REMOTE SKIN DISEASES DIAGNOSIS SYSTEM USING MACHINE LEARNING TECHNIQUES

BY

RANA MOHAMMED ALHADI ABD ALKAREEM ALSHEIKH EDREES

INDEX NO. 124050

Supervisor

Prof. Shareef Fadul Babikir

A REPORT SUBMITTED TO

University of Khartoum

In partial fulfillment of the requirements for degree of

B. Sc (HONS) Electrical and Electronic Engineering

(SOFTWARE ENGINEERING)

Faculty of Engineering

Department of Department of Electrical and Electronic Engineering

October 2017

DECLARATION OF ORGINALITY

I declare this report entitled "**REMOTE SKIN DISEASES DIAGNOSIS SYSTEM USING MACHINE LEARNING TECHNIQUES**" *is my own work except as cited in references. The report has been not accepted for any degree and it is not being submitted currently in candidature for any degree or other reward.*

Signature: ____________________

Name: _______________________

Date: ________________________

ACKNOWLEDGMENT

I would like to express my deep gratitude and appreciation to my supervisor, **Prof. Shareef Fadul Babikir**, who has been extremely supportive and quite encouraging throughout the different phases of this project. His guidance and support have always been a motivation for me to work hard.

Also, special thanks to **Mr. Marwan Awad**, for his excellent support and contribution in the project.

A deeper gratitude to my project partner, **Abubaker Tagelsir,** for his unlimited support and patience throughout the project phases.

DEDICATION

To My Parents

To My Brothers & Sisters

To My Lovely Grandfather

To my Uncle

To My Friends

ABSTRACT

Skin diseases have become one of the most common diseases all over the world, beside their painful effects they are spreading very fast to cover a larger area and also have a psychological effect to the patients, the diagnosis of the skin diseases requires a high level of expertise and they are subjective to the dermatologist, so computer aided skin diseases diagnosis system is proposed to provide more objective and reliable solution to this problem.

This project aims to develop skin diseases diagnosis system with a mobile interface, the system is built on a machine learning model to classify the infected images using Bag of Features Model with SVM classifier and develop an ANDROID interface application to capture the images, the designed model has successfully able to classify the infected images of 3 sample classes with accuracy 94% of cross-validation method and 85% of holdout method.

The system is built successfully and the interface application is communicating properly with the server established, and all the system functionalities is working properly, despite that there are some problems occurred through the development of the system starting with the data collected that are distorted by a watermark that obstacle the classification process and the synchronization between the server and the client sides of the system,

The system developed is using a single server, which by the increase of the number of the users will face a performance issue, and also the system interface is available for the android users only, these are the proposed future improvements of the system.

المستخلص

الامراض الجلدية أصبحت واحدة من أكثر الأمراض انتشارا في العالم, اضافة للألم الذي تسببه فهي تنتشر بسرعة لتغطي مساحة أوسع من الجلد, علاوة على ذلك لديها تأثيرات نفسية على المريض.

عملية تشخيص الأمراض الجلدية تتطلب مستوى عال من الخبرة و المعرفة و هي تعتمد على وجهة نظر الطبيب لحظة التشخيص, اذا تشخيص الأمراض الجلدية بمساعدة الحاسوب اقترح ليعطي نظام تشخيص اكثر موضوعية و موثوقيه.

هذا المشروع يهدف لتطوير نظام تشخيص للأمراض الجلدية مع واجهة مستخدم للهاتف النقال, وهو مبني على طريقة تستخدم التعلم التلقائي للالة لتصنيف المرض المعني في الصورة باستخدام نموذج حقيبة الخصائص مع المصنف داعم المتجهات بالاضافة الى تطوير برنامج للهواتف النقالة لتسهيل استخدام النظام و التقاط صور لمنطقه الجلد المصابه.

هذا النظام المطور قد نجح في تصنيف ثلاثة من الامراض المقترحة بدقة بلغت 94% عند استخدام حزء من البيانات في التدريب والجزء الاخر في التقييم, ايضا بلغت دقه النموذج 85% عند استخدام صورمن مصادر اخرى خارجية.

تم بناء النظام بنجاح, عمليه الاتصال بين العميل والمخدم اكتملت ايضا بنجاح, جميع الوظائف المتوقعه من النظام تعمل بكفاءة رغم عن كل المشاكل التي حدثت اثناء بناء النظام مثل عدم توفر بيانات ملائمة بصورة صحيحة و مشكلة التزامن في العمل بين اطراف النظام.

النظام في الوقت الحالي يعمل بكفاءة لكن باستخدام مخدم واحد فقط, و مع زيادة عدد المستخدمين سيتسبب ذلك في مشاكل في أداء النظام, كما انه يعمل في الاجهزة الداعمة لنظام تشغيل أندرويد فقط. و هذه المشاكل مقترحة لتحل في التطويرات المستقبلية للنظام.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Overview

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 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 [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.

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.

1.3 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.
    - Be applicable locally in Sudan.
2. Develop a mobile interface that:
    - Capture the image, send it to the server and receiving the results back.

1.4 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.

On the other hand, smartphones have become one of our lives basis, they could be used in many ways, Skin Diseases Diagnosis System built-in smartphones would be very accurate by implementing machine learning and artificial intelligence techniques in addition its available, reliable and very easy to use.

1.5 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 drmnet.com, medicine.uiowa.edu, dermnetnz.org and others; we choose to classify between 3 famous Diseases which are Acne, Eczema and Melanoma.

Bag-of-features model using Support Vector Machine as a classifier is used to train our data, which is very powerful in object and texture recognition.

An android application is used as interface to the user, this application communicates with MATLAB application through apache server, the user simply captures the image, a pre-trained classifier in MATLAB is used to define the Skin Disease of this image, then the results sent back to the android application. The

Figure 3.3.1.1-1 shows the overall overview of the system.

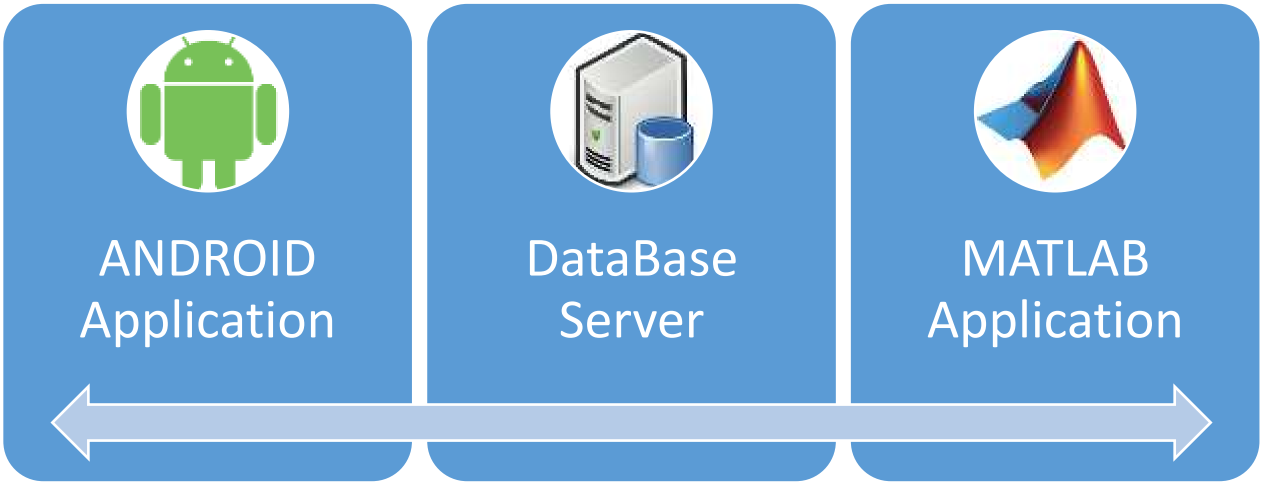


Figure 3.3.1.1-1System Overview

1.6 Thesis Layout

This thesis is composed of five chapters that are organized as follows:

Chapter 2 (LITERATURE REVIEW): this chapter presents some related theories, also an overview of previous skin diseases diagnosis systems and the technologies implemented in these systems are discussed.

Chapter 3 (METHODOLOGY): this chapter presents a detailed look at the design process of the learning model of the system, and the development processes of the applications used in the system, and explains the functionalities of each application and the problems during the development process.

Chapter 4 (RESULTS AND DISCUSSION): this chapter presents the results of the final learning model of the system, and the overall integrated system performance.

