

DETECTING SKIN CANCER BASED ON AUGUMENTED REALITY

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering degree in Computer Science and Engineering

BY

BATCHU ASHIK (37110080)

&

DILEEP BANDA (37110074)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC I 12B Status

by UGC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600 119

March – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC | Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Mr. **BATCHU ASHIK** (17SCS8015), **MR. DILEEP BANDA** (17SCS0040) who carried out the project entitled "**DETECTING SKIN CANCER BASED ON AUGMENTED REALITY**" under my supervision from November 2020 to March 2021.

Internal Guide

(Dr. USHA NANDINI M.E., Ph.D .,)

Head of the Department
Dr.L.Lakshmanan M.E., Ph.D.,
Dr.S.Vigneshwari M.E., Ph.D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **BATCHU ASHIK** and **DILEEP BANDA** hereby declare that the Project Report entitled " **DETECTING SKIN CANCER BASED ON AUGUMENTED REALITY**" done by us under the guidance of **Dr. USHA NANDINI M.E., Ph.D .**,Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** 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 and **Dr.L.Lakshmanan M.E., Ph.D.**, and **Dr.S.Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering 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. USHA NANDINI 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 Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This project discussing on the implementation of skin cancer detection using the SVM algorithm. There are so many skin cancers but here in this project we mainly focus on 3 diseases are Acne, Eczema, and Melanoma. laptop vision will play a vital role in Medical Image identification and it's been proved by several existing systems. For the detection of skin cancer, we divide our project into three modules. Initially, skin pictures were preprocessed and here we use Gaussian blur filter before edge detection aims to remove noise and irrelevant background by filtering and transformation. Then the preprocessed image was done by method Gabor filter then it was introduced to segmentation. Where in segmentation the image was converted into a grayscale image for easy extract the properties. Then we convert the image into a conventional matrix. With the help of the conventional matrix, we define the type of cancer by using the SVM (Super Vector Machine) classifier. The experimental results demonstrate the effectiveness and practicability of the planned methodology.

TABLE OF CONTENTS

	ABSTRACT	v
	LIST OF FIGURES	Viii
	LIST OF TABLES	ix
CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	
	1.1 INTRODUCTION	1
	1.2 OBJECTIVES	1
	1.3 METHODOLOGY	2
2.	AIM AND SCOPE	
	2.1 AIM	3
	2.2 MOTIVATION	3
	2.3 PROBLEM STATEMENT	3
	2.4 LITERATURE REVIEW	4
3.	SYSTEM DESIGN & METHODOLOGY	
	3.1 SYSTEM REQUIREMENTS	5
	3.2 DATA SET	5
	3.3 SERVER SIDE	7
	3.3.1MATLAB APPLICATION	7
	3.3.1.1 LEARNING MODEL	7
	3.3.1.2 SERVER CONNECTION	7
	3.3.1.3 PREPROCESSING AND IMAGE	7
	MANIPULATION	
	3.3.2 MATLAB APPLICATION FUNCTIONALITY	8

4.	SOFTWARE DEVELOPMENT METHODOLOGY	
	4.1 SYSTEM ARCHITECTURE	10
	4.2 MODULES	10
	4.2.1 PREPROCESSING	11
	4.2.1.1 GAUSSIAN BLUR	11
	4.2.2 SEGMENTATION	16
	4.2.2.1 GABOR FILTERS	16
	4.2.3 FEATURE EXTRACTION & CLASSIFICATION	20
	4.2.3.1 SVM	20
	4.2.3.1.1 SELECTING THE BEST HYPER- PLANE	21
	4.2.3.1.2 SVM KERNEL	24
	4.2.3.1.3 ADVANTAGES OF SVM	24
5.	RESULTS AND DISCUSSION	
	5.1 LEARNING MODEL RESULTS	25
	5.2 DATA QUANTITY AND VARIATION EFFECT ON ACCURACY	26
	5.3 APPLICATION SNAPSHOTS	28
6.	CONCLUSION AND FUTUREWORK	
	6.1 CONCLUSION	31
	6.2 FUTURE WORK	31
	REFERENCES	33
	APPENDIX	
	A. SOURCE CODE	36
	B. SCREEN SHOTS	42
	C. JOURNAL PUBLICATION	45
	D. PLAGIARISM REPORT	48

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	System Architecture	10
4.2	Gaussian Distribution	12
4.3	Gaussian distribution with mean	13
4.4	Discrete Approximation to Gaussian Function	14
4.5	One of the pair of 1-D convolution kernels	15
4.6	Real Part and Imaginary Part of The Impulse Response of a Gabor Filter	17
4.7	Magnitude of a Gabor Filter Set	18
4.8	Half-Value Plot of The Gabor Filters	19
4.9	Sublimates of The Lenna-Picture and Magnitudes of The Corresponding Filter Results	20
4.10	Linearly Separable Data Points	21
4.11	Selecting the Best Hyper-Plane	22
5.1	Input Image	28
5.2	Inserting Image to Application	28
5.3	Preprocessing	29
5.4	Segmentation	29
5.5	Feature Extraction	30
5.6	Classification	30

LIST OF TABLES

TABLE No.	TABLE NAME	PAGE No.
5.1	Confusion matrix of Cross-Validation method	25
5.2	Confusion matrix of the Holdout Method	26
5.3	Effect of Dataset Quantity on the Accuracy	27
5.4	Results of preprocessed images	27

CHAPTER 1

INTRODUCTION

N

1.1 INTRODUCTION

The skin is the largest organ in the human body and has numerous potential abnormalities, there are about 1500 distinct skin diseases. We are relatively ignorant about the symptoms of the majority of these diseases although knowledge is rapidly increasing, however, that makes it a challenge for dermatologist to diagnose them.

Nowadays technologies have changed our day-to-day life in all aspects and the medical field is not an exception, many medical systems have been developed to help both patients and doctors in different ways, starting from registration process ending with the use of technologies for diagnosing diseases.

This chapter gives an overview about the problem statement of the project, the objectives to be achieved, motivation of the project and brief view about the methods used in the implementation. In addition, an overview of report layout will be presented.

1.2 OBJECTIVES

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

1. Develop machine learning application that in general, has the ability to:
 - Determine the affected areas in the image.
 - Determine the disease in the specified region.
 - Determine the type of disease.
2. Develop a desktop application:

- Take the image and given the disease type.

1.3 METHODOLOGY

This project will implement machine learning techniques for diagnosing Skin Diseases, the input data is an image of the infected skin, the system will determine the type of the disease.

Large number of Skin Diseases images were collected from many online resources such as dermnet.com, medicine.uiowa.edu, dermnetnz.org and others; we choose to classify between 3 famous Diseases which are Acne, Eczema and Melanoma.

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The aim of this project is to develop and implement a model to get output for specific disease by using SVM algorithm.

2.2 MOTIVATION

Sudan is one of the developing countries that need a lot of attention in order to improve the life style of its citizens. Technology needs to be improved on many aspects, especially in the medical field, because of its sensitivity and effect in human lives which requires accurate and objective diagnosis, one of the important and common regions is the branch of medicine that dealing with the skin, nails, hair and its diseases which is called Dermatology.

2.3 PROBLEM STATEMENT

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.

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.

Since most of the Sudanese have dark skin then most of the applications that have been developed in this region are not properly applicable locally in Sudan. On the other hand, everywhere desktops have become one of our life's bases, they could be used in many ways, this software would be very accurate by implementing machine learning and artificial intelligence techniques in addition its available, reliable and very easy to use.

2.4 LITERATURE REVIEW

Skin diseases, in the last few decades, become very common all over the world, for example number of skin cancer infections has been doubled in the past 15 years [2], with this extensive spreading and the severity of large number of the skin diseases arise the need for radical solutions for these diseases, 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. This motivates 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, different machine learning and data science techniques were implemented in those systems because of their ability to recognize patterns in objective manner.

CHAPTER 3

SYSTEM DESIGN & METHODOLOGY

3.1 SYSTEM REQUIREMENTS

1 Software

Matlab2013

2 Hardware

intel i5

Ram – 8gb

Windows

10

AR (Augmented reality) set

3.2 DATA SET

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 whole class of diseases at once, so we were 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, that will be explained later,

this is another reason why we download more data from other sites.

3.3 SERVER SIDE

The server side of the skin diseases diagnosis system consist of a MATLAB application running in the server environment, which receives the input image from the MATLAB application and return back the result through the server.

3.3.1 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:

3.3.1.1 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 Image Datastore object and feed it to the model to be trained as explained in section.

3.3.1.2 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.

3.3.1.3 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.

3.3.2 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.

Then at each side, the client and the server side some certain libraries are used to establish a connection, volley for the android application and JDBC connector for the MATLAB application.

Ports

There are two connection entities in the server each one of them uses specific ports for the apache server there are port 443 and 808, the default ports are 80 and 443 but has been changed to resolve the conflict with other applications that uses the port 80 as default too, and for MySQL there is port 3306.

Sequence flow of data between the client and server

This part describe the sequence of actions to perform a connection between the client and server applications, first at the server side the MATLAB application establish the connection to the server and keep listening to the changes in the database, when a user need to diagnose an image, the android application captures the image and uploaded it to pre- specified file in the server, then write the image name and path to the database, when the MATLAB application detect the data written in the database it's triggered to perform a classification, so it reads the new image path from the database, and upload the image by its path from the server file, then process the image and write the results back to the database column classification, then the android application read the result outcome from the database and view it to the user.

CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

4.1 SYSTEM ARCHITECTURE

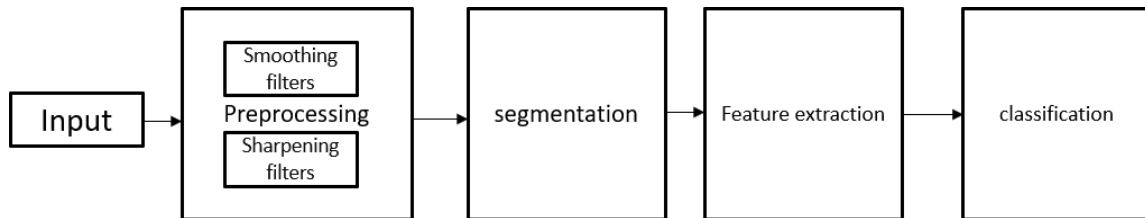


Fig:4.1: SYSTEM ARCHITECTURE

4.2 MODULES

- Preprocessing
- Segmentation
- Feature extraction
- Classification

4.2.1 PREPROCESSING

- Taking input image from the user, and then the image was preprocessing.
- Here we use gaussian filter algorithm for preprocessing.
- The algorithm is worked on Using a **Gaussian Blur filter** before edge detection aims to reduce the level of noise in the image.
- which improves the result of the following edge-detection algorithm.
- It has two states.

a. **Smoothing filters:** Smoothing filters are basically used for noise reduction, those filters could be used to remove small undesired details from the image, such as hair, salt and pepper noise, air bubbles and background noise

b. **Sharpening filters:** Sharpening filters are used to highlight fine details such as the edges or to enhance details that have been blurred in the image, also sharpening filters are applied in both spatial and frequency domains.

4.2.1.1 Gaussian Blur

A Gaussian filter is a linear filter. It's usually used to blur the image or to reduce noise. If you use two of them and subtract, you can use them for "unsharp masking" (edge detection). The Gaussian filter alone will blur edges and reduce contrast.

The Median filter is a non-linear filter that is most commonly used as a simple way to reduce noise in an image. It's claims to fame (over Gaussian for noise reduction) is that it removes noise while keeping edges relatively sharp.

I guess the one advantage a Gaussian filter has over a median filter is that it's faster because multiplying and adding is probably faster than sorting.

The Gaussian smoothing operator is a 2-D convolution operator that is used to 'blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian ('bell-shaped') hump. This kernel has some special properties which are detailed below.

The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where σ is the standard deviation of the distribution. We have also assumed that the distribution has a mean of zero (i.e. it is centred on the line $x=0$). The distribution is illustrated in Figure 1.

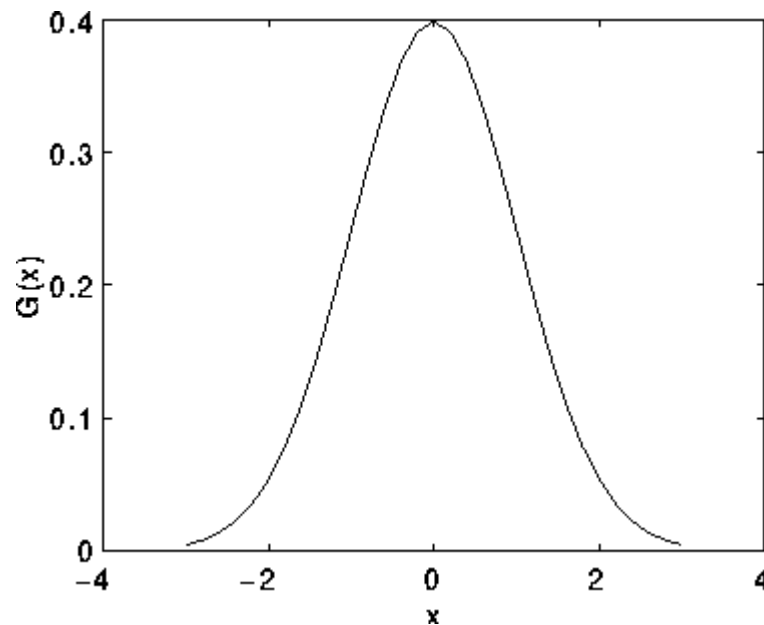


Fig:4.2: 1-D Gaussian distribution with mean 0 and infinity =1

In 2-D, an isotropic (*i.e.* circularly symmetric) Gaussian has the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This distribution is shown in Figure 2.

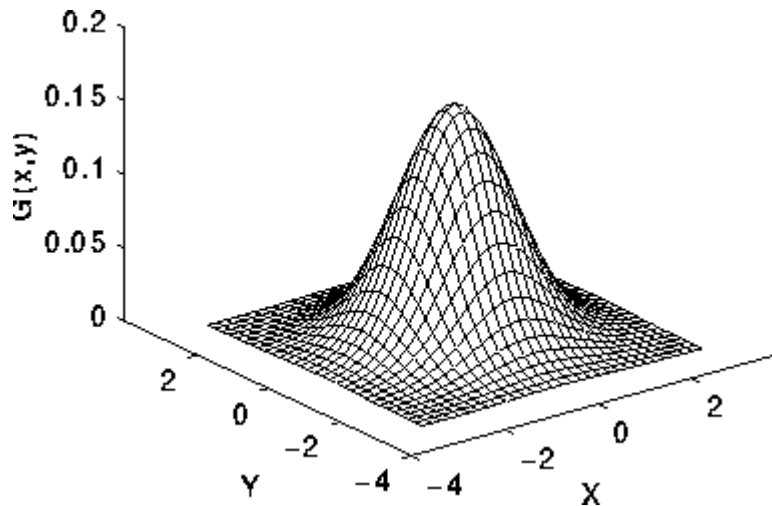


Fig:4.3: 2D Gaussian distribution with mean (0,0) and infinity =1

The idea of Gaussian smoothing is to use this 2-D distribution as a 'point-spread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. Figure 3 shows a suitable integer-valued convolution kernel that approximates a Gaussian with a infinity of 1.0. It is not obvious how to pick the values of the mask to approximate a Gaussian. One could use the value of the Gaussian at the centre of a

pixel in the mask, but this is not

accurate because the value of the Gaussian varies non-linearly across the pixel. We integrated the value of the Gaussian over the whole pixel (by summing the Gaussian at 0.001 increments). The integrals are not integers: we rescaled the array so that the corners had the value 1. Finally, the 273 is the sum of all the values in the mask.

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Fig:4.4: Discrete approximation to Gaussian function with infinity =1.0

Once a suitable kernel has been calculated, then the Gaussian smoothing can be performed using standard convolution methods. The convolution can in fact be performed fairly quickly since the equation for the 2-D isotropic Gaussian shown above is separable into x and y components. Thus the 2-D convolution can be performed by first convolving with a 1-D Gaussian in the x direction, and then convolving with another 1-D Gaussian in the y direction. (The Gaussian is in fact the *only* completely circularly symmetric operator which can be decomposed in such a way.) Figure 4 shows the 1-D x component kernel that would be used to produce the full kernel shown in Figure 3 (after scaling by 273, rounding and truncating one row of pixels around the boundary because they mostly have the

value 0. This

reduces the 7x7 matrix to the 5x5 shown above.). The y component is exactly the same but is oriented vertically.

.006	.061	.242	.383	.242	.061	.006
------	------	------	------	------	------	------

Fig:4.5: One of the pair of 1-D convolution kernels used to calculate the full kernel shown in Figure 3 more quickly.

A further way to compute a Gaussian smoothing with a large standard deviation is to convolve an image several times with a smaller Gaussian. While this is computationally complex, it can have applicability if the processing is carried out using a hardware pipeline.

The Gaussian filter not only has utility in engineering applications. It is also attracting attention from computational biologists because it has been attributed with some amount of biological plausibility, e.g. some cells in the visual pathways of the brain often have an approximately Gaussian response.

4.2.2 SEGMENTATION

- After preprocessing the out image was goes to segmentation process.
- For segmentation, we are using the **Gabor filter**
- **Features** are **extracted** directly from gray-scale character images by **Gabor** filters which are specially designed from statistical information of character structures.
- An adaptive sigmoid function is applied to the outputs of **Gabor** filters to achieve better performance on low-quality images.

4.2.2.1 GABOR FILTERS:

Gabor filters are bandpass filters which are used in image processing for feature extraction, texture analysis, and stereo disparity estimation. The impulse response of these filters is created by multiplying a Gaussian envelope function with a complex oscillation. Gabor showed that these elementary functions minimize the space (time)- uncertainty product. By extending these functions to two dimensions it is possible to create filters which are selective for orientation. Under certain conditions the phase of the response of Gabor filters is approximately linear. This property is exploited by stereo approaches which use the phase-difference of the left and right filter responses to estimate the disparity in the stereo images. It was shown by several researchers that the profile of simple-cell receptive fields in the mammalian cortex can be described by oriented two-dimensional Gabor functions.

Let $\mathbf{x}=[x_1 \ x_2]^T$ be the image coordinates. The impulse response of a Gabor filter $g(\mathbf{x})$ is then given by:

$$g_{mn}(\mathbf{x}) = \frac{1}{2\pi a_n b_n} e^{-\frac{1}{2}\mathbf{x}^T \mathbf{A}_{mn} \mathbf{x}} e^{jk_{0mn}^T \mathbf{x}}$$

Here, the matrix $\mathbf{A}$ determines the bandwidth and the orientation selectivity of the filter.

$$A_{mn} = \begin{bmatrix} \cos\phi_m & -\sin\phi_m \\ \sin\phi_m & \cos\phi_m \end{bmatrix} \begin{bmatrix} a_n^{-2} & 0 \\ 0 & b_n^{-2} \end{bmatrix} \begin{bmatrix} \cos\phi_m & \sin\phi_m \\ -\sin\phi_m & \cos\phi_m \end{bmatrix}$$

If the modulation frequency vektor $\mathbf{k}_0$ is in the same direction as one of the envelopes axes, with

$$\mathbf{k}_{0mn} = k_{0n} \begin{bmatrix} \cos\phi_m \\ \sin\phi_m \end{bmatrix}$$

then the real part and the imaginary part of the filter's impulse response are looking as follows:

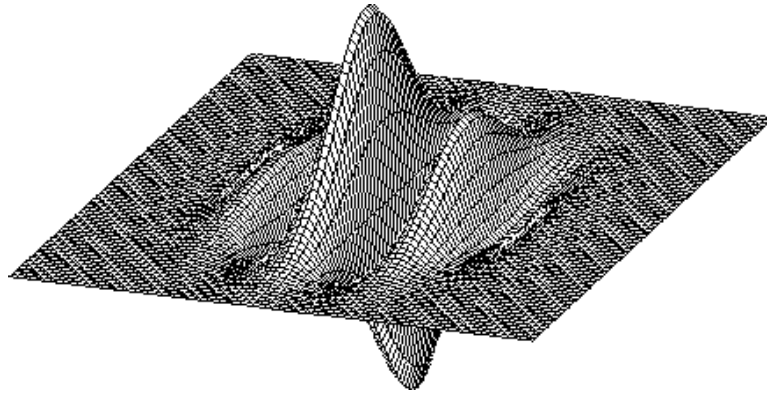


FIG:4.6: REAL PART AND IMAGINARY PART OF THE IMPULSE RESPONSE OF A GABOR FILTER

The transfer function $G(\mathbf{k})$ of a Gabor filter (Fourier transform of the impulse response) is given by:

$$G_{mn}(\mathbf{k}) = e^{-\frac{1}{2}(\mathbf{k} - \mathbf{k}_{0mn})^T (\mathbf{A}_{mn}^{-1})^T (\mathbf{k} - \mathbf{k}_{0mn})}$$

where $\mathbf{k} = [k_1 \ k_2]^T$ is the spatial frequency. In order to establish a multi-resolution strategy the image can be filtered with a set of N Gabor filters with different

bandwidths and modulation frequencies. If the modulation frequencies are given by

$$k_{0n} = \frac{\pi}{2^{n+1}}; \quad n \in [0 \dots N-1]$$

and the relative bandwidth is chosen to be constant for all filters the image is decomposed into octaves.

