

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By

ABIRAMI RASU

38120006



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND

TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119. TAMILNADU

JANUARY 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of who carried out the project entitled “**DETECTION OF SKIN DISEASE AND PREVENTIVE MEASURE USING MACHINE LEARNING**” under our supervision from August 2020 to November 2020.

Internal Guide

Dr. G. Mathivanan B.Tech., M.E., Ph.D.,

Head of the Department

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I hereby declare that the Project Report entitled “**DETECTION OF SKIN DISEASE AND PREVENTIVE MEASURE USING MACHINE LEARNING**“ done by me under the guidance of Dr.G.Mathivanan B.Tech., M.E., Ph.D., at Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 21/01/2021

Place: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D. ,** Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.G.Mathivanan B.Tech.,M.E.,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.



RETECH
SOLUTIONS PVT. LTD.
An ISO 9001:2015 certified company

INT: RE20S045

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **ABIRAMI R** student of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, BTECH/IT** has undergone the Internship Training through online in our concern entitled **PYTHON** from **12th Aug 2020** to **26th Aug 2020** in relevant departments related to their academic studies.

During the above period, the performance was good & we wish great success in all your future endeavours.

J. meru

Technology drives the world. We are the people who are behind the technology.

Authorised Signature



No-31, 1st Floor, Alagesan Street, Tambaram West, Chennai-45.

Office : 044-42077204 | +91-8122266653

www.retechsolutions.in



08/27/2020

Abirami R

has successfully completed

Introduction to HTML5

an online non-credit course authorized by University of Michigan and offered through Coursera

Colleen van Lent

Colleen van Lent, Ph.D.
Lecturer
School of Information, University of Michigan

Charles Severance

Charles Severance
Clinical Professor, School of Information
University of Michigan

**COURSE
CERTIFICATE**



Verify at coursera.org/verify/CPKCVRU99YPN
Coursera has confirmed the identity of this individual and
their participation in the course.

ABSTRACT

Skin diseases are hazardous and often contagious, especially melanoma, eczema, and impetigo. These skin diseases can be cured if detected early. The fundamental problem with it is, only an expert dermatologist is able to detect and classify such disease. Sometimes, the doctors also fail to correctly classify the disease at early stages. Our research proposes a skin disease detection method based on Image Processing and machine learning Techniques and it include diseases like eczema, psoriasis, acne, rosacea, ichthyosis, vitiligo, hives, seborrheic dermatitis Our system is Personal Computer based so can be used even in remote areas . The patient needs to provide the image of the infected area and it is given as an input to the application. Image Processing and machine Learning techniques process it and deliver the accurate output. In this research, we present algorithm that use Support Vector Machine (SVM), which gives the accuracy in detecting the skin diseases.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	xi
1.	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 SCOPE OF THE PROJECT	2
	1.3 DOMAIN OVERVIEW	3
	1.3.1 MACHINE LEARNING	3
	1.3.2 TYPES OF MACHINE LEARNING	4
2.	LITERATURE SURVEY	7
	2.1 LITERATURE SURVEY 1	7
	2.2 LITERATURE SURVEY 2	7
	2.3 LITERATURE SURVEY 3	8
	2.4 LITERATURE SURVEY 4	8
	2.5 LITERATURE SURVEY 5	9
	2.6 LITERATURE SURVEY 6	10
3	SYSTEM ANALYSIS	11
	3.1 EXISTING SYSTEM	11
	3.1.1 CONSTRAINTS	11
	3.2 PROPOSED SYSTEM	12
	3.2.1 ADVANTAGES	13

4	REQUIREMENT SPECIFICATION	15
	4.1 INTRODUCTION	15
	4.2 SOFTWARE REQUIREMENTS	15
	4.2.1 OS: WINDOWS 7/8/8.1/10	15
	4.2.2 MATLAB	16
	4.2.2.1 THE MATLAB SYSTEM	16
	4.2.2.2 DEVELOPMENT ENVIRONMENT	17
	4.2.2.3 MATLAB APPLICATION	18
	4.2.2.4 MATLAB APPLICATION FUNCTIONALITY	19
5	SYSTEM DESIGN	20
	5.1 INTRODUCTION	20
	5.2 ARCHITECTURE DIAGRAM	20
	5.3 DATA FLOW DIAGRAM	21
	5.4 UML DIAGRAMS	22
	5.4.1 USE CASE DIAGRAM	22
	5.4.2 CLASS DIAGRAM	23
	5.4.3 SEQUENCE DIAGRAM	23
	5.4.4 ACTIVITY DIAGRAM	24
	5.4.5 COMPONENT DIAGRAM	25
6	SYSTEM IMPLEMENTATION	27
	6.1 INTRODUCTION	27
	6.2 LIST OF MODULES	28
	6.2.1 MODULE IMPLEMENTATION	28
	6.2.2 IMAGE EXTRACTION AND PREPROCESSING	30
	6.2.3 SEGMENTATION	31
	6.2.4 FEATURE EXTRACTION	33
	6.2.5 DETECTION SYSTEM	35

7	SOFTWARE TESTING	38
	7.1 INTRODUCTION	38
	7.2 OBJECTIVES	38
	7.3 TYPES OF TESTING	38
	7.3.1 UNIT TESTING	39
	7.3.2 INTEGRATION TESTING	39
	7.3.3 VALIDATION TESTING	40
	7.3.4 ACCEPTANCE TESTING	40
	7.3.5 PERFORMANCE TESTING	40
	7.3.6 FUNCTIONAL TESTING	41
	7.3.7 SYSTEM TESTING	42
	7.3.8 WHITE BOX TESTING	42
	7.3.9 BLACK BOX TESTING	43
8	CONCLUSION AND FUTURE ENHANCEMENT	44
	8.1 CONCLUSION	44
	8.2 FUTURE ENHANCEMENT	44
	APPENDIX I (SCREENSHOTS)	46
	APPENDIX II (SAMPLE CODING)	47
	REFERENCES	55

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	ARCHITECTURE OF MACHINE LEARNING	3
1.2	CLASSIFICATION OF MACHINE LEARNING	4
1.3	SUPERVISED LEARNING	5
1.4	UNSUPERVISED LEARNING	5
3.1	CNN ARCHITECTURE	11
5.1	ARCHITECTURE DIAGRAM O	21
5.2	DATA FLOW DIAGRAM	21
5.3	USE CASE DIAGRAM	22
5.4	CLASS DIAGRAM	23
5.5	SEQUENCE DIAGRAM	24
5.6	ACTIVITY DIAGRAM	25
5.7	COMPONENT DIAGRAM	26
6.1	RGB IMAGE CONVERTED TO GRAY IMAGE	31
6.2	DIFFERENT STAGES OF SEGMENTED IMAGE	31
6.3	CLASSIFY AND SELECTION PROCESS	35
6.4	CLASSIFICATION USING SVM	36
6.5	SVM PROCESS	37
7.1	WHITE BOX TESTING	42
7.2	BLACK BOX TESTING	43

A.1.1	ADMIN PAGE	46
A.1.2	DETECTION SYSTEM	46

LIST OF ABBREVIATIONS

S. NO	ABBREVIATION	EXPANSION
1	WHO	World Health Organization
2	SVM	Support Vector Machine
3	MATLAB	MATrix LABoratory
4	ML	Machine Learning
5	AI	Artificial Intelligence
6	MCNN	Multiple Convolution Neural Networks
7	ISIC	International Skin Imaging Collaboration
8	ROC	Receiver Operating Characteristic
9	FFNN	Feed-forward neural network
10	GLCM	Grey level Co-occurrence matrix
11	CNN	Convolutional Neural Network
12	GMM	Gaussian Mixture Model
13	OS	Operating System
14	GUI	Graphical User Interface
15	MS-DOS	Microsoft Disk Operating System
16	DFD	Data Flow Diagram
17	UML	Unified Modelling Language
18	CBD	Component based Development
19	KNN	K-Nearest Neighbourhood
20	ANN	Artificial Neural Network

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Skin diseases rate has been increasing for past few decades, many of these diseases are very dangerous, particularly if not treated at early stages. In Sudan skin diseases are big issue, according to the latest WHO data published in May 2014 Skin Diseases Death in Sudan reached 1,974 or 0.76% of total death. The age adjusted death rate is 9.81 per 100,000 of population, this results ranks Sudan number 8 in the world [1]. In addition, dermatologist use variety of visual clues such as color, scaling and arrangement of lesions, the body site distribution and others, when these individual components are analyzed separately, the recognition of the disease can be quite complex that requiring high level of experience. Diagnosis by humans depends on subjective judgment of the dermatologists so it's hardly reproducible, unlike computer aided diagnostic systems which are more realistic and reliable. The traditional diagnosis of skin diseases requires high level of expertise in the domain and a large amount of knowledge to differentiate between the large number and similar diseases, also it depends on the visual aspects of the physicians, and some of them can't be distinguished just with the human eye and requires further tools or tests, so the traditional skin diseases diagnosis is subjective and may be unreliable, then more proper solutions are needed. It makes researchers to propose computer aided systems for diagnosing Skin Diseases, especially with the increasing role of the machine learning and data science implementations in the medical field and it implemented in those systems because of their ability to recognize patterns in objective manner. This introduces you with some of the techniques that could be implemented to develop Skin Diseases Diagnosis System based on machine learning approach. SVM (support Vector Machine) as a classifier is used to train our data, which is very powerful in object and texture recognition.

An windows tablet application is used as interface to the user, this application communicates with MATLAB application.

1.2 SCOPE OF THE PROJECT

The objective of this project can be summarized into the following points:

1. Develop Deep learning application that in general, has the ability to:
 - Determine the affected areas in the image.
 - Determine the disease in the specified region.
 - Be applicable locally in India.
2. Develop a Android Tablet interface that:
 - Capture the image; send it to the server and receiving the results back.

1.3 DOMAIN OVERVIEW

1.3.1 MACHINE LEARNING

Machine learning (ML) is a subfield of artificial intelligence (AI). Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: ***The ability to learn.*** Machine learning is actively being used today, perhaps in many more places than one would expect. It provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning** focuses on the development of computer programs that can access data and **use** it learn for themselves. **Machine learning** is a method of data analysis that automates analytical model building. It is a branch of artificial **intelligence** based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

How does Machine Learning Work?