Chapter 5 (CONCLUSION): the final conclusion of the project is represented in this chapter, illustrating the project objective, how far they were met, features and limitation of the application and the future work that can be implemented to improve the system.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

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.

This chapter introduce you with some of the techniques that could be implemented to develop Skin Diseases Diagnosis System based on machine learning approach, in addition provide and discuss the previous attempts to develop Skin Diseases Diagnosis Systems using machine learning or any similar techniques.

2.2 Dataset

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.

2.3 Preprocessing and Image Enhancement

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 form 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.

2.3.1 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.

Many filters could be applied for smoothening the image, in both spatial domain and frequency domain of the image. The most commonly used smoothing filter in the spatial domain is the Median Filter, another filter that could be used are Arithmetic Mean filter, Contra-harmonic Mean filter, Alpha-trimmed Mean filter and Adaptive Median filter.

In the frequency domain, Gaussian filter is commonly used for smoothing because it has no ringing artifacts which are not acceptable in the medical fields.

2.3.2 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.

Sobel operators and the Laplacian filter are both examples of sharpening filters in the spatial domain, in the frequency domain sharpening can be achieved using high-pass filters which attenuate low frequency components without affecting high frequency components of the image –such as edges, High-pass Gaussian Filters are commonly used in sharpening in the frequency domain.

2.3.3 Histogram processing

Histograms show the distribution of gray levels of the image, they are the basis for many spatial operation, they are massively used in segmentation.

The most commonly used process is histogram equalization simply used to equalize frequencies in order to improve dark or washed-out images, another example are histogram matching and local enhancement.

2.4 Feature Extraction

Feature extraction is the process of detecting a certain features of interest within an image in order to use it for further processing, generally, this is a critical step in the image processing solutions because it marks the transition between pictorial to non-pictorial data representation, and the resulting outcome of the feature extraction process can be subsequently used as an input to a further pattern recognition or classification technique in order to label, classify or recognize the content of the input image [4].

The process of finding the corresponding correlation between two images of the same scene acquired by the same or different sensors, at different time or from different viewpoint is called image registration, there are several steps involve in the registration process:

- Feature Detection: Salient and distinctive objects in both reference and sensed images are detected.

- Feature Matching: The correspondence between the features in the reference and the sensed image established.
- Transform Model Estimation: The type and parameters of the so-called mapping functions, aligning the sensed image with the reference image, are estimated.
- Image Resampling: The sensed image is transformed by means of the mapping functions.

Using the image features there are two ways to identify the matching regions from the input images, block matching and feature-point matching, Block matching algorithms calculate the correlation between regular-sized blocks generated in sequential images. Such methods include either Normalized Cross-Correlation or phase correlation using a Fast Fourier Transform, these methods involve a series of complex calculations and they are very sensitive to the slight distinction between images, in general, feature based methods extract distinctive features from each image and matches these features to establish a global correspondence. Here are some of the most common feature extraction techniques [5]in image processing:

Harris Corner Detection

Harris corner detector is a well-known interest key point detector due to its invariance to rotation, illuminations variation and image noise. It is based on local auto-correlation function of a signal where local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions.

FAST

Presented by Trajkovic and Hedley to detect corners, the detection of corner was prioritized over edges in FAST as corners were found to be the good features to be matched, because it shows a two dimensional intensity change, and thus well distinguished from the neighboring points. FAST incremented the computational speed required in the detection of corners, also it is an accurate and fast algorithm that yields good localization (positional accuracy) and high point reliability.

SURF

SURF algorithm is a fast and robust algorithm for local, similarity invariant representation and comparison of images, that interest points of a given image are defined as salient features from a scale-invariant representation [6], it can be used for tasks such as object recognition, image registration, classification or 3D reconstruction.

SIFT

The Harris operator is not invariant to scale and correlation is not invariant to rotation, for better image matching, SIFT algorithm was developed by David Lowe in 1999. Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters. Scale Invariant Feature Transform (SIFT) consists of four stages as below:

- Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.
- Key-point localization: At each candidate location, a detailed model is fit to determine location and scale. Key-points are selected based on measures of their stability.
- Orientation assignment: One or more orientations are assigned to each key-point location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
- Key-point descriptor: The local image gradients are measured at the selected scale in the region around each Key-point. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination [5].

HOG

Local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. In practice this is implemented by dividing the image window into small spatial regions (“cells”), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram “energy” over somewhat larger spatial regions (“blocks”) and using the results to normalize all of the cells in the block. We will refer to the normalized descriptor blocks as Histogram of Oriented Gradient (HOG) descriptors [7].

2.5 Classification

Data mining is the process of extracting patterns from data, its referred to as knowledge discovery from databases, classification techniques are widely used in data mining to classify data among various classes [8], they are considered as a type of supervised learning when both input features and the target output are represented during the learning process, this technique typically used to predict group membership for data instances.

In Skin diseases diagnosis systems, although the features that extracted from the image represent the image in a simpler way, but it will be useful to implement the classification techniques in the prediction of the specific disease (predicted output) which has specific group of features (inputs), here some of the classification techniques will be introduced.

2.5.1 Decision trees

Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch presents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values [9], Advantages of Decision trees that they are simple to understand and interpret, require little data and are able to handle both numerical and categorical data [8]. It is possible to validate a model using statistical tests. They are robust in nature; they perform well even if its assumptions are somewhat violated by the true model from which the data were generated they perform well with large data in a short time. But practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node, which doesn’t guarantee to find the optimal solution, the search will stop when finding a solution regardless of whether it's the optimal solution or not, this reduce the generalization of the learning process. Figure 3.3.1.1-1 shows decision tree sample.

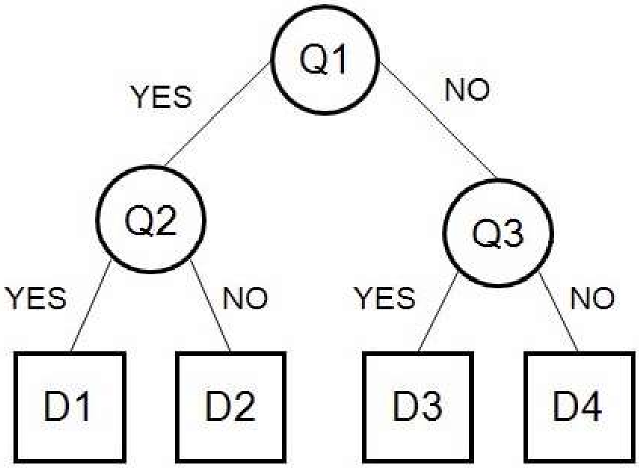


Figure 3.3.1.1-1 Decision tree sample

2.5.2 Support vector machines

Support vector machine were first introduced to solve classification and regression problems by Vapnik and his colleagues, viewing input data as two sets of vectors in an n-dimensional space, an SVM will construct a separating hyper-plane in that space, one which maximizes the margin between the two data sets [8].

To calculate the margin, two parallel hyper-planes are constructed, one on each side of the separating hyper-plane, which are "pushed up against" the two data sets. A good separation is achieved by the hyper-plane that has the largest distance to the neighboring data points of both classes, since in general the larger the margin the lower the generalization error of the classifier, this hyper-plane is found by using the support-vectors and margins [8]. Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error [10].

Nevertheless, most real-world problems involve non-separable data for which no hyperplane exists that successfully separates the positive from negative instances in the training set. One solution to the inseparability problem is to map the data onto a higher-dimensional space and define a separating hyperplane there. This higher-dimensional space is called the feature space, as opposed to the input space occupied by the training instances [10].

Finally, the training optimization problem of the SVM necessarily reaches a global minimum, and avoids ending in a local minimum, which may happen in other search algorithms such as neural networks. However, the SVM methods are binary, thus in the case of multiclass problem one must reduce the problem to a set of multiple binary classification problems. Discrete data presents another problem, although with suitable rescaling good results can be obtained.

2.5.3 Artificial neural networks

Artificial Neural Networks are complex data processing model that tries to mimic the way that a human brain functions, its main objective is to find a function that maps given inputs to desired output, they use a huge interconnection of simple processing units called neurons, these units receives inputs from previous unit to be processed then sends the processed output to other neurons. These units are connected through synaptic weights which determine how strong these neurons are affected by each other. Usually ANNs constructed in layers, input layer receives input from the environment, hidden layer(s) and output layer. Generally, properly determining the size of the hidden layer is a problem, because an underestimate of the number of neurons can lead to poor approximation and generalization capabilities, while excessive nodes can result in overfitting and eventually make the search for the global optimum more difficult [10]. Synaptic weights between the connected neurons are updated using different learning algorithms such as Hebbian learning, Competitive learning and Gradient Descent rule either online (after each input case), or offline (after all the data to be processed by the network). The implemented neurons could be linear neurons, sigmoid neurons, binary threshold neurons or even radial basis function neurons.