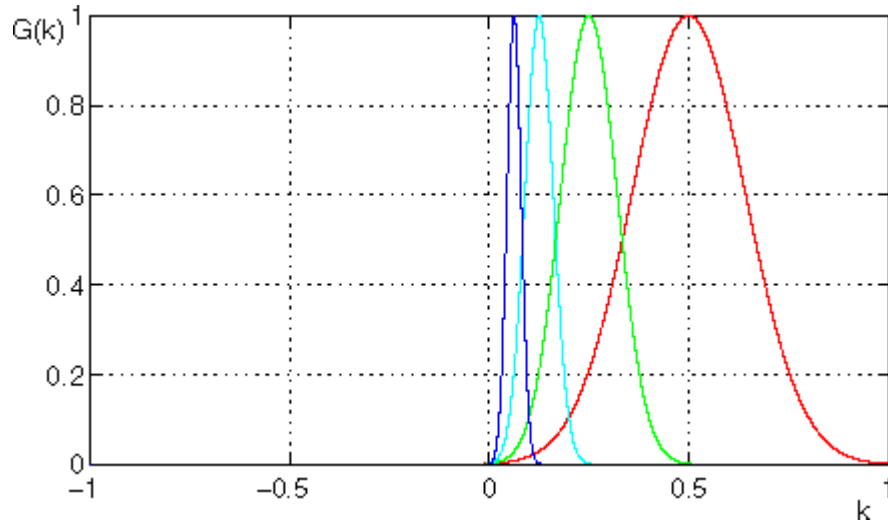


FIG:4.7 MAGNITUDE OF A GABOR FILTER SET FOR N=4 IN DIRECTION OF THE MODULATION FREQUENCY

In the figure shown above the transfer functions of the filters are chosen to overlap at a value of 0.5. Under this condition the DC component in the images and the frequency components lying above the frequency twice as large as the modulation frequency are damped by at least -54dB. Thus the filter results can be sub-sampled by a factor

$$s_n = 2^n; \quad n \in [0 \dots N-1]$$

with neglectable aliasing effects. Furthermore the images can be decomposed into M channels of different orientations by using filters tuned to different angles:

$$\phi_m = m\Delta\phi; \quad m \in [0 \dots M-1]$$

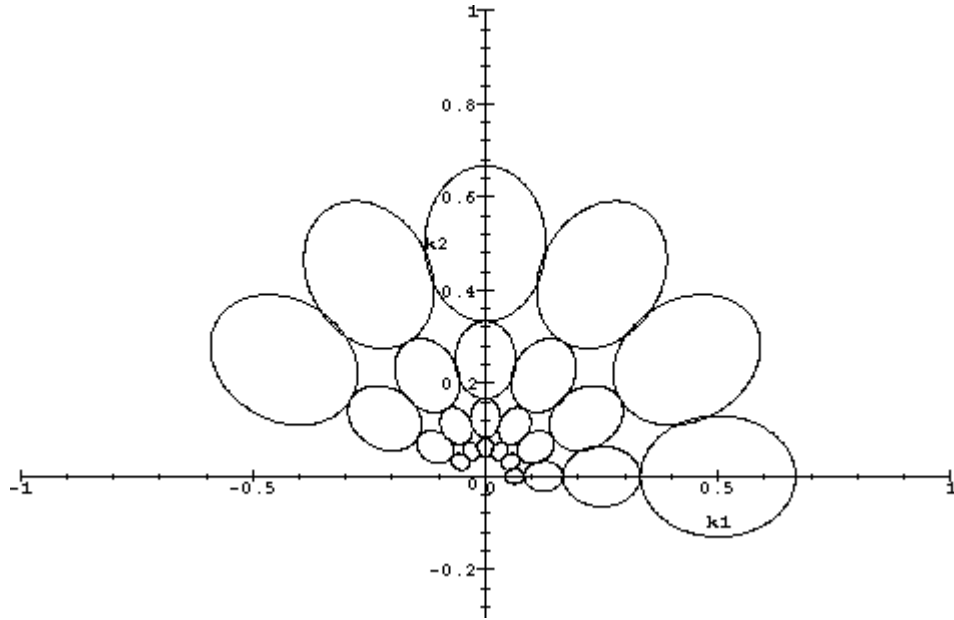


Fig:4.8: Half-value plot of the Gabor filters in the frequency plane tuned to different frequencies and orientations (30degree resolution)

The responses of the filters lying in the lower frequency half-plane need not to be calculated explicitly, because they can be generated by the complex-conjugation of the filter response lying on the other side of the origin. The following images show the original Lenna picture and Gabor filter results for different sampling factors. The size of the input image is chosen such that the filter results have always the same size after subsampling. This strategy leads to a multi resolution architecture which is also called Gabor pyramid. In the best resolution channel it is possible to extract fine image structures of a small image region whereas in the lower resolution levels coarse image structures can be extracted over large regions. This can be exploited in coarse-to-fine strategies. On the left side of the following pictures the original input (sub) images and the superposed filter results for all orientations are shown. On the right side the subsampled filter result is shown for each orientation.

$$s = 1$$

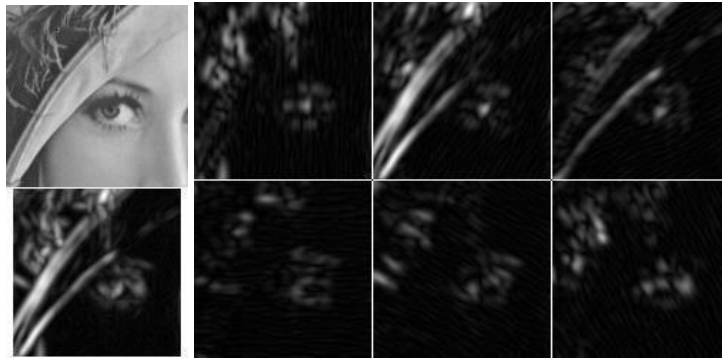


Fig:4.9: Subimages of the Lenna-picture and magnitudes of the corresponding filter results

4.2.3 Feature extraction & classification:

- The grey image that was taken from the segmentation, which was further goes through the feature extraction.
- For the feature extraction we have so many algorithms like Corner Detection and SVM, CNN.
- By Using SVM algorithm we convert the image data in to conventional matrix.
- In conventional matrix we taken the values of Contrast, Correlation, Energy, Homogeneity, Mean, Standard Deviation, Entropy, RMS, Variance, smoothness, Kurtosis, Skewness, IDM.
- With the help of the conventional matrix we generate the accuracy of the output. With the help of output, we compare with data set values.
- After that, it will display which type of disease.

4.2.3.1 SVM

support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The

dimension of

the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Let's consider two independent variables x_1 , x_2 and one dependent variable which is either a blue circle or a red circle.

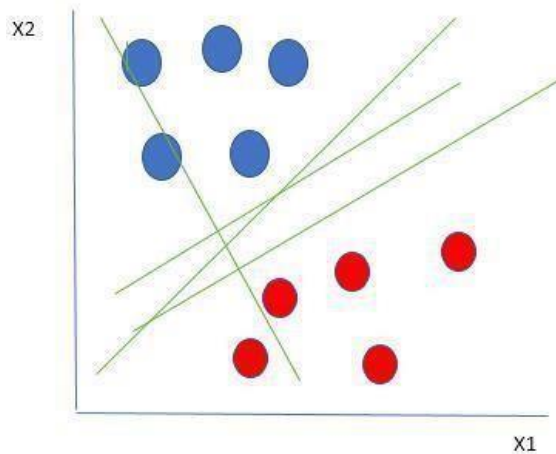


Fig:4.10: Linearly Separable Data points

From the figure above its very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x_1 , x_2) that segregates our data points or does a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points.

4.2.3.1.1 SELECTING THE BEST HYPER-PLANE

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.

