

# A Decision Support System for Symptom-based Common Diseases and Image-based Skin Diseases Detection

by

Moinul Alam Joy  
19166022

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
M.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
May 2021

© 2021. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**



---

Mohammad Moinul Alam Joy  
19166022

# Approval

The thesis/project titled “A Decision Support System for Symptom-based Common Diseases and Image-based Skin Diseases Detection” submitted by

1. Mohammad Moinul Alam Joy

Of spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science on May 06, 2021.

## Examining Committee:

Supervisor:  
(Member)



---

Dr. Md. Golam Rabiul Alam  
Associate Professor  
Dept. of Computer Science and Engineering  
BRAC University

Program Coordinator:  
(Member)



---

Dr. Amitabh Chakrabarty  
Associate Professor  
Dept. of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)



---

Sadia Hamid Kazi, Ph.D  
Chairperson and Associate Professor  
Department of Computer Science and Engineering  
Brac University

## Abstract

Nowadays, it has become difficult to get in touch with a doctor for the current pandemic situation. Patients need to wait several days to get an appointment from the doctor and visiting the hospitals in recent times is also very risky. So, The main purpose of this application is to give a patient the basic treatment by given his symptoms so that he/she can receive medical services in home. This project focuses on two portions of disease detection. One portion is for common diseases and another one is for skin diseases. We have used different training algorithm for both portions of disease detection. An online API has been used along with machine learning library like tensorflow to produce the results. This mobile application not only shows the probability of the diseases, but also gives information about the cure or solutions.

**Keywords:** Disease Prediction;Support Vector Machine; Machine Learning; Skin Disease; MobileNet; Collaborative filtering; Android Application.

## Acknowledgement

Firstly, I would like to thank the Almighty God who provided us the opportunity to study at BRAC University, and blessed me with the most precious gift of physical and mental health during the whole study period. I would like to take the opportunity to express my humble gratitude to my honorable supervisor, Dr. Md. Golam Robiul Alam, Associate Professor, Dept. of Computer Science and Engineering, BRAC University. Under His supervision, I have completed this project successfully. His constant guidance and willingness made me understand this project and its manifestation in great depth of knowledge also helped me to complete the assigned task. His endless support, cooperativeness, inspiration, outstanding guidance, constructive criticism, and fantabulous suggestions throughout the progress of our work helped me make this work far better. I am cordially grateful to him and I feel blessed to have him as my supervisor.

# Table of Contents

Declaration	1
Approval	2
Abstract	3
Acknowledgment	4
Table of Contents	5
List of Figures	7
List of Tables	8
Nomenclature	8
<b>1 Introduction</b>	<b>9</b>
1.1 Introduction . . . . .	9
1.2 Motivation . . . . .	9
1.3 Problem Background . . . . .	10
1.4 Objectives . . . . .	10
<b>2 Related Work</b>	<b>12</b>
<b>3 Methodology</b>	<b>15</b>
3.1 Proposed System Diagram . . . . .	15
3.1.1 Dataset for Common Disease . . . . .	17
3.1.2 Data Visualization . . . . .	17
3.1.3 Implementation Using Web Scraping . . . . .	19
3.1.4 Data Preprocessing . . . . .	19
3.1.5 Intelligent Smart Symptom Prediction . . . . .	19
3.1.6 User Clustering . . . . .	19
3.1.7 Smoothing . . . . .	21
3.1.8 New Ratings . . . . .	21
3.1.9 The Dense User-Item Matrix . . . . .	22
3.1.10 Item Clustering . . . . .	22
3.1.11 Algorithm . . . . .	22
3.1.12 Selecting Clustering Centers . . . . .	23
3.1.13 Selecting Neighbors . . . . .	24
3.1.14 Producing Recommendations . . . . .	24

3.1.15	Support Vector Machine . . . . .	26
3.1.16	Decision Boundary . . . . .	26
3.1.17	Equation of Hyperplane . . . . .	27
3.1.18	Distance Measure . . . . .	27
3.1.19	Optimal Hyperplane . . . . .	27
3.1.20	SVM Representation . . . . .	28
3.1.21	Kernel Trick . . . . .	28
3.1.22	Kernel Functions . . . . .	29
3.2	Skin Disease Prediction . . . . .	29
3.2.1	Dataset . . . . .	30
3.2.2	Data Visualization . . . . .	30
3.2.3	MobileNet Description . . . . .	31
3.2.4	MobileNet Architecture . . . . .	32
3.3	Image Preprocessing . . . . .	35
3.4	Training Algorithm . . . . .	35
3.5	Inference Algorithm . . . . .	36
3.6	Final Result . . . . .	37
<b>4</b>	<b>Implementation and Result Analysis</b>	<b>38</b>
4.1	Implementation . . . . .	38
4.1.1	Diagnosis . . . . .	39
4.1.2	Diagnosis Sample Response . . . . .	39
4.1.3	Proposed Symptoms Sample Response . . . . .	40
4.1.4	SVM Accuracy . . . . .	40
4.2	Implementation of skin disease prediction . . . . .	40
4.2.1	Confusion Matrix . . . . .	42
4.2.2	Accuracy . . . . .	42
4.3	Result . . . . .	42
4.3.1	Common Disease Prediction . . . . .	44
4.4	Skin Disease Prediction . . . . .	47
<b>5</b>	<b>Conclusion</b>	<b>50</b>
5.1	Conclusion . . . . .	50
5.1.1	Limitations . . . . .	50
5.1.2	Future Work . . . . .	50
	<b>Reference</b>	<b>52</b>

# List of Figures

1.1	A general view of the project . . . . .	11
3.1	Proposed System Model . . . . .	16
3.2	Dataset for common disease . . . . .	17
3.3	Data visualization of common disease prediction . . . . .	18
3.4	Collaborative Filtering . . . . .	20
3.5	User Clustering . . . . .	20
3.6	Item Clustering . . . . .	22
3.7	Result for common disease prediction . . . . .	25
3.8	How SVM works . . . . .	26
3.9	Kernel Trick . . . . .	28
3.10	Different type of skin disease . . . . .	30
3.11	Data visualization for skin disease . . . . .	31
3.12	Block diagram of MobileNet . . . . .	33
3.13	Inference algorithm . . . . .	36
3.14	Skin disease output . . . . .	37
4.1	Tensor flow Lite image . . . . .	41
4.2	Confusion Matrix . . . . .	42



# List of Tables

3.1	Common parameters . . . . .	19
3.2	Proposed Symptoms . . . . .	24
3.3	MobileNet Architecture . . . . .	34
3.4	Training MobileNet . . . . .	36
4.1	Diagnosis contents . . . . .	39
4.2	SVM Accuracy . . . . .	40
4.3	MobileNet Accuracy . . . . .	42

# Chapter 1

## Introduction

### 1.1 Introduction

It is important to stay healthy and fit. But as time goes on, life is becoming more and more hectic by the day to reach out for medical appointments. The sole purpose of this android project is to provide quick, easy and comfortable medical aid to people who might need it. The patient can get the primary knowledge about their diseases by checking symptoms. They can get the probability with percentage of every disease which is depending on their symptoms. In case of skin diseases, the patients can see the probability of the occurrence of skin disease by live streaming, taking a picture through the application or by selecting a picture of their skin disease from the gallery option in their phone. In this Covid-19 pandemic situation, this is making a risk factor while people are going to the hospital to meet doctors. Several people are getting infected everyday by covid-19 by visiting the hospitals. To get medical attention, the patients have to go through the ordeal of going to the hospital, getting an appointment, waiting in line in a queue, all of which might be inconvenient for a patient in need of immediate medical treatment. If by some inconvenience the doctor has to cancel the appointment, the patients then find himself in a very bothersome situation. Besides, people are not informed which disease they are suffering from by checking their symptoms. They have to wait for the doctor's confirmation. So we are introducing this project so that the patients can get medical care in their home in this difficult times. The proposed work is an android application that will help people to find out any kind of disease by checking their symptoms. They can be notified about their disease and they will know about the reason and cure of the disease. In case of skin disease, we tried to detect and predict 7 common skin diseases by taking skin disease image data from public websites and manually taken images. Then the images are classified into the correct skin disease group by training in MobileNet CNN. [19].

### 1.2 Motivation

In this pandemic situation, getting an appointment of a doctor, especially in the hospital is very difficult and risky given the circumstances. To decrease this risk, people should take a step to consult or getting suggestions from a doctor from their places. Besides, several times doctors cannot get free time. In that time, if people can get suggestions just by selecting symptoms then it would be much pleasant and

beneficial for them. Figure 1.1 gives us a general view of the project. These are the works to had to offer to the project purpose,

- We proposed and developed a machine learning based framework for predicting symptom-based common diseases. The developed framework will support physicians and patients in recognizing common diseases.
- We proposed and developed a machine learning based framework for Image-analysis based real-time skin diseases detection. It is also a decision support system that may help dermatologists for recognizing basic skin diseases also in remote setting or telemedicine.

### 1.3 Problem Background

At present, most people are not able to know about their disease by checking symptoms. For this reason, they suffer physical difficulties for the lack of medical attention. They can't get services from doctors for different reasons, many can't even afford the service of a doctor. Many people also suffer or in worst case die for mistreatment. Some take medicines based on their own knowledge or by taking suggestions from their neighbors without discussing with any doctor. And besides, this covid-19 pandemic situation has made it more difficult to get in touch with a doctor. So patients with diseases other than covid-19 are struggling to get medical attention.

### 1.4 Objectives

The aim of this project is to deliver medical assistance to patients anywhere and anytime. The main objectives of this projects are as follows:

- To develop a system which can diagnose common diseases from the patient's symptoms.
- To inform the patients about the reasons and cure of their diseases.
- This android application can predict 42 different disease.
- To predict skin disease from image.
- To predict skin disease via camera in real-time.
- Can predict related symptoms according to the patient's given symptoms.
- To predict common disease and skin disease, we have used Support vector machine and Convolutional neural network respectively.

Here, **Figure 1.1** tries to give a visual representation of the general view of the project.

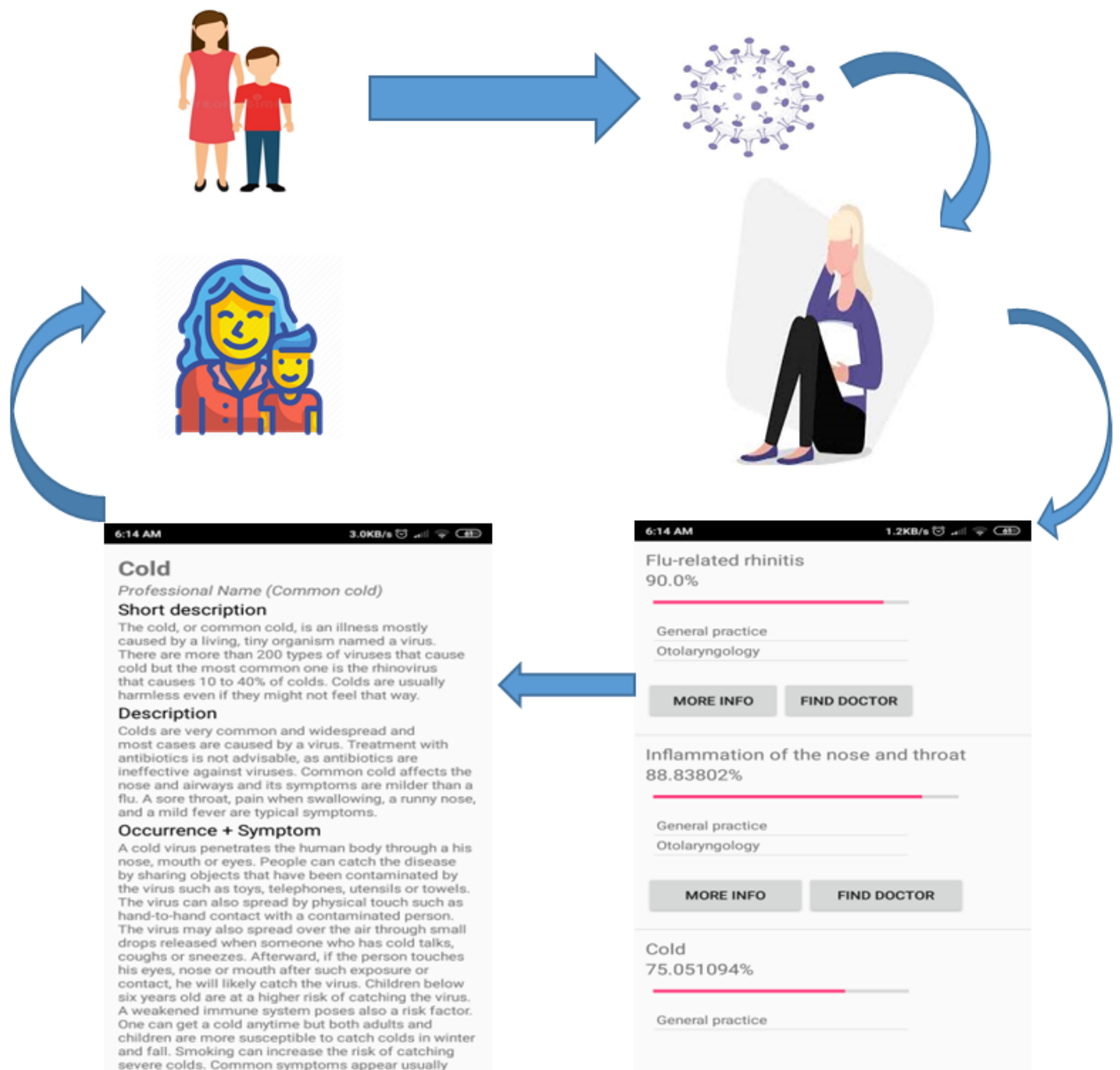


Figure 1.1: A general view of the project

# Chapter 2

## Related Work

The related projects and researches to point out the research gaps are briefly discussed in this particular chapter. A Smartphone-Based Skin Disease "Classification Using MobileNet CNN" is a similar mobile application which detects only skin diseases using MobileNet model [19]. They used transfer learning method using the MobileNet model on seven skin diseases. This resulted in creating a skin disease classifier system on an android application. The authors gathered 3,406 number of images and it was observed that the classes contained unequal number of images. So it was considered as imbalanced dataset. To improve the accuracy, they used different preprocessing and sampling method on input data.