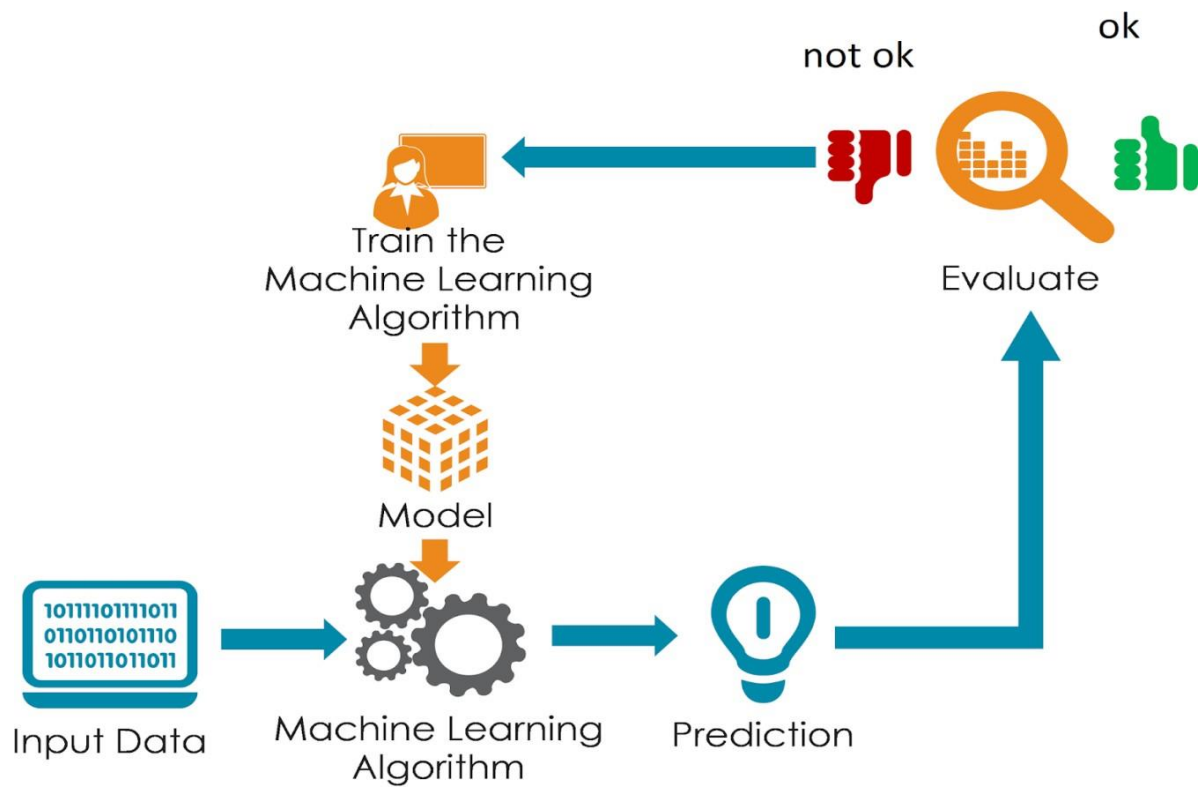


Fig 1.1 Architecture of Machine Learning

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model.

The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set. This is just a very high-level example as there are many factors and other steps involved.

1.3 .2 TYPES OF MACHINE LEARNING

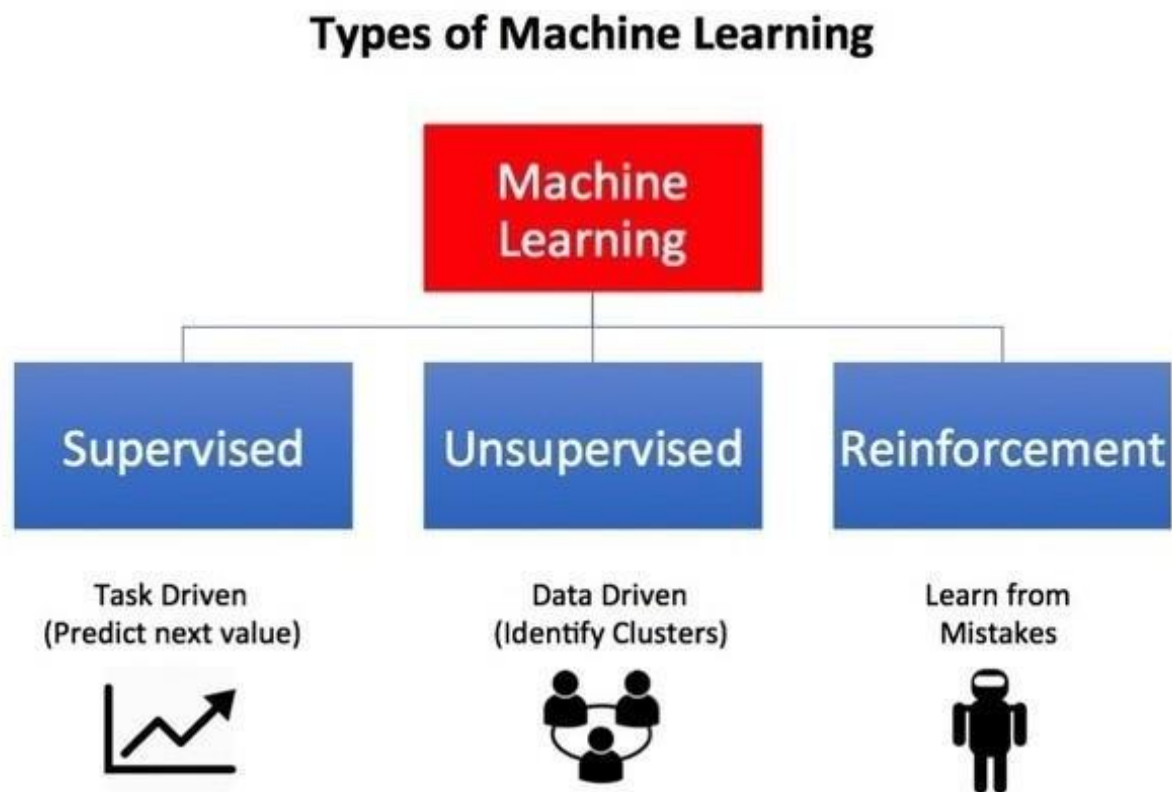


Fig 1.2 classification of machine learning

Supervised learning : It is the most popular paradigm for machine learning. It is the easiest to understand and the simplest to implement. It is very similar to teaching a child with the use of flash cards. Given data in the form of examples with labels, we can feed a learning algorithm these example-label pairs one by one, allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not. Over time, the algorithm will learn to approximate the exact nature of the relationship between examples and their labels. When fully-trained, the supervised learning algorithm will be able to observe a new, never-before-seen example and predict a good label for it.

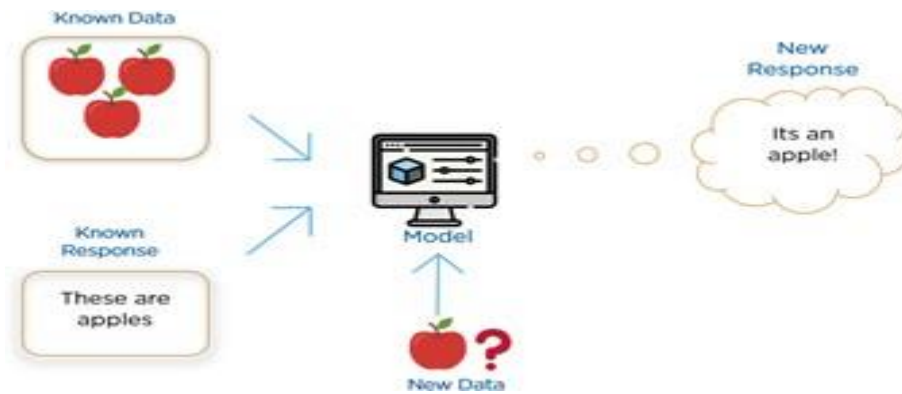


Fig 1.3 Supervised learning

Supervised learning is often described as task-oriented because of this. It is highly focused on a singular task, feeding more and more examples to the algorithm until it can accurately perform on that task.

Unsupervised learning: It is very much the opposite of supervised learning. It features no labels. Instead, our algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and/or organize the data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data.

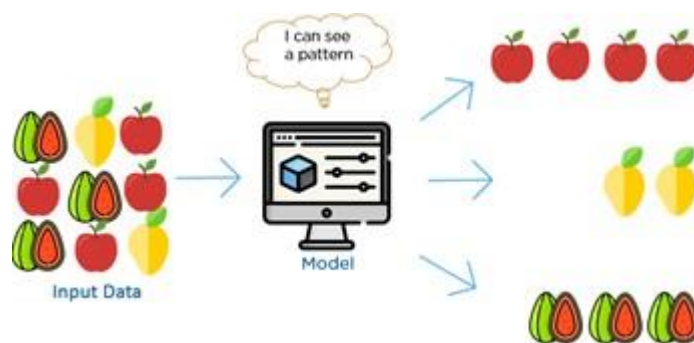


Fig 1.4 UnSupervised learning

Having intelligent algorithms that can take our terabytes and terabytes of unlabeled data and make sense of it is a huge source of potential profit for many industries. That alone could help boost productivity in a number of fields. The unsupervised

learning is based upon the data and its properties, we can say that unsupervised learning is data-driven.

Reinforcement learning is fairly different when compared to supervised and unsupervised learning. Where we can easily see the relationship between supervised and unsupervised (the presence or absence of labels), the relationship to reinforcement learning is a bit murkier. Some people try to tie reinforcement learning closer to the two by describing it as a type of learning that relies on a time-dependent sequence of labels. Reinforcement learning as learning from mistakes. Place a reinforcement learning algorithm into any environment and it will make a lot of mistakes in the beginning. So long as we provide some sort of signal to the algorithm that associates good behaviors with a positive signal and bad behaviors with a negative one, we can reinforce our algorithm to prefer good behaviors over bad ones. Over time, our learning algorithm learns to make less mistakes than it used to.

Reinforcement learning is very behavior driven. It has influences from the fields of neuroscience and psychology. If you've heard of Pavlov's dog, then you may already be familiar with the idea of reinforcing an agent, albeit a biological one.

CHAPTER 2 LITERATURE

SURVEY

2.1 Multiple Convolutional Neural network for skin Dermoscopic Image classification

Melanoma classification is a serious stage to identify the skin disease. It is considered a challenging process due to the intra-class discrepancy of melanomas, skin lesions' low contrast, and the artifacts in the dermoscopy images, including noise, existence of hair, air bubbles, and the similarity between melanoma and non- melanoma cases. To solve these problems, we propose a novel multiple convolution neural network model (MCNN) to classify different disease types in dermoscopic images, where several models were trained separately using an additive sample learning strategy. The MCNN model is trained and tested using the training and validation sets from the International Skin Imaging Collaboration (ISIC 2016), respectively. The classification accuracy and receiver operating characteristic (ROC) curve are used to evaluate the performance of the proposed method. The values of AUC (the area under the ROC curve) were used to evaluate the performance of the MCNN.