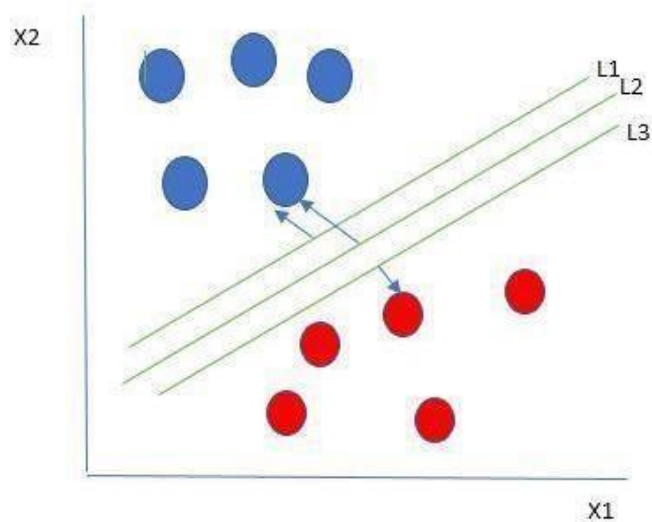
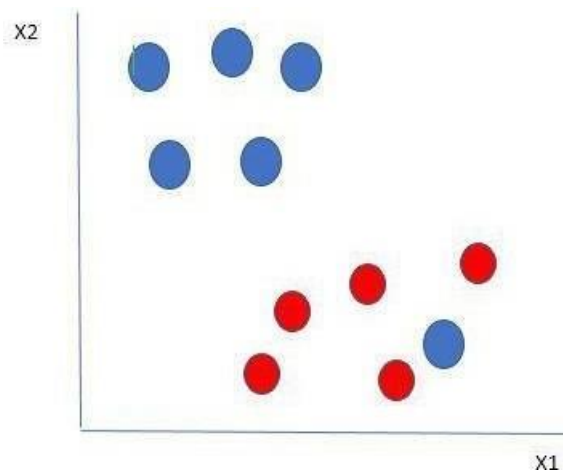


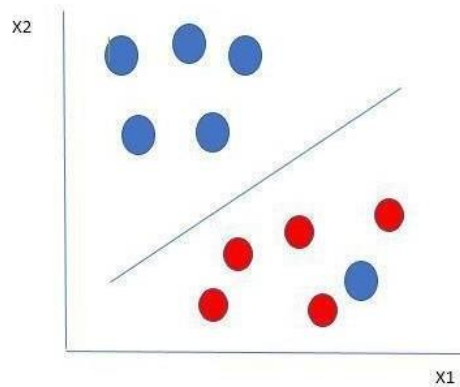
Fig:4.11: Selecting the best hyper-plane

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So from the above figure, we choose L2.

Let's consider a scenario like shown below

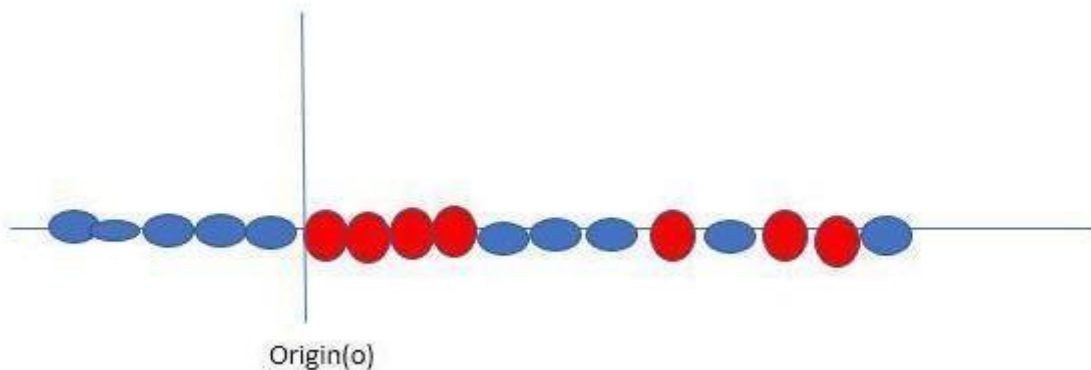


Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It's simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.



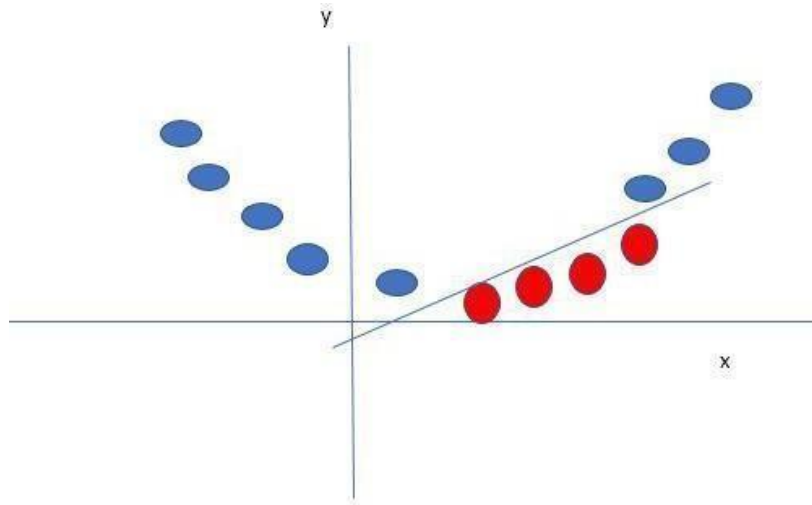
So in this type of data points what SVM does is, it finds maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these type of cases are called soft margin. When there is a soft margin to the data set, the SVM tries to minimize $(1/\text{margin} + \lambda (\sum \text{penalty}))$. Hinge loss is a commonly used penalty. If no violations no hinge loss. If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data (the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is like shown in the figure above. SVM solves this by creating a new variable using a kernel. We call a point x_i on the line and we create a new variable

y_i as a function of distance from origin 0. so if we plot this we get something like as shown below



In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as kernel.

4.2.3.1.2 SVM KERNEL

The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.

4.2.3.1.3 Advantages of SVM

- Effective in high dimensional cases
- Its memory efficient as it uses a subset of training points in the decision function called support vectors
- Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 LEARNING MODEL RESULTS

The learning model of the system is built based on Bag of Features model, which uses an SVM classifier, the values of the parameters of the model are tuned such that the Data split ratio is 60%, Number of visual vocabularies are 2000, Grid step is 10 by 10, Strongest Features Ratio is equal to 90%, SVM kernel is RBF and Box constraint is equal to 2.

The final result of the learning model of the system is successfully classify the images of the skin diseases of the selected classes (Acne, Melanoma & Eczema) with cross validation accuracy of 94% and resultant in 85% for holdout method accuracy.

The confusion matrices of the both methods are shown in Table 5.1 and Table 5.2, which illustrate the resultant accuracy per class.

Table 5.1: Confusion matrix of Cross-Validation method

Predicted \ Actual	Acne	Eczema	Melanoma
Unknown			
Acne	0.94	0.03	0.03
Eczema	0.03	0.91	0.06
Melanoma	0.04	0.00	0.96

Table 5.2: Confusion matrix of the Holdout Method

Predicted \ Actual	Acne	Eczema	Melanoma
Unknown			
Acne	0.80	0.07	0.13
Eczema	0.19	0.75	0.06
Melanoma	0.00	0.00	1.00

In table Table 5.1, we notice that there is almost no correlation between the separate 3 classes, the largest error rate occurred was 0.06 for eczema images that are classified as Melanoma, so it gives a very good accuracy, but we could not rely on the results of cross-validation method because the testing data was taken by a percentage from the dataset so the data used for testing are similar to the training data.

In Table 5.2, we notice that the correlation between classes are increased because the error ratios are increased, which means more images are classified to incorrect labels, but the maximum error ratio is 19% which is for eczema images that are recognized as acne images, despite the increase in the error ratio the overall accuracy of the system is still suitable for the system, also we notice that the error ratio for melanoma classes is zero.

5.2 Data Quantity and Variation Effect on Accuracy

Table 5.1 shows that providing more data for training the model, produces higher accuracies for both Cross-validation and Holdout method, that's is because more features that representing the class of the images are extracted, which increase the generalization of the model, number of 15 images are used in Holdout method.

Table 5.3: Effect of Dataset Quantity on the Accuracy

No. of Images in the dataset	Cross-validation accuracy	Holdout accuracy
50	72%	66%
100	84%	83%

Then more data images were downloaded from different resources, about 200 images of each class were used, these images were processed (converted to gray and powered image of 2 were obtained), results obtained from the model trained by different processed images shown Table5.2.

Table 5.4: Results of preprocessed images

Type of Images in the Dataset	Cross-validation accuracy	Holdout accuracy
Colored images	79%	83%
Gray images	77%	83%
Power of 2 image	85%	40%

Both colored and gray images provide reasonable results, but powered images of 2 provides acceptable results in Cross-validation method, 85%, but the results of the Holdout method were poor, that means using power of 2 images provide bad generalization, it also provides results that are worse than using the original data because the watermark becomes more clear. To increase our data and for better results both colored and gray images were combined together as input for training the Bag of Features model.

5.3 Application Snapshots

- The below Fig 5.1 is initial application snapshot while the project is open.

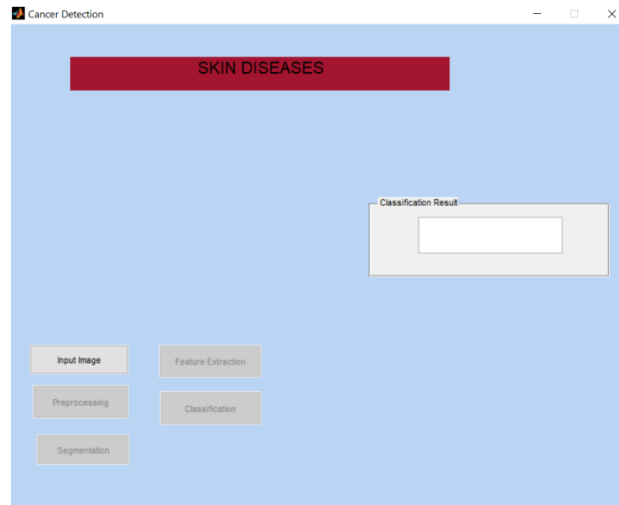


Fig:5.1: Input image

- It will ask the input image for the skin cancer.
- After giving image to the system, it will enable the preprocessing part.