Another related work is "Mobile Application for Preliminary Diagnosis of Disease" which focuses on detecting common diseases and providing recommendations. This particular application contains an information system which can analyze the symptoms of patient's disease and can also determine preliminary diagnosis. It can also recommend doctors of a particular specialization, which actually helps the patients find their desired doctor's service which is suitable for their current condition [13].

Another paper, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques" [16] is about predicting heart diseases. A heart disease can cause a lot of suffering to a person. It can cut a person's life span in half if infected. It is a very critical challenge to predict cardiovascular disease in the world of clinical data analysis. Nowadays, machine learning is giving effective assistance in decision making and predictions which are produced from large number of data. This data are produced from large number of data. This data are provided by the health care industry. The authors observed use of machine learning techniques in recent developments in different sections of IoT(Internet of things). identifying a heart disease is not easy because of many risk factors like diabetes, abnormal pulse rate, high bp (blood pressure), high amount of cholesterol and many more. In order to find the severity of a heart disease in human body, different type of techniques in the field of data mining along with neural network has been waged. K-nearest neighbour (KNN), Decision tree, Genetic algorithm, Naive bias algorithms can classify the severity of the infected heart disease. The nature of heart disease is complex so it must be handled with great care, otherwise it may be the cause of premature death. A good method to investigate data and predict heart disease is data mining with classification. This paper introduces computer aided decision support system

in the world of research and medicine. The authors proposed to diagnose heart disease by applying genetic algorithm (GA). This method uses tournament selection, mutation and crossover; techniques provided by GA to construct newly proposed fitness function. Cleveland dataset collected from UCI machine learning repository is used for the experimentation. They also generated some rules of "Particle swarm optimization" (PSO) for heart disease. These rules along with encoding techniques resulted an improvement in the overall accuracy. Symptoms, pulse rate, sex, age and many other parameters are used to predict heart disease [16].

"Disease Prediction in Data Mining Technique – A Survey" focuses on comparing multiple techniques in detecting and predicting various disease. Data mining shifts through anormous amount of data to gain valuable information. Association rules, classification, sequential patterns, prediction, clustering, etc are among some of the methods data mining offers. This methods can be used in many causes and applications. Data mining plays a role of great importance and significance by reducing the number of tests required for the patient to detect a certain disease. This also effects performance and time. But like other methods, This one also has pros and cons. In this paper, the authors analyzed various types of data mining techniques to predict various kind of diseases. The paper reviewed research papers gthat mainly focuses on predicting diabetes, breast cancer and heart disease. Applying data mining in medical research requires a hypothesis. Later, the results are adjusted according to the hypothesis. Although, this technique is unlike normal data mining method where the process starts with datasets relying on no particular hypothesis. A traditional data mining method mainly concentrates on patterns and trends consisting in a dataset, which are unlikely to be confronted in medical data mining purpose. Clinical decisions rests in the hands of the doctors advice and suggetion. Excessive medical cost, unwanted bias, errors can cause low quality of service. Generating a knowledge base environment can be achieved by applying data mining [18].

This paper focuses on Identifying research contributions which consists of application of multiple supervised learning algorithms to detect a certain disease. This paper gives us a summary of the corresponding performance evaluation of different versions of supervised learning algorithms which are used for predicting disease. This valuable piece of information can help fellow researchers to decide which supervised machine learning algorithms should they choose to use for their purpose. In the recent years, the data science researchers have given a great amount of attention to disease prediction along with medical information. The main concept of this paper is to shed light on the performance comparison on different versions of supervised learning algorithms. The availability of huge health databases, and wide spread use of computer based technology has encouraged data science researchers to study more on the health service field [18].

This paper introduced a new approach which relies on coactive neuro fuzzy inference system (CANFIS). This method is used for predicting heart disease. The proposed model was integreted with GA (genetic algorithm) combined with fuzzy logic approach and neural network adaption capacities to diagnose a disease. The model was proven to be of great importance and showed potentiality in predicting the heart disease. The models performance was evaluated based on classification accuracy

and training performance. Fuzzy inference systems combines the evaluative nature of rules with capabilities of neural networks. To quickly and accurately approximate complex functions the CANFIS model integrates a modular network consisting of fuzzy inputs. When the underlying function is variable, these type of networks solve problems more swiftly and correctly. Use of genetic algorithm to search for the best optimal number of MF for inputs were applied to improve the learning rate of the model. Learning rate, momentum coefficient were used for this purpose. This method also allows to select the most relevant features from the training data. This results in producing a smaller network which is less complicated; that contains the ability to generate freshly uploaded data, because of removing redundant data. To find the best solution, the GA combines mutation operators, crossovers, selection by searching and it keeps up the process until the specified goal is achieved. This solution is denoted as chromosome; it is a collection of genes. Genes act as CANFIS parameters, which are to be optimized. In search of the best network parameters, the GA evolves the population corresponding to multiple generations. It does so by creating an population and evaluating it by training network for chromosomes [5].

This particular paper proposes a system for early disease prediction. The proposed model consists of isolation forest which uses outlier detection method for removing outlier data. To balance distribution of data and synthetic minority oversampling method is used along with ensemble approach to predict the disease. To extract the most significant risk factors and to build the proposed model, four datasets were used. The authors also applied the model in an mobile application for practical usage. The mobile app collects the risk factor data to send it to a remote server. With this approach, the user's current condition will be diagnosed with the proposed model. After that, the prediction result will be back to the mobile app so that the user may know and take necessary steps for prevention about any disease occurrence at an early stage [14].

# Chapter 3

## Methodology

We are working on an application which can predict diseases given certain symptoms and can also propose more symptoms as input. This application has two main features, one is only for skin disease prediction and the other one is for common diseases. For this purpose, we have used two algorithms. In this portion, we will try to discuss about this algorithms and how they work.

### 3.1 Proposed System Diagram

In this section, **Figure 3.1** tries to represent the project through a diagram. As the diagram shows,

- After starting the application in their mobile, the users will be presented with two options.
- They can select “common disease” or “skin disease” according to their choice.
- If they select “common disease”, the application would request for the patient’s symptoms along with age and gender to be entered as input.
- If any invalid symptom is given by the user, then the application will go to the previous process and request a valid symptom.
- After entering a symptom, the application will suggest some more symptoms according to relevance by collaborative filtering.
- After getting all the symptom information from the user, the application will analyze the data through SVM and as a result show users the predicted disease with it’s cure and full details.
- On the other hand, if skin disease option is selected, the application will offer the user three options.
- They can take a photo manually for scanning any skin abnormality, they can select a previously taken skin disease photo from their phone or they can live stream and see the prediction in real-time.
- After the above step, the image is preprocessed for better detection. Then the preprocessed image will be segmented according to it’s luminance, color and texture.



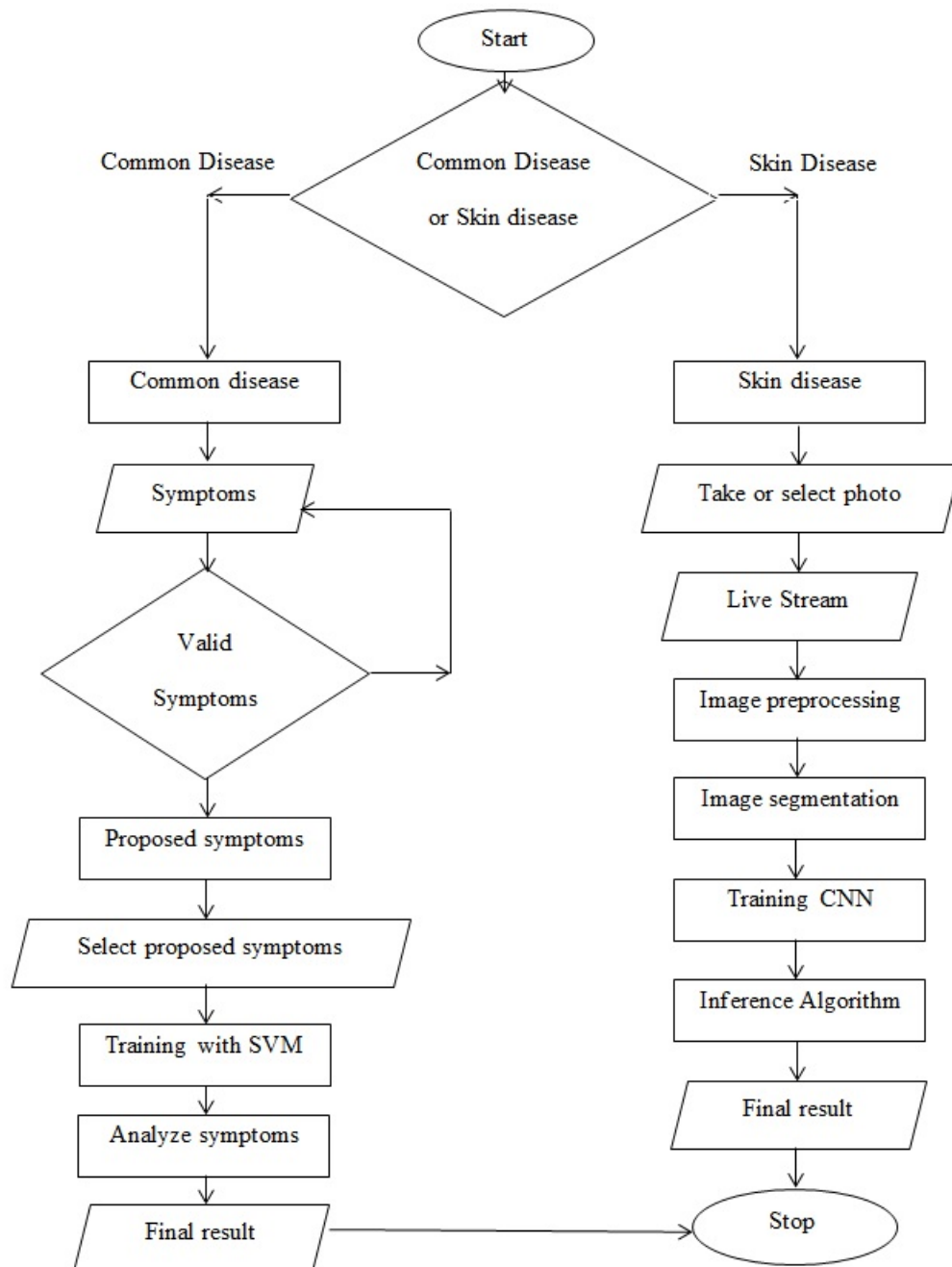


Figure 3.1: Proposed System Model

- Then the image will be trained by CNN algorithm for the skin disease prediction.
- Next an inference algorithm is applied where it takes the help of tensorflow to utilize the MobileNet model into android application properly.
- Finally the predicted disease is shown to the user.

### 3.1.1 Dataset for Common Disease

The dataset used for common disease prediction is given below :  
This dataset has 133 total columns, 132 parameters on which 42 different diseases experienced by patients can be predicted.