ANNs’ effectiveness in recognizing patterns and relations is a reason why they are being used to aid doctors in solving medical problems. Until 1996 they have not been used in this area frequently; they have been generally applied in radiology, urology, laboratory medicine and cardiology, Recently ANNs have proven to be useful in many other fields of medicine including dermatology. They have shown large efficiency not only in diagnosis but also in modelling parts of the human body [2].

2.5.4 K-nearest neighbor classifiers

Nearest neighbor classifiers are based on learning by analogy, each sample represents a point in an n-dimensional space. In this way, all of the training samples are stored in an n-dimensional pattern space. When given an unknown sample, a k-nearest neighbor classifier searches the pattern space for the k training samples that are closest to the unknown sample. "Closeness" is defined in terms of Euclidean distance or other distance measures, such as the Manhanttan distance could be used [9]. The unknown sample is assigned the most common class among its k nearest neighbors, nearest neighbor classifiers are instance-based or lazy learners in that they store all of the training samples and do not build a classifier until a new(unlabeled) sample needs to be classified. An expected lazy learning methods are faster data training than eager methods, but slower at classification since all computation is delayed to that time, Unlike decision tree induction and backpropagation, nearest neighbor classifiers assign equal weight to each attribute. This may cause confusion when there are many irrelevant attributes in the data. The k-nearest neighbors' algorithm is amongst the simplest of all machine learning algorithms, it also can be used in regression problems as well as classification.

The choice of which specific learning algorithm we should use is a critical step. The classifier's evaluation is most often based on prediction accuracy (the percentage of correct prediction divided by the total number of predictions), so both overfitting and under fitting of the input data should be avoided, several techniques are used, one of them is stopping the training process before the overfitting occurred.

2.6 Related Works

Skin diseases are among the most common health problems worldwide, a great work has been done by many researchers to develop computer aided systems to diagnose many types of Skin Diseases, various techniques were successfully implemented, as example in 2009 a framework for diagnosing skin diseases using feedforward artificial neural networks was designed by L. G. Kabari and F. S. Bakpo [11], the data used as input to train the artificial neural networks was the patient complaints, patient vital signs, patient demographics and presence of specific symptoms, although the system accuracy was 90%, but its training based on verbal words, computer vision techniques were not implemented in this system, so it was not sufficient enough.

In the year 2014 there was many attempts for developing better Skin Diseases Diagnosis systems, for example Sanjay Jaiswar, Mehran Kadri and Vaishali Gatty present a computer aided method to detect Melanoma skin cancer using image processing techniques [12], the image samples were provided to the system, many preprocessing techniques are applied, these techniques were image illumination equalization, color range normalization, image scale fitting and image resolution normalization, then images are passed through a segmentation phase. There were 3 algorithms used for segmentation, threshold based segmentation, clustering techniques and edge detection based, then features -Asymmetry index, Border Irregularity, Color index and Diameter- were extracted from the images to be used as basis for the detection, then TDS index is calculated from the extracted features to determine the existence of a cancer disease or not according to the value of the TDS index, the output given by this system will help the dermatologist to detect the lesion and its type, but it can’t be implemented as standalone equipment for the diagnosis process.

Also in May 2014 Delia-Maria FILIMON and Adriana ALBU attempt to develop a System that suggest a diagnosis regarding skin diseases from erythemato-scuamous class [13], the system was developed in Matlab environment, its neural network has a hidden layer with 10 neurons, output layer with 6 neurons which has been trained using backpropagation learning algorithm, the system also has 33 inputs of clinical and histopathological feature of the patients, based on these features and the predicted disease the system provides suggestions of the medical treatments of the patient. Although this system got an accuracy that is almost 94% but it doesn’t use features that are extracted from an image of the infected area of the skin, it completely depends on clinical features, it also used to diagnose uncommon diseases with a threshold output.

But in September-October 2014 Sarika Choudhari and Seema Biday have proposed a computer aided diagnosis system for Skin Cancer [14], based on dermascopy images of the skin that were enhanced by applying many preprocessing techniques, Dull Razor Filter to remove hair, obtaining the grayscale of the image, contrast enhancement and median filter to remove noise, the infected area was segmented from the rest of the skin according to its binarized gray level by applying Maximum Entropy. Thresholding, Level Co-occurrence Matrix (GLCM) was implemented to extract features from the enhanced image, multi-layers Feedforward Artificial Neural Network with backpropagation learning algorithm was used as a classifier that trained based on the extracted features, their Methodology has got 86.66% as accuracy, but this system only detects whether it's a Melanoma or non-melanoma so it couldn't differentiate between different types of skin diseases despite of their seriousness.

Few months later, in December 2014 Rahat Yasir, Md. Ashiqur Rahman, and Nova Ahmed have proposed a method that uses computer vision based techniques to detect different diseases from color images, the system successfully detects 9 different types of dermatological skin diseases with an accuracy rate of 90% [15], they have used 10 feature, 7 features were extracted from user's input (liquid type, liquid color, elevation, duration, feeling, gender, age) and 3 features were extracted from the image(color, area, edge), those features were extracted using eight different types of algorithms which are grey image, sharpening filter, median filter, smooth filter, binary mask, histogram, YCbCr and sobel operator. They have trained these 10 feature as input to train and test into a feed forward back propagation artificial neural network to identify the dermatological disease. Although this model has high accuracy but it depends on only 10 features only 3 of them are extracted from the image, and the rest are taken as input from the user, which might not be accurate enough (take feeling as example). making it more human dependent and less automated. In 2015 they have implemented the same model above that have android interface to the system functions, making it portable and more friendly in use [16].

In January 2015 A.A.L.C. Amarathunga, E.P.W.C. Ellawala, G.N. Abeysekara and C. R. J. Amalraj have proposed System enables users to recognize only 3 Skin Diseases and provides advises or treatments [16], also many image processing and data mining techniques have been implemented in this system, skin images were enhanced using both median and Gaussian filters, thresholding segmentation was applied, then Morphological the enhanced image were extracted, these features along with external information from the user present the input to many classifiers (AdaBoost, BayesNet, J48, MLP, NaiveBayes), both MLP and J84 were better than the rest with accuracy more that 85%, but the system can only recognize 3 diseases(Eczema, Impetigo and Melanoma), also the distance between camera lens and affected skin was 5cm in addition it only developed for windows operating system.

In 2016, Pravin S. Ambad1, A. S. Shirsat have develop an Image analysis system to detects skin diseases, they develop a system to be used for early detection and prevention of the skin diseases and they target 3 main diseases skin cancer, psoriasis and dermatophilosis [17], the disease diagnosis and classification is built on statistical parameter analysis. Statistical parameters include: Entropy, Texture index, Standard deviation, Correlation, the user of the system will able to take images of different moles or skin patches. Then the system will analyze and process the image and classifies the image to normal, melanoma, psoriasis or dermo case based extracting the image features.an alert will be provided to the user to seek medical help if the mole belongs to the atypical or melanoma category, the input images firstly passed through a median filter to remove a remove the noise, then apply the image enhancement and the statistical analysis techniques, then two-level classifier is used the first level is to specify if the image is either normal or abnormal and the second level is to classify into specified category: Melanoma, Psoriasis or dermo, the system is classify the images with accuracy 90%.

Vinayshekhar Bannihatti Kumar, Sujay S Kumar and Varun Saboo then provided an approach to detect 6 different skin diseases using smartphones in 2016, they have implemented dual stage approach combines machine learning and computer vision [18]. The computer vision consists of two stages in the first stage eight preprocessing techniques were implemented in order to extract features of the image namely converting to grey scale image, sharpening filter, median filter, smooth filter, binary mask, RGB extraction, histogram and sobel operator, the extracted features are used as input for training two different models in the second stage, these models are Maximum Entropy model and

Feedforward Artificial Neural Network with two hidden layers and Softmax output layer that learned using Backpropagation learning algorithm, this stage was developed for users that couldn't access the histopathological attributes. In the Machine Learning stage the histopathological attributes entered by the user combined with the features have been extracted from the image were used as input to train three different training models, Decision Tree, Feedforward Neural Network same as in [17], and K'th Nearest Neighbor. The novel method of using a dual stage system has given very promising results in identification of skin diseases with accuracies of up to 95%, although they got high accuracy but it decreases when tested with varying skin colors.