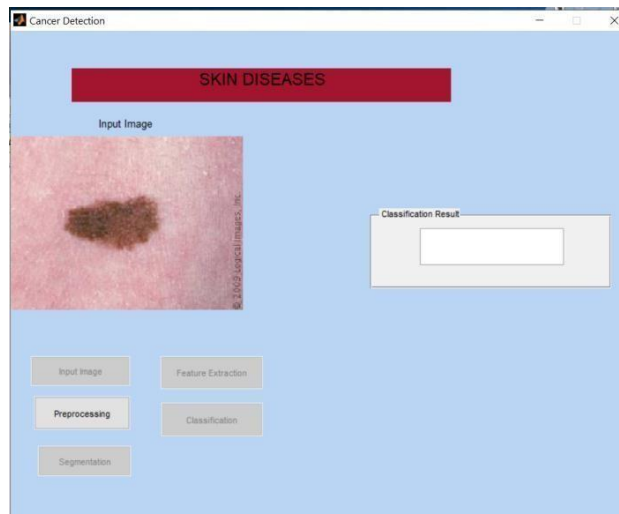


Fig:5.2: inserting image to application

- On clicking on preprocessing button, it will show the preprocessing image.
- After that it will give permission for the segmentation process. By enabling the segmentation button.

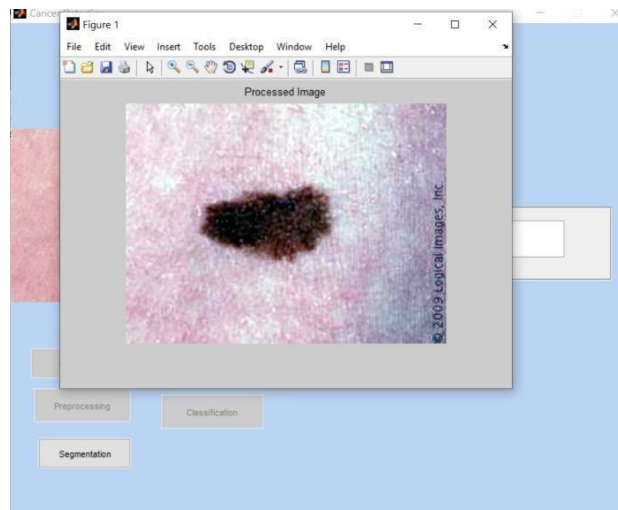


Fig:5.3: Preprocessing

- By clicking of segmentation button, it will show the segmented image in the form of gray scale.
- After that it will enable the feature extraction button.

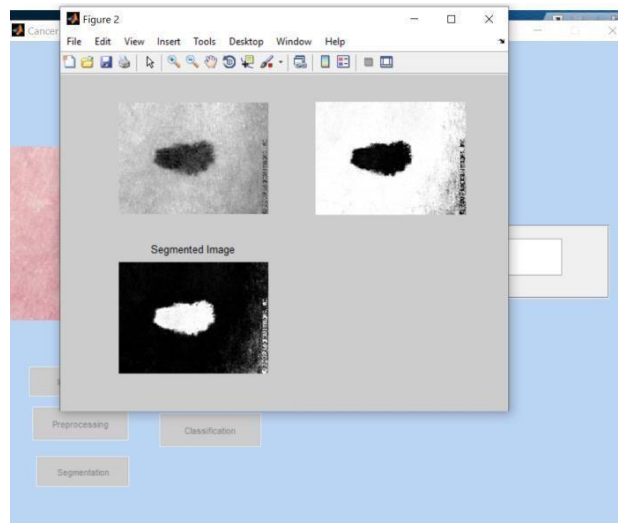


Fig:5.4: segmentation

- By clicking the feature extraction button it will show two images in the form of dots to represented the features of the image.

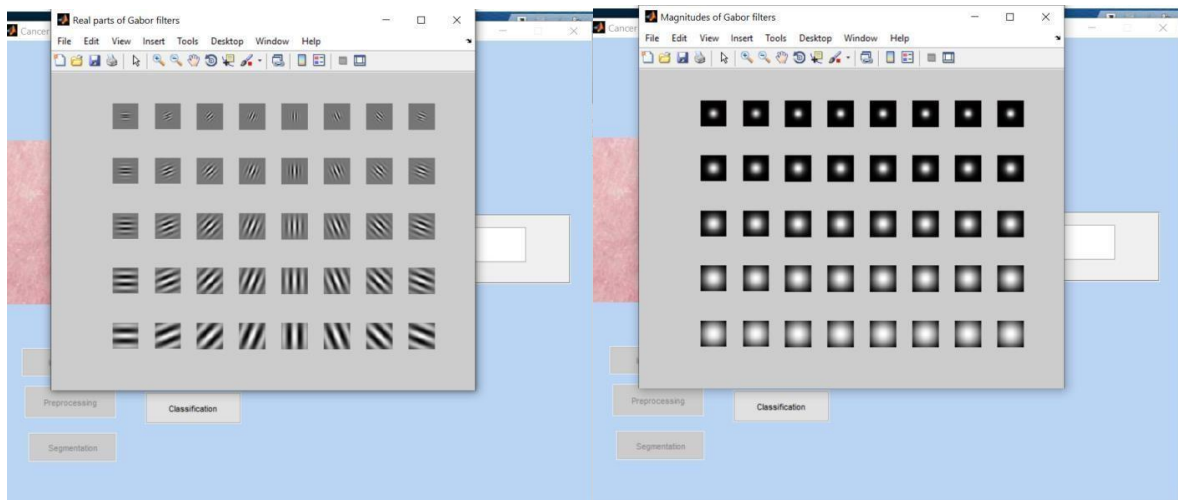


Fig:5.5: feature extraction

- This the final stage, here the classification button is enabling while clicking on it.
- It will show, the final output which tells the type cancer for the inputted image.

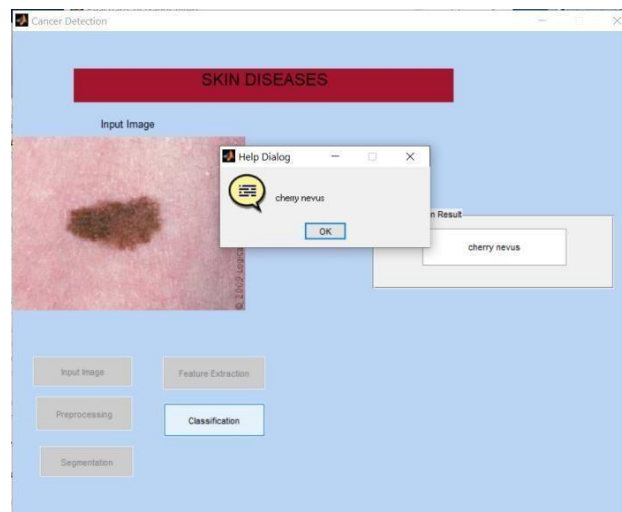


Fig:5.6: classification

CHAPTER 6

CONCLUSION AND FUTUREWORK

6.1 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 SVM as a classifier for feature extraction with an interface on desktop, the core model is developed using MATLAB and the interface is developed using desktop.

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.

6.2 FUTURE WORK

6.2.1 AR Implementation for Skin Disease's Information

- The 3D object model created will be filled with information about these three skin diseases.
- In the design for the 3D model itself, it is done using Blender software, which for information displayed, uses the text tool on Blender, and is converted to Mesh so that it can be displayed on the application.
- Once converted, the 3D model object is exported into the

".glTF" extension, and imported into the Android application.

- To import this extension file, use the tool in ARCore, which is Scene form,

where the file becomes a new extension called ".obj".

- This extension means that the 3D model object that has been imported has turned into an object that can be detected on Android so that it will produce a new file again with the extension ".sfa" where this file is a file that contains information ranging from name, base color, opacity, sources, and materials used to create the 3D object model.
- To use this model, Scene form will compile the file into a new file again with the extension ".sfb". This file is a binary asset Scene form that can be used to be built into an android application and loaded at runtime to create the render able so that when AR is run, the 3D model object can be displayed in the real world.

REFERENCES

- [1] A. A. A. a. H. A. A. M. Hassaballah, Image Features Detection, Description and Matching, Springer International Publishing Switzerland, 2016.
- [2] A. A. Delia-Maria FILIMON, "Skin diseases diagnosis using artificial neural," ResearchGate, 2014
- [3][3] A. S. Suneel Kumar, "Image Processing for Recognition of Skin Diseases," *International Journal of Computer Applications* (0975 – 8887), vol. 149, 2016
- [4] A. S. S. Pravin S. Ambad, "A Image analysis System to Detect Skin Diseases," www.iosrjournals.org, Vadgaon, Pune, India, 2016.
- [5][5] A. P. J. a. A. P. V. A. Dilipsinh Bheda, "A Study on Features Extraction Techniques for Image Mosaicing," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 3, 2014
- [6] C. Tomasi, "pdfs.semanticscholar.org,"[Online]. Available: <https://pdfs.semanticscholar.org/8086/99bacf0cc91a38fa77b6af5a9f91dc25d127.pdf>. [Accessed 10 5 2017].
- [7] E. E. G. A. C. R. J. A. A.A.L.C. Amarathunga, "Expert System For Diagnosis Of Skin Diseases," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, vol. 4, no. 01, 2015.

- [8] F. B. Ledisi G. Kabari, "Diagnosing Skin Diseases Using an Artificial Neural Network," in *ResearchGate*, 2009.
- [9] Google, "Developer.Android," Google, [Online]. Available: <https://developer.android.com/training/volley/index.html>.

- [10] H. Liao, "A Deep Learning Approach to Universal Skin Disease Classification," University of Rochester, 2015+.
- [11] J. R. Edouard Oyallon, "An Analysis of the SURF Method," 2015.
- [12] K. M. S.-E.-A. a. A. R. C. Mir Anamul Hasan, "Human Disease Diagnosis Using a Fuzzy Expert System," *JOURNAL OF COMPUTING*, vol. 2, no. 6, 2010.
- [13] "kom.aau.dk," [Online]. Available: http://kom.aau.dk/group/04gr742/pdf/kmeans_worksheet.pdf.
[Accessed 10 10 2017].
- [14] M. A. R. a. N. A. Rahat Yasir, "Dermatological Disease Detection using Image Processing And Artificial Neural Network," in *8th International Conference on Electrical and Computer Engineering*, 2014.
- [15] M. I. Antkowiak, "Artificial Neural Networks vs. Support Vector Machines for Skin Diseases Recognition," 2006.
- [16] M. K. V. G. Sanjay Jaiswar, "Skin Cancer Detection Using Digital Image," *International Journal of Scientific Engineering and Research (IJSER)*, 2014.
- [17] M. S. I. N. a. N. A. Rahat Yasir, "A Skin Disease Detection System for Financially Unstable People in Developing Countries," *Global Science and Technology Journal*, 2015.
- [18] R. E. Woods, *Digital Image Processing*, Second Edition, 2001.
- [19] S. B. K. . I. D. Z. . P. E. Pintelas, "Machine learning: a review of classification and combining techniques," 2007.