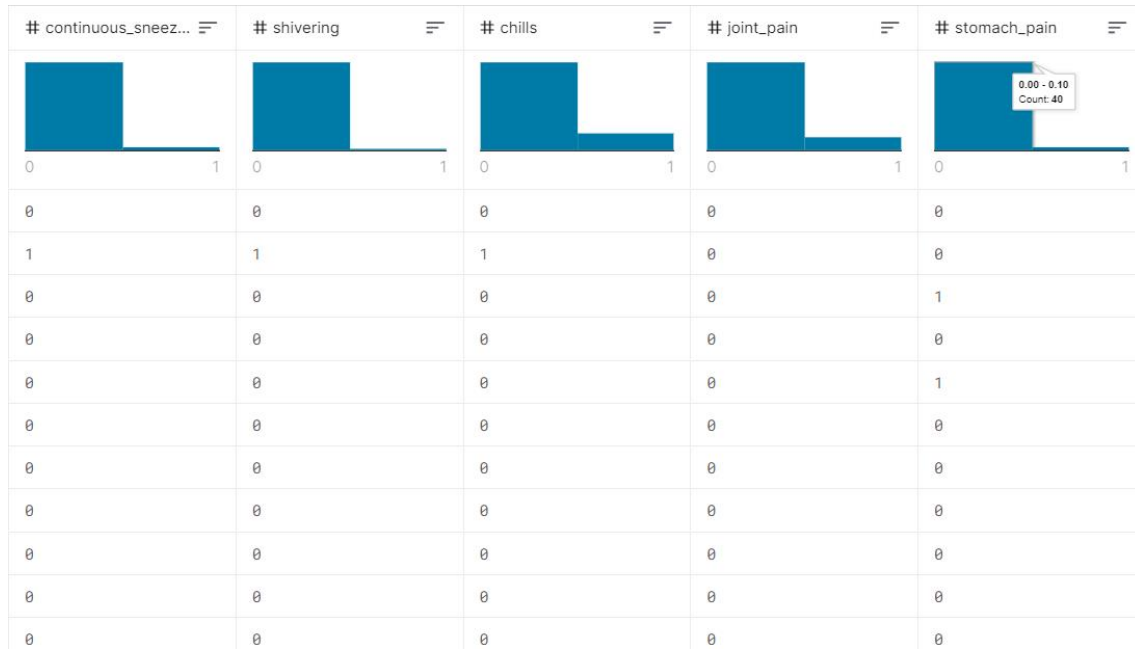


Figure 3.2: Dataset for common disease

Every symptoms probability of occurrence for several diseases are not the same. It varies from disease to disease. Using SVM, we classified the probability of the diseases.

### 3.1.2 Data Visualization

In this figure we have used heatmap for data visualization. Heatmap is a graphical description by which each value of a matrix is denoted with colors. Heatmap helps to visualize values between dimensions of a matrix. This helps searching for patterns and represents outlook of depth.

In **Figure 3.3**, both axis contain the parameters of the dataset. The values represent the correlation between the parameters, in this case the diseases. The bar at the right side represents the correlation between diseases with colors. The blue boxes containing value 1 represents high possibility of close correlation among diseases.

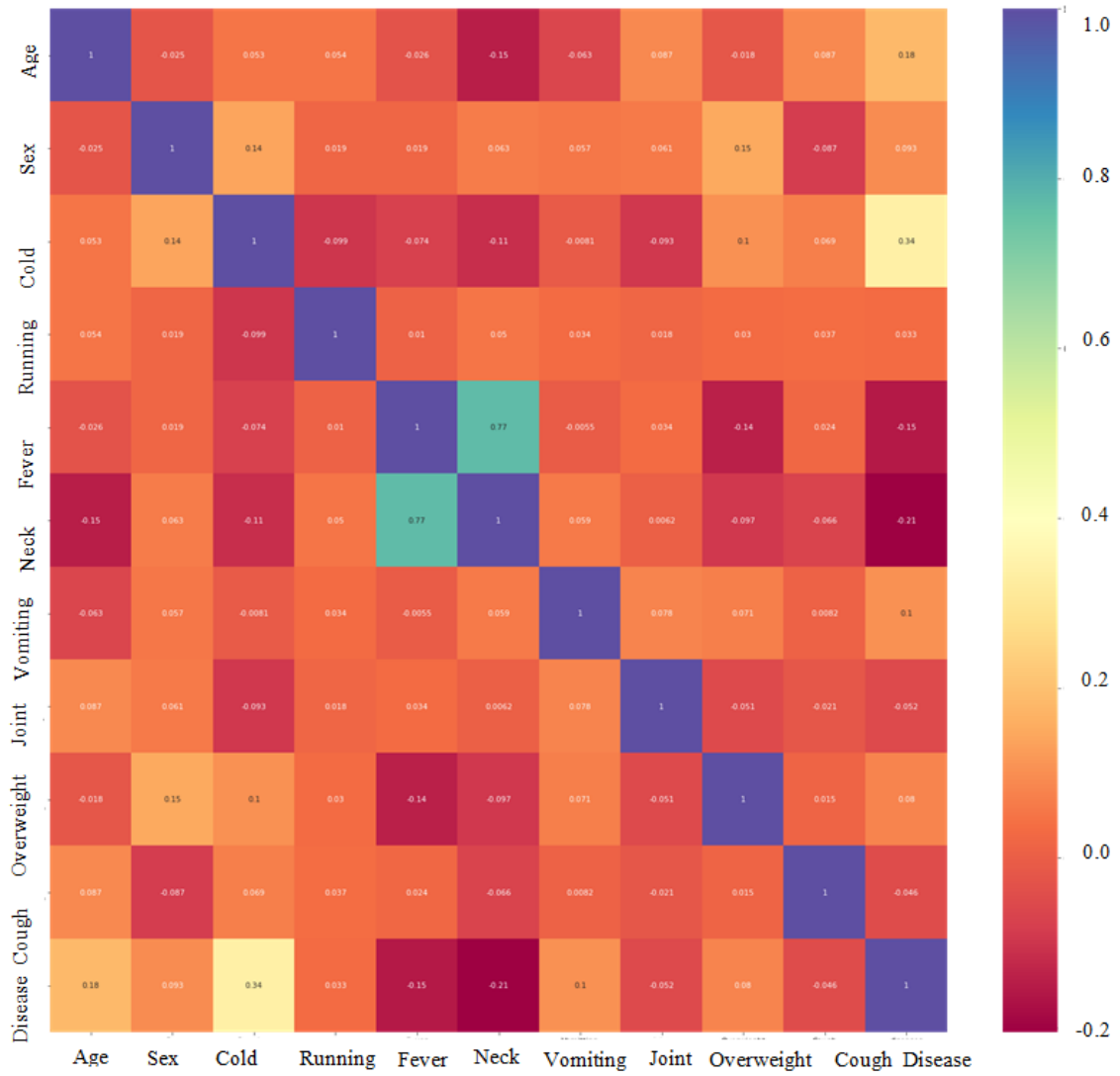


Figure 3.3: Data visualization of common disease prediction

Yellow boxes containing value 0.34 represent moderate correlation between diseases and so on.

### 3.1.3 Implementation Using Web Scraping

Web scrapping is the process which extracts and combines data collected from the web. [9]. We used webscraping to extract data from a web API for our prediction results.

Parameter	Type	Values
Token	String	Security token received from <a href="https://authservice.priaid.ch/login">https://authservice.priaid.ch/login</a>
Format (optional)	String	json.xml

Table 3.1: Common parameters

### 3.1.4 Data Preprocessing

Machine learning's rate of succes on any task can be influenced by various factors. Good quality of data and proper representation is top prioroty. The training phase of the algorithm turns out to be worse and complex if the dataset consists of redundant data. Data preprocessing is a great solution to this kind of issues for it's feature extraction, transformation, data cleaning, normalization properties. This method produces the final training set of data [4]. From the data preprocessing step we can extract the data which is needed for the prediction process.

### 3.1.5 Intelligent Smart Symptom Prediction

The necessity of effective recommender system increases every day rapidly as more and more users are joining in the internet and also because the internet is expanding massively in all sectors of life. [8].

Collaborative filtering technique performs collaboration among data instances, sources, viewpoints to filter information. This method is very popular and widely used for analyzing any user's interest on a subject depending on the data of other similar user's view.

### 3.1.6 User Clustering

User clustering methods identifies groups of consumers who have interest on similar subjects or topics. Predictions for a user can be created by producing the average of opinions of other users in a cluster; this step occurs after the cluster is created.

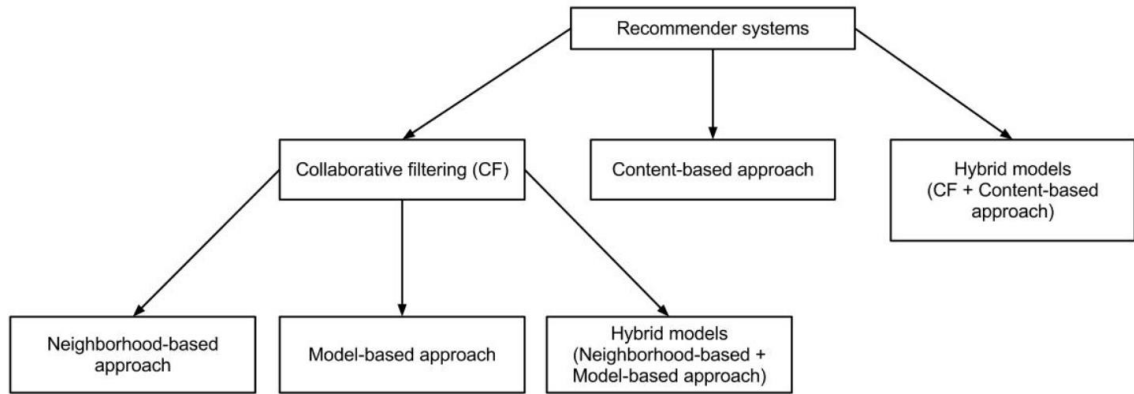


Figure 3.4: Collaborative Filtering

Representing users containing partial participation in multiple clusters are seen in some clustering methods. The clustering performance depends on the size of the target group for analyzing. **Figure 3.4** shows the concept of this method where collaborative filtering system users are separated by user clustering. The algorithm may generate fixed or variable sized partition depending on few similarity limits.

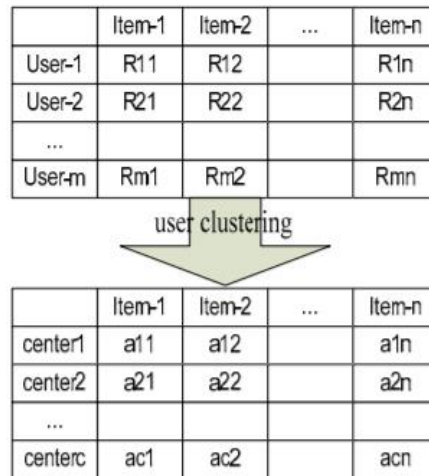


Figure 3.5: User Clustering

Here  $R_{ij}$  is the rating of user  $i$  respect to item  $j$ ,  $\bar{r}_i$  denotes average rating of the center of user  $i$  to item  $j$ ,  $n$  represents number of items,  $m$  represents the number of all users and user centers are denoted by  $c$ .

### 3.1.7 Smoothing

K means clustering algorithm is used in this purpose to form clustering centers by clustering users into few groups.

The algorithm is as follows:

---

**Algorithm 1** Smoothing Algorithm

---

```

1: Input the clustering number k, user-item grading matrix
2: Give the smoothing grading as output
3: Start
4: Select set of user  $Y=Y_1, Y_2, \dots, Y_z$ ;
5: Select set of item  $E=E_1, E_2, \dots, E_v$ ;
6: Select top k rated users for clustering
7:  $CL = CL_1, CL_2, \dots, CL_k$ ;
8: Clustering center k null if  $d = d_1, d_2, \dots, d_k$ ;
9: do
10: for each user  $Y_i \in Y$ 
11:   for each cluster center  $CL_j \in CL$ 
12:     calculate the  $\text{sim}(Y_i, CL_j)$ ;
13:   end for
14:  $\text{sim}(Y_i, CL_m) = \max \text{sim}(Y_i, CL_1), \text{sim}(Y_i, CL_2), \dots, \text{sim}(Y_i, CL_k)$ ;
15:  $cp = cp \cup Y_i$ 
16: end for
17: for each cluster  $cp \in c$ 
18:   for each user  $Y_j \in Y$ 
19:      $CL_i = \text{mean}(d_i, Y_j)$ ;
20:   end for
21: end for
22: while C is unchanged
23: Stop =0

```

---

### 3.1.8 New Ratings

Sparsity of data is one of the major problems faced in collaborative filtering. Plain use of item clusters for prediction implementation was made so that unfilled values which are part of user-item rating data can be predicted. Derived from the user-item clustering result, prediction mechanism is applied on free rating data as given below,

$$R_{ij} = \begin{cases} R_{ij} & \text{if user } i \text{ rate the item } j \\ c_j & \text{else} \end{cases} \quad (3.1)$$

### 3.1.9 The Dense User-Item Matrix

Dense ratings from the users given to the items were obtained after applying the clustering algorithm. By this procedure, we get the dense user item matrix from the original scattered user item matrix.

