AN IDENTIFICATION AND MONITORING SYSTEM FOR THERAPEUTIC INTERVENTION FOR CHILDREN UNDER CARE

Trisha Melani Amarasekara

(179303T)

M.Sc. in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2019

AN IDENTIFICATION AND MONITORING SYSTEM FOR THERAPEUTIC INTERVENTION FOR CHILDREN UNDER CARE

Trisha Melani Amarasekara

(179303T)

This dissertation submitted in partial fulfillment of the requirements for the Degree of MSc in Computer Science specializing in Software Architecture

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2019

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ……………… Date: ……………….

Name: A.A.T.M Amarasekara

The supervisor/s should certify the thesis/dissertation with the following declaration.

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for evaluation for the MSc Thesis (CS6997).

Signature of the supervisor: …………………………. Date: ………………..

Name: Dr. Indika Perera

ABSTRACT

Nowadays, everyone owns a mobile device or access to one. With the massive usage of computers, HCI based applications are in high demand. HCI based applications can be used effectively in Medical and healthcare sector, especially to diagnose diseases. There are several genetic disorders and among them, Down syndrome is the most common genetic disorder. Earlier identification and therapies are very important, since early treatments help children to grow more normally. In this problem background, this research is mainly focused on developing a HCI based, identification and monitoring system for therapeutic intervention for children under care. Since, children with Down syndrome have distinct facial features than others, image processing based approach is used to support the identification of the disorder.

This approach is based on the client server architecture. Here, the client is the mobile application and in server side, there is a combination of the web service and the database. The REST API is for the purpose of Down syndrome detection. To implement this web service, 20 face samples were gathered, including 10 Down syndrome face samples and 10 healthy face samples. By using these samples, two datasets were created for Down syndrome and none syndrome. Each dataset have 10 data for each facial landmark, which includes, jaw, left eye, left eyebrow, right eye, right eyebrow, mouth and nose. After creating the dataset, it was trained using LBP. Based on this trained dataset the web service has been implemented. This web service mainly consists of three phases, face detection, facial feature extraction and classification.

The mobile application consists with three main functionalities, which are Down syndrome detection test, Strengths and Difficulties Questionnaire (SDQ) and Progress evaluation based on the SDQ. In Syndrome detection test, once the parent browse or capture an image of the child, it is passed to the web service as a HTTP POST request and the response from the web service is displayed to the parent as the result. The evaluation of the Detector test has been done by using a test dataset which includes, 30 images and shown that it has 90% of accuracy level, 87.5% precision, 93.3% recall and 90.3% f1-score. Parents can perform SDQ test which includes a number of questions to identify mental and health problems of children between 4 to 17 years old. After completing the test, the application displays the result which includes total difficulties score, emotional symptom scale, hyperactivity scale, peer problem scale, pro social scale and the impact score by examining whether the scores are normal, borderline or abnormal based on the standard scoring scheme. Further, from the second attempt of the SDQ, there is a progress evaluation tool. The application keeps track of all the historical records of the child. These two functionalities also have been evaluated based on feedbacks from few doctors as well as few parents.

ACKNOWLEDGEMENT

First, I wish to express my great gratitude and appreciation to my project supervisor, Dr. Indika Perera, for the guidance, suggestions and encouragements throughout this research project which helped me to complete this work, in time.

I convey my special thanks to Dr.Apeksha, Dr.G.D Indika Kumara, Dr.S.R Pathirana and medical staff members of Lady Ridgeway hospital, for knowledge, and valuable assistance in initialization of this research.

Then, my utmost gratitude is extended to my husband for his encouragement and guidance throughout my academic life. Next, my friends, especially to Mr.Tiran Wijesekara, Miss. Niroshini Ranaweera, Mr.Ryan Benjamin, Mr. Vilochane Vidyarathne, Mr.Lakshan Costa and Miss. Nithila, whose friendship, hospitality and for support in one way or another in the preparation and completion of this study.