Then Suneel Kumar and Ajit Singh in the same year, also develop a Computer based skin disease detection system using digital image processing techniques for the classifications of the infected skin [19], the unique features of the images were extracted using two algorithms HSV-histogram and SURF algorithm, then the extracted features were fed in a K-NN classifier to classify the image to normal skin or infected, 5 classes were used in which 5 shows the normal skin and 1 to 4 is showing the infected skin (i.e. 1 for bloody, 2 for burned, 3 for cancer and 4 for allergic skin), this model got good accuracy, but it only classify the images into a general classification level and do not has a further detailed disease classes, but it could be very useful in medical field to see the clear image of the infected part in the skin as well as the parts that are not visible by human eyes.

Finally, Haofu Liao investigate the feasibility of constructing a universal skin disease diagnosis system using deep convolutional neural network (CNN) [20], the dataset used in the model is from two main sources, Dermnet dataset which include 23,000 skin disease images and more than 600 skin diseases divided into 23 main classes of diseases and OLE dataset which contains more than 1300 skin disease images and 19 skin diseases, the convolutional neural network is built on the Dermnet dataset and the classes taken is the main 23 classes, then the system is tested using the Dermnet dataset and the OLE dataset, the resultant accuracy of the test using the Dermnet dataset was 73.1% for the top-1 accuracy and 91% for the top-5 accuracy, and the accuracy when testing with OLE dataset was 24.8% top-1 accuracy and 61.7% top-5 accuracy, the accuracy decreased due to the lack of the broader variance in the training set, so when increasing the variance in the training set the accuracy improved to 31.1% top-1 accuracy and 69.5% top-5 accuracy. This model can be improved by collecting data from extra resources, their system in addition to its low accuracy, it’s of high computational cost since training CNN requires computational resources of high performance, or it will be very time consuming.

METHODOLOGY AND DESIGN

3.1 Overview

This chapter presents each step of the design and implementation of the skin diseases diagnosis system and discuss the methods used in each step, and provide figures and tables from the implementation process for more explanation, the chapter is divided into three main parts the first one is discussing how the data has been gathered, the second one is discussing the classification model design and implementation and the third part is considering the system development and integration.

3.2 Dataset

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, shown in the Figure 3.3.1.1-1, 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.

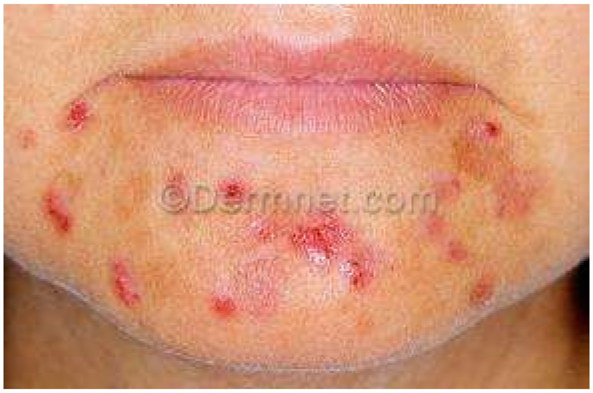


Figure 3.3.1.1-1Dernmet image sample

3.3 Learning Model

3.3.1 Images Preprocessing

Before using the images to train our model, series of preprocessing have been applied to our data to enhance the images also to increase our data for better generalization. All these processes were implemented using MATLAB image processing toolbox.

3.3.1.1 Resizing the image

At first all images were resized to be 293*192, resizing the image is important to have a uniform size for all images because the number of features that will be extracted from each image must be unified, we choose this size to reduce the computational efficiency, after resizing then list of preprocessing are applied to the image.

3.3.1.2 Gray images

Images were converted from RGB – red, green and blue- type to gay scale images. Sample is shown in Figure 3.3.1.2-1 below.

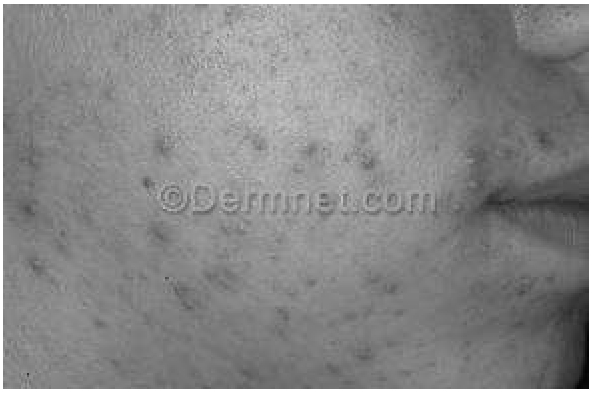


Figure 3.3.1.2-1 Grayscale image sample

3.3.1.3 Powered images

Since Negative images are useful for enhancing white or grey detail embedded in dark regions of an image, then we convert these gray images to their negative, then power of two is applied to the negative image to darken the image. Sample is illustrated in Figure 3.3.1.3-1, notice that the injected area here is more visually appealing, but also the water mark become more clearly.

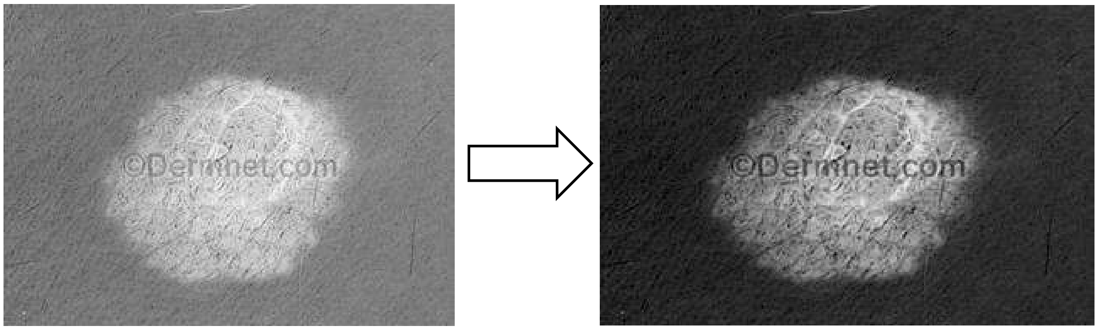


Figure 3.3.1.3-1 From Negative Image to Power of two Image

3.3.2 Learning Model Selection

Before using Bag-of-Features as basis for training the data, list of techniques for classification was used, *Table 3.3-1* shows the results of these models using jkkjk200 images as input for training and evaluation, only Cross-Validation is applied.

Convolutional Neural Networks in addition to its low accuracy, it requires very high computational resources (when using GPU, it requires GPU computational capabilities greater than 3.0 which was not available for us), or then when using CPUs, it will take very long time for training our data.

Then we attempt to extract HOG features of each image and classify these features using SVM classifier, the result was insufficient because HOG features are not suitable to present our data.

Table 3.3-1 Different Classification Techniques

Learning Model	**Associated accuracy**
CNN	66%
Bag-of-Features	83%
HOG features with SVM classifier	55%

Based on these promising results, bag-of-features was selected to be the model to train our Skin Diseases Diagnosis System's data.

3.3.3 Bag of Features Model

Bag of Features approach in computer vision in the past few decades has been used a lot in many applications. Bag of Features (BoF) methods have been applied to image classification, object detection, image retrieval, and even visual localization for robots.

In our System BoF approach is implemented to train our data, since it is used to classify images based on its texture. BoF approaches are characterized by the use of an order less collection of image features. Lacking any structure or spatial information this eliminates the effect of the water mark in our images which increase the accuracy compared with other learning models such as Convolutional Neural Networks and manually HOG features extraction along with SVM classification**Error! Reference source not found.**.

All preprocessed images are combined together along with their labels (Acne, Eczema and Melanoma) to form the input data to lean the BoF model, to implement BoF model approach, three steps must be followed. The first step is to extract features from the images, interest point must be detected and described in this step, step two is Quantization, finally the last step is the Classification of the quantized vectors.

3.3.3.1 Step One: Feature Extraction

This step is the base for the coming steps, the features that will be used to train the classifier will be extracted at this step, to achieve this the interest points must both be detected and described.

Interest points detection can be achieved in several ways. Dense feature could be used, also one of the feature extraction techniques such as Harris Corner Detection, FAST, SURF and SIFT that described in 2.4 .