### 3.1.10 Item Clustering

Item clustering methods are used to identify groups of items which has identical ratings. Predictions for a particular item can be made after the clusters are ready. It is done by taking the average of opinions of the other items in the same cluster. The prediction then is measured with degree of participation. .

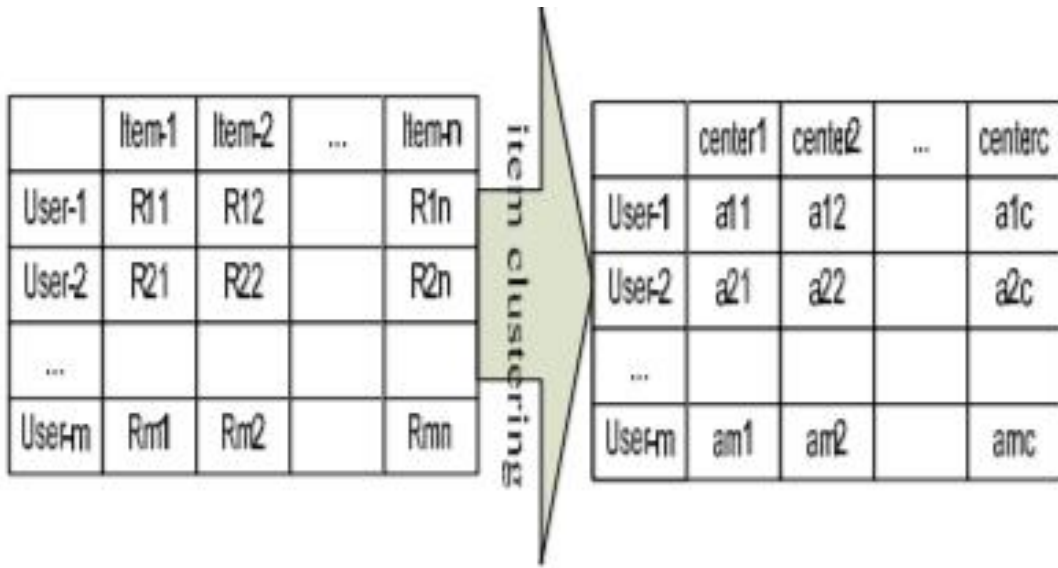


Figure 3.6: Item Clustering

### 3.1.11 Algorithm

There are several algorithms that are applicable for item clustering. The k means clustering algorithm is used for this task. Here, k is an input that represents the number of clusters we want. At first, for the k unique clusters, k items are taken as cluster centers. Each of the remaining items is then compared to the closest center. Then the cluster centers are re-evaluated depending on cluster centers which were created in the last pass and the cluster integration is evaluated once more. The algorithm :

The following is pearson's correlation formula, which measures the linear correlation among two vectors of ratings. This rating is represented as the target item  $q$  and the remaining item  $e$ .

---

**Algorithm 2** Item Clustering

---

```
1: Input the clustering number k, user-item grading matrix
2: Give the item center matrix as output
3: Start
4: Select set of user Y=Y1,Y2, ..., Yz;
5: Select set of item H = H1, H2, ..., Hv;
6: Select top k rated users for clustering
7: CH=CH1, CH2, ..., CHk;
8: Clustering center k null if d = d1, d2, ..., dk;
9: do
10: for each item
11:  $Hi \in H$ 
12: for each cluster center
13:  $CHj \in CH$ 
14: evaluate the sim(Hi, CHj);
15: end for
16:  $\text{sim}(Hi, CHx) = \max(\text{sim}(Hi, CH1), \text{sim}(Hi, CH2), \dots,$ 
17:  $\text{sim}(Hi, CHk);$ 
18:  $dx = dx \in Hi$ 
19: end for
20: while CY and d are unchanged
21: for each cluster
22:  $di \in d$ 
23: for each user
24:  $Hj \in H$ 
25:  $CHi = \text{mean}(di, Hj);$ 
26: end for
27: end for
28: Stop =0
```

---

$$\text{sim}(q, e) = \frac{\sum_{k=1}^j (R_{ik} - A_e)(R_{ek} - A_q)}{\sqrt{\sum_{k=1}^j (R_{ik} - A_q)^2 \sum_{k=1}^j (R_{ek} - A_e)^2}} \quad (3.2)$$

Here,  $R_{iq}$  is the rating of our item  $q$  by user  $k$ .  $R_{ie}$  denotes rating of the last item  $e$  by user  $k$ .  $A_t$  is the mean rating of the target item  $q$  for every co-rated users,  $A_e$  is the mean rating of the last item  $e$  for every co-rated users, and  $j$  represents all rating users respect to the item  $q$  and item  $e$ .

### 3.1.12 Selecting Clustering Centers

Searching neighbors of the target item is a necessary step in collaborative filtering. Memory based collaborative filtering suffers from bad scalability if large amount of users and items are added in ratings database. The items center is obtained by clustering it. This center represents average rating of all items present in the cluster. After evaluating similarity between item and centers using pearson's correlation



formula, the items of the most similar centers are selected.

### 3.1.13 Selecting Neighbors

Calculation of the similarity among the target item and selected clustering center items needs to be done after selecting the target item's nearest cluster centers. Based on the cosine measure, the top K most similar items were selected. The formula below states the angle between two vectors of rating as target item  $t$  and remaining item  $r$ .

$$sim(h, e) = \frac{\sum_{i=1}^n (R_{ih} R_{ie})}{\sqrt{\sum_{i=1}^n R_{ih}^2 \sum_{i=1}^n R_{ie}^2}} \quad (3.3)$$

Here  $R_{in}$  denotes the target item  $t$ 's rating by user  $i$ ,  $R_{ie}$  denotes the remaining item  $e$ 's rating by user  $i$ , the number of total rating  $n$ , to the item  $h$  and item  $e$ .

### 3.1.14 Producing Recommendations

Calculation of the weighted mean of the neighbor's ratings, weighted by each others similarity to the target item is possible after getting the membership of item. Target user  $j$ 's rating to the target item  $h$  is given below,

$$Q_{jh} = \frac{\sum_{i=1}^n R_{ji} \times sim(h, i)}{\sum_{i=1}^n sim(h, i)} \quad (3.4)$$

Here  $R_{ij}$  denotes the rating of target user  $j$  respect to neighbor item  $i$ ,  $sim(h, i)$  represents similarity between target item  $h$  and the neighbor user  $i$  for every relevant items. Lastly,  $n$  is the total number of rating users respect to item  $h$  and item  $v$  [6]. This option is given to the application so that users don't have to input symptoms manually every time. It requests for proposed symptoms using the parameters below. It distinguishes a symptoms by it's corresponding SL no. Symptoms with similar effects are proposed to the user for multiple selection.

Source	<a href="https://healthservice.priaid.ch/symptoms/proposed">https://healthservice.priaid.ch/symptoms/proposed</a>
Parameter	key,symptoms,yob,gender
Access	Token

Parameter	Type	Values
Symptom	int[] array encoded by json	Array contains sorted <i>SLno.</i> in json code, for example [3,4,5]
Gender	String	male, female
yob	int	

Table 3.2: Proposed Symptoms

After executing this steps, the result below is generated :

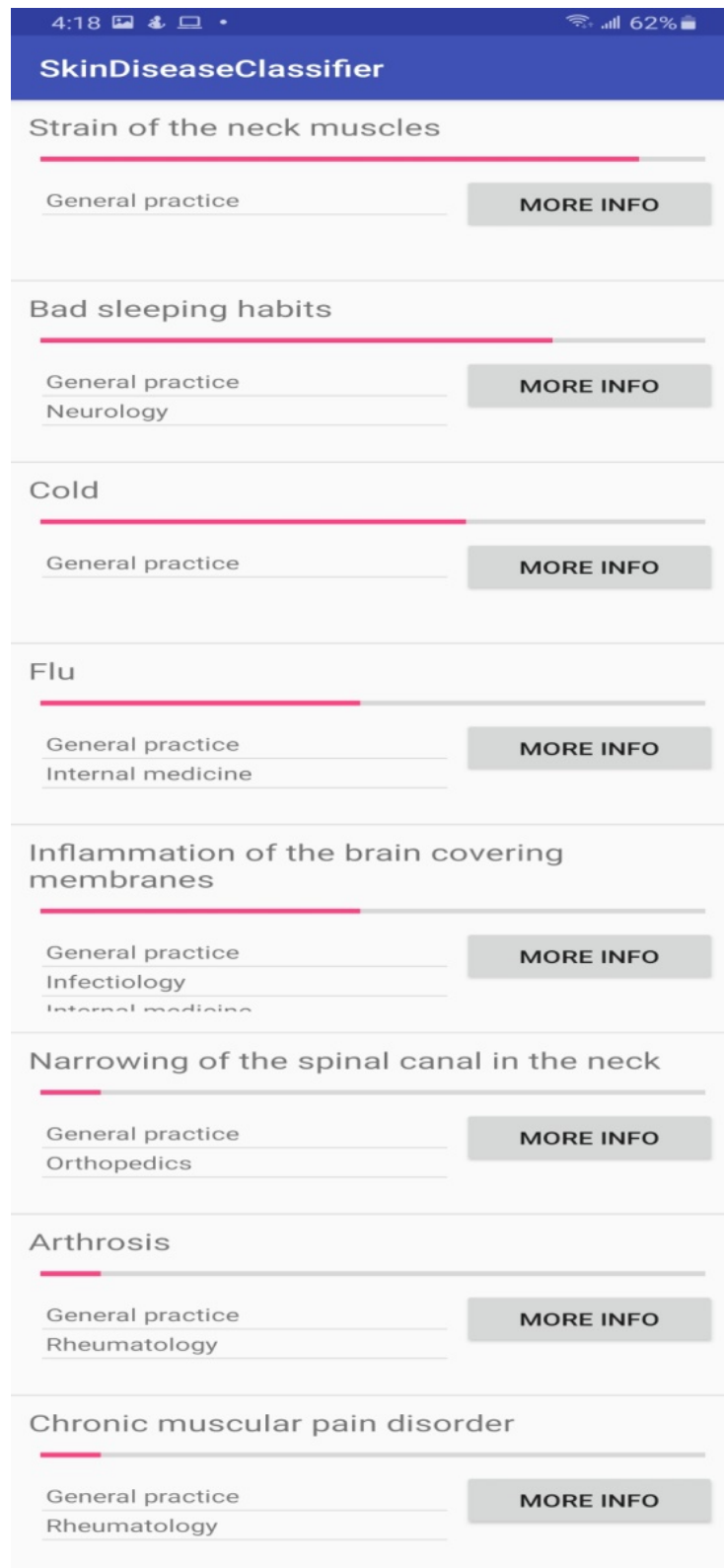


Figure 3.7: Result for common disease prediction

Finally, this is the prediction result generated by our application. Results will depend on the given symptoms. If there are several symptoms, then the prediction become more accurate [6].

### 3.1.15 Support Vector Machine

In case of solving classification problems containing huge data, support vector machine might be one of the best choices. In a big data environment, it makes multidomain tasks relatively easy. But it is computationally costly and difficult to execute [12].

SVM models work by dividing data into different classes consisting in a hyperplane using decision boundary. The hyperplane acts in an iterative manner to avoid errors. SVM's main objective is to separate the classes gained from the dataset for the purpose of finding maximum marginal hyperplane.

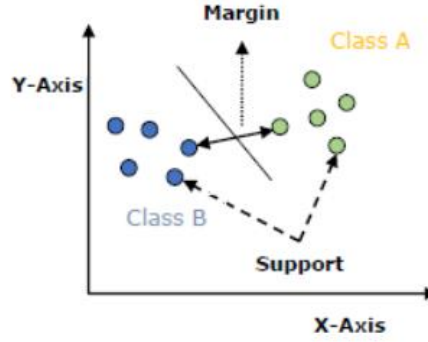


Figure 3.8: How SVM works

Suppose we have dataset and we want SVM to classify male and female genders. This action can be performed by first studying relevant characteristics of both genders. For unseen data, it needs to label them according to their gender. The characteristics that help SVM classify data are called features. Defining a feature in real space represents its domain. Mapping a function  $b = f(a)$  gives us the range and co-domain. Basically a data is presented to us to be divided by SVM. These data are denoted as unique point in a feature space and every single point is denoted by  $a$  which are denoted as feature vectors.

$$a \in R^F \quad (3.5)$$

Further mapping points into feature space  $a$ ,

$$\phi(a) \in R^N \quad (3.6)$$

We map the transformed basis vector  $\phi(a)$  for the transformed feature space,

$$\phi(a) : R^F = R^M \quad (3.7)$$

### 3.1.16 Decision Boundary

SVM includes different kinds of variations like adaptive margin classifier, kernel trick, soft margin classifier etc. A constructive hyperplane's job is to separate classes. This is done by putting margins between each class. Even if classes contain noise and are overlapped even a slightest bit, they have similar properties [2]. On the other hand, the decision boundary acts as the main divider to distribute features to their relevant classes.