- [20] S. B. Sarika Choudhari, "Artificial Neural Network for SkinCancer," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 3, no. 5, 2014.
- [21] S. S. K. a. V. S. Vinayshekhar Bannihatti Kumar, "Dermatological Disease Detection Using Image Processing And Machine Learning," IEEE, 2016.
- [22] "skincancer.org," 2017. [Online].Available: <http://www.skincancer.org/skin-cancer-information/melanoma>. [Accessed 2 October 2017].
- [23] T. N. Phyu, "Survey of Classification Techniques in Data Mining," in *International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2009.
- [24] "World life Expectancy,"[Online]. Available: <http://www.worldlifeexpectancy.com/sudan-skin-disease>.

APPENDIX

A. Source Code

```
function varargout = main(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @main_OpeningFcn,
    ... 'gui_OutputFcn', @main_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
function main_OpeningFcn(hObject, eventdata, handles, varargin)
guidata(hObject, handles);
axes(handles.axes1); axis off
function varargout = main_OutputFcn(hObject, eventdata, handles)
function pushbutton1_Callback(hObject, eventdata, handles)
[filename, pathname, filterindex]=uigetfile( ...
    {'*.jpg','JPEG File (*.jpg)'; ...
```

.', 'Any Image file (*.*)'}, ...

'Pick an image file');

```

var=strcat(pathname,filename);
    k=imread(var);
    handles.YY = k;
    set(handles.edit1,'String',var);

z=k;
guidata(hObject,handles);
axes(handles.axes1);
    imshow(k);
    title('Input Image');
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton16,'enable','on');
function pushbutton2_Callback(hObject, eventdata, handles)
k=handles.YY;
j=rgb2gray(k);
handles.XX=j;
figure,subplot(2,2,2),imshow(j);title('gray Image');
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','off');

set(handles.pushbutton4,'enable','on');

function pushbutton3_Callback(hObject, eventdata, handles)
str=get(handles.edit1,'String');
I = imread(str);
h =
ones(5,5)/25; I2
= imfilter(I,h);
figure
imshow(I2)
title('Filtered Image')

```

```
set(handles.pushbutton3,'enable','off');
```

```

set(handles.pushbutton12,'enable','on');
function pushbutton4_Callback(hObject, eventdata, handles)
ei=25;
st=35;
k=ei*st;
I=handles.YY
;
h = ones(ei,st) / k;
I1 = imfilter(I,h,'symmetric');
IG=rgb2gray(I1);
I11 = imadjust(IG,stretchlim(IG),[]);
level = graythresh(I11);
BWJ = im2bw(I11,level);
dim = size(BWJ)
IN=ones(dim(1),dim(2));
BW=xor(BWJ,IN); %inverting
figure,subplot(2,2,2), imshow(BW), title('Black and White');
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','off');
set(handles.pushbutton3,'enable','on');
set(handles.pushbutton4,'enable','off');
function pushbutton6_Callback(hObject, eventdata, handles)
I=handles.YY;
[y,x,z]=size(I);
myI=double(I)
;
H=zeros(y,x);
S=H;
HS_I=H;
for i=1:x
    for j=1:y

```

```
HS_l(j,i)=((myl(j,i,1)+myl(j,i,2)+myl(j,i,3))/3);  
S(j,i)=1-3*min(myl(j,i,:))/(myl(j,i,1)+myl(j,i,2)+myl(j,i,3));
```

```

        if ((myl(j,i,1)==myl(j,i,2))&(myl(j,i,2)==myl(j,i,3)))
            Hdegree=0;
        else
            Hdegree=acos(0.5*(2*myl(j,i,1)-myl(j,i,2)-myl(j,i,3))/((myl(j,i,1)-
myl(j,i,2))^2+(myl(j,i,1)-myl(j,i,3))*(myl(j,i,2)-myl(j,i,3)))^0.5);
        end
        if (myl(j,i,2)>=myl(j,i,3))
            H(j,i)=Hdegree;
        else
            H(j,i)=(2*pi-Hdegree);
        end
    end
end
end

```

```

Hth1=0.9*2*pi; Hth2=0.1*2*pi;
Nred=0;
for i=1:x
    for j=1:y
        if ((H(j,i)>=Hth1)|| (H(j,i)<=Hth2))
            Nred=Nred+1;
        end
    end
end
end

```

```

Ratio=Nred/(x*y);

```

```

if (Ratio>=0.6)
    Red=1
else

```

```
    Red=0  
end
```

```
HS_I=uint8(HS_I);
```

```
figure(1);  
imshow(I);  
figure(2);  
imshow(HS_I);
```

```
str=get(handles.edit1,'String');  
I1 = imread(str);  
I = im2double(I1);  
HSV = rgb2hsv(I);  
H = HSV(:,:,1); H = H(:);  
S = HSV(:,:,2); S = S(:);  
V = HSV(:,:,3); V = V(:);  
idx = kmeans([H S V], 4);  
imshow(I1);  
figure,imshow(ind2rgb(reshape(idx, size(I,1), size(I, 2)), [0 0 1; 0 0.8  
0])) if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function [ Unow, center, now_obj_fcn ] = FCMforImage( img, clusterNum )
```

```
if nargin < 2
```

```
    clusterNum = 2; % number of  
cluster end
```

```
[row, col] = size(img);  
expoNum = 2; % fuzzification  
parameter epsilon = 0.001; % stopping  
condition  
mat_iter = 100; % number of maximum iteration
```

```
Upre = rand(row, col, clusterNum);  
dep_sum = sum(Upre, 3);  
dep_sum = repmat(dep_sum, [1,1, clusterNum]);  
Upre = Upre./dep_sum;
```

```
center = zeros(clusterNum,1);
```

```
for i=1:clusterNum  
    center(i,1) = sum(sum(Upre(:, :, i).*img))/sum(sum(Upre(:, :, i)));  
end
```

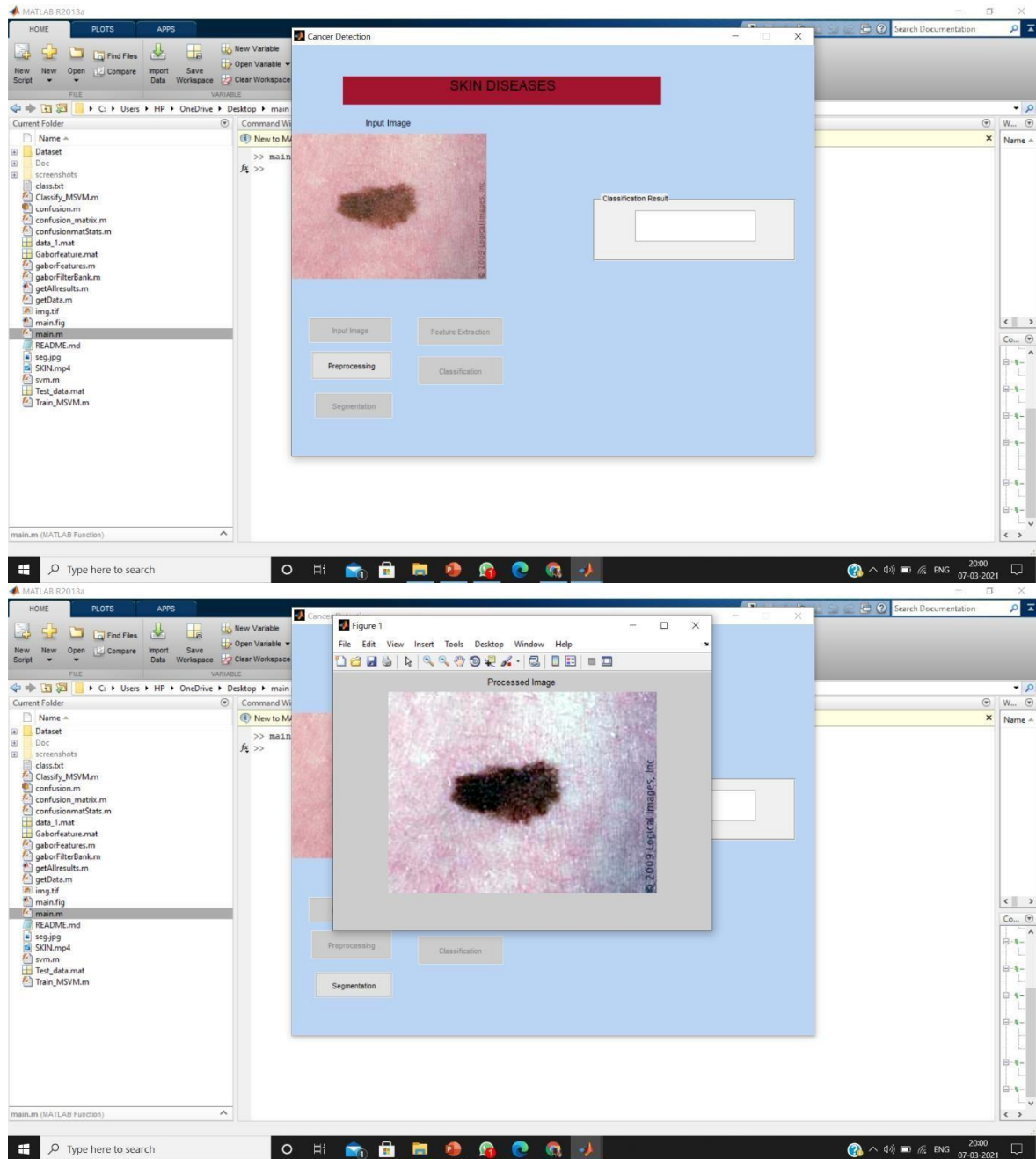
```
pre_obj_fcn = 0;  
for i=1:clusterNum  
    pre_obj_fcn = pre_obj_fcn + sum(sum((Upre(:, :, i) .*img - center(i)).^2));  
end  
fprintf('Initial objective fcn = %f\n', pre_obj_fcn);
```