At first we attempt to use SURF features to detect interest points but it was not sufficient, because the result list of the interest points was crossed and concentrated together ignoring large part of the image that considered as important points also. The next figure shows the strongest 10 points extracted using SURF detector.

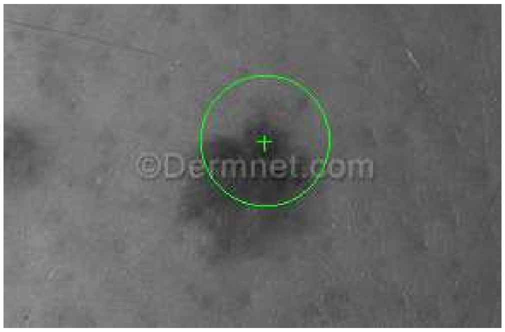


Figure 3.3.3.1-1 Strongest 10 points extracted using SURF

For better results we choose dense features, every part in the image contributes and effects on the features selected to describe the image, the image is simply divided in regular grid, when the interest points are detected then features are extracted based on these points.

In our model we select SURF algorithm to extract features from each block of each interest point. These features represent what is called visual words describing the image.

3.3.3.2 Step Two: Quantization

All feature of all images (visual of words) are quantized, i.e. using clustering technique to cluster these visual words to specific number of clusters (visual vocabulary), the image is represented as distribution of these words. This is done by k-means clustering to represent group of similar visual words as single cluster (visual vocabulary). The number of desired clusters is selected manually, optimal selection of the number of visual words vocabulary depends on two factors, the first one if it's too long then the computational cost will increase, the second one is that when number of visual words vocabulary is too short then no proper discrimination between features will be obtained. In our model number of 2000 visual words is selected to be the number of visual words.

Figure 3.3.3.2-1, Figure 3.3.3.2-2 and Figure 3.3.3.2-3 show the histogram of visual vocabularies of samples from the three classes of the model, notice the different histogram distribution of each class, which can be used as basis for the classification.

K-means Clustering Algorithm

K-means is a method of clustering observations into a specific number of disjoint clusters. The ”K” refers to the number of clusters specified. Various distance measures exist to determine which observation is to be appended to which cluster. The algorithm aims to minimize the measure between the centroid of the cluster and the given observation by iteratively appending an observation to any cluster and terminate when the lowest distance measure is achieved [22]. K-means clustering is one of the simplest clustering techniques, typical steps of the algorithm are:

- The sample space is initially partitioned into K clusters and the observations are randomly assigned to the clusters.
- For each sample Calculate the distance from the observation to the centroid of the cluster. IF the sample is closest to its own cluster THEN leave it ELSE select the closest cluster.
- Repeat steps 1 and 2 until no observations are moved from one cluster to another.

Common distance measures include the Euclidean distance, the Euclidean squared distance, The Euclidean measure corresponds to the shortest geometric distance between two points, a faster way of determining the distance is by use of the squared Euclidean distance. Euclidian distance d between n points is determined by:

$$d = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2} \quad Eq. 3.4\ 1$$

Sample of Visual Vocabularies Histograms for the model classes

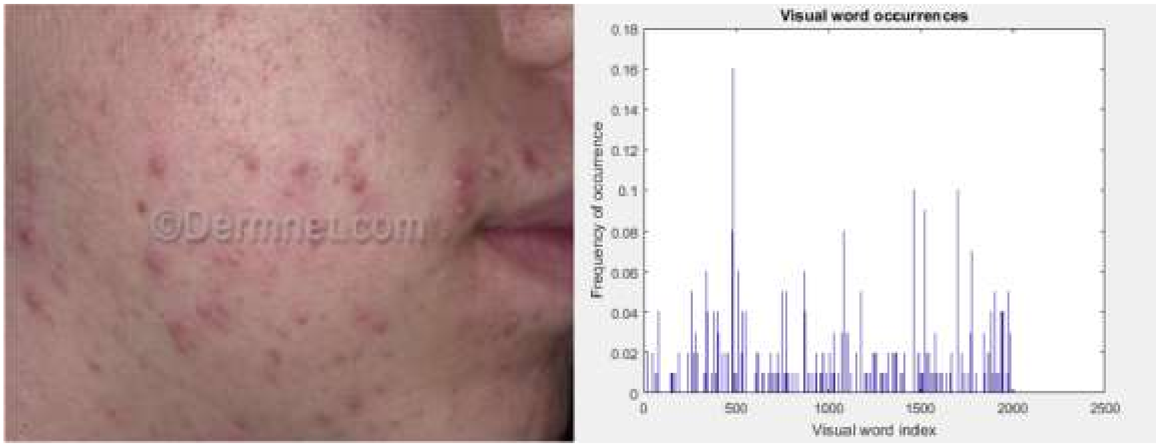


Figure 3.3.3.2-1Sample Image for class 1 Acne & Its Histogram of Visual vocabularies

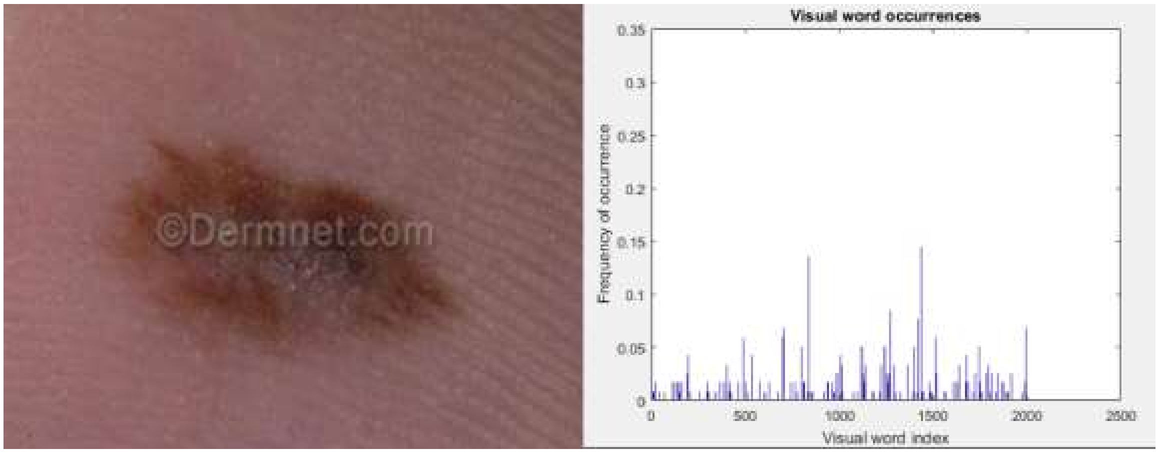


Figure 3.3.3.2-2Sample Image for class 2 Melanoma & Its Histogram of Visual vocabularies

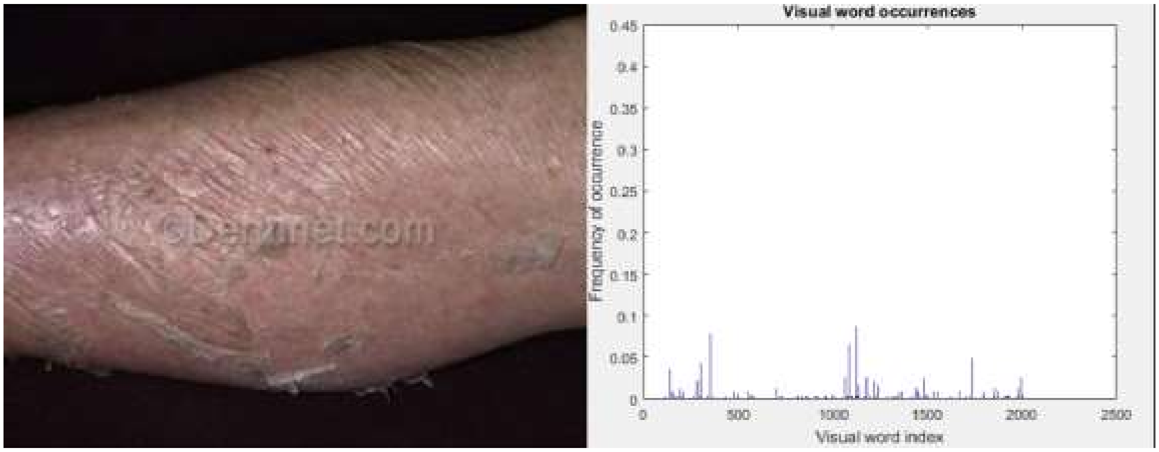


Figure 3.3.3.2-3Sample Image for class 3 Eczema & Its Histogram of Visual vocabularies

3.3.3.3 Step Three: Classification

When different visual vocabularies are obtained then each image is described using these vocabularies, the histogram of each visual vocabulary is determined and stored in what is called feature vector, all vectors of all images represent the input to the classifier. There are many classification techniques as mentioned in 2.5.