2.2 Classification of Multiple Diseases based on Wavelet Features

This study presents an efficient disease classification approach based on medical images. The approach is more efficient as it reduces the computational complexity. The implementation uses only two wavelet filters in selecting the texture features as compared with five filters used in the earlier research works. The computed average and energy features are fed to feed-forward neural network (FFNN) and support vector machine (SVM) classifiers. The SVM is proved as a better classifier than the FFNN for all the three diseases related to skin, breast and

retina with an improved accuracies of 89%, 92% and 100%, respectively. Also, a graphical user interface is developed useful for various disease classification based on the whole dataset of size 100

2.3 Skin Disease Recognition Method Based on Image Color and Texture Features

Skin diseases have a serious impact on people's life and health. Current research proposes an efficient approach to identify singular type of skin diseases. It is necessary to develop automatic methods in order to increase the accuracy of diagnosis for multitype skin diseases. In this paper, three type skin diseases such as herpes, dermatitis, and psoriasis skin disease could be identified by a new recognition method. Initially, skin images were preprocessed to remove noise and irrelevant background by filtering and transformation. Then the method of grey- level co-occurrence matrix (GLCM) was introduced to segment images of skin disease. The texture and color features of different skin disease images could be obtained accurately. Finally, by using the support vector machine (SVM) classification method, three types of skin diseases were identified. The experimental results demonstrate the effectiveness and feasibility of the proposed method.

2.4 Skin Disease Classification versus Skin Lesion Characterization: Achieving Robust Diagnosis using Multi-label Deep Neural Networks

In this study, we investigate what a practically useful approach is in order to achieve robust skin disease diagnosis. A direct approach is to target the ground truth diagnosis labels, while an alternative approach instead focuses on determining skin lesion characteristics that are more visually consistent and discernible. We argue that, for computer-aided skin disease diagnosis, it is both more realistic and more useful that lesion type tags should be considered as the target of an automated diagnosis system such that the system can first achieve a high accuracy in describing skin lesions, and in turn facilitate disease diagnosis using lesion characteristics in

conjunction with other evidence. To further meet such an objective, we employ convolutional neural networks (CNNs) for both the disease-targeted and lesion-targeted classifications. We have collected a large-scale and diverse dataset of 75,665 skin disease images from six publicly available dermatology atlantes. Then we train and compare both disease-targeted and lesion-targeted classifiers, respectively. For disease-targeted classification, only 27.6% top-1 accuracy and 57.9% top-5 accuracy are achieved with a mean average precision (mAP) of 0.42. In contrast, for lesion-targeted classification, we can achieve a much higher mAP of 0.70.

2.5 Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer

Melanoma is considered a fatal type of skin cancer. However, it is sometimes hard to distinguish it from Nevus due to their identical visual appearance and symptoms. The mortality rate because of this disease is higher than all other skin related consolidated malignancies. The number of cases is growing amongst young people but if it is diagnosed at its earlier stage then the survival rates become very high. The cost and time required for the doctors to diagnose all patients for Melanoma are very high. In this research work, we propose an intelligent system to detect and distinguish Melanoma from Nevus by using state of the art image processing techniques. At first, Gaussian Filter is used for removing noise from the skin lesion of the acquired images followed by the use of improved K-mean clustering to segment out the lesion. A distinctive hybrid super feature vector is formed by the extraction of textural and color features from the lesion. Support Vector Machine (SVM) is utilized for the classification of skin cancer into melanoma and nevus. Our aim is to test the effectiveness of the proposed segmentation technique, extract the most suitable features and compare the classification results with the other techniques present in the literature. The proposed methodology is tested on DERMIS dataset having a total number of 397 skin cancer images where 146 are melanoma and 251 are nevus skin lesions. Our proposed methodology archives encouraging results having 96% accuracy.

2.6 Analysis of Deep Networks with Residual Blocks and Different Activation Functions: Classification of Skin Diseases

Deep convolutional neural networks have been implemented for image classification tasks and achieved promising results in recent years. Particularly, ResNETs have been used since they can eliminate vanishing gradient problem in very deep networks. However, ResNET architectures with different activation functions, batch sizes, number of images in the testing and training stages can cause different results. Therefore, the effect of residual connections and activation functions image classification is still unclear. Also, in the literature, ResNET based models have been trained and tested with data sets having different characteristics. However, to make meaningful evaluations of the results obtained from different ResNET models, the same data sets should be used. Therefore, in this work, four network models have been implemented to analyze the effect of two activation functions (ReLU and SELU) and residual learning for image classification using the same data sets. To evaluate performances of these models, a real world issue, which is automated skin disease classification from colored digital images, has been handled. Experimental results and comparative analyses indicated that the ResNET with SELU and without residual block yields in the highest validation accuracy (97.01%) for image classification.

CHAPTER 3

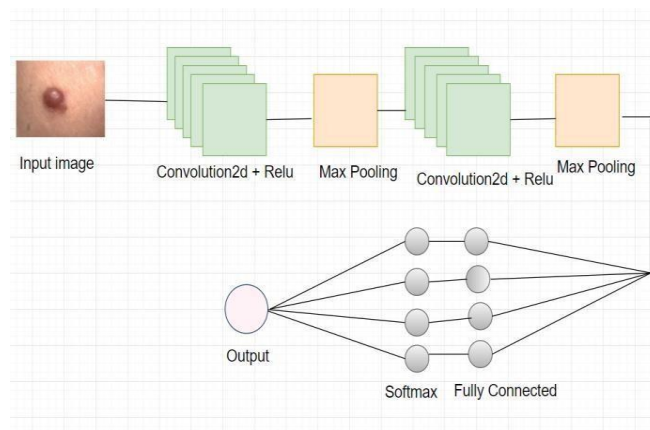
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Existing system investigates the application of the Gaussian mixture model (GMM) based on the analysis and classification of skin diseases from their visual images using a Mahalanobis distance measure. The GMM has been preferred over the convolution neural network (CNN) because of limited resources available within the mobile device. Gray-level co-occurrence matrix (GLCM) parameters contrast, correlation, energy, and homogeneity derived from skin images have been used as the input data for the GMM. The analysis of the results showed that the proposed method is able to predict the classification of skin diseases with satisfactory efficiency. It was also observed that different diseases occupy distinct spatial positions in multidimensional space clustered using the Mahalanobis distance

Fig 3.1 CNN Architecture

measure. This preferred over the CNN because of limited resources are available with in the



mobile device.

3.1.1 CONSTRAINTS

- The Gaussian mixture model does not show better accuracy than SVM.
- It uses euclidean distance it leads to limited accuracy for

different weighted coordinate axis .

- The main limitations of CNN is the code complexity and the requirement of large amounts of input data for training.
- These applications under-perform with the images taken in poor lighting conditions and generally lead to wrong diagnoses . The problem is more severe in the diseases with similar symptoms.
- Gaussian mixture model does not contain the spatial information.
- The main limitation of the GMM algorithm is that, for computational reasons, it can fail to work if the dimensionality of the problem is too high.
- GMM algorithm is that the user must set the number of mixture models that the algorithm will try and fit to the training dataset.

3.2 PROPOSED SYSTEM

In proposed system we use support vector machine, it used to analyze data for classification and regression. It gives the most accurate value and it easy to indentify the type. User uploads image of the affected area using the mobile application Image processing unit receives uploaded image at the backend where following steps will be performed on image.

- Pre-processing of image
- Segmentation to extract skin lesion
- Feature Extraction to extract the features
- Classification model uses extracted features for detection of affected area.
- Result will be shown to the user in mobile application

Image Acquisition: Images are acquired through a camera or locally stored device. Images are obtained from surveys and websites.

Image Pre-processing: For Pre-processing of Image, Filtering is performed on image which is a non-linear process used for enhancing the overall image by preserving the edges of the image. Median filtering is used especially to reduce impulsive, salt-

pepper noise. In this, each pixel value in an image is replaced with the median value of its neighboring pixels including itself.

Image Segmentation: Image segmentation is performed to separate suspicious lesion from normal skin. This is implemented through MATLAB. For image segmentation, multilevel thresholding using Otsu method is performed where image is segmented into 3 levels using IM Quantize with 2 threshold level. The Segmented image is converted into a color image using label2rgb().

Feature Extraction: Unique features of skin lesion are extracted. Features are extracted using the 2D Wavelet Transform. Features extracted using the wavelet transform are Entropy, Mean, Mean Absolute Deviation, Median Absolute Deviation, Energy, Standard deviation, L1 norm, L2 norm, Kurtosis, Skewness. Texture Features extracted using GLCM are Contrast, Correlation, Energy and Homogeneity. dataset comprises of visual features (features extracted from images using image processing). For skin disease classification, Support Vector Machine (SVM) classifier is used

Classification: After the feature extraction process, the extracted features are directly fed into the SVM classifier. The process involves two phases:

- Training Phase: 408 images of eczema, impetigo, melanoma, and others are used for training.
- Testing Phase: In this phase, test images are given to the classifier and the classifier uses knowledge gained during the training phase to classify the test image. Finally in this phase classify the disease.

3.2.1 ADVANTAGES:

- Increased accuracy is obtained by adding additional parameters
- Avoids human effort saves on manpower.

- This system efficient for resource-poor environment, where health workers have even lesser expertise specially in dermatology, need more convenient and innovative measures for the proper diagnosis of the skindiseases.
- Early detection of skin diseases can prove to be cost effective and can be accessible in remote areas. Identifying the infected area of skin and detecting the type of disease is useful for early awareness
- The skin disease is identified in its early stage and it can be cured as fast as possible.
- Through this we can cure many diseases and it is beneficial for people who stay in rural areas.

CHAPTER 4 REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

The requirements specification is a specification of technical requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

4.2 SOFTWARE REQUIREMENTS

- OS: WINDOWS 7/8/8.1/10
- MATLAB

4.2.1 OS: WINDOWS 7/8/8.1/10

The first version of Windows, released in 1985, was simply a GUI offered as an extension of Microsoft's existing disk operating system, or MS-DOS. Based in part on licensed concepts that AppleInc. had used for its Macintosh System Software, Windows for the first time allowed DOS users to visually navigate a virtual desktop, opening graphical "windows" displaying the contents of electronic folders and files with the click of a mouse button, rather than typing commands and directory paths at a text prompt.

Subsequent versions introduced greater functionality, including native Windows File Manager, Program Manager, and Print Manager programs, and a more dynamic interface. Microsoft also developed specialized Windows packages,

including the networkable Windows for Workgroups and the high-powered Windows NT, aimed at businesses. The 1995 consumer release Windows 95 fully integrated Windows and DOS and offered built-in Internet support, including the World Wide Web browser Internet Explorer.

4.2.2 MATLAB

The skin diseases diagnosis system mainly consists of two parts that represents the server side and the client side each one contains a separate application and linked together over a shared network. The server side is a MATLAB application that contains the main machine learning model which implement the training and classification task. The client side is an Android application that acts as an interface to the MATLAB application, its main task is to receive the input from the user and pass it to the server and return the output of the server.