Last but not least, my sincere thanks also forwarded to my family. I am thankful to my Mother, whom I dedicate my work, for her love and best wishes. It is a pleasure to express my gratitude wholeheartedly to my father and my loving sisters for their inseparable support and care in the successful completion this research and for supporting me throughout the toughest years of my academic life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
HCI	Human Computer Interaction
CLM	Constrained Local Model
PCA	Principal Component Analysis
SVM	Support Vector Machines
RBF	Radial Basis Function
k-NN	K-Nearest Neighbor
RF	Random Forest
GWT	Gabor Wavelet Transform
LBP	Local Binary Patterns
API	Application Programming Interface
SDK	Software Development Kit
OpenCV	Open Source Computer Vision Library
HOG	Histogram of Oriented Gradients
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
SDQ	Strengths and Difficulties Questionnaire
HTTP	HyperText Transfer Protocol

Chapter 1

INTRODUCTION

1.1 Background

Nowadays, everyone has a mobile device or access to one and it has become a necessity. The variety of people's aspects can be represented using mobile applications. With the massive usage of computers, in most of the aspects of our day-to-day works, Human Computer Interaction (HCI) is becoming very important. The main purpose of HCI is to design user-friendly as well as well-functioning system which is matched to user requirements by using new input and output technologies. Usability is a very important concept in HCI and is concerned with developing systems as easy to use and easy to learn with minimum error rate [1]. Effective use of Information Technology (IT) can be seen as a key success factor of our growing world.

Medicine and Health Care is a key field where we can use IT effectively. To increase health and safety of Medicine and Health Care sector, new IT technologies can be used effectively. Especially for clinical decision support systems [2]. Medical diagnosis is the main area in the field and HCI based systems can play an important role in this area. In the world, there are many births with genetic disorders. Among these genetic disorders Down syndrome is the most common genetic disorder.

1.2 Down syndrome

Humans are made up of cells and averagely adult human body has around 37.2 trillion cells [3]. Each cell has 23 pairs of chromosomes, 22 pairs of them are called as autosomes and these pairs are same for males and females. 23rd pair is differ for males and females, since it is the sex chromosomes. Females have XX and males have XY [4]. Human chromosomes lined up is shown in Figure 1.1.

If there is an additional copy of chromosome 21, it causes to Down syndrome. Genetic basis of Down syndrome is shown in figure 1.2. In this situation there are three copies instead of normal two. This additional genetic material causes development changes since cells cannot properly control protein production. Therefore, they have a huge risk of medical significant problems [5] [6]. They have difficulty in learning to walk, talk and also take care of them self as others do.

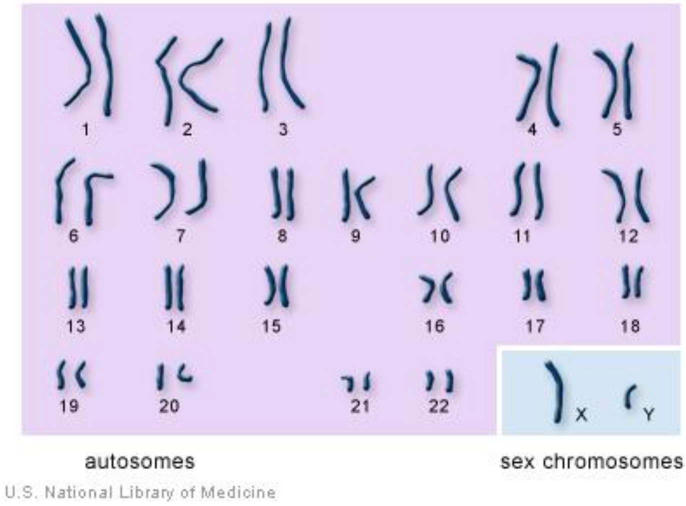


Figure 1.1: Chromosome pairs *[7]*

They have different facial features than other normal people. A flat face, upward slanting eyes, a small broad nose, abnormally shaped ears and a large tongue [8] [9].