Here Support Vector Machine -SVM- classifier is used, since the input data is complicated and nonlinearly separable then SVM with Radial Basis Function kernel is used.

There are many types of radial basis functions such as Gaussian radial basis function, Multi-Quadric Functions and Thin Plate Spline Function, Gaussian function is the most commonly used.

Gaussian Function:

$$\varphi(r) = exp\left({-r^2}/{2\sigma^2}\right) \quad 3.3\text{-}1$$

For further information about SVM see Appendix A.

An overall view of Bag-of-Features model is illustrated in the Figure 3.3.3.3-1.

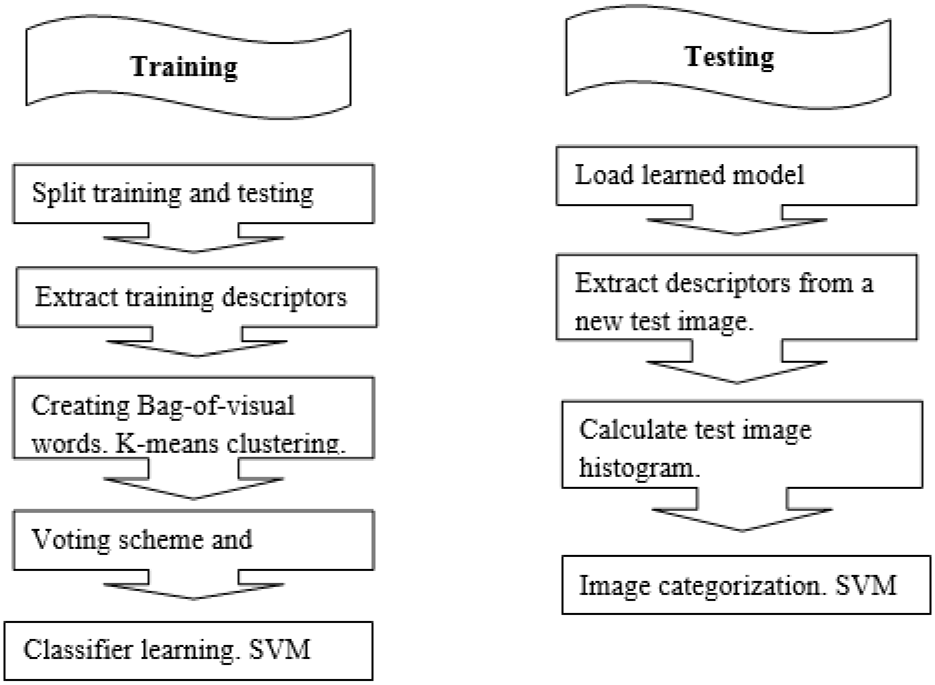

Training
Split training and testing
Extract training descriptors
Creating Bag-of-visual words. K-means clustering.
Voting scheme and
Classifier learning. SVM
Testing
Load learned model
Extract descriptors from a new test image.
Calculate test image histogram.
Image categorization. SVM

Figure 3.3.3.3-1 Bag-of-Features Model

3.3.4 Model Training and Evaluation

In the training, we use 400 images of each class, 200 of them were colored images, 200 were gray images, so we get overall 1200 images, the dataset was separated that 60 percent of the images were used for training and 40 percent used for the evaluation.

In the training process, bag of features model is used, implemented in MATLAB, many parameters were tuned such as the size of visual vocabulary, grid step, box constraint (A parameter that controls the maximum penalty imposed on margin-violating observations, and aids in preventing overfitting) and the percentage of total extracted features, to determine the optimal or close to optimal of values of these parameters.

To evaluate the model, the accuracy has been used, the accuracy of the model is the number of positive predictions divided by the total number of positive class values predicted.

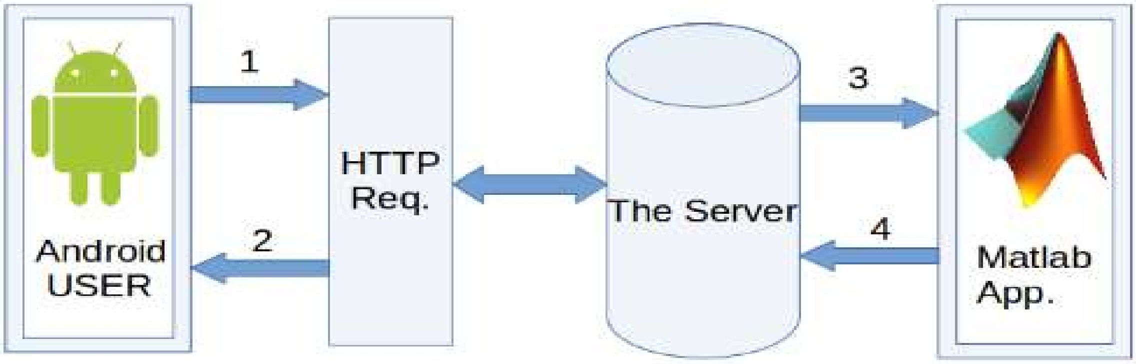
We use Cross-Validation method, that is randomly dividing the data set to training and testing datasets based on percentage, our data was divided to 60% images for training, and

40% images for testing and evaluation. Also additional data is obtained to test our model that's called Holdout evaluation method.

3.4 System Design

In this part, we discuss the overall system architecture, and explain in details the design and development process of each part of the system, and the tools and methods used to do that.

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 MATLAP 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. The two sides are connected together through apache server and SQL database server using HTTP protocol.



3.4.1 Client Side

3.4.1.1 Android Application Interface

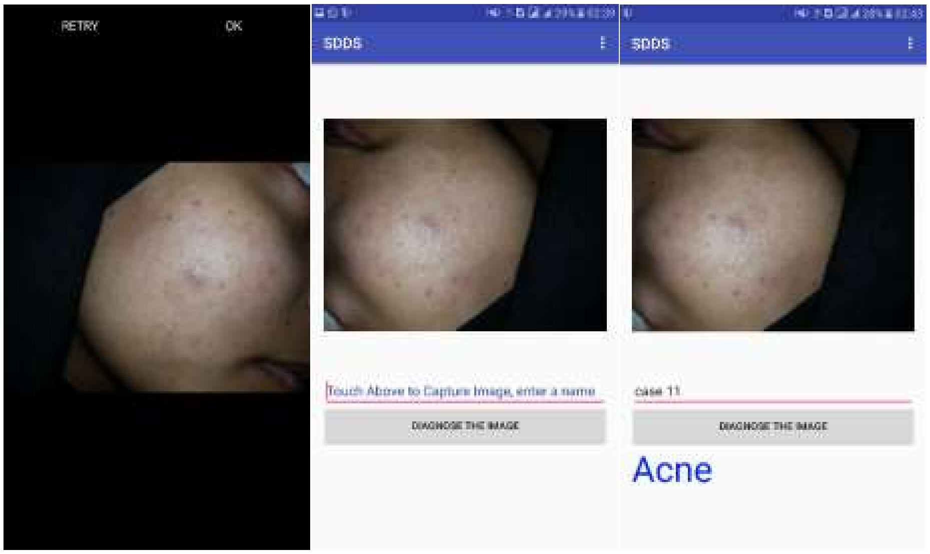


Figure 3.4.1.1-1 Android Application Interface

Figure 3.4.1.1-1 Android Application Interface presents the layout of the application, the application has only one screen which provides the input required views, the image to be uploaded, the name of the image and a button to send the image to the server, also there is a text view to print the resultant outcome when the output is received, another part of the interface is the option menu part at the top right corner of the screen which include an option to configure the server address, when this option is selected it pops out a dialog bar asking to enter the server address and confirm it to be configured for the connection.

3.4.1.2 Development Tools and Libraries

The tools used in the development process of the android application.

Android Software Development Kit (ADK)

It provides the developers the API libraries and developer tools necessary to build, test, and debug applications for Android, enabling them to create android virtual devices (AVD) for testing, helping the developers to monitor and control AVDs and physical devices connected to the computer.

Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

All these features make android studio the best development tools for developing android applications, the version used to develop skin diseases diagnosis system is android studio 2.3.

VOLLEY LIBRARY

Volley is an HTTP library that makes networking for Android apps easier and faster, it provides many benefits such as [23]:

- Automatic scheduling of network requests.
- Multiple concurrent network connections.
- Transparent disk and memory response caching with standard HTTP cache coherence.

- Support for request prioritization.
- Cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel.
- Ease of customization, for example, for retry and back off.
- Strong ordering that makes it easy to correctly populate your UI with data fetched asynchronously from the network.
- Debugging and tracing tools.

Testing Device

In the android development there are two options generally for building and running your project, the first one is using AVD to create a virtual device within your PC and run the application into it, or you can use an external physical device that using Android OS to run your application into using physical connection, this option is a very efficient compare to the first one, even though it requires providing an external device but it give a better performance than the first option, the external device must be configured first to be able to run the application into it. For that the second option is selected and a SAMSUNG device has been used for the testing and debugging operations of the android development.

3.4.1.3 Android Application Development

The development of the android application of the skin diseases diagnosis system is done using android studio IDE installed on Ubuntu operating system, generally in the android applications there are basic functions that must be in each application, and each one of them is called at specific time these functions are explained in the reference [23], also there are some specific function for this application each one is developed for specific task, which is called along the application running to do the certain task and get the result then the application continue through the lifecycle of the basic functions, these specific functions are:

- **getStringImage:**

this function take the image captured by the user and convert it to array of bytes stream class and encode the image using base64 encoder to make it suitable to be send to the server.

- **uploadImage**:
  this function is to take the image capture by the user and make string request to the server to send the image to the server using the PHP code saved in the server side.
- **getResult – Android side**:
  this function is called after the response of the upload image is received, it creates another string request to the server, and execute the getResult PHP script in the server, then it get the result from the server and write it down to a text view field to print it to the user.
- **savePicture – PHP side**:
  this function is on the server side but it's called from the client side whenever an image has to be uploaded, it's a PHP script that receives parameters from the android request through the volley, which represent the image name and the encoded image, then it decode the image using base64 decoder and using the image name it create a path for the image in the pictures file on the server, then it save the decode image to that path, then create a database connection by defining the database name and the user name and the password, select the database and the table for skin diseases and write the image name and path to new row in the database, then close the connection.
- **getResult** – PHP side:
  this function is also on the server side and is called to receive the result from the server after the first request has been sent, it's a PHP script that create the database connection similar as the previous one in the save Picture script and read the result classification from the classification column in the database.

Android Device Permissions Policy

The new android versions from android 6.0 or higher, has a new permissions policy for granting the access to the device resources, simply they don't accept the general access permission from the android manifest file, as applied for the older versions, they require to grant a permission whenever you need to access a device resource, so an online permission algorithm has been implemented to check the permission grants whenever there's a need.

Synchronization between the Android and MATLAB

The MATLAB requires a time to read the image, process it and write the result back to the server, this time cannot be in a single connection request because the android will shut down any connection that exceed a certain limit of a time, so there is a synchronization required between the two application to organize the overall operation of the system and to resolve any chance of conflict between the processes, a VOLLEY library has been used to manage the connection for the android, and two separate requests for the input and the result has been used to resolve the long time of the connection, and the second request is executed in the response of the first one to save the user time and do not make it wait longer time, also do not require any addition actions from the user to get the result back for good user experience policy.

3.4.1.4 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.

Send the image to the server

This function is used to send the input image to the server using HTTP request.

Receive the results from the server

This function receives the result data from the server and store it in the android application.

Show the results

This function is used to view the result data that has been received from the server to the user which contains the classification output class.

Menu options

It appears when the option button is clicked. It’s a small menu consist of only one menu buttons. This button leads to the next function.

Set the server address

This option provides the ability to change the server address used to establish the connection, necessary function specially in the development process but it can be discarded by using the domain name which is constant, instead of the server address which changes from network to network, or from server to server.

3.4.2 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.

3.4.2.1 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:

- 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.

3.4.2.2 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.

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.

3.4.3 Client Server Connection

3.4.3.1 Tools

The tools used for establishing a connection between the client and server are apache server and Database server which are provided as an open source package from XAMP.

XAMP is completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl, so it’s easy to be used as a single server application to set up the whole server environment.

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.

3.4.3.2 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.

3.4.3.3 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.

3.4.4 Integrated System Testing

The system has been tested against the following requirement: connection establishment, image upload speed, the correct information flow, the processes execution sequence and to the database accessibility, also test the overall system emerging properties, system performance and the system security.

The overall system has been tested using XAMP server installed in WINDOWS operating system where the MATLAP application is installed, and the android application is installed on a SAMSUNG device with android OS 6.0 to test the new permission policy too, then the two devices are connected together using D-link router wireless network and also using a mobile hotspot tethering, the system works properly for all specified features.

RESULTS AND DISCUSSION

4.1 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 4.1-1 and Table 4.1-2, which illustrate the resultant accuracy per class.

Table 4.1-1Confusion matrix of Cross-Validation method

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

Table 4.1-2Confusion matrix of the Holdout Method

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

In table Table 4.1-1, 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 4.1-2, 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.

4.2 Data Quantity and Variation Effect on Accuracy

Table 4.2-1 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 4.2-1 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 Table 4.2-2.

Table 4.2-2 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. These data were used to train the final model.

4.3 Integrated System Results

The skin diseases diagnosis system is successfully built with all the specified functionalities, giving the expected outcome at each step, the data successfully flow between client and server without problems.

The image is captured using a mobile camera and is sent to the server successfully, the image information is recorded in the database, then the MATLAB script read the data immediately and load the image and perform the classification with the pre-trained model, then writing the result back to the database, then the mobile application read the result directly from the database and print it to the user.

The system also performs the task with a good performance, an addition its easily used requiring uncomplicated configurations to be used, so it’s a user-friendly application.

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 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 Ubuntu 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.

5.1 Future Work

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
- The system may be combined with other medical systems to propose an integrated medical care services, the suggested system is TBibak.

REFERENCES

[1] "World life Expectancy," [Online]. Available: http://www.worldlifeexpectancy.com/sudan-skin-disease.

[2] M. l. Antkowiak, "Artificial Neural Networks vs. Suport Vector Machines for Skin Diseases Recognition," 2006.

[3] R. E. Woods, Digital Image Processing, Second Edition, 2001.

[4] A. A. A. a. H. A. A. M. Hassaballah, Image Features Detection, Description and Matching, Springer International Publishing Switzerland, 2016.

[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 Comunication Engineering,* vol. 2, no. 3, 2014.

[6] J. R. Edouard Oyallon, "An Analysis of the SURF Method," 2015.

[7] C. Tomasi, "pdfs.semanticscholar.org," [Online]. Available: https://pdfs.semanticscholar.org/8086/99bacf0cc91a38fa77b6af5a9f91dc25d127.pdf. [Accessed 10 5 2017].

[8] N. A. Megha Gupta, "CLASSIFICATION TECHNIQUES ANALYSIS," in *NCCI 2010 -National Conference on Computational Instrumentation*, Chandigarh, 2010.

[9] T. N. Phyu, "Survey of Classification Techniques in Data Mining," in *International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2009.

[10] S. B. K. ·. I. D. Z. ·. P. E. Pintelas, "Machine learning: a review of classification and combining techniques," 2007.

[11] F. B. Ledisi G. Kabari, "Diagnosing Skin Diseases Using an Artificial Neural Network," in *ResearchGate*, 2009.

[12] M. K. V. G. Sanjay Jaiswar, "Skin Cancer Detection Using Digital Image," *International Journal of Scientific Engineering and Research (IJSER),* 2014.

[13] A. A. Delia-Maria FILIMON, "Skin diseases diagnosis using artificial neural," ResearchGate, 2014.

[14] S. B. Sarika Choudhari, "Artificial Neural Network for SkinCancer," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),* vol. 3, no. 5, 2014.

[15] 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.

[16] 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.

[17] A. S. S. Pravin S. Ambad, "A Image analysis System to Detect Skin Diseases," www.iosrjournals.org, Vadgaon, Pune, India, 2016.

[18] S. S. K. a. V. S. Vinayshekhar Bannihatti Kumar, "Dermatological Disease Detection Using Image Processing And Machine Learning," IEEE, 2016.

[19] A. S. Suneel Kumar, "Image Processing for Recognition of Skin Diseases," *International Journal of Computer Applications (0975 – 8887),* vol. 149, 2016.