- **Application Functionality Capture an Image :** This function is used to capture an image using the mobile camera and feed it as input to the android application, which contains the skin disease image to be classified.
- **Load image from storage:** This function holds the other option instead of capturing the image, which is loading an image stored in the device storage or any associated SD card.

4.2.2.1 The MATLABSystem

The MATLAB system consists of five main parts:

Development Environment: This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, and browsers for viewing help, the workspace, files, and the search path.

The MATLAB Mathematical Function Library: This is a vast collection of computational algorithms ranging from elementary functions like sum, sine,

cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

The MATLAB Language: This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

Handle Graphics: This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API): This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

4.2.2.2 DEVELOPMENT ENVIRONMENT

Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

Starting and Quitting MATLAB

Starting MATLAB:

On a Microsoft Windows platform, to start MATLAB, double-click the

MATLAB shortcut icon on your Windows desktop.

On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations.

Quitting MATLAB:

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

4.2.2.3 MATLAB Application

The MATLAB application is consisted of a several files, each one is responsible of a certain functionality, and it's divided into three parts:

- **Learning Model:** This part is related to the learning model training and evaluation, and is executed once a new model to be trained and it contains loading the data using ImageDatastore object and feed it to the model to be trained as explained in section.
- **Server Connection:** This part is related to establishment of the connection between the MATLAB and the server, and is consist of two parts the first one for establishing the connection, called once at the beginning and is used to set the host address, the port used, username, the password, java database connector path and string, then use the JDPC driver to connect to the database server.
- **Preprocessing and image manipulation:** This part includes several preprocessing functions which is applied to the data before the training or for

the data to be classified, and contains functions for.

4.2.2.4 MATLAB application functionality

These are the main functionalities in the MATLAB application used in the server side:

- **Check the database server for any input :** The MATLAB application is running always on the server side and this function is listening to any changes in the database server, any addition in the database server means that the android application wrote a new input, then this function triggers the MATLAB application to process the new input.
- **Read the image uploaded to the server :** When the MATLAB discovers a new input from the database server, this function read the image path from the database and use that path to load the image into the application to perform a classification task.
- **Image resizing:** When the image is uploaded to the MATLAB application, this function changes the size of the image to a certain size that is configured in the model to be suitable for the classification.
- **Image classification:** This task is to load a pre-trained model and use it to classify the input data into a suitable class of skin diseases, this function is the main function of the application.
- **Write the results back to the server:** This function is performed after the classification process and get the result, it's written in the database server to be received at the client end.

CHAPTER 5

SYSTEM DESIGN

5.1 INTRODUCTION

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a product that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate the requirements into finished product.

5.2 ARCHITECTURE DIAGRAM

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. The architecture diagram of a system in which the principal parts of functions are represented by blocks connected by the lines that show the relationships of the blocks. The architecture diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation. It shows the relationship between different components of a system. System architecture can comprise system components, the externally visible properties of those components, the relationships between them.

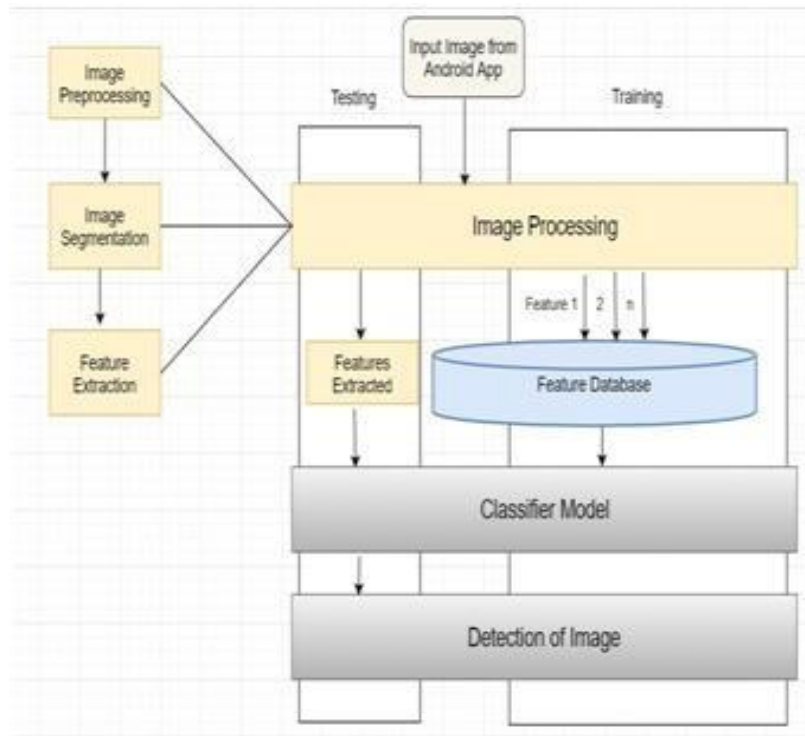


Fig 5.1 Architecture diagram of detection of skin disease

5.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modelling its process aspects. It concerns things like where the data will come from and go to as well as where it will be stored. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects

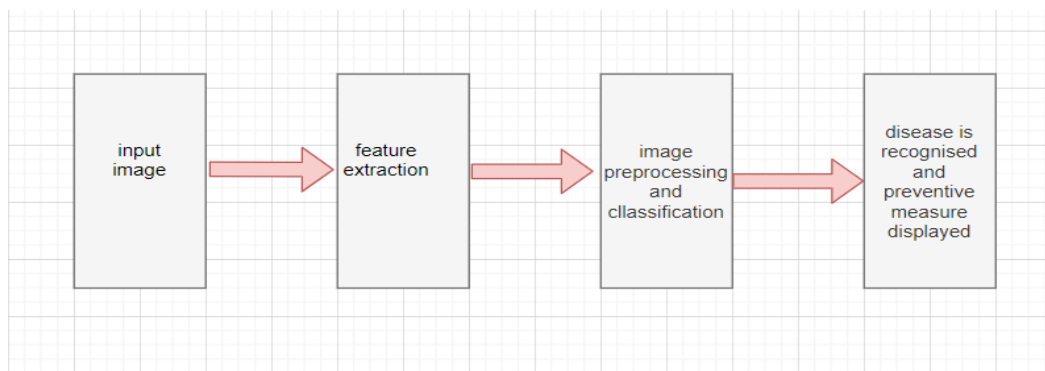


Fig :5.2 Data flow diagram

5.4 UML DIAGRAMS

The Unified Modelling Language (UML) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. It can be described as a general purpose visual modelling language to visualize, specify, construct and document software system. Although UML is generally used to model software systems but it is not limited within this boundary. It is also used to model non software systems as well as process flow in a manufacturing unit etc.

5.4.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The purpose of use case diagram is to capture the dynamic aspect of a system. When a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

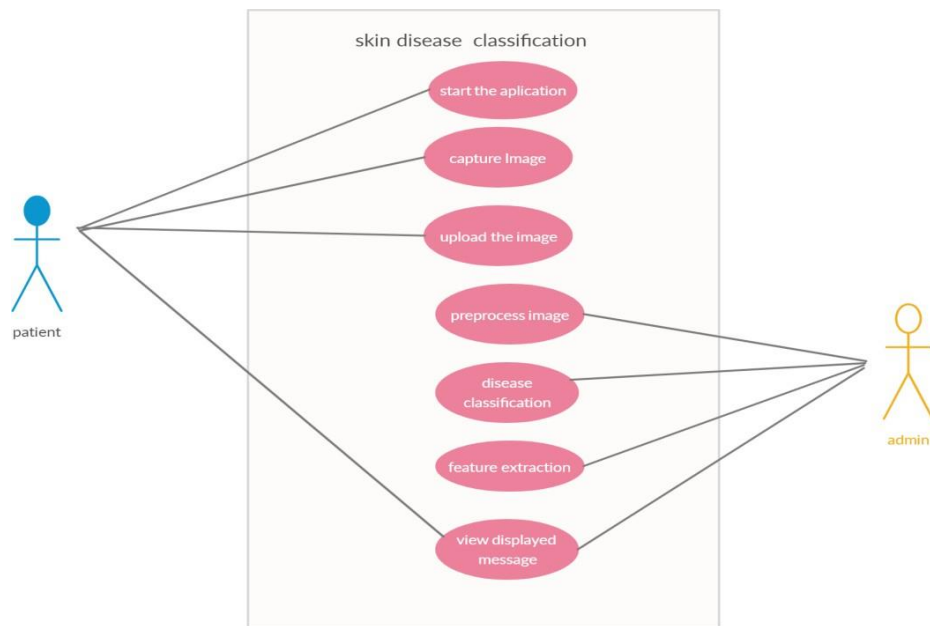


FIG 5.3 use case diagram

5.4.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Each of the components in this system has a unique identifier. In case of BS there is an additional id called pbsid to store and forward the potential BS id to the MS. BS has an array of MS identifiers and BSC has an array of BS connected to it. The BS organizes this array for better results. MS finds location and sends it to BS which finds the triangle and forwards it to BSC which gives the potential BS information to MS and msid to potential BS. The associativity is similar to that of architecture diagram.

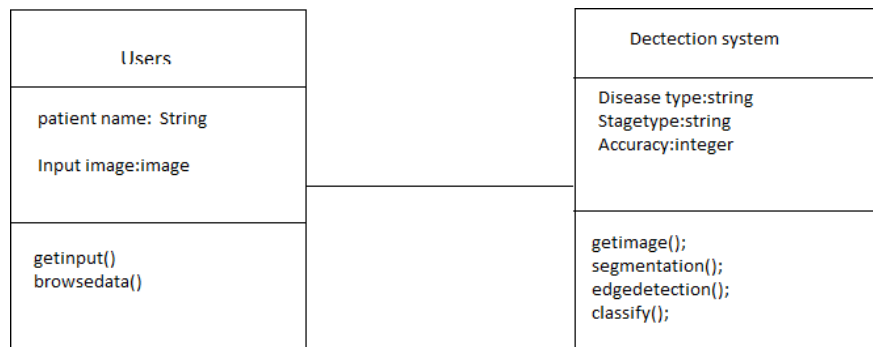


Fig 5.4 Class diagram

5.4.3 SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the

scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or even scenarios.