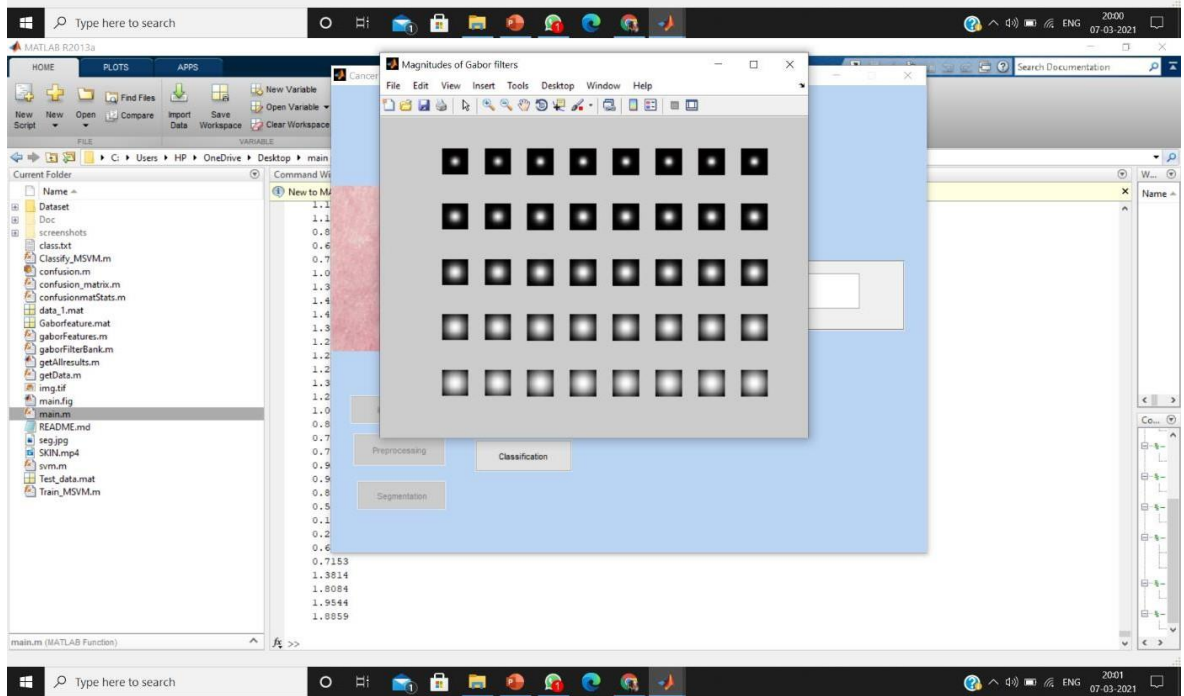
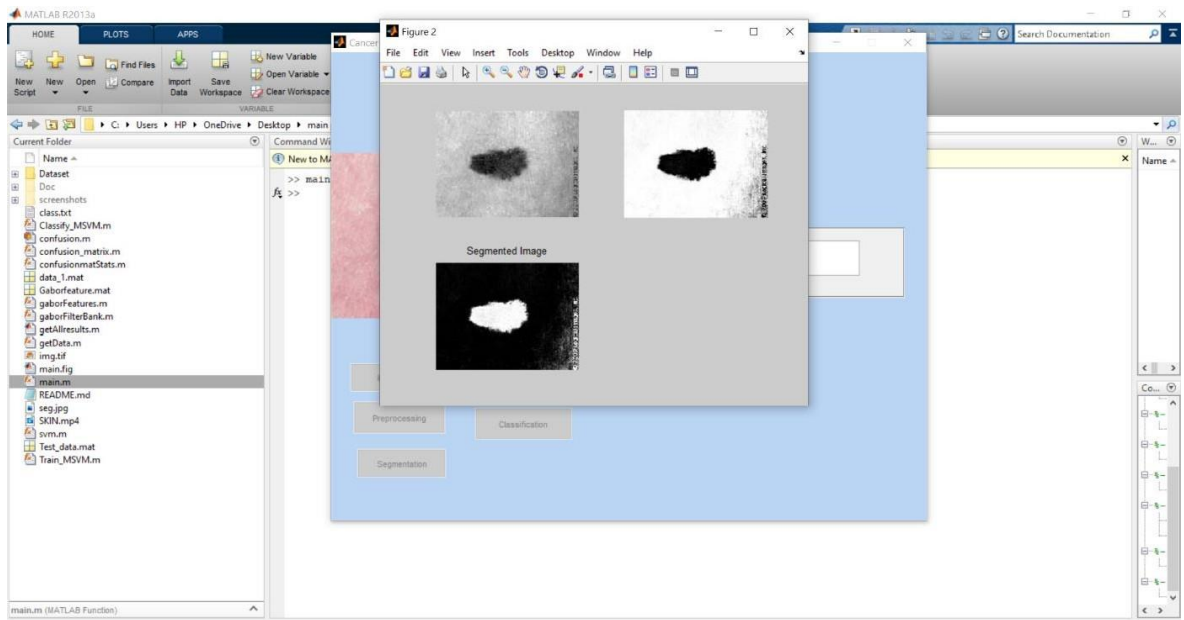
```
for iter = 1:mat_iter
```

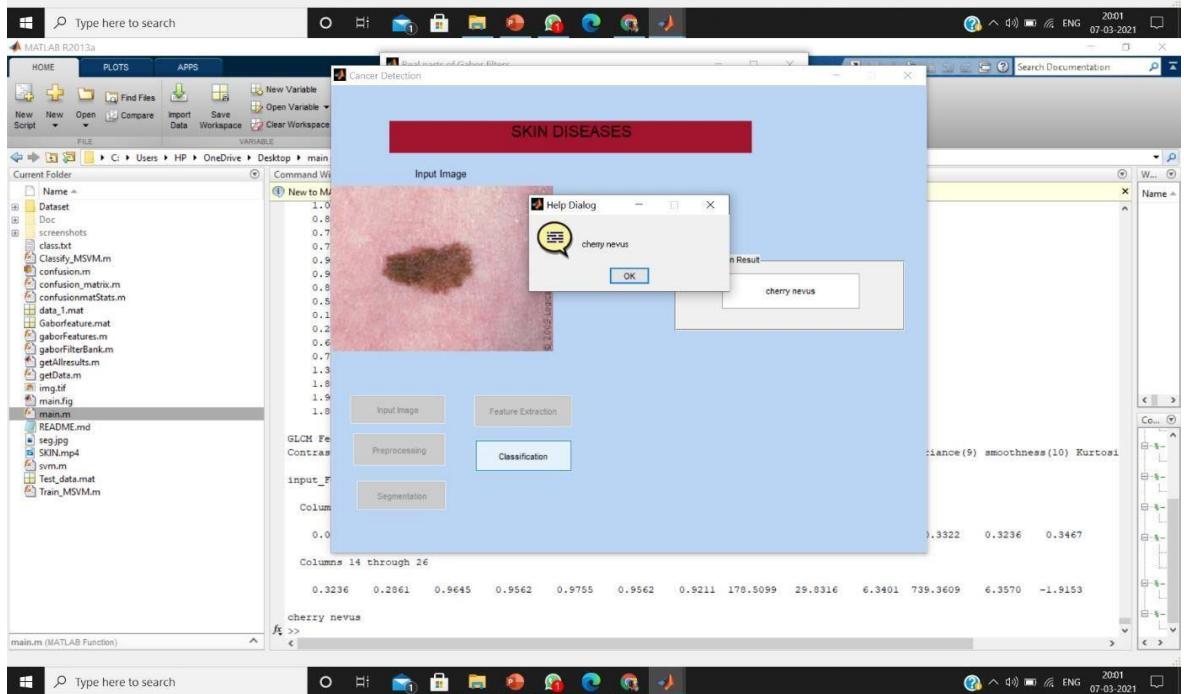
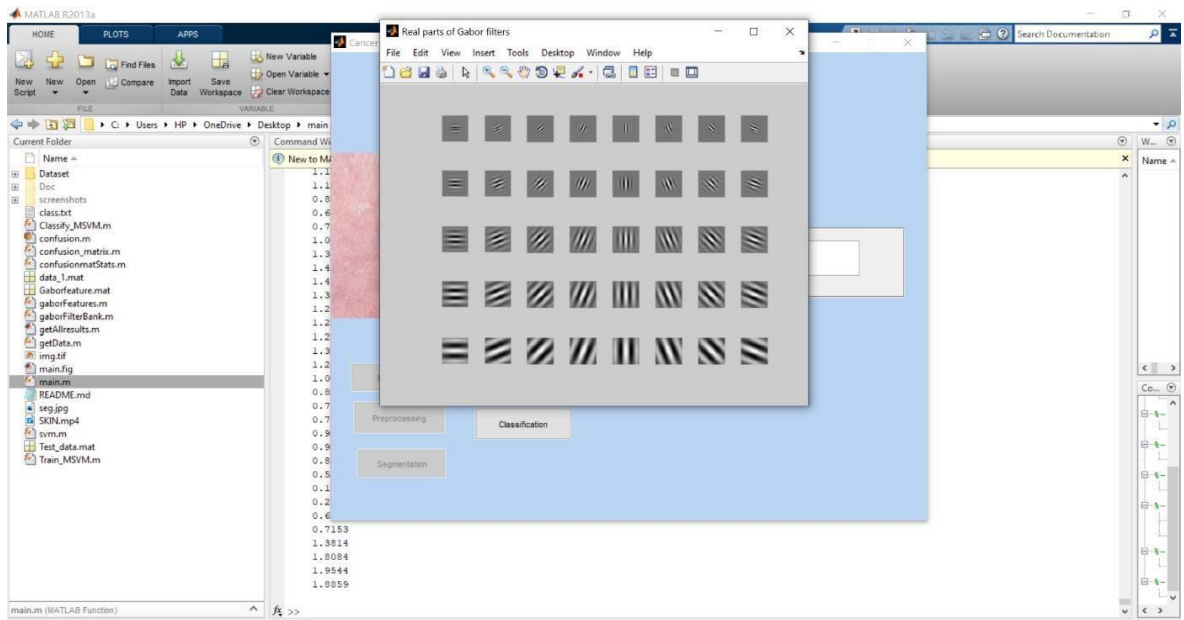
```
    Unow = zeros(size(Upre));  
    for i=1:row
```

```
for j=1:col
```