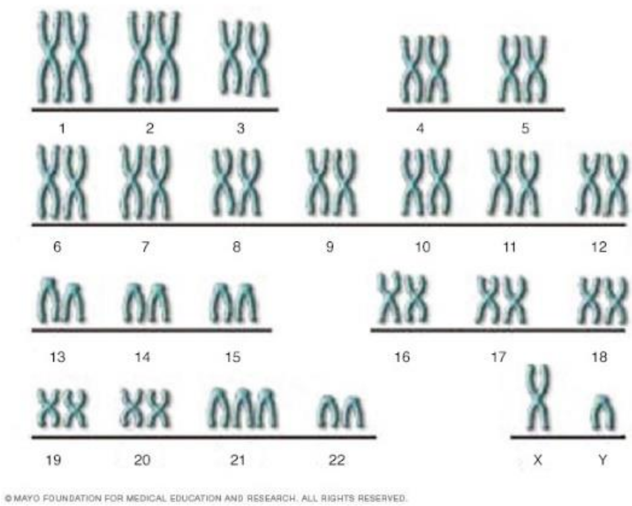


Figure 1.2: The genetic basis of Down syndrome *[10]*

1.3 Problem statement and Motivation

Among the children under care, most of the children have Down syndrome. Earlier detection of Down syndrome and therapies are very important. There are several ways to diagnose this disorder. Screening tests can be performed during pregnancy to identify whether the child is with Down syndrome or not. But these tests are performed only if there is an issue in the scanning test. There are many cases, Down syndrome is not diagnosed in the womb. If it is so, usually doctors can identify it after the birth by using facial appearance of the baby. But in most cases it is difficult to identify these symptoms as soon as the child is born. However, when the child is growing up these distinct facial features can be identified if the parents are aware about it. But the problem is parents are not aware about this.

This disease cannot be cured, but if it is identified earlier, therapies can help these children to grow more normally [8]. Early treatments can help these children to develop more normally. There are a variety of therapies for the children with Down syndrome. Some of these therapies are physical therapy, speech-language therapy, occupational therapy and emotional and behavioral therapies. In addition to therapies, there are some drugs to affect and improve their brain activities and also there are some assistive devices which can help these children to enhance their learning skills and to solve their problems [11].

Since, the people with Down syndrome have a greater risk of having several health problems. Many of these problems require immediate treatments, occasional treatments or long term treatments, therefore, earlier detection is very important. If it detects after long time, it is difficult to make their lives productive even using therapies and other treatments.

In such a background, the focus of this study is to develop an automated Down syndrome detection supporting tool which can also be used as a progress evaluation tool, if it detects as Down syndrome.

1.4 Objectives and Aims

- Identify distinct facial features of children with Down syndrome.
- Implement a web service for Down syndrome detection based on their distinct facial features by using image processing and machine leaning techniques.
- Implement a mobile application which has following functionalities.
    - Down syndrome detection supporting tool.
    - SDQ scoring tool.
    - Progress evaluation tool based on SDQ scores and child history records.

1.5 Scope of the project

There are several types of genetic disorders and many of them result from unbalanced chromosome abnormalities [4]. Among all the genetic disorders, Down syndrome is the most common disorder. Therefore, by considering the scope of the project, this study mainly focused on children with Down syndrome. Since facial appearances vary with the nationality, this implementation only considers about the Sri Lankan domain.

1.6 Overview of the project

The purpose of this project is to develop an identification and monitoring system for therapeutic intervention for children under care. By considering the scope of the project, this research mainly forces on children with Down syndrome since it is the most commonly occurring genetic disorder. This will help parents to identify this disorder earlier by simply uploading an image to the application. Once a parent uploads an image of the child to the application, it will analysis the image and display the result. According to the result if there is a possibility of having Down syndrome, parents can immediately contact a doctor and get treatments.

This approach is based on the client server architecture. Here the client is the mobile application and in server side there is a combination of script and database.

Figure 1.3: Basic architecture

In addition to the Down syndrome detection part, there are two functionalities in the application. Which are SDQ test and progress evaluation tool. Parents can give answers to the SDQ test and the application evaluates the answers and displays the result which includes all the scores required by doctors. The progress evaluation tool keeps track of all the history records of the child and evaluates the progress and graphically represents the evaluation summary.

Chapter 2

LITERATURE REVIEW

2.1 Symptoms of Down syndrome

Down syndrome is a chromosome level genetic disorder. It causes from an additional copy of chromosome 21. This happens because of the non-disjunction, which means a pair of chromosomes fails to separate during mother's egg or father's sperm formation. Mostly there is a risk of having a baby with Down syndrome, if the mothers age 35 or older [8] [11].

People with Down syndrome have a huge risk of having medically significant problems, including heart defects, vision problems, hearing loss, infections, blood disorders and hypothyroidism. They are developing more slowly than others and they have difficulty in walking, talking and take care of them self.

Their distinct facial features are listed below.