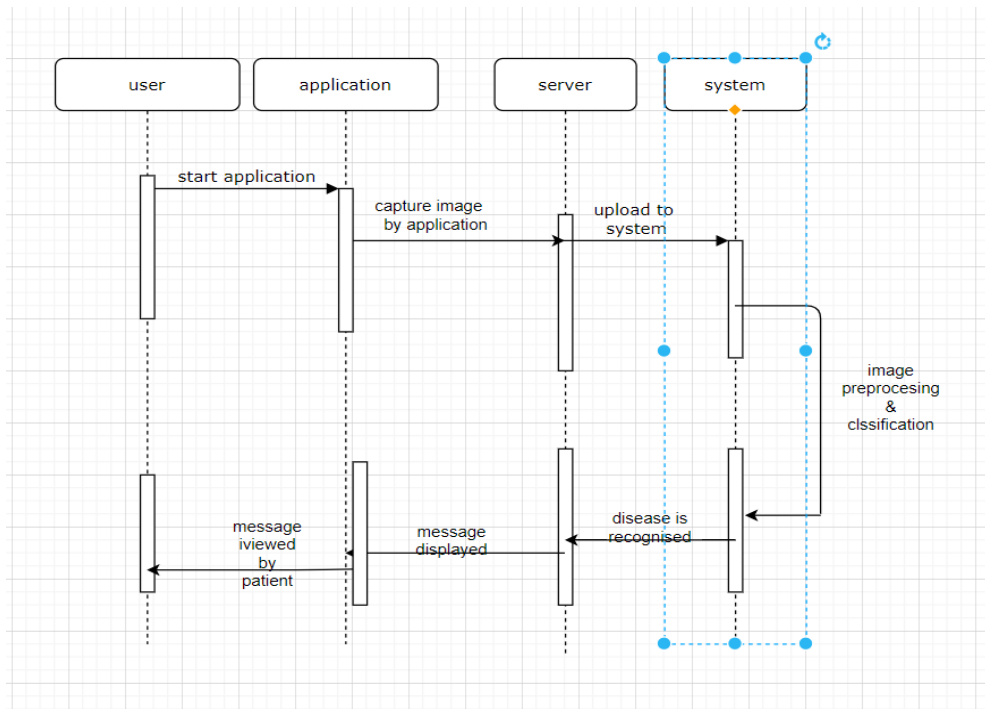


Fig 5.5 sequence diagram

5.4.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows consisting of stepwise activities and actions. Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- Rounded rectangles represent actions;
- Diamonds represent decisions;
- Bars represent the start (split) or end (join) of concurrent activities;
- A circle represents the start (initial state) of the workflow;
- An encircled circle represents the end (final state).



Fig: 5.6 Activity diagram

5.4.5 COMPONENT DIAGRAM

Component-based development (CBD) and object-oriented development go hand-in-hand, and it is generally recognized that object technology is the preferred foundation from which to build components. UML 2 component diagrams are used as architecture-level artefact, either to model the business software architecture, the technical software architecture, or more often than not both of these architectural aspects. Component diagrams are particularly useful with larger teams. Initial architectural modelling efforts during cycle 0 should focus on identifying the initial architectural landscape for your system. UML component diagrams are great for doing this as they enable you to model the high-level software components, and more importantly the interfaces to those components. Once the interfaces are

defined, and agreed to by your team, it makes it much easier to organize the development effort between sub teams.

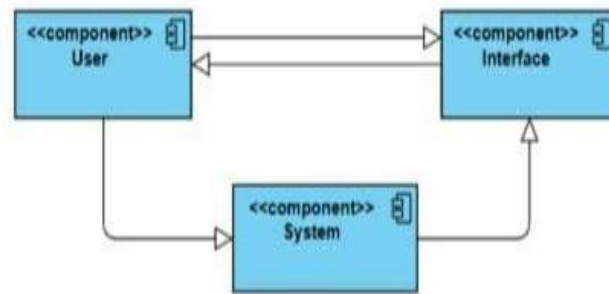


Fig 5.7 component diagram

CHAPTER 6 SYSTEM IMPLEMENTATION

6.1 INTRODUCTION

A software application in general is implemented after navigating the complete life cycle method of a project. Various life cycle processes such as requirement analysis, design phase, verification, testing and finally followed by the implementation phase result in a successful project management. System implementation is an important stage of theoretical design is turned into practical system. Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user and so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly. The final stage is to document the entire system which provides components and the operating procedures of the system.

6.1 LIST OF MODULES

- Collection of Dataset
- Image extraction and Pre-processing

- Segmentation
- Feature extraction
- Detection system

6.2 MODULE IMPLEMENTATION

6.2.1 COLLECTION OF DATASET

The datasets are collected from various trusted online resources. These data sets includes t parameters which are related body skin lesion of the patient. One of the major headache issues when developing machine learning or data science application is the data to be used, because it's the base that the application will be built on, so the data must be correct and large enough to develop a good model. To solve this common issue there are many websites that provide datasets for machine learning applications, they gather a large amount of data in different domains, organize and categorize them in a way that to be useful for the implementation in these applications, for example there are KAGGLE, UCI and OLE. For skin diseases diagnosis systems, there are DERMNET & OLE they have a large amount of skin diseases images, and these images are categorized into some general taxonomy of the skin diseases, but they don't provide a direct link for downloading these images, so the data is still headache for most of the developers in this domain.

To get data for structuring skin diseases diagnosis systems there are other options beside the online sources, for example, the database stored in the hospitals and the healthcare centers about their patients may provide an alternate resource, or the data may be collected manually from patients individually, which require large effort and is very time consuming. Generally, collecting data that fit your application is one of the most difficult steps in developing a machine learning application, so for developing the Skin Diseases Diagnosis system based on captured images, the required data are images with a labeled classes of skin diseases, so we had two choices either to collect the images manually from hospitals, healthcare centers and individual patients or using online resources for skin diseases images database. For

the manual collection of the data, we face two major problems to apply this option, the first one is that there is lack of data collection and documentation in most of the local hospitals, neither digital data nor hard copies of the data, the second problem is that if there is a data available, it's hardly reachable because it's considered private for the hospitals and require permissions from the local health authorities, also the data is not sufficiently enough for the training of the model, not well prepared and require a lot of work to be ready for usage, so we choose to search for another option to collect the data.

For the online resources option, there are a lot of researches are done to obtain resources for skin diseases diagnosis images. DERMNET.COM was the dominant resource for the system data, it is one of the largest dermatology photos resource that are available publicly. Although it has more than 23,000 skin disease images on a wide variety of skin conditions, but there was not direct way to download a forced to download each single image every time. in contrast this data couldn't be downloaded at once, we were forced to download each image individually.

To increase the reliability and generalization of our model, it must be trained on different images with different characteristic such as background color of different resources, to achieve this more images were downloaded from other multiple resources such as medicine.uiowa.edu, dermnetnz.org, dermquest.com, dermquest.com along with other websites available at dermweb.com. Most of those images were normally captured by camera with acceptable resolution that more probably will match the type of photos that will be captured by the users.

Our system has been designed to detect 3 types of skin diseases namely, Eczema, Melanoma and Acne. These classes were chosen because Acne and Eczema are of the most common skin diseases that have multiple effects on the patients on different aspects, they are painful in addition they have psychological effect caused by the changes that happen in skin specially for teenagers. Melanoma is the most dangerous form of skin cancer, most often caused by ultraviolet radiation from sunshine or tanning beds [21], If melanoma is recognized and treated early, it is

almost always curable, but if it is not, the cancer can advance and spread to other parts of the body, where it becomes hard to treat and can be fatal.

For the training of our system, 200 images for each of Acne, Eczema and Melanoma were collected, although most of the images were downloaded from Dermnet about 140 of each, but they were signed with watermark in the middle of the image which restrict us in applying different preprocessing techniques.

6.2.2 IMAGE EXTRACTION AND PREPROCESSING

Changing the nature of the image in order to be more useful is one of the basic steps before using the image in machine learning applications, it's important to enhance images for highlighting interesting details in the image, removing noise from the image or to make images more visually appealing [3]. One of the simplest forms of image preprocessing is cropping the area of interest, resizing the image or even converting the image from RGB image to grayscale image, here some examples of images preprocessing and enhancement techniques that could be used to enhance Skin Diseases images. Any image considered for the purpose of cancer detection has to go through a pre- processing stage to mainly rid it of noise. It is mandatory to study and separate abnormalities in the background on the result [24]. And by noise, we mean parts of the image that are not required for detection purposes. Any redundant parts in the background will also be cut off, helping improve the image for the next process by retaining only the required information. Using a good preprocessing technique means better accuracy . The pre-processing stage involves three processes, that of image enhancement, image restoration and hair removal. For Pre-processing of Image, Filtering is performed on image which is a non-linear process used for enhancing the overall image by preserving the edges of the image shown in Fig3. Median filtering is used especially to reduce impulsive, salt-pepper noise. In this, each pixel value in an image is replaced with the median value of its neighbouring pixels including itself.

Gray images

Images were converted from RGB – red, green and blue- type to gray scale images



Fig 6.1 RGB image is converted to gray image

6.2.3 SEGMENTATION:

Image segmentation is performed to separate suspicious lesion from normal skin. This is implemented through MATLAB. For image segmentation, multilevel thresholding using Otsu method is performed where image is segmented into 3 levels using IM Quantize with 2 threshold level. The Segmented image is converted into a color image using function label2rgb(). Multilevel thresholding is a process that segments a gray level image into several distinct regions. This technique determines more than one threshold for the given image and segments the image into certain brightness regions, which correspond to one background and several objects.



Fig -2: Original Image



Fig-3: Filtered Image



Fig -4: Segmented Image

Fig 6.2 Different stages of segmented image

A major challenge for research and development in this field includes segmentation

of skin cancer images. This process is essential in digital image processing as it is ultimately used for image description and classification. Various properties like shape, brightness, colour and/or texture are applied to assist in the segmentation of skin lesion. In the past few decades, many algorithms have been proposed for the detection of lesions in skin cancer images.

categorized the segmentation methods into:

- Histogram thresholding which separates the area of interest (ROI) and background using one or more threshold values.
- Region-based methods which incorporate the pixels into their similar regions using region-splitting and region-merging algorithms.
- Edge-based methods in which the edges of lesions are determined using edge operators .
- Active-contour methods in which the contours in the shape are made to evolve using curve evolution techniques.
- Morphological methods determine the seeds and employ watershed transform to identify contours of an object.
- Colour-clustering methods employ unsupervised clustering algorithms to generate homogeneous areas by separating the colour space.
- Soft-computing methods employ different soft-computing techniques to classify the pixels.
- Model-based methods, in which the image is considered as a random field and the model is parameterized using optimisation methods .

In segmentation, sets of objects with similar characteristics are classified into different groups. This process called clustering has been widely applied in many areas such as image processing, machine learning, pattern recognition, data mining and statistics. Recently, clustering algorithms found crucial applications in medical imaging field.

6.2.4 FEATURE EXTRACTION:

Unique features of skin lesion are extracted. Features are extracted using the 2D Wavelet Transform. Features extracted using the wavelet transform are Entropy, Mean, Mean Absolute Deviation, Median Absolute Deviation, Energy, Standard deviation, L1 norm, L2 norm, Kurtosis, Skewness. Texture Features extracted using GLCM (Grey Level Co-occurrence Matrix) are Contrast, Correlation, Energy and Homogeneity.