### 3.1.17 Equation of Hyperplane

In SVM, the support vectors determine the classification hyperplane. Furthermore, the classifier is not affected by other samples [15].

The equation of a straight line with slope  $p$  and intercept  $i$  is :  $p + i = 0$ .

The hyperplane equation is given below

$$H : y^D(a) + d = 0 \quad (3.8)$$

In this equation,  $d$  denotes intercept and also bias value of the equation.

### 3.1.18 Distance Measure

Based on previous topics, we now have a clear idea of fitting a separating line among points, data points etc. While we fit the separating line, we except it to avoid misclassification, errors while isolating the data points for the feasible way for classification. For this, we should have a clear idea of the distance among data points and the divider. The distance of a line  $dx + ey + f = 0$  from an input point  $(p_0, q_0)$  is given by M [23].

In the same way, the distance of hyperplane equation  $y^D\phi(a) + d = 0$  from an input point vector  $\phi(a_0)$  and it is as follows :

$$m_H(\phi(a_0)) = \frac{|y^D(\phi(a_0)) + d|}{\|y\|_2} \quad (3.9)$$

$\|y\|$  denotes the euclidean norm for length of  $y$ ,

$$\|y\|_2 = \sqrt{y_1^2 + y_2^2 + y_3^2 + \dots y_n^2} \quad (3.10)$$

### 3.1.19 Optimal Hyperplane

If a linear discriminant function, whose sign is similar to the class of every training examples, the training set is considered as linearly separable. In this stage, the training set are said to be found among a huge number of separating hyperplane. Equation of optimum hyperplane is given below :

$$\min Q(a, d) = \frac{1}{2}y^2 \quad (3.11)$$

$$\text{subject to } \forall \alpha_m \geq 0, \sum_v y_v x_v = 0 \quad (3.12)$$

Because of the constraints being quite difficult, it is hard to solve the problem directly. The given approach helps to the solution of the dual problem,

$$\max M(\alpha) = \sum_{v=1}^h \alpha_v - \frac{1}{2} \sum_{v,j=1}^h w_v x_v w_j x_j \phi(a_v)^D \phi(a_j) \quad (3.13)$$

$$\text{subject to } \forall \alpha_v \geq 0, \sum_v w_v \alpha_v = 0 \quad (3.14)$$

### 3.1.20 SVM Representation

The QP formulation in case of SVM classification is given below :

SVM classification :

$$\min_{e, \xi_v} ||e||_k^2 + G \sum_{v=1}^l \xi_v w_v e(a_v) \geq 1 - \xi_v, \text{ for all } v \xi_v \geq 0 \quad (3.15)$$

For Dual formulation:

$$\min_{x_v} \sum_{v=1}^l \alpha_v - \frac{1}{2} \sum_{v=1}^l \sum_{s=1}^l \alpha_v \alpha_s y_v y_s K(a_v, a_s) \quad 0 \leq \alpha_v \leq G, \text{ for all } v; \sum_{v=1}^l \alpha_v w_v = 0 \quad (3.16)$$

The variables assess the mistake made at point  $(a_v, w_v)$ . Large number of training points makes the training procedure of SVM complex.

### 3.1.21 Kernel Trick

Kernel support vector machine is a great tool to use in non-linear classification. This classification can be applied in kernel space based on linear discriminant function. Although SVM is a good choice for applications consisting of high-dimensional space, low-dimensional spaces require kernel SVM [11].

It uses a method called kernel trick to get the ability of working in the input space rather than dealing with high-dimensional kernel space. It's popularity has spread it's use among other pattern recognition and machine learning tasks.

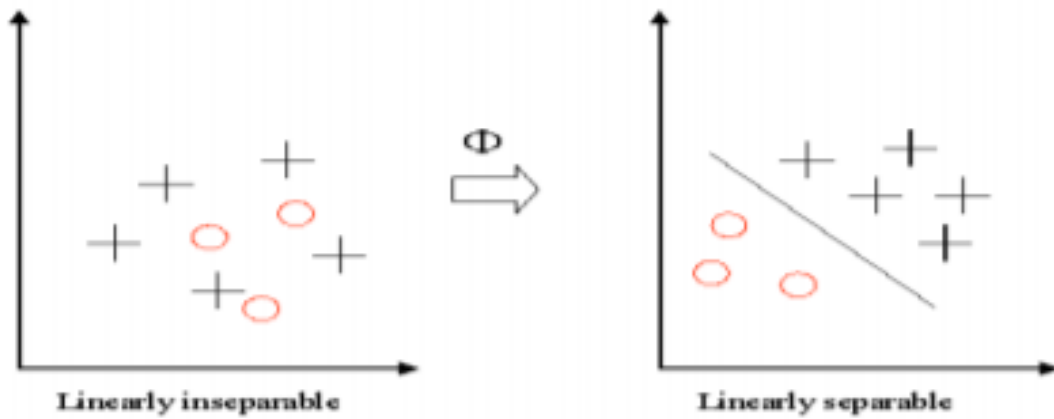


Figure 3.9: Kernel Trick

The kernel defines the mapping as :

$$K(a, b) = \phi(a) \cdot \phi(b) \quad (3.17)$$

### 3.1.22 Kernel Functions

The kernel function's motive is to grant the ability to perform operations in the input space instead of infinite dimensional feature space. The functions are required to perform mapping on the attributes of the currently working input space in the feature space. It also plays a vital role on SVM performance and is based on reproducing Kernel Hilbert Spaces.

$$K(a, a') = (\phi(a), \phi(a')) \quad (3.18)$$

Mercer's condition could be achieved if  $K$  is a symmetric positive definite function

$$K(a, a') = \sum_n^{\infty} x_n \phi_n(x) \phi_n(a'), \quad x_n \geq 0, \quad (3.19)$$

$$\iint K(a, a') y(a) y(a') da da' \geq 0, \quad y \in L_2 \quad (3.20)$$

After this, the kernel is able to produce a correct inner product in the feature space. The feature space consists of linearly separable training set. Variations of kernel function are given below :

**Polynomial** : Polynomial mapping is used for non-linear modeling. The second kernel avoids problems and is therefore, more preferable.

$$K(a, a') = (a, a')^d \quad (3.21)$$

$$K(a, a') = ((a, a') + 1)^d \quad (3.22)$$

This form of function uses Gaussian formulation :

$$K(a, a') = \exp\left(-\frac{\|a - a'\|^2}{2\sigma^2}\right) \quad (3.23)$$

**Exponential Radial Basis Function**: This functions are preferable in a situation where discontinuities are acceptable. It produces linear solutions one by one.

$$K(a, a') = \exp\left(-\frac{\|a - a'\|}{2\sigma^2}\right) \quad (3.24)$$

**Multi-Layer Perceptron**: This function consists of a single hidden layer along with a valid kernel representation :

$$K(a, a') = \tanh(\rho(a, a') + \varrho) \quad (3.25)$$

Fourier, spline, additive kernels, B-splines tensor products are also part of this clustering[3].

## 3.2 Skin Disease Prediction

Skin disease can occur for various skin abnormalities. The surroundings around us can cause skin disease and infections too. Hidden bacteria, fungus forming over the skin, allergic reactions are part of skin abnormalities that causes skin disease [24]. Environmental pollution, ozone layer, sun burn are also responsible for causing

skin disease. There are many classification algorithm which are developed or under-development to detect and predict skin disease [20]. Machine learning algorithms are also designed to predict skin disease at an early stage. For this, the disease attributes needs to be studied thoroughly [21].

Tensorflow is a public software library accesible to anyone for machine learning applications. It mainly focuses on inference and training deep neural networks.



Figure 3.10: Different type of skin disease

### 3.2.1 Dataset

The dataset we used for this purpose consists of publicly accessible medical skin disease photos, manually taken photos and photos from dermatology repositories. The consisting images of the dermatology repositories are examined and valited by professional dermetologists. The dataset consists of herpes, acne, warts, eczema, actinic keratosis and cellulitis impetigo.

### 3.2.2 Data Visualization

In **Figure 3.11**, we have used a Piechart for data visualization. A piechart represents the data consisting in a dataset by dividing them into distinct classes and putting them in a circle chart while dividing each class with radial slices. The classes' size is proportional to the amount of data it has in the original dataset. This figure represents the percentage of skin diseases present in our preferred dataset. We can see that the percentage of “acne” is 18.8 percent, “actinic keratosis” is 15.4 percent, ”eczema” is 12.5 percent and so on. With this we can see a clear representation of the present skin diseases in the dataset.

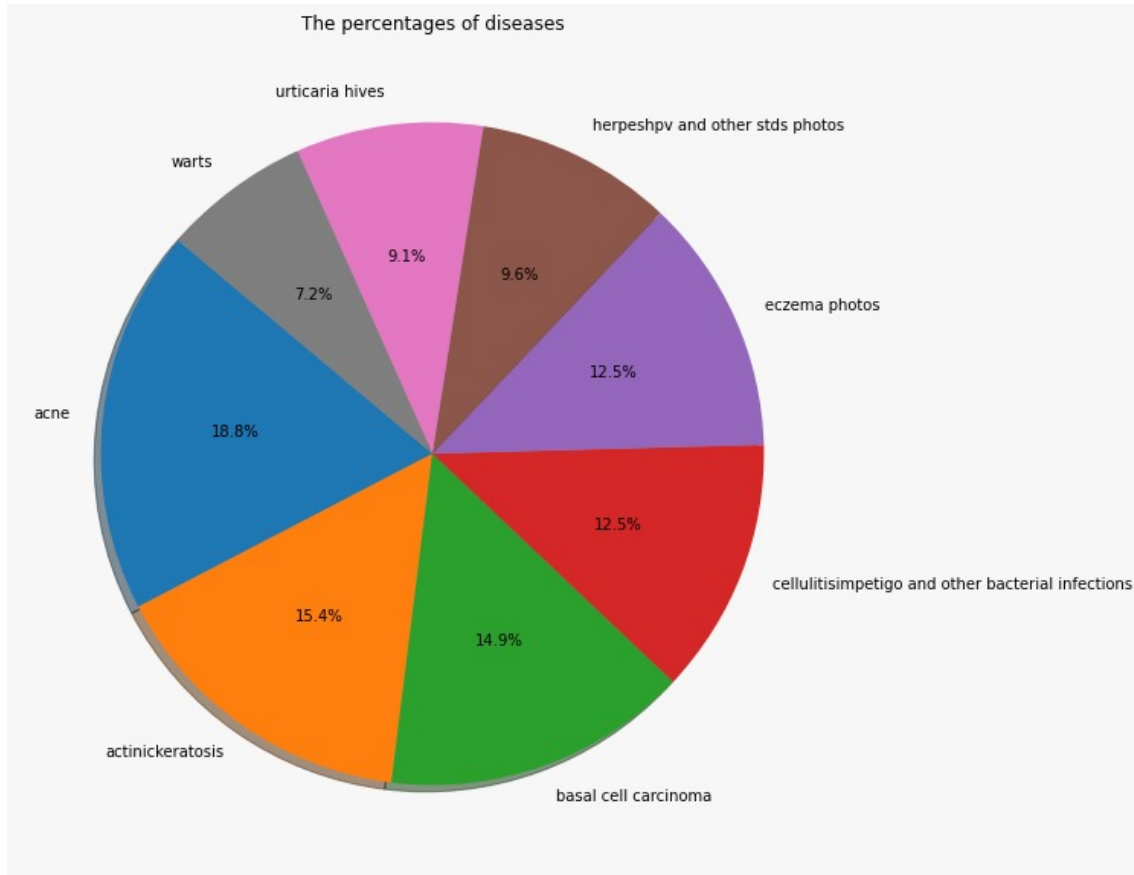


Figure 3.11: Data visualization for skin disease

### 3.2.3 MobileNet Description

CNN's are used frequently in computer vision operations. This networks have huge number of parameters and operations. For this reason, it is very difficult to apply them into mobile devices. And this is where MobileNet comes in. It is a version of CNN with less operators and parameters and a great option for using in mobile applications [22].

It is also a computer vision model developed by tensorflow. MobileNet uses depth wise separable convolution to run it's operations. It has less amount of parameters compared to regular convolutions with same depth. Depthwise separable convolution consists of two main operators :

- Depthwise convolution.
- Pointwise convolution.

MobileNet is a good option to train small and efficient classifiers.