B.SCREEN SHOTS







B. Journal Publication

DETECTION OF SKIN CANCER BASED ON SVM

Dr. Usha Nandhini
Faculty of Computing, Sathyabama University, Chennai, India-600119
usha.cse@sathyabama.ac.in

Authors Batchu Ashik (Author)
Sathyabama institute of science and technology
UG student School of computing, 2017-2021
Chennai, India- 600119
ashik7787@gmail.com

Authors Dileep Banda (Author)
Sathyabama institute of science and technology
UG student School of computing, 2017-2021
Chennai, India- 600119
dileep.banda84@gmail.com

Abstract— In this paper we discussed about on our project which was implemented for skin cancer detection. There are so many skin cancers are there we mainly focus on 3 diseases that are Acne, Eczema and Melanoma. For the detection of skin cancer, we divide our project into 3 modules. Initially skin picture were preprocessing, and then the preprocessed image was taken and goes to segmentation. Where in segmentation the image was converted into gray scale image for easy to extract the properties. And finally, with the help of the gray scale image we define the which type of cancer by using SVM (super vector machine) classifier. (Abstract)

Keywords—SVM; Matlab; Data Set (key words)

I. INTRODUCTION (HEADING 1)

The Skin is the largest organ in human body. There are nearly 1500 types of skin diseases. Normally we are ignoring this type of diseases. But there are some dangerous skin cancers which leads to death. For the safety concern we make an automated software to detect the type of skin cancer.

Due to very less in doctor ratios as per the patients. There will be formed a gap between the chain. To reduce this space for medical concern we develop this system.

Now a days there is drastic change of technology everyone was upgraded as per technology to going fast procedure. On the hands every area has systems or laptops for using those equipment's we easily find the type of cancer.

II. PROPOSED SYSTEM (HEADING 2)

The input was taken from the user. Some of the images there will be a noise on it. So that we used preprocessed the image by removing the unwanted data and then highlighted the portion of the effected skin. By using of preprocessed image,

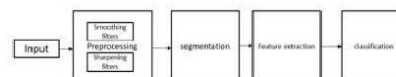
it converts into gray scale image which help to easily for classification. The gray scale image was classified by using SVM with the help of dataset. And finally, it shows the output from the system.

III. DATA SET

Generally, SVM is machine learning algorithm, for that we need the dataset for classification. The main part of this project is data set, without data set we didn't find the type of cancer. There is 2 ways of data collection one is gather the images manually from hospitals, individual patients or else from data centers for medical purpose. For manual collection we are facing problems like taking permission from higher authorities or else there is lack of knowledge to collect the data from each patient and also it is most sensitive data for the patients. So, we decided to take data from the net resources.

For the data set there are more resources which was available from the internet. We take the data from Kaggle and GitHub where it contains so many resources for medical filed. We collected nearly 1800 images for this project. Those images are divided into two sets train and test. For the good approach we allocated the train folder with 70% of the images. Remaining 30% of the images are consider as test images.

IV. SYSTEM ARCHITECTURE



Above diagram is our system architecture, where it contains different steps for implementing the project.

V. MODULES

The modules that are used in our project are,

- Preprocessing
- Segmentation
- Feature extraction
- Classification

• Preprocessing:

Taking input image from the user, and then the image was preprocessing. Here we use gaussian filter algorithm for preprocessing. The algorithm is worked on Using a **Gaussian Blur filter** before edge detection aims to reduce the level of noise in the image, which improves the accuracy of the classification there are two methods which was done in this stage.

One is Smoothing filters are primarily used for reduce noise, it will help to remove unwanted details from the image, reminiscent of hair, salt and pepper noise, air bubbles and background.

And other is Sharpening filters are wont to highlight fine details reminiscent of the perimeters or to reinforce details that are blurred within the image, additionally sharpening filters are applied in each special and frequency domain.



Fig: preprocessing

• Segmentation:

After preprocessing the out image was goes to segmentation process. For segmentation, we tend to are mistreatment the Dennis Gabor filter. Features are extracted directly from gray-scale character pictures by Gabor filters that are specially designed from applied math info of character structures. A sigmoid perform is applied to the image in Gabor filters to attain higher performance on low-quality images.

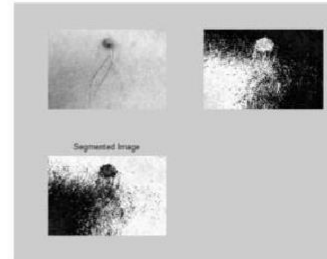


Fig: segmentation

• Feature extraction:

The gray image that was taken from the segmentation, that was more goes through the feature extraction. For the feature extraction we've such a large number of algorithms like Corner Detection and SVM, CNN. By mistreatment SVM formula we have a tendency to convert the image information in to traditional matrix.

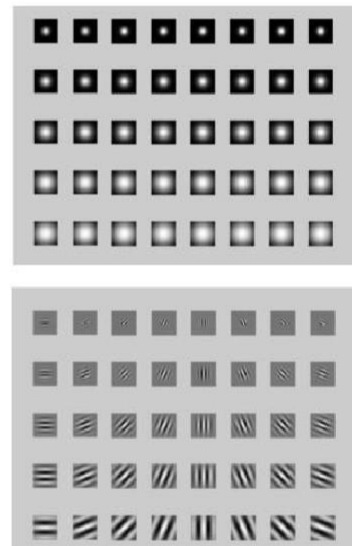


Fig: Feature extraction

• Classification:

In typical matrix we have a tendency to taken the values of Contrast, Correlation, Energy, Homogeneity, Mean, customary Deviation, Entropy, RMS, Variance, smoothness, Kurtosis, Skewness, IDM. With the assistance of the standard matrix, we

have a tendency to generate the accuracy of the output. With the assistance of output, we have a tendency to compare without knowledge set values. After that, it'll show which kind of disease.

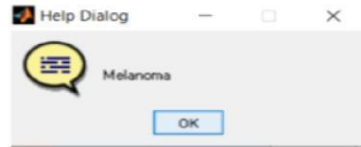


Fig: final output

VI. RESULT AND DISCUSSION

The learning model of the system is built based on an SVM classifier. The values of the parameters of the model are tuned such that the Data split ratio is 60%, Number of visual vocabularies are 2000, Grid step is 10 by 10, Strongest Features Ratio is equal to 90%, SVM kernel is RBF and Box constraint is equal to 2. The final result of the learning model of the system is successfully classify the images of the skin cancer of the selected classes Acne, Melanoma & Eczema with cross validation accuracy of 90%.

VII. CONCLUSION

Difficulties to determine the skin diseases arise because of the spreading of the skin diseases all over the world, which makes us challenge to the dermatologist to recognize the different skin diseases easily. A computer-based system is proposed to resolve these difficulties, so a machine learning model such as SVM, it is a classifier for feature extraction with an interface on desktop, the core model is developed using MATLAB and the interface is developed using desktop.

VIII. FEATURE WORK

AR implementation for Skin Disease's Information
The 3D object model created will be filled with information about these three skin diseases. In the design for the 3D model itself, it is done using Blender software, which for information displayed, uses the text tool on Blender, and is converted to Mesh with the help it can be displayed on the application. Once converted, the 3D model object is exported into the ".glTF" extension, and imported into the Android application. To import this extension file, use the tool in ARCore, which is Scene form, where the file becomes a new extension called ".obj". This extension means that the 3D model object that has been imported has turned into an object that can be detected on Android so that it will produce a new file again with the extension ".sfa" file, where it contains information

1 ranging from name, base color, opacity, sources, and materials used to create the 3D object model. To use this model, Scene form will compile the file into a new file again with the extension ".sfb". This file is a binary asset Scene form that can be used to be built into an Android application and loaded at runtime to create the renderable so that when AR is run, the 3D model object can be displayed in the real world.

IX. REFERENCES

- [1] "World life Expectancy," [Online]. Available: <http://www.worldlifeexpectancy.com/sudan-skin-disease>.
- [2] <https://www.kaggle.com/fanconic/cnn-for-skin-cancer-detection>
- [3] <https://github.com/0x5eba/Skin-Cancer-Segmentation>
- [4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", Google, 2019.
- [5] Mohd Azlan Abu, Nurul Hazirah Indra, Abdul Halim Abd Rahman, Nor Amalia Sapiee and Izzanordina Ahmad, "A study on Image Classification based on Deep Learning and Tensorflow", Universiti Kuala Lumpur British Malaysian Institute, Malaysia, Vol. 12, No 4, pp. 563-569, 2019.

C. Plagiarism report

DETECTION OF SKIN CANCER BASED ON SVM ALGORITHM

ORIGINALITY REPORT

9%

SIMILARITY INDEX

0%

INTERNET SOURCES

9%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

Mizanuddin Auliya Iqbal, Akuwan Saleh,

8%

Haryadi Amran Darwito. "Implementation of The Introduction of Skin Diseases Based on Augmented Reality", 2020 International Electronics Symposium (IES), 2020

Publication

Shihab A. Hameed, Alaa Haddad, Mohamed Hadi habaebi, Ali Nirabi. "Dermatological diagnosis by mobile application", Bulletin of Electrical Engineering and Informatics, 2019

1%

Publication

Exclude quotes On

Exclude matches Off

Exclude bibliography On