Entropy

$$H [x] = - \sum_x (x) \log_2 (x)$$

The entropy or average information of an image is a measure of the degree of randomness in the image. The entropy is useful in the context of image coding: it is a lower limit for the average coding length in bits per pixel which can be realized by an optimum coding scheme without any loss of information

Mean

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Image processing is a somewhat broad term in modern IT that refers to using various means to process or enhance images.

Mean Absolute Deviation

$$MAD = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

To find the mean absolute deviation of the data, start by finding the mean of the data set. Find the sum of the data values, and divide the sum by the number of data values.

Median Absolute Deviation

$$MAD = \frac{1}{T} \sum_{t=1}^T d_t$$

The mean absolute deviation of a dataset is the average distance between each data point and the mean

Energy

$$E = \sum_{k=-\infty}^{\infty} |x[k]|^2$$

Energy is a fairly loose term used to describe any user defined function (in the image domain). The motivation for using the term 'Energy' is that typical object detection/segmentation tasks are posed as a Energy minimization problem.

Standard deviation

$$SD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

In statistics, the standard deviation (SD, also represented by the lower case Greek letter sigma σ for the population standard deviation or the Latin letter s for the sample standard deviation) is a measure that is used to quantify the amount of variation or dispersion of a set of data values

L1 norm

$$L1 = [\sum_i |e_i|^1]$$

Also known as Manhattan Distance or Taxicab norm . L1 Norm is the sum of the magnitudes of the vectors in a space. It is the most natural way of measure distance

between vectors that is the sum of absolute difference of the components of the vectors.

L2 norm

$$L2 = \left[\sum_i |e_i|^2 \right]^{1/2}$$

The notation for the L2 norm of a vector is $\| \cdot \|_2$. As such, it is also known as the Euclidean norm as it is calculated as the Euclidean distance from the origin. The result is a positive distance value. The L2 norm is calculated as the square root of the sum of the squared vector values.

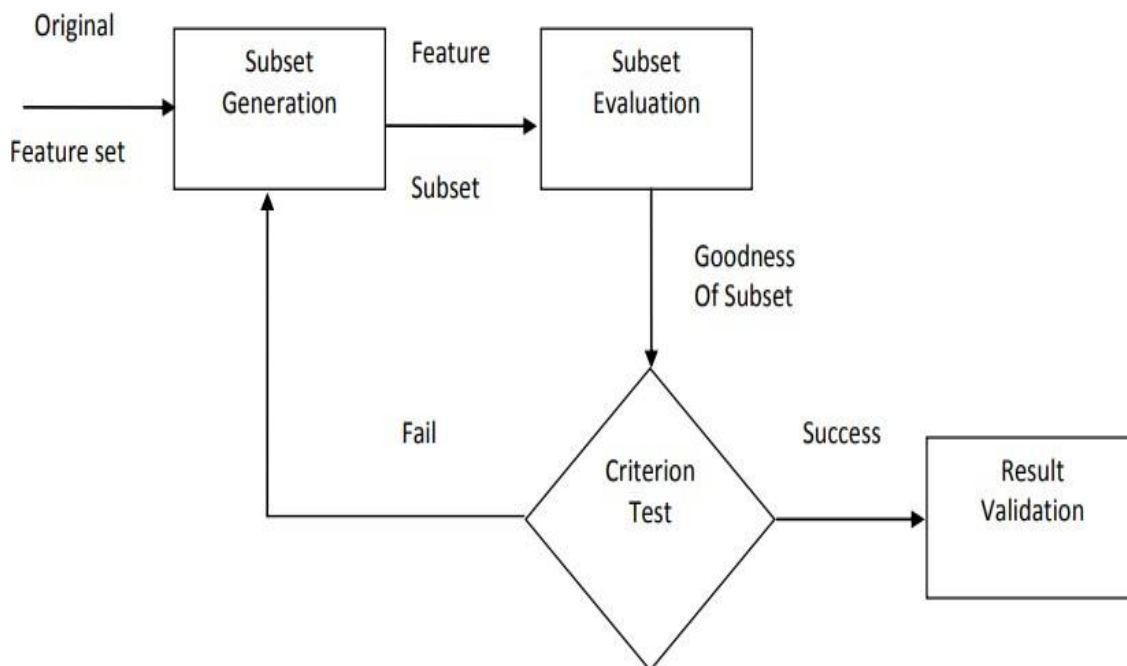


Fig:6.3 classify and selection process

6.2.5 DETECTION SYSTEM:

The final step is computerized analysis of melanoma detection, to estimate if a lesion is malignant or benign, is Lesion classification. To carry out the classification task, existing systems utilize different classification methods with feature descriptors that were extracted in the previous stage. The efficiency of these

methods appertains to both extracted descriptors and selected classifier [73]. There are different classifiers that currently exist namely Discriminant Analysis, Artificial Neural Network(ANN), K-Nearest Neighbourhood(KNN), Support Vector Machine (SVM), Decision Trees and Self-Advising SVM.

SUPPORT VECTOR MACHINE

The support vector machine is a supervised learning model used for optimization. It is a unified framework in which different learning machine architecture can be generated through an appropriate choice of kernels. The principal used in SVM is statistical and structural risk minimization. The SVM is already a ready-to-use available classifier in MATLAB. After the feature extraction process, the extracted features are directly fed into the SVM classifier.

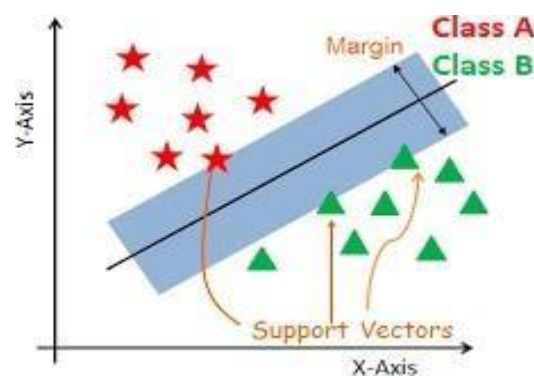


Fig 6.4 classification using SVM The

process involves two phases:

- A. Training Phase:** Huge images of eczema, impetigo, melanoma, and others are used for training.
- B. Testing Phase:** In this phase, test images are given to the classifier and the classifier uses knowledge gained during the training phase to classify the test image.

The support vector machine is a supervised learning model used for optimization. It is a unified framework in which different learning machine architecture can be

generated through an appropriate choice of kernels. The principal used in SVM is statistical and structural risk minimization. The SVM is already a ready-to-use available classifier in MATLAB. After the feature extraction process, the extracted features are directly fed into the SVM classifier

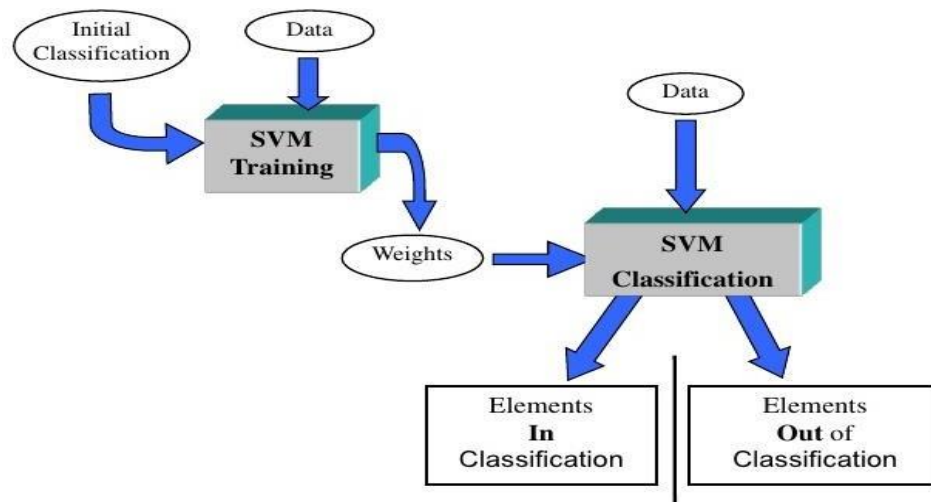


Fig : 6.5 SVM PROCESS

In SVM classifier, an image needs to be processed using image processing unit and then given for the classification to SVM classifier. CNN classifier is implemented in such a way where there is no need of image processing module. CNN classifier is a layered architecture where multiple layers perform various operations to train and test the image data.

CHAPTER 7

SOFTWARE TESTING

7.1 INTRODUCTION

Testing is a very important phase of software development which involves the process of detecting errors. In other words, testing is a process of executing a program with the intent of finding an error. Testing plays a critical role in assuring quality and ensuring the reliability of software. The results of testing are used later on during maintenance. The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies. Various testing strategies were applied at various levels (functional, modular and combined) to get a better simulation.

7.2 OBJECTIVES

- To uncover a host of errors, systematically and with minimum effort and time.
- A good test case is one that has a high probability of finding error, if it exists.
- The tests are inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

7.3 TYPES OF TESTING

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided its design and development. System testing makes a logical assumption that if all the parts of the system are correct, the goal will

be successfully achieved. There are a lot of testing types, few of which are discussed here. The discussed ones were done on the project at various stages.

7.3.1 UNIT TESTING

Unit testing, as the name suggests, is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. Unit testing is performed by using the White Box Testing method. Unit Testing is normally performed by software developers themselves or their peers. In rare cases it may also be performed by independent software testers. The testing was done in parts function wise and module wise. Various errors occurred and were mitigated.

7.3.2 INTEGRATION TESTING

It is quite obvious that the data can be lost across the interface; one module can have an adverse effect on others. **Integration testing** tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. It conducts tests to uncover errors associated within the interface conducted. The objective is to take unit tested modules and combine them test it as a whole. Thus, in the integration-testing step all the errors uncovered are corrected for the next testing steps.

The various integration testing techniques are

- Big Bang integration testing.
- Top down integration testing.
- Bottom up integration testing.

The integration was done within modules and with all modules together.

7.3.3 VALIDATION TESTING

The term validation means to check whether the specified requirements are met. Therefore, Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed on appropriate environment. It answers to the question, Is the right product built? Each function and module was validated separately and as a whole.

7.3.4 ACCEPTANCE TESTING

It is a welcomed fact that user acceptance of a system is the key factor for the success of any system. **Acceptance Testing** is a level of the software testing where a system is tested for acceptability .The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.This is done in regard to the following.

Input screen design and output screen design. An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive report is prepared. The images and symbols used in this simulation were tested for acceptance.

7.3.5 PERFORMANCE TESTING

Performance testing is a software testing practice which mainly aims at testing the responsiveness and stability of a system under various workloads. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage. The various types of performance testing includes

- **Load testing**

It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored. Huge datasets were given. Thus the system was subjected to huge loads and errors were mitigated.

- **Stress testing**

It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum. The system worked fine for huge data set which is bounded by the hardware limitations.

- **Soak testing**

Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use. Huge data sets were given for a longer period of time and were tested.