- Upward slanting eyes with small skin folds on the inner corner of the eyes

Figure 2.1: Symptom 1 - Upward slanting eyes with skin folds *[12]*

- A small broad nose

Figure 2.2: Symptom 2- Small broad nose *[12]*

- A flat face

Figure 2.3: Symptom 3 - Flat face *[12]*

- Abnormally shaped ears

Figure 2.4: Symptom 4 - tiny ears *[12]*

- A small mouth

Figure 2.5: Symptom 5 - Small mouth *[12]*

2.2 Similar Products

There are several medical diagnosis applications and few of them are syndrome related applications. But most of them are text based and users have to insert answers to questions which are asked by the application. Then the application displays results based on the user answers. Face2Gene app is based on the image processing.

2.2.1 Face2Gene

Face2Gene is a mobile application designed for healthcare professionals and to use this, users should have proper medical training. Once a photo of a patient's face is updated by a geneticist, the application analyzes data and displays a list of possible syndromes. This application converts images to data, based on measurements, including distance between eyes, the length of the face and other ratios. This application contains three main parts, clinic, forum and library. In the clinic section, there are a number of features including detect dimorphic features, reveal related traits and discover disorders. In the forum section, users can share cases and comment on other cases. Library section can be used to search for syndromes, review photos and features [13].

2.3 Research studies based on Down syndrome detection

A number of research papers have been published and presented by several researchers regarding Down syndrome detection. Some of these researches are based on the facial photographs' analysis.

2.3.1 Automated Down Syndrome Detection using Facial Photographs [5]

The authors of this research proposed a navel strategy based method to detect Down syndrome using some machine learning techniques along with facial images of children. In this research, they have used a dataset of 100 photos including 50 Down syndrome patients' photos and 50 healthy persons' photos in age range from 0 to 10. Their work has been mainly divided into four parts, which are landmark detection, feature extraction, classification and evaluation [5].

2.3.1.1 Landmark detection

A constrained local model (CLM) has been used to locate landmarks. 44 anatomical landmarks and 37 pseudo-landmarks have been defined as shown in figure 2.6 (a). CLM has been optimized by dividing into two parts which are model building and searching [5].

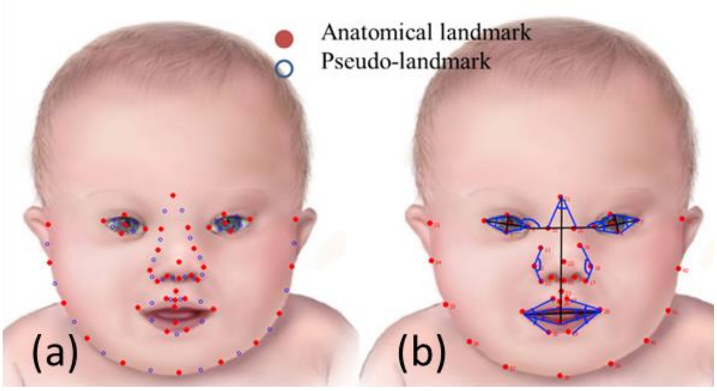


Figure 2.6: Facial landmarks: (a) Face annotated using 44 anatomical landmarks And 37 pseudo-landmarks; (b) the illustration of geometric landmarks *[5]*

There are two parts in their model building part, which are shape model and patch model [5]. The shape model describes the variation of face shape by using principal component analysis (PCA) and the patch model describes facial feature surrounding look. First three principal modes of PCA are shown in figure 2.7 and the second principle shows the normal model [5].

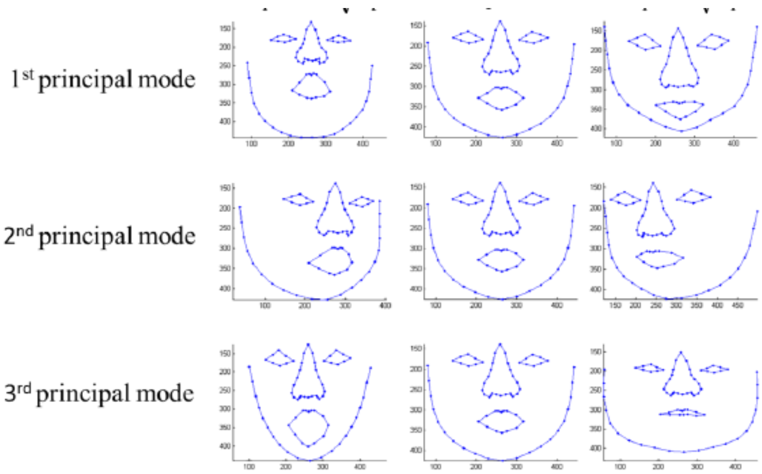


Figure 2.7: First three principles of (PCA) *[5]*

2.3.1.2 Feature extraction

After locating landmarks, then geometric features are extracted from these landmarks and texture features. For that they have defined seven vertical distances and 13 angles as shown in figure 2.6 (b). A local binary pattern (LBP) histogram has been used by the author to extract features [5].