**Spatial separable convolutions** Spatial Separable convolutions work with the image's height and width. It works by dividing kernels into smaller kernels. Lets say, a 3\*3 kernel is given and it can be divided as 3\*1 and 1\*3 by convolution.

To make the network efficient, we can divide the multiplications among convolutions. Furthurmore, this step also reduces computational complexity.



$$\begin{vmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{vmatrix} = \begin{vmatrix} 3 \\ 4 \\ 5 \end{vmatrix} * \begin{vmatrix} 1 & 2 & 3 \end{vmatrix}$$

$$\begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 \\ 2 \\ 1 \end{vmatrix} * \begin{vmatrix} -1 & 0 & 1 \end{vmatrix}$$

But still, not all kernels are separable despite of achieving less computation power. Spatial separable convolutions are not effective in this scenerio.

### Depthwise Seperable Convolutions

The above mentioned issue can be resolved by using Depthwise separable convolution. MobileNet architecture is based on this method. It works with kernels that can't be divided and also contains depth dimensions. It consists of a 1\*1 convolution called pointwise convolution. It can factorize a normal convolution into depthwise convolution because of it's factorizing properties. In order to filter and combine the kernel, depth wise convolution divides it into two seperable kernels. Here pointwise convolutions play the role of combining whereas depthwise convolution does the filtering operation.

Total computation for a standard convolution is :  $R_K.R_K.M.N.R_F.R_F$ , ;  $M$  denotes the number of input channels,  $N$  represents number of output channel,  $R_K$  is the size of the kernel and  $R_F$  denotes size of the feature map. The resulting computation we get by combining and filtering is presented below :

$$\frac{R_K.R_K.M.R_F.R_F}{R_K.R_K.M.N.R_F.R_F} \quad (3.26)$$

which is equivalent to

$$\frac{1}{N} + \frac{1}{R_k^2} \quad (3.27)$$

Which explains that the computational cost can be reduced to 8 or 9 times when  $R_K * R_K$  is 3\*3.

### 3.2.4 MobileNet Architecture

Among machine learning algorithms, embedded vision CNN's plays a crucial role in object detection alongside recognition also. MobileNet, for it's small size due to less parameters, is an efficient light-weight machine learning model for mobile applications [17].

- The first layer of MobileNet is a full convolutional layer. Except that, the other layers are depthwise separable convolutions.
- Every layer ends with a batch normalization and ReLU. But the last layer uses softmax classification and it is fully connected convolution that has no non-linearity.
- Strid convolution is used for every layer to perform down sampling.
- MobileNet has a total of 28 layers, including depthwise and pointwise convolutions.

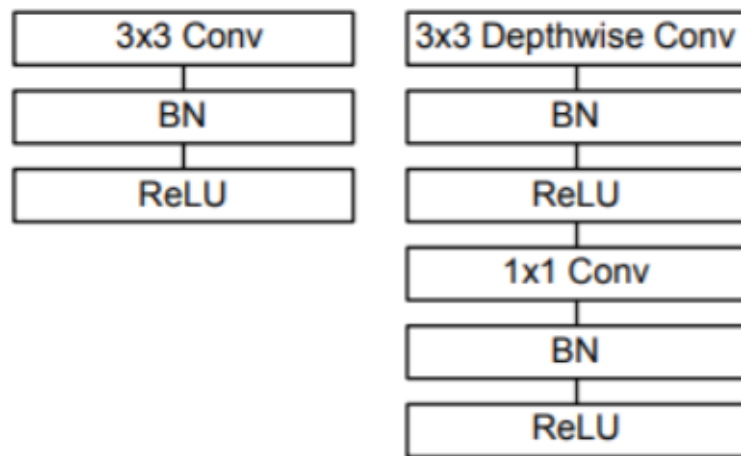


Figure 3.12: Block diagram of MobileNet

MobileNet architecture is shown below:

Sort/Stride	Refine Size	Given Shape
CoLayer / s2	3 x 3 x 3 x 32	224 x 224 x 3
CoLayer dpws / s1	3 x 3 x 3 x 32 dpws	112 x 112 x 32
CoLayer / s1	1 x 1 x 32 x 64	112 x 112 x 32
CoLayer dpws / s2	3 x 3 x 64 dpws	112 x 112 x 64
CoLayer / s1	1 x 1 x 64 x 128	56 x 56 x 64
CoLayer dpws / s1	3 x 3 x 128 dpws	56 x 56 x 128
CoLayer / s1	1 x 1 x 128 x 128	56 x 56 x 128
CoLayer dpws / s2	3 x 3 x 128 dpws	56 x 56 x 128
CoLayer / s1	1 x 1 x 128 x 256	28 x 28 x 128
CoLayer dpws / s1	3 x 3 x 256 dpws	28 x 28 x 256
CoLayer / s1	1 x 1 x 256 x 256	28 x 28 x 256
CoLayer dpws / s2	3 x 3 x 256 dpws	28 x 28 x 256
CoLayer / s1	1 x 1 x 256 x 512	14 x 14 x 256
5 x CoLayer dpws / s1	3 x 3 x 512 dpws	14 x 14 x 512
CoLayer / s1	1 x 1 x 512 x 512	14 x 14 x 512
CoLayer dpws / s2	3 x 3 x 512 dpws	14 x 14 x 512
CoLayer / s1	1 x 1 x 512 x 1024	7 x 7 x 512
CoLayer dpws / s2	3 x 3 x 1024 dpws	7 x 7 x 1024
CoLayer / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Mean Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Classifier	1 x 1 x 1000

Table 3.3: MobileNet Architecture

### Width Multiplier to achieve Thinner Models

To make MobileNet more smaller and less costly to minimize computational cost, another model needs to be constructed. In this model, a separate parameter is used and is denoted as  $\alpha$  and is known as width multiplier. It assists to make the model slimmer uniformly. The number of input channels are denoted by  $C$  and transforms into  $\alpha C$ . The number of output channels are denoted as  $H$  and transforms into  $\alpha H$ . The computational cost that we get after this procedure is presented by the equation below,

$$R_K.R_K.\alpha C.R_F.R_F + \alpha C.\alpha H.R_F.R_F \quad (3.28)$$

A width multiplier's purpose is to reduce the structure of a model which needs training from scratch. It can define a smaller model by doing so resulting better size, accuracy and latency.

### Resolution Multiplier for reduced representation

This is another parameter used for the same purpose as width multiplier and that is to reduce computational expense. It is denoted by  $\rho$  and by applying it to the input image, each layer's internal representation is reduced. The computational cost is presented by the equation given below:

$$R_K.R_K.\alpha C.\rho R_F.\rho R_F + \alpha C.\alpha H.\rho R_F.\rho R_F \quad (3.29)$$

The base MobileNet is represented by  $\rho = 1$  and  $\rho < 1$  represents the reduced computational MobileNets.

## 3.3 Image Preprocessing

Digital image processing has two very important steps which are image preprocessing and feature extraction [7].

Image preprocessing is a method that explores different types of image in order to detect certain objects or patterns to give the desired output that exists within the image. At first, the image's quality needs to be enhanced for better performance and it is done by converting it to a standard size. This helps the model for better generalization. In case of skin disease images, features like hair and pigments are filtered off for the purpose of enhancing the capability of detection.

### Image Segmentation

Image segmentation is also of great importance for medical-image applications [1]. In this process, the result of image preprocessing is divided into disjoint regions which are homogeneous and properties like texture, color, luminance are chosen for the sole purpose of making the analyzing process more meaningful.

## 3.4 Training Algorithm

Based on the given table below, we used the MobileNet CNN model for skin disease prediction. The input images were preprocessed by converting size to 224x224x3,

which is the ideal measurement of image MobileNet is capable of working with.

For fine-tuning process, the last layer of the model was retrained alongside the

Model	Weight Size	Loading Time (seconds)	Accuracy(Percent)
MobileNet	16.821MB	4.837	84.28

Table 3.4: Training MobileNet

removal of the final classification layer. The model converts the input images in a meaningful form so that the processing part does not get complicated. After that, the first layer goes to the second layer (max-pooling) till the fully connected neural network can be achieved.

### 3.5 Inference Algorithm

After completing the training phase of the model, it's weight and architecture is saved as a Keras file which contains an extension known as .h5. Then the keras file is converted to protobuf file to install the MobileNet model into the android application. The protobuf file extension is .pb. In this stage, tensorflow plays the role of succesfully loading the android application with the model. It does so by freezing the protobuf files graph and producing a text file that contains the labels of the model. A tensorflow function called "optimize for inference" is applied to use the model entirely for inference [19]. The figure below shows the necessary steps of the procedure.

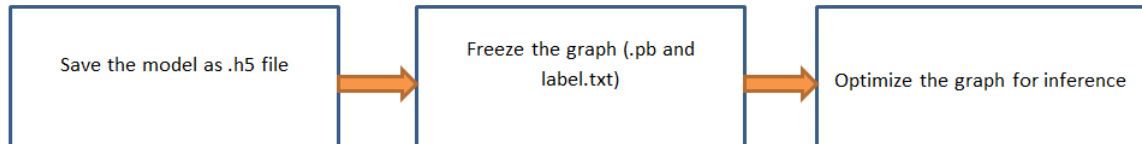


Figure 3.13: Inference algorithm

## 3.6 Final Result

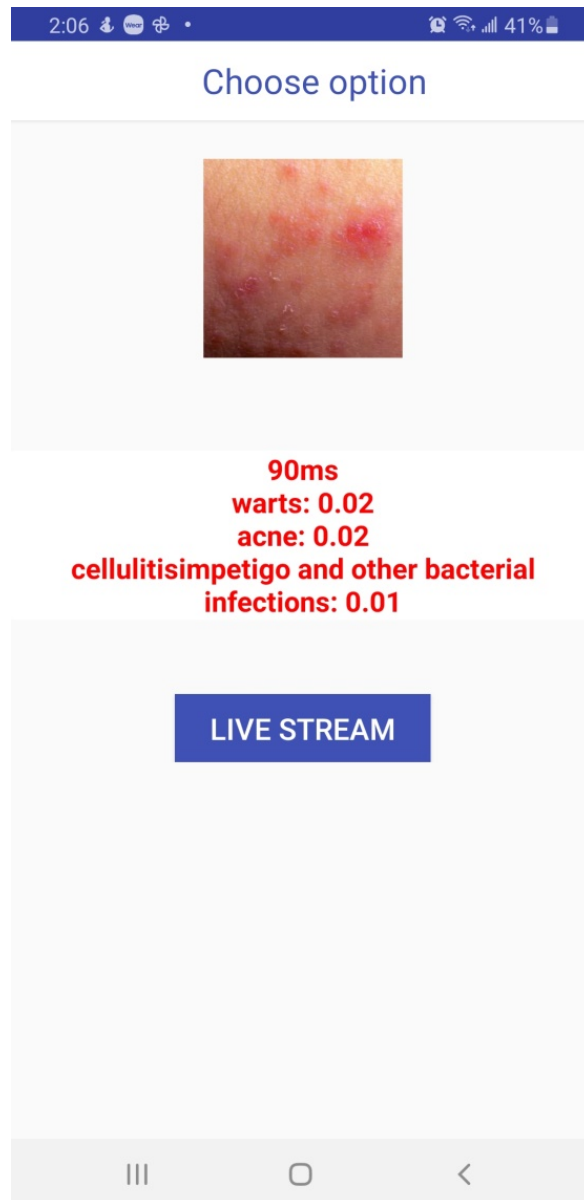


Figure 3.14: Skin disease output

# Chapter 4

## Implementation and Result Analysis

### 4.1 Implementation

This section discusses about the implementation process of the application which includes datasets, data visualization, accuracy and many more.

#### Sample response by scraping symptoms

Type of content : json for android application

```
[
{ "SL no.":288,
"Label": "Conjunctivity"
},
{
"SL no.":138,
"Label": "Yellow skin"
},
{
"SL no.":674,
"Label": "Red eyes"
},
{
"SL no.":34,
"Label": "Decreased appetite"
},
{
"SL no.":171,
"Label": "Choking while eating"
},
{
"SL no.":50,
"Label": "Pain while taking breath"
},
...
]
```

] Every symptom has a unique SL number. Like for “Yellow skin”, the SL no. is “138”. The data is coming from the api as json array. We had to scrap the data from the array using retrofit.

## Retrofit

Retrofit is a network library which helps in the webscrapping process. It works for java and android. It’s job is to retrieve and upload files, in our case json files by a webservice that is REST based.

### 4.1.1 Diagnosis

This is the main function of the android application. It works on symptoms given by the user and gives the appropriate disease prediction. It recognizes the symptoms via SL no. of the corresponding symptoms which are stored in json array.

