsher, and Amin M. Abbosh, "Millimeter-wave Substrate Integrated Waveguide Probe for Near-Field Skin Cancer Detection,2018.

[9] He Huang, Pegah Kharazmi, David I. McLean, Harvey Lui, "Automatic detection translucency using a deep learning method from patches of clinical basal cell carcinoma images,2018.

[10] A.Prabhu Chakkaravarthy, Dr. A.Chandrasekar, “An Automatic Segmentation of Skin Lesion from Dermoscopy Images using Watershed Segmentation,2018.

[11] Alper ARIK, Mesut GÖLCÜK, “Derin Ögrenme Yöntemleri ile Cilt Kanseri Te¸shis Learning Based Skin Cancer Diagnosis,2017.

[12] Ms. Amulya P M, Asst Prof Mr. Jayakumar T V, “A Study on Melanoma Skin Cancer Detection Techniques”,2017.

[13] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6091241/

[14] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3052811/

[15]https://www.aafp.org/afp/1998/0915/p873.html

[16] Jemal A, Siegel R, Ward E, Murray T, Xu J, Smigal C, et al. Cancer statistics, 2006. CA Cancer J Clin. 2006;56:106–30. doi: 10.3322/ canjclin.56.2.106.

[17]www.cancer.org/cancer/skincancermelanoma/detailedguide/melanoma-skin-cancerkey statistics.

[18] Celebi ME, Kingravi HA, Iyatomi H, Aslandogan YA, Stoecker WV, Moss RH, et al. Border detection in dermoscopy images using statistical region merging. Skin Res Technol. 2008;14:347-53. doi. org/10.1111/j.1600-0846.2008.00301.x. PubMed PMID: 19159382. PubMed PMCID: 3160669.