2.3.1.3 Classification and evaluation

After the feature extraction process, Down syndrome specific features are selected based on the shape model which they have built for normal and abnormal cases. Figure 2.8 shows the statistical point distribution of the training data and the mean shapes of the two groups [5].

Finally, they have compared performance using four classifiers, which are support vector machines (SVM) with radial basis function (RBF) kernel, linear SVM, k-nearest neighbor (k-NN), and random forest (RF), regarding accuracy, precision and recall. They have shown 94.6% accuracy when using local texture features over geometric features [5].

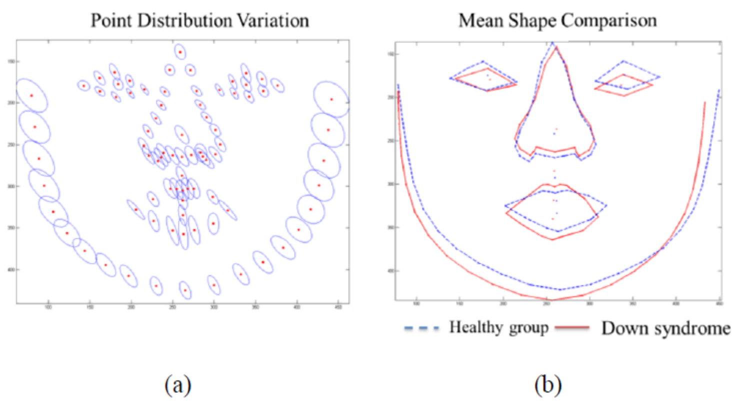


Figure 2.8: Shape model: (a) point distribution of the training data; (b) mean shape of Down syndrome and healthy groups. *[5]*

2.3.2 Down syndrome diagnosis based on Gabor Wavelet Transform [14]

The authors of this research also proposed a novel method to detect Down syndrome, but Gabor Wavelet Transform (GWT) has been used to extract textures. Their dataset consists of healthy and Down syndrome children's images in age range from 1-12 years. Each group consists of 15 images, including both girls and boys [14].

2.3.2.1 Standardization and selection of images

In order to feature extraction process, they have standardized the images. This process includes rotating, cropping, scaling, gray scaling and histogram equalization. Rotation process is to adjust visual direction of images. Cropping process is to focus on the actual area by deleting unneeded parts of images. Scaling process is to adjust the resolution of the images in order to Gabor Wavelet Transformation. Histogram equalization contrast differences in images [14].

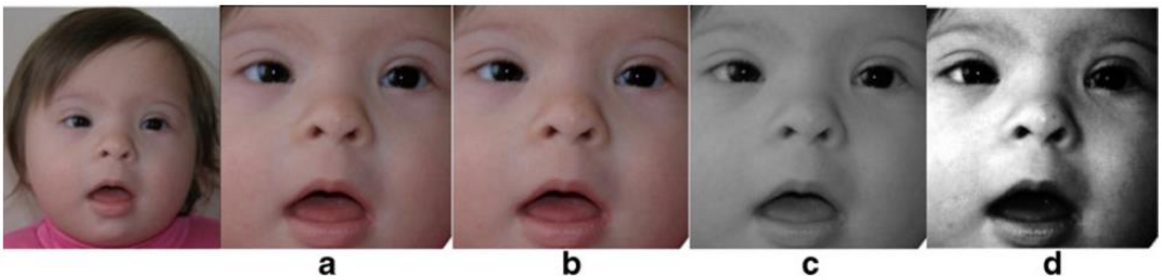


Figure 2.9: Preprocessing of images. a Crop b Scale c Convert RGB image to grayscale d Histogram Equalization *[14]*

2.3.2.2 Feature extraction of images using Gabor Wavelet Transform

The feature extraction process has been done using Gabor Wavelet Transform (GWT). Wavelet transforms are performed analysis based on the time and frequency domain [14]. In this process important facial landmarks are detected and local features such as distance and angle between these points, some quantitative measures are extracted. For the face recognition, above mentioned features are used. They have convolved 40 different wavelets for all images in the database and magnitudes of convolution results of each image are collected in each vector. Figure 2.10 shows the convolution result of Down syndrome's face with 4o Gabor wavelets [14].