Source	<a href="https://healthservice.priaid.ch/diagnosis">https://healthservice.priaid.ch/diagnosis</a>
Parameter	token,year of birth, gender, symptoms,
Access	Key

Parameter	Type	Values
Symptom	int[] array with json format	SL no. of corresponding symptoms sorted in json array for instance, [155,156,157]
Gender	String	male,female
<i>yob</i>	integer	

Table 4.1: Diagnosis contents

The endpoint’s output is generated as an array containing health diagnosis. The parameters are (id, name, icd, profname, icdname). The elements have the corresponding accuray presented in percentage.

### 4.1.2 Diagnosis Sample Response

Type of content : json for android application

```
]
},
{
  "Problem":{
    "SL no.": 252,
    "Label": "Bleeding internally",
    "Medname": "Internal hemorrhage"
    "bcf" : "B04.8",
    "BcfLabel" : "Ruptured blood",
    "Precision": 20
  },
}
```



### 4.1.3 Proposed Symptoms Sample Response

Type of content : json for android application

### 4.1.4 SVM Accuracy

By dividing accurately classified instances with the complete number of instances existing in the dataset, we can attain the accuracy.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

According to the equation above TP is denoted as True positive, FP is denoted as False positive, FN represents False negative, and TN denotes True negative  
**TP Rate** This term is also known as sensitivity in machine learning. For searching the high true positive rate, this ability is used.

$$TPR = \frac{TP}{TP + FN}$$

**Precision** It presents the number of units classified as faulty. It needs the number of incorrectly classified and number of correctly classified modules to generate output.

$$Precision = \frac{TP}{TP + TF}$$

**F Measure** This is the combined term of Re-call and precision. It is used to measure the classification performance of an algorithm [10].

$$F - Measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Table 4.2 represents the accuracy measure of the datasets using SVM

Algorithm	Accurately Classified Instances (%)	Inaccurately Classified Instances (%)	TP Rate	Precision	F Measure	Recall
SVM	78.52	23.68	0.763	0.820	0.213	0.173

Table 4.2: SVM Accuracy

## 4.2 Implementation of skin disease prediction

### Tensorflow

Google developed tensorflow as a deep learning frame work. It contains suitable libraries for image processing and it can control every node in a network. To achieve

the best performance, the neural networks existing weights can be adjusted as per requirement.

00000000:	18 00 00 00 54 46 4C 33 00 00 0E 00 18 00	04 00 08 00 0C 00 10 00 14 00 0E 00 00 00
0000001C:	03 00 00 00 9C 3C 02 01 0C 00 00 00 10 00	00 00 20 00 00 00 01 00 00 00 B4 33 02 01
00000038:	0F 00 00 00 54 4F 43 4F 20 43 6F 6E 76 65	72 74 65 64 2E 00 5B 00 00 00 88 33 02 01
00000054:	58 33 02 01 28 B6 01 01 74 A6 01 01 64 96	01 01 54 86 01 01 44 76 01 01 34 6E 01 01
00000070:	24 66 01 01 14 5E 01 01 04 56 01 01 F4 4D	01 01 E4 45 01 01 D4 3D 01 01 C4 35 01 01
0000008C:	B4 2D 01 01 A4 29 01 01 94 25 01 01 84 21	01 01 74 1F 01 01 64 1D 01 01 54 1B 01 01
000000A8:	44 19 01 01 34 18 01 01 A4 17 01 01 14 17	01 01 0C 17 01 01 04 17 01 01 F4 0E 01 01
000000C4:	EC 0E 01 01 E4 0E 01 01 DC 0E 01 01 D4 0E	01 01 C4 0A 01 01 BC 0A 01 01 B4 0A 01 01
000000E0:	AC 0A 01 01 A4 0A 01 01 94 C2 00 01 8C C2	00 01 7C 32 C2 00 6C 32 B2 00 64 32 B2 00
000000FC:	54 EA B1 00 4C EA B1 00 3C EA A1 00 2C CA	A1 00 1C C9 A1 00 0C C9 91 00 FC C0 91 00
00000118:	EC C0 81 00 DC 78 81 00 D4 78 81 00 C4 78	7D 00 34 74 7D 00 24 74 75 00 94 66 75 00
00000134:	84 5E 75 00 74 3A 75 00 64 3A 73 00 54 BA	72 00 44 B1 72 00 3C B1 72 00 2C B1 71 00
00000150:	1C 9F 71 00 14 9F 71 00 04 7B 71 00 FC 7A	71 00 F4 7A 71 00 E4 32 71 00 DC 32 71 00
0000016C:	D4 32 71 00 C4 A2 70 00 B4 A2 60 00 AC A2	60 00 9C A2 40 00 94 A2 40 00 8C A2 40 00
00000188:	84 A2 40 00 7C A2 40 00 74 A2 40 00 6C A2	40 00 5C 5A 40 00 4C 48 40 00 44 48 40 00
000001A4:	34 48 00 00 2C 48 00 00 24 48 00 00 14 00	00 00 0C 00 00 00 04 00 00 00 E8 A0 FD FE
000001C0:	EC A0 FD FE 1E CD FD FE 04 00 00 00 00 48	00 00 25 16 48 3F B8 D2 50 BE 7A 3F 3D 3E
000001DC:	90 C2 32 BD 73 26 FA BD 01 12 9E 3D 6C 63	F5 3B E4 B3 AD BE 0B 75 1A BE 33 F7 C1 BC
000001F8:	5B 99 30 3E CC 88 64 3D 31 4D 63 3E C0 0F	20 3E 33 12 EA BD 93 1E 6E 3E DA D7 D0 BE
00000214:	28 58 79 3C 16 62 AB 3D 5E CB 4E 3E A7 1D	FC BE 7A 6B 40 3E 37 B6 88 3E 4C 13 A6 BD
00000230:	1E DE 08 3E CB EA D5 3E FE DF 91 3D 48 A9	53 3E B5 F7 E7 BC E4 58 41 BE E3 13 89 BD
0000024C:	95 D7 F0 3D 90 4A C4 BE CE 84 03 BE DC 56	12 BE 3C 1E 43 3E FF E4 DB 3E 17 74 77 3D
00000268:	25 57 84 3C 96 71 E7 3D AA 22 86 3E 99 78	B1 3E 59 18 A9 3D FB F5 98 BD CF 28 9E BE

Figure 4.1: Tensor flow Lite image

For mobile or embedded devices with limited memory assets, tensorflow lite is a great option for operating the models smoothly with efficiency. In case of storing models, special formats are required for better efficiency. TensorFlow models needs to be transformed into this format before Tensorflow Lite can use them.

To gain optimization while not losing accuracy, models needs to be converted. This procedure also reduces the size of the files. The TensorFlow Lite converter can reduce the file even more to increase application speed but this option comes with some trade-offs..

### 4.2.1 Confusion Matrix

The preprocessed imbalanced dataset was used to train our MobileNet model . Figure represents the confusion matrix for this application that achieved an accuracy of 84.28%.

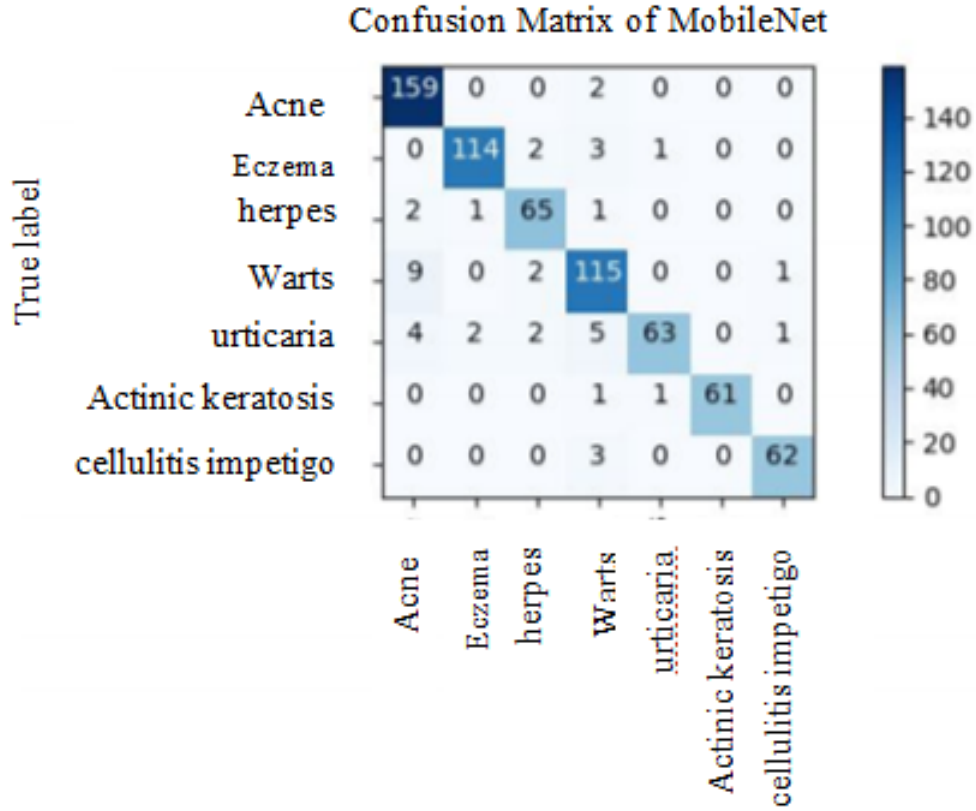


Figure 4.2: Confusion Matrix

### 4.2.2 Accuracy

We evaluated the performance of the SVM model by depending on F-score, accuracy, precision. True positives, true negatives, false positives, false negatives are parameters used for the evaluation process.

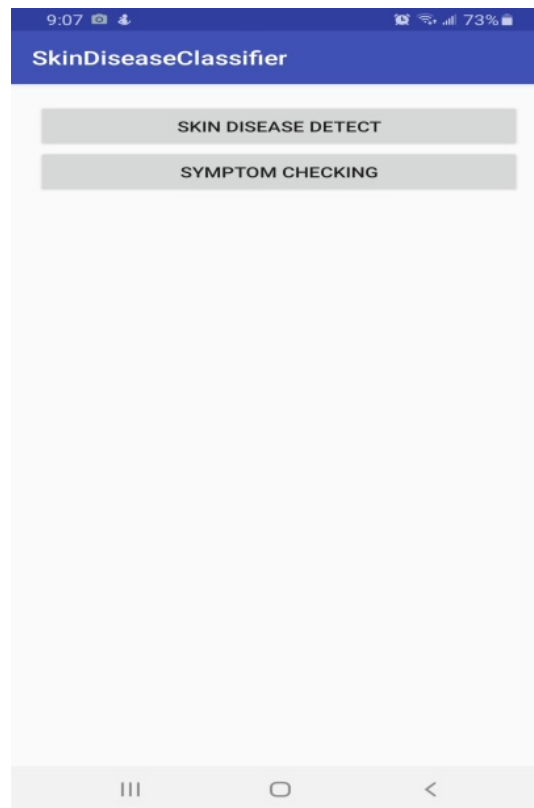
Algorithm	Accuracy	Precision	Recall	F-Score
MobileNet	84.28	92.6	71.4	80.6

Table 4.3: MobileNet Accuracy

## 4.3 Result

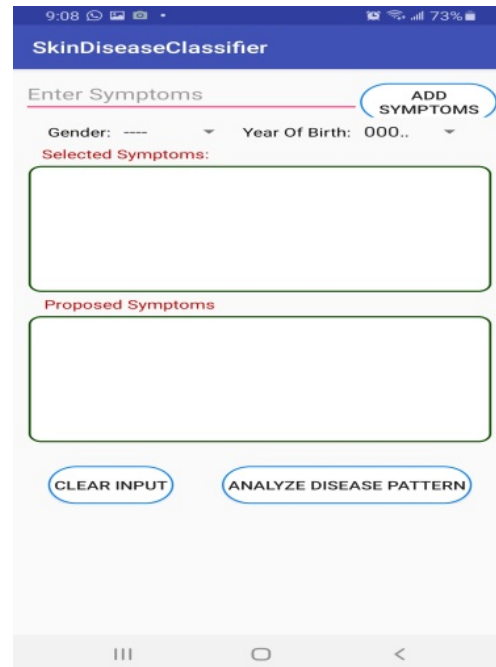
In this section, we will show how the application operates. After opening the application, the users will be presented with two options. One is for common disease

prediction using symptom checking and the other one is for skin disease prediction only.