- **Spike testing**

Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload. Sudden increase in number of MS also resulted in success.

7.3.6 FUNCTIONAL TESTING

Functional tests provide systematic protests that functions tested are accessible as stated by the business and technical necessities, system certification and user guides. Functional difficult is centered on the subsequent items:

- **Valid Input** is used to identified classes of valid input must be accepted.
- **Invalid Input** is used to identified classes of illegal input must be disallowed.
- **Functions** is used to identified purposes must be exercised.
- **Output** is used to classify modules of request outputs.
- **Systems/Procedures** is used to interfacing systems or events must be appealed.

Organization and grounding of functional tests is focused on supplies, key functions, or special test belongings. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and succeeding processes must be well-thought-out for testing. Before functional testing is complete, supplementary tests are identified and the operative value of recent tests is resulted.

7.3.7 SYSTEM TESTING

System testing confirms that the entire combined software system meets supplies. It tests a configuration to ensure known and predictable outcomes. An example of system testing is the arrangement oriented system mixingtest. System testing is based on procedure similes and flows, emphasizing pre-driven process links and addition points.

7.3.8 WHITE BOX TESTING

White Box Testing is a challenging in which the software tester has information of the inner workings, construction and language of the software, or at least its drive. It is used to test areas that cannot be stretched from a black box level.

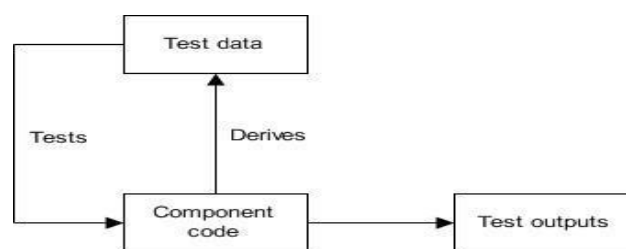


Fig 7.1 White Box testing

7.3.9 BLACK BOX TESTING

Black Box Testing is testing the software short of any knowledge of the inner mechanisms, building or language of the module actuality tested. Black box tests, as most other kinds of tests, must be printed from a final source document, such as requirement or necessities file, such as specification or requirementfile. It is a testing in which the software below test is treated, as a black box you **cannot “see” into it. The test provides inputs and respondto outputs without** seeing how the software works.

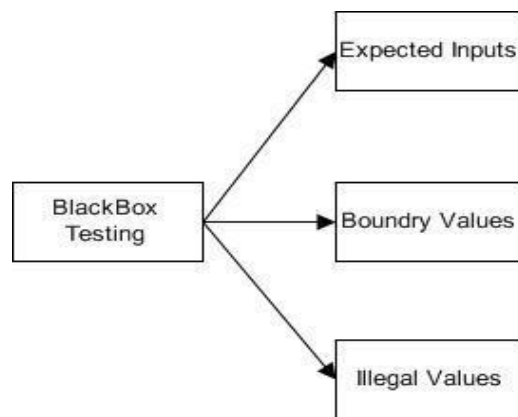


Fig:7.2 Black Box testing

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION:

Difficulties in the diagnosing skin diseases arise because of the spreading of the skin diseases all over the world, which make it a challenge to the dermatologist to recognize the different skin diseases easily, a computer aided system is proposed to resolve these difficulties, so a machine learning model based on bag of features algorithm is designed which use CNN as a classifier and SURF for feature extraction with an interface on mobile applications is developed for android devices, the core model is developed using MATLAB and the interface is developed using android studio on windows OS. The developed system performs the required work with accuracy 94% within the dataset and 85% with the external data, and the integrated system is working properly, although the system works fine there are some points that bound the performance of the system such as that, there is a single processor of the system that is used to classify the images, which will reduce the performance of the whole system with the increase of the requests from multiple users.

8.2 FUTURE ENHANCEMENT

The proposed modifications of the skin diseases diagnosis system are generally to increase the performance of the system, resolve the system limitations, or to increase its capability. So, there are several suggested modifications to both the system core model and the system mobile interface:

- Increase the training data used for training the model, not only in term of quantity but also obtaining more data from different resources namely collecting data from hospitals and healthcare centers, to increase the learning model generalization.
- Apply better preprocessing techniques to resolve the images distortions.

- Apply training data of more classes, that the model will be capable to recognize and diagnose more diseases.
- Develop a cross-platform application to work on different mobile platforms, which will increase the number of system users.
- Enhance the application interface to be more user friendly for better user experience policies.
- Develop a distributed system for skin diseases diagnosis to resolve the single server limitation, and increase the processing capabilities.
- Enhance the functionality of the system to be more useful by giving advices for the users about the disease treatment.

APPENDIX-I (SCREENSHOTS)

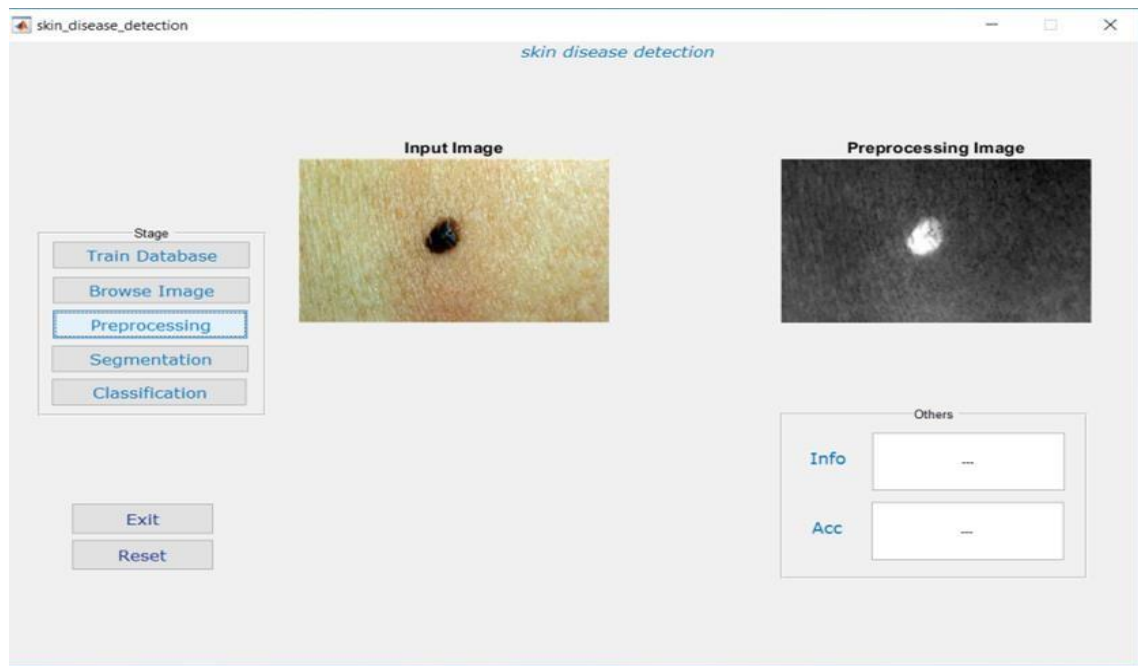


FIG: A 1.1 Admin page

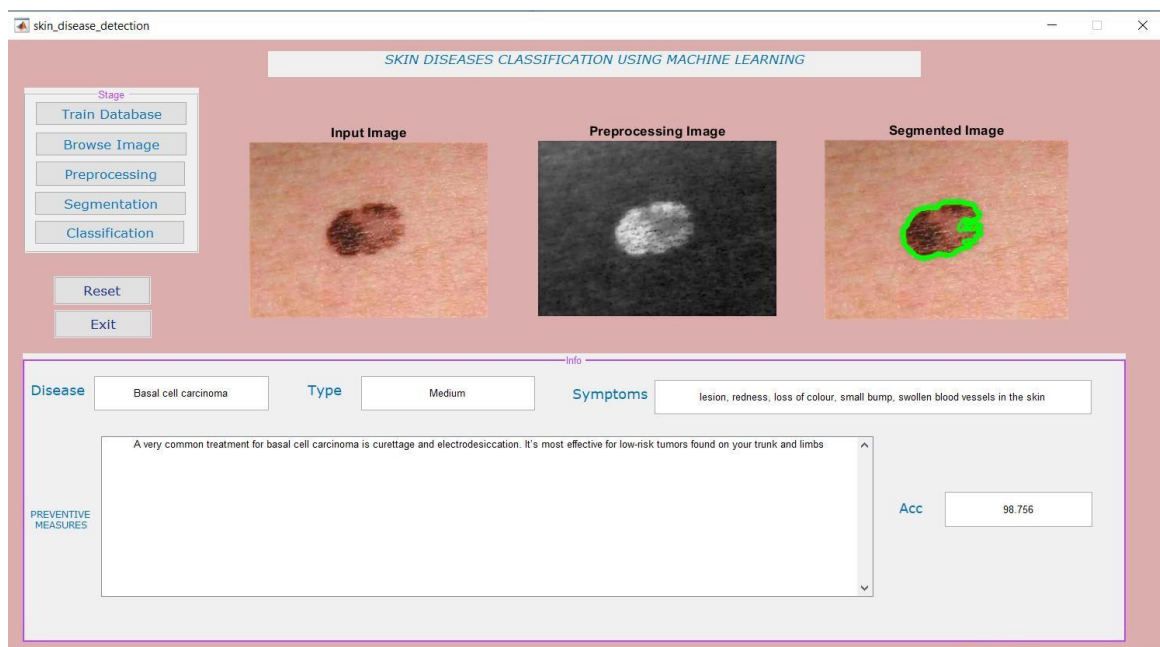


FIG: A 1.2 Detection system

APPENDIX -II(SAMPLE CODE)

Skindisease Detection

```
function varargout = skin_disease_detection(varargin)
% SKIN_DISEASE_DETECTION MATLAB code for skin_disease_detection.fig
%     SKIN_DISEASE_DETECTION, by itself, creates a new
%     SKIN_DISEASE_DETECTION or raises the existing
%     singleton*.
%
%
%     H = SKIN_DISEASE_DETECTION returns the handle to a new
%     SKIN_DISEASE_DETECTION or the handle to
%     the existing singleton*.
%
%     SKIN_DISEASE_DETECTION('CALLBACK',hObject,eventData,handles,...) calls the
%     local
%     function named CALLBACK in SKIN_DISEASE_DETECTION.M with the given input
%     arguments.
%
%     SKIN_DISEASE_DETECTION('Property','Value',...) creates a new
%     SKIN_DISEASE_DETECTION or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before skin_disease_detection_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to skin_disease_detection_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help skin_disease_detection

% Last Modified by GUIDE v2.5 07-Feb-2020 13:03:57

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @skin_disease_detection_OpeningFcn, ... 'gui_OutputFcn',
                  @skin_disease_detection_OutputFcn, ...
```

```

        'gui_LayoutFcn', [] , ...
        'gui_Callback',    []); if
nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1}); end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:}); else
    gui_mainfcn(gui_State, varargin{:}); end
% End initialization code - DO NOT EDIT

% --- Executes just before skin_disease_detection is made visible.
function skin_disease_detection_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to skin_disease_detection (see VARARGIN)

% Choose default command line output for skin_disease_detection
handles.output = hObject;
axes(handles.axes1);  axis  off
axes(handles.axes2);  axis  off
axes(handles.axes3); axis off

set(handles.edit1,'String','---');
set(handles.edit8,'String','---');
set(handles.edit9,'String','---');
set(handles.edit7,'String','---');
set(handles.edit10,'String','---');

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes skin_disease_detection wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = skin_disease_detection_OutputFcn(hObject, eventdata, handles)