[20] H. Liao, "A Deep Learning Approach to Universal Skin Disease Classification," University of Rochester, 2015+.

[21] "skincancer.org," 2017. [Online]. Available: http://www.skincancer.org/skin-cancer-information/melanoma. [Accessed 2 October 2017].

[22] "kom.aau.dk," [Online]. Available: http://kom.aau.dk/group/04gr742/pdf/kmeans_worksheet.pdf. [Accessed 10 10 2017].

[23] Google, "Developer.Android," Google, [Online]. Available: https://developer.android.com/training/volley/index.html.

[24] 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.

[25] 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.

[26] L. G. K. a. F. S. Bakpo, "Diagnosing Skin Diseases Using an Artificial," IEEE, 2009.

[27] V. K. N. U. s. Nisha Yadav, "Skin Diseases Detection Models using Image Processing: A Survey," *International Journal of Computer Applications (0975 – 8887),* vol. 137, 2016.

[28] O. O. O. a. S. A. O. Damilola A. Okuboyejo, "Automating Skin Disease Diagnosis Using Image Classification," in *World Congress on Engineering and Computer Science*, San Fransisco, 2013.

APPENDIX A

SVM Theory

We have L training points, where each input xi has D attributes (i.e. is of dimensionality D) and is in one of two classes yi = -1 or +1, i.e our training data is of the form:

$$\{x_i\ y_i\}\quad where\ i = 1,2 \ldots L,\ y_i\ \in \{-1, 1\}, x_i \in \mathcal{R}^D$$

Here we assume the data is linearly separable, meaning that we can draw a line on a graph of x_1 vs x_2 separating the two classes when D = 2 and a hyperplane on graphs of x_1, x_2 … x_{D}, for when D > 2.

This hyperplane can be described by w.x + b = 0 where:

- w is normal to the hyperplane.
- $\frac{b}{||\,w||}$ is the perpendicular distance from the hyperplane to the origin.

Support Vectors are the examples closest to the separating hyperplane and the aim of Support Vector Machines (SVM) is to orientate this hyperplane in such a way as to be as far as possible from the closest members of both classes.

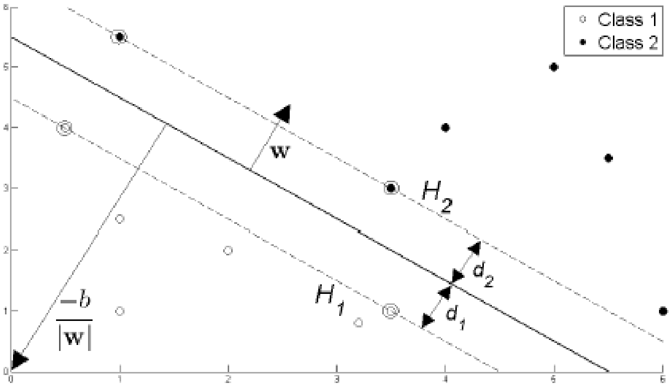


Figure 2 : Hyperplane through two linearly separable classes

Referring to Figure 2, implementing a SVM boils down to selecting the variables w and b so that our training data can be described by:

$$x_i.w + b \geq +1 \ for \quad y_i = +1$$

$$x_i.w + b \leq -1 \ for \quad y_i = -1$$

These equations can be combined into:

$$y_i(x_i.w + b) - 1 \ \geq 0 \ \forall_i \ (1.3)$$

If we now just consider the points that lie closest to the separating hyper-plane, i.e. the Support Vectors (shown in circles in the diagram), then the two planes H_1 and H_2 that these points lie on can be described by:

$$x_i.w + b = +1 \quad for \quad H_1$$

$$x_i.w + b = -1 \quad for \quad H_2$$

Referring to Figure 2, we define d_1 as being the distance from H_1 to the hyperplane and d_2 from H_2 to it. The hyperplane's equidistance from H_1 and H_2 means that $d_1 = d_2$ - a quantity known as the SVM's margin. In order to orientate the hyperplane to be as far from the Support Vectors as possible, we need to maximize this margin.

the Support Vectors as possible, we need to maximize this margin. Simple vector geometry shows that the margin is equal to $\frac{1}{||w||}$ and maximizing it subject to the constraint in (1.3)is equivalent to finding:

$$\min||w|| \quad such \ that \quad y_i(x_i.w + b) - 1 \ \geq 0 \ \forall_i$$

Minimizing $||w||$ is equivalent to minimizing $\frac{1}{2} \ ||w||^2$ and the use of this term

makes it possible to perform Quadratic Programming (QP) optimization later on. We therefore need to find:

$$\min\frac{1}{2} \ ||w||^2 \quad such \ that \quad y_i(x_i.w + b) - 1 \ \geq 0 \ \forall_i$$

In order to cater for the constraints in this minimization, we need to allocate them Lagrange multipliers α ,where $\alpha_i \geq 0 \ \forall_i$:

$$LP \equiv \frac{1}{2}\left||w|\right|^{2} - \alpha[\ y_i(x_i.w+b)-1\ \forall_i]$$

$$\equiv \frac{1}{2}\left||w|\right|^{2} - \sum_{i=1}^{L} \alpha_i[\ y_i(x_i.w+b)-1\ \forall_i]$$

We wish to find the w and b which minimizes, and the α which maximizes (1.9) (whilst keeping $\alpha_i \geq 0\ \forall_i$). We can do this by differentiating L_P with respect to w and b and setting the derivatives to zero:

$$\frac{\partial\, L_{\ p}}{\partial w} = 0 => w = \sum_{i=1}^{L} \alpha_i\ \ y_i x_i\ \ (1.10)$$

$$\frac{\partial\, L_{\ p}}{\partial b} = 0 => \sum_{i=1}^{L} \alpha_i\ \ y_i = 0\ (1.11)$$

Substituting (1.10) and (1.11) into (1.9) gives a new formulation which, being dependent on α, we need to maximize:

$$LD = \sum_{i=1}^{L} \alpha_i - .5 \sum_{i,j} \alpha_i\ \alpha_j\ y_i y_j x_i\,.x_j, \qquad \alpha_i \geq 0\forall_i\ and \sum_{i=1}^{L} \alpha_i y_i = 0$$

$$= \sum_{i=1}^{L} \alpha_i - .5 \sum_{i,j} \alpha_i\ H_{ij}\alpha_j \qquad , where\ H_{ij} = y_i y_j x_i.x_j \geq 0\forall_i\ and \sum_{i=1}^{L} \alpha_i y_i = 0$$

$$= \sum_{i=1}^{L} \alpha_i - .5 \sum_{i,j} \alpha^T\ H_\alpha \qquad , \qquad \alpha_i \geq 0\forall_i\ and \sum_{i=1}^{L} \alpha_i y_i = 0$$

This new formulation LD is referred to as the Dual form of the Primary LP . It is worth noting that the Dual form requires only the dot product of each input vector xi to be calculated.

Having moved from minimizing LP to maximizing LD, we need to find:

$$\max(\alpha) \left[\sum_{i=1}^{L} \alpha_i - .5 \sum_{i,j} \alpha^T\ H_\alpha \qquad \right]\ such\ that\ \alpha_i \geq 0\forall_i\ and \sum_{i=1}^{L} \alpha_i y_i = 0$$

This is a convex quadratic optimization problem, and we run a QP solver which will return α and from (1.10) will give us w What remains is to calculate b.

.

Any data point satisfying (1.11) which is a Support Vector xs will have the form:

$$y_s(x_s.w + b) = 1$$

Substituting in (1.10):

$$y_s(\sum_{m \in s} \alpha_m \ y_m x_m . x_s + b) = 1$$

Where S denotes the set of indices of the Support Vectors. S is determined

by finding the indices i where α_i > 0. Multiplying through by ys and then

using ys^2 = 1 from (1.1) and (1.2):

$$y_s^2(\sum_{m \in s} \alpha_m \ y_m x_m . x_s + b) = y_s$$

$$b = \ y_s - \sum_{m \in s} \alpha_m \ y_m x_m . x_s$$

Instead of using an arbitrary Support Vector xs, it is better to take an average over all of the Support Vectors in S:

$$b = \frac{1}{N_s} (y_s - \sum_{m \in s} \alpha_m \ y_m x_m . x)$$

We now have the variables w and b that define our separating hyperplane's

optimal orientation and hence our Support Vector Machine.

APPENDIX B

The code of the system applications (MATLAB and android) attached in the CD.