### 4.3.1 Common Disease Prediction

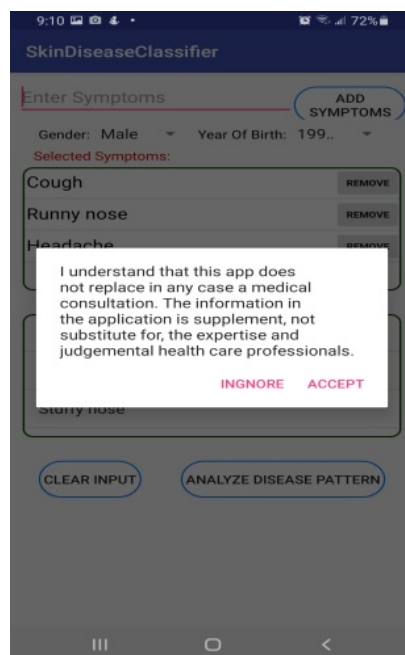
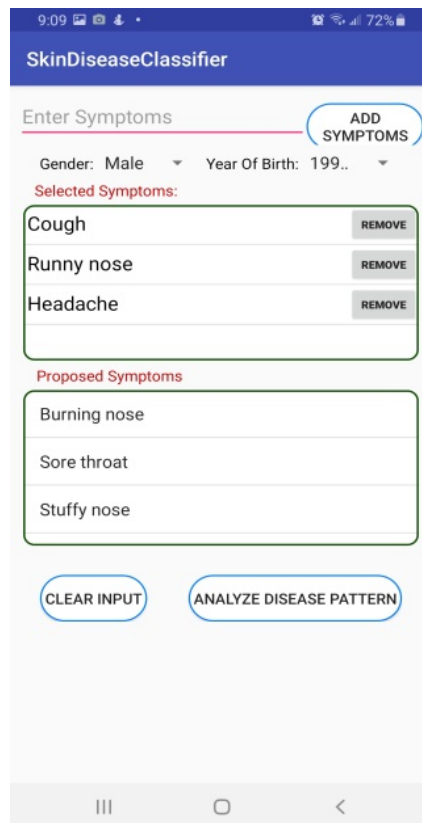
After hitting the symptom checking option, the application will take the user to the next page where they will have to give their information for the disease prediction process.



The screenshot shows a mobile application interface titled "SkinDiseaseClassifier". At the top, there is a status bar with the time 9:08, signal strength, and 73% battery. Below the title bar, there is a section labeled "Enter Symptoms" with a red underline. To the right of this section is a blue button labeled "ADD SYMPTOMS". Below the "Enter Symptoms" section, there are two dropdown menus: "Gender: ---" and "Year Of Birth: 000..". Below these is a red label "Selected Symptoms:" followed by a large empty rectangular box. Below this box is a red label "Proposed Symptoms:" followed by another large empty rectangular box. At the bottom of the form, there are two blue buttons: "CLEAR INPUT" and "ANALYZE DISEASE PATTERN". The bottom of the screen shows a standard Android navigation bar with three icons: a square, a circle, and a triangle.

The users will have to enter their symptoms along with their gender and year of birth. After entering this information they need to hit the “ADD SYMPTOM” button to input these information. The application will then propose some more symptoms related to the previous given one for the user’s ease of usage.

The user can also remove any symptoms if need be. The user can also input symptoms manually if he/she thinks that their occurring symptoms are not given in the “Proposed Symptoms” box. After entering symptoms, the user will need to press the “ANALYZE DISEASE PATTERN” button so that they can get a prediction given the symptoms.



By hitting the analyze button the users will get a message where they will be notified that this applications job is to give basic medical attention to patients, it is not substitute for the expertise and judgmental health care professionals.

4:18
62%

SkinDiseaseClassifier

Strain of the neck muscles

General practice
MORE INFO

Bad sleeping habits

General practice
Neurology
MORE INFO

Cold

General practice
MORE INFO

Flu

General practice
Internal medicine
MORE INFO

Inflammation of the brain covering membranes

General practice
Infectiology
Internal medicine
MORE INFO

Narrowing of the spinal canal in the neck

General practice
Orthopedics
MORE INFO

Arthrosis

General practice
Rheumatology
MORE INFO

Chronic muscular pain disorder

General practice
Rheumatology
MORE INFO

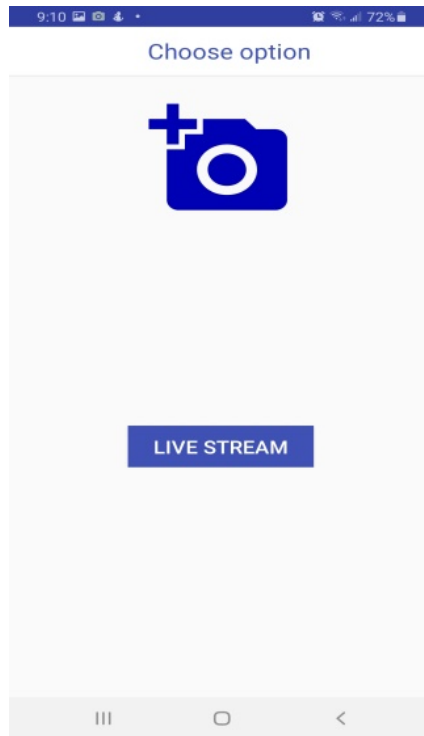
Then the users will get the prediction based on their given symptoms. If they press the “MORE INFO” button beside any disease, then they will be presented with detailed information about the disease, how does it occur, what are it’s cure etc.



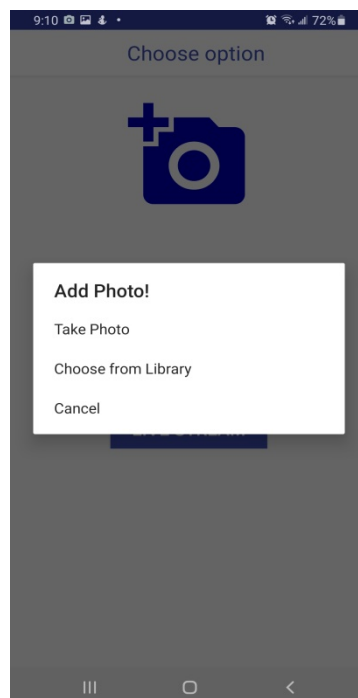
## 4.4 Skin Disease Prediction

The second portion of this application is skin disease prediction where the application can scan any skin abnormalities via camera in real-time or from an image taken by the camera or even from an image of skin disease selected from the phone.

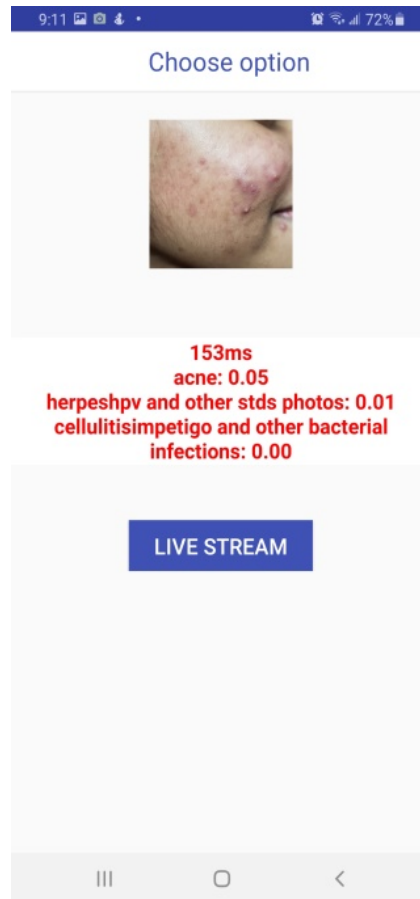




The skin disease page contains three options by which the user can see the prediction of their probable skin disease. If they press the “LIVE STREAM” button, then the application will go to camera mode to scan any skin abnormalities via camera in real-time.



If they press the “blue camera” icon, then they will be presented with two options. They can take a photo by camera to see the skin disease prediction or they can select a previously taken photo from the phone and see the results. The prediction gets more accurate if the picture quality is clear enough.



# Chapter 5

## Conclusion

### 5.1 Conclusion

The proposed android application that we constructed is reliable and secure medium for predicting patients' diseases. It also has a user-friendly interface requiring minimal learning and IT skills. One of the most satisfactory feedbacks of our system is it saves so much time and energy. It requires less manpower and is very useful for patients who can't afford medical treatment. Also users can get basic health tips and medical attention anywhere anytime. Which makes it a really handy and useful resource given the current covid-19 pandemic situation.

#### 5.1.1 Limitations

- Users can't search anything manually.
- The user cannot locate any doctor nearby.
- Requires internet connection to update the database.
- Some symptoms may not match with others in case of rare diseases.

#### 5.1.2 Future Work

- We can add an FQ option to get user feedback.
- We can give more advanced features for the system including more facilities according to user demand.
- We will try to add an online drug-selling platform.
- We will implement a GPS tracking system for users so that they can find any doctor nearby.
- We can develop a system to suggest doctors with their chamber locations and contact information according to the patients selected disease after prediction.
- Will try to add some premium features to the registered members.

# Bibliography

- [1] D. L. Pham, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.
- [2] R. Koggalage and S. Halgamuge, “Reducing the number of training samples for fast support vector machine classification,” *Neural Information Processing-Letters and Reviews*, vol. 2, no. 3, pp. 57–65, 2004.
- [3] V. Jakkula, “Tutorial on support vector machine (svm),” *School of EECS, Washington State University*, vol. 37, 2006.
- [4] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised learning,” *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.
- [5] L. Parthiban and R. Subramanian, “Intelligent heart disease prediction system using canfis and genetic algorithm,” *International Journal of Biological, Biomedical and Medical Sciences*, vol. 3, no. 3, 2008.
- [6] S. Gong, “A collaborative filtering recommendation algorithm based on user clustering and item clustering,” *JSW*, vol. 5, no. 7, pp. 745–752, 2010.
- [7] S. Bhattacharyya, “A brief survey of color image preprocessing and segmentation techniques,” *Journal of Pattern Recognition Research*, vol. 1, no. 1, pp. 120–129, 2011.
- [8] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, “Recommender systems,” *Physics reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [9] D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato, and F. Fdez-Riverola, “Web scraping technologies in an api world,” *Briefings in bioinformatics*, vol. 15, no. 5, pp. 788–797, 2014.
- [10] S. Vijayarani, S. Dhayanand, *et al.*, “Data mining classification algorithms for kidney disease prediction,” *Int J Cybernetics Inform*, vol. 4, no. 4, pp. 13–25, 2015.
- [11] M. Murty and R. Raghava, “Kernel-based svm,” in *Support vector machines and perceptrons*, Springer, 2016, pp. 57–67.
- [12] S. Suthaharan, “Support vector machine,” in *Machine learning models and algorithms for big data classification*, Springer, 2016, pp. 207–235.
- [13] E. Vasilevskis, I. Dubyak, T. Basyuk, V. Pasichnyk, and A. Rzhenskyi, “Mobile application for preliminary diagnosis of diseases,” in *IDDM*, 2018, pp. 275–286.

- [14] N. L. Fitriyani, M. Syafrudin, G. Alfian, and J. Rhee, "Development of disease prediction model based on ensemble learning approach for diabetes and hypertension," *IEEE Access*, vol. 7, pp. 144 777–144 789, 2019.
- [15] J. Hamidzadeh and S. Moslemnejad, "Identification of uncertainty and decision boundary for svm classification training using belief function," *Applied Intelligence*, vol. 49, no. 6, pp. 2030–2045, 2019.
- [16] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," *IEEE Access*, vol. 7, pp. 81 542–81 554, 2019.
- [17] D. Sinha and M. El-Sharkawy, "Thin mobilenet: An enhanced mobilenet architecture," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE, 2019, pp. 0280–0285.
- [18] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC medical informatics and decision making*, vol. 19, no. 1, pp. 1–16, 2019.
- [19] J. Velasco, C. Pascion, J. W. Alberio, J. Apuang, J. S. Cruz, M. A. Gomez, B. Molina Jr, L. Tuala, A. Thio-ac, and R. Jorda Jr, "A smartphone-based skin disease classification using mobilenet cnn," *arXiv preprint arXiv:1911.07929*, 2019.
- [20] A. K. Verma and S. Pal, "Prediction of skin disease with three different feature selection techniques using stacking ensemble method," *Applied biochemistry and biotechnology*, pp. 1–20, 2019.
- [21] A. K. Verma, S. Pal, and S. Kumar, "Comparison of skin disease prediction by feature selection using ensemble data mining techniques," *Informatics in Medicine Unlocked*, vol. 16, p. 100 202, 2019.
- [22] D. Wu, Y. Zhang, X. Jia, L. Tian, T. Li, L. Sui, D. Xie, and Y. Shan, "A high-performance cnn processor based on fpga for mobilenets," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, 2019, pp. 136–143.
- [23] R. KUNCHHAL, *The mathematics behind support vector machine algorithm (svm)*, <https://www.analyticsvidhya.com/blog/2020/10/the-mathematics-behind-svm/>, Oct. 2020.
- [24] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm," *Sensors*, vol. 21, no. 8, p. 2852, 2021.