```

```

% varargin cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure varargin{1}
= handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA) global
im;global a;
[fname,path]=uigetfile('*.','Browse Ear'); if
fname~=0
    img=imread([path,fname]);
    axes(handles.axes1); imshow(img);title('Input Image');
    im=img;a=im;
meann = mean2(a); y =
    round(meann);
    disp(y)
    tts('Input Image is selected'); else
    warndlg('Please Select the necessary Image File');
    tts('Please Select the necessary Image File');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA) global
im;global I;global K;global BW1;global B;
global f;global C;global S;global se;global close;global gray;global wlab;global Skin;
im = im2double(im); lab
= rgb2lab(im);
f = 0;
wlab = reshape(bsxfun(@times,cat(3,1-f,f/2,f/2),lab),[],3); [C,S] =
pca(wlab);
S = reshape(S,size(lab)); S =
S(:, :,1);

```

```

Skin=86;
gray = (S-min(S(:)))/(max(S(:))-min(S(:))); se =
strel('disk',1);
close = imclose(gray,se); K=
imcomplement(close)
[cA,cH,cV,cD] = dwt2(K,'bior1.1');
thresh1 = multithresh(cA);
thresh2 = multithresh(cH);
thresh3 = multithresh(cV);
thresh4 = multithresh(cD);
level = (thresh1 + thresh2 + thresh3 + thresh4)/2; X =
idwt2(cA,cH,cV,cD,'bior1.1') BW=imquantize(X,level);
BW1 = edge(edge(BW,'canny'), 'canny'); axes(handles.axes2);
imshow(K);title('Preprocessing Image'); tts('Preprocess is done');

```

```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA) global
im;global B;global BW1;global gk;global Skin; tts('Segmentation is Started.
');
BW3 = imclearborder(BW1); CC
= bwconncomp(BW3);
S = regionprops(CC, 'Area'); L =
labelmatrix(CC);
BW4 = ismember(L, find([S.Area] >= 100)); BW5
= imfill(BW4,'holes');
[B,L,N] = bwboundaries(BW5);
axes(handles.axes3); imshow(im);title('Segmented Image');hold on; for
k=1:length(B),
    boundary = B{k};
    plot(boundary(:,2),...
        boundary(:,1),'g','LineWidth',2);
end
gk=8;

```

```

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
msgbox('Ready to Create the database..... ');
tts('Ready to Create the database..... ');
pause(1) msgbox('Process
Start. .... ');
tts('Process Start. .... ');
pause(1)

```

```

s=0;
for i=1:6 j=num2str(i);
    imgname=strcat(j, '.jpg');
    b=imread(imgname);
    b=imresize(b,[250 250]); save
    database b
end
msgbox('Database Created..... ');
tts('Database Created..... ');

```

```

function edit1_Callback(~, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

```

% -----Executes during object creation, after setting all properties.

```

function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white'); end

```

SVM(classification)

```
global gk;global a;
meann = mean2(a);
y = round(meann); disp(y)
if (y==163 && gk==8) || (y==148 && gk==8)
set(handles.edit1,'String','Basal cell carcinoma');
set(handles.edit8,'String','Medium');
set(handles.edit9,'String','lesion, redness, loss of colour, small bump, swollen blood vessels in the
skin');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','A very common treatment for basal cell carcinoma is curettage and
electrodesiccation. It's most effective for low-risk tumors found on your trunk and limbs');

elseif (y==161 && gk==8) set(handles.edit1,'String','squamous
cell carcinoma'); set(handles.edit8,'String','High');
set(handles.edit9,'String','A firm, red nodule and A flat sore with a scaly crust. '); set(handles.edit7,
'Min', 0, 'Max', 2);
set(handles.edit7,'String','Mohs Surgery. Mohs surgery has the highest cure rate of all therapies
for squamous cell carcinomas. Curettage and Electrodesiccation. This very common treatment
for squamous cell carcinoma is most effective for low-risk tumors');

elseif (y==111 && gk==8) set(handles.edit1,'String','acne');
set(handles.edit8,'String','High');
set(handles.edit9,'String','A firm, red nodule and A flat sore with a scaly crust. '); set(handles.edit7,
'Min', 0, 'Max', 2);
set(handles.edit7,'String','First-line treatment for mild acne vulgaris includes benzoyl peroxide or
a topical retinoid, or a combination of topical medications including topical antibiotics.
Tetracyclines are the preferred oral antibiotic, and doxycycline and minocycline have been
shown to be more effective than tetracyclineFirst-line treatment for mild acne vulgaris includes
benzoyl peroxide or a topical retinoid, or a combination of topical medications including topical
antibiotics. Tetracyclines are the preferred oral antibiotic, and doxycycline and minocycline have
been shown to be more effective than tetracycline');

elseif (y==165 && gk==8) || (y==67 && gk==8)
set(handles.edit1,'String','BASAL CANCER ');
set(handles.edit8,'String','Small');
```

```

set(handles.edit9,'String','This cancer typically appears as a white, waxy lump or a brown, scaly
patch on sun-exposed areas. ');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','A very common treatment for basal cell carcinoma is curettage and
electrodesiccation. It's most effective for low-risk tumors found on your trunk and limbs');

```

```

elseif (y==170 && gk==8) set(handles.edit1,'String','psoriasis');
set(handles.edit8,'String','Small');
set(handles.edit9,'String','This cancer typically appears as a white, waxy lump or a brown, scaly
patch on sun-exposed areas. ');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','Salicylic acid: This can help reduce swelling and remove scales, often in
people with scalp psoriasis. ');

```

```

elseif (y==137 && gk==8) || (y==185 && gk==8)
set(handles.edit1,'String','melanoma ');
set(handles.edit8,'String','Medium');
set(handles.edit9,'String','unusual growth or a change in an existing mole. Melanomas can
occur anywhere on the body');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','Surgery to remove affected lymph nodes. If melanoma has spread to
nearby lymph nodes, your surgeon may remove the affected nodes');

```

```

elseif (y==137 && gk==8)
set(handles.edit1,'String','melanoma ');
set(handles.edit8,'String','Medium');
set(handles.edit9,'String','unusual growth or a change in an existing mole. Melanomas can
occur anywhere on the body');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','Surgery to remove affected lymph nodes. If melanoma has spread to
nearby lymph nodes, your surgeon may remove the affected nodes');

```

```

elseif (y==136 && gk==8) set(handles.edit1,'String','rosacea');
set(handles.edit8,'String','Medium');
set(handles.edit9,'String','unusual growth or a change in an existing mole. Melanomas can
occur anywhere on the body');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','For mild to moderate rosacea, your doctor may prescribe a cream or gel
that you apply to the affected skin. Brimonidine (Mirvaso) and

```

oxymetazoline (Rhofade) reduce redness by constricting blood vessels. You may see results within 12 hours after use.');

```
y=y+20;
elseif (y==181 && gk==8) set(handles.edit1,'String','eczema
'); set(handles.edit8,'String','small');
set(handles.edit9,'String','A rash that typically appears onthe arms and behind the knees');
set(handles.edit7, 'Min', 0, 'Max', 2);
set(handles.edit7,'String','Applying moisturizer within 3 minutes of bathing to “lock in” moisture,
moisturizing every day, wearing cotton and soft fabrics, and avoiding rough, scratchy fibers and
tight-fitting clothing')
```

```
y=y-30;
elseif (y==201 && gk==8)
set(handles.edit8,'String','Small');
set(handles.edit9,'String','***');
set(handles.edit1,'String','Normal Issue '); y=y-
30;
```

```
else
    tts('Contact Admin .....');
end run('SVM1.m');
```

REFERENCES

- [1].Nisha Yadav, Virender Kumar Narang, Utpal shrivastava, “Skin Diseases Detection Models using Image Processing”, International Journal of Computer Applications (0975 – 8887) Volume 137 – No.12, March 2016. [2]
- [2] Er.Shrinidhi Gindhi, Ansari Nausheen, Ansari Zoya, Shaikh Ruhin, “An Innovative Approach for Skin Disease Detection Using Image Processing and Data Mining”, “International Journal of Innovative Research in Computer and Communication Engineering (IJRCCE)”, April 2017. [3]
- [3] A.A.L.C.Amarathunga, E.P.W.C. Ellawala, G.N. Abeysekara, C. R. J. Amalraj, “Expert System For Diagnosis Of Skin Diseases”, “International Journal Of Scientific & Technology Research Volume 4, (IJSTR)”, January 2015.
- [4] Amrutha Ravi, Sreejith S, “A Review on Brain Tumour Detection Using Image Segmentation”, “International Journal of Emerging Technology and Advanced Engineering (IJETAEE)”, June 2015.
- [5] Aswin.R.B, J. Abdul Jaleel, Sibi Salim³, “Implementation of ANN Classifier using MATLAB for Skin Cancer Detection”, International Journal of Computer Science and Mobile Computing (IJCSMC), December 2013.
- [6] Rahat Yasir, Md. Ashiqur Rahman, Nova Ahmed, “Dermatological Disease Detection Using Image Processing And Artificial Neural Network”, 8th International Conference On Electrical & Computer Engineering, December 2014
- [7]. Nazia Hameed , Antesar M. Shabut , M. A. Hossain ”Multi Class Skin Disease classification Using Deep Convolutional Neural Network and Support Vector Machine” , 2018 12th International Conference on Software, Knowledge, Information Management & Applications
- [8] Jingyi Lin ,Zijian Guo , Dong Li , Xiaorui Hu , Yun Zhang, “Automatic Classification of Clinical Skin Disease Images with Additional High-Level Position Information”

- [9] K. Melbin ; Y.Jacob Vetha Raj “An Enhanced Model for Skin Disease Detection using Dragonfly Optimization based Deep Neural Network” ,2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)
- [10] Md. Zahid Hasan ; Shadman Shoumik ; Nusrat Zahan,”Integrated Use of Rough Sets and Artificial Neural Network for Skin Cancer Disease Classification” 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering
- [11] Nikos Petrellis ,”Using color signatures for the classification of skin disorders” 2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)