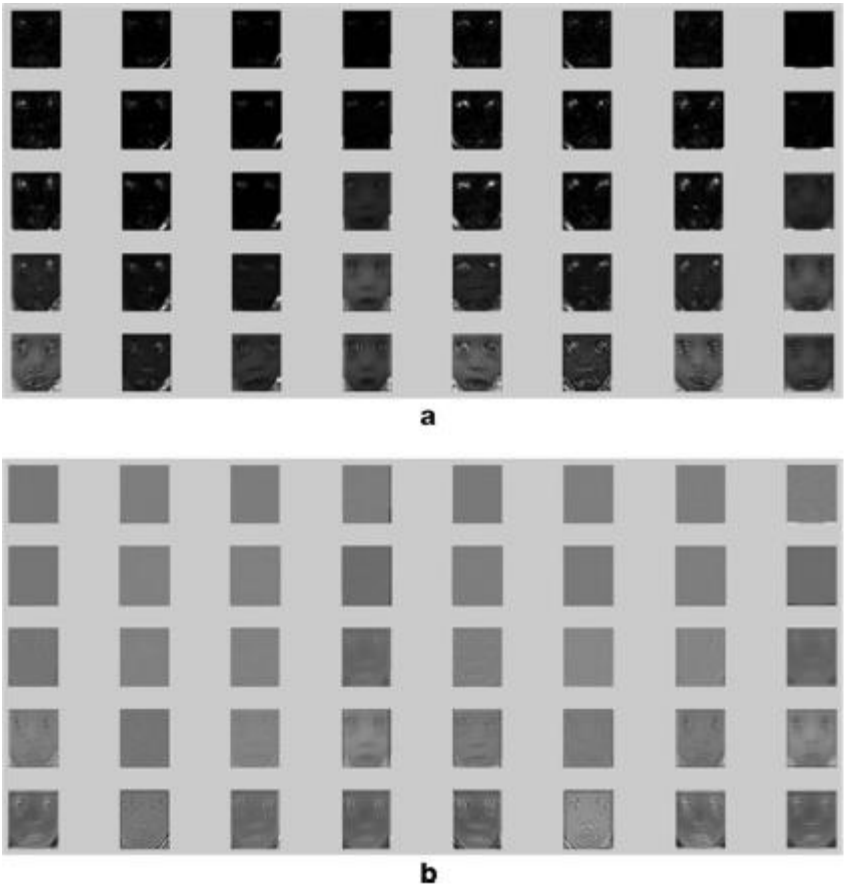


Figure 2.10: Gabor wavelet representation of a face image. a magnitude b the real part *[14]*

2.3.2.3 Statistical analysis of features

For the first time, the most valuable feature selection is done in this research. Dimension reduction is done by applying Principal Component Analysis (PCA) to find most valuable 2000 components. Then Linear Discriminant Analysis (LDA) has been used to derive most important information.

2.3.2.4 The statistical evaluation of classification

K-nearest neighbor (KNN) and Support Vector Machine (SVM) methods have been used for classification. For both methods, accuracy is given as 96% and 97.34% respectively.

2.3.3 Down syndrome recognition using local binary patterns and statistical evaluation of the system [15]

The authors of this research used a dataset with 51 Down syndrome and 56 healthy face images in age range 0 – 50 years. LBP has been used to extract facial features. Euclidean distance and Changed Manhattan distance methods have been used to implement classification process. In this research, the authors have considered about many differences such as race, skin tone, facial wear, age, pose etc. [15] .

2.3.3.1 Selection of images

Since their classification process has been done using template matching based approach, training sets have been selected for both Down syndrome and healthy groups. Their training sample consists of 20 face samples, including 10 Down syndrome face samples and 10 healthy face samples [15].

2.3.3.2 Feature extraction

After the training sets defining process, LBP features have been extracted from each sample and then for each set, LBP feature vectors have been defined by obtaining the average of the LBP features [15].

2.3.3.3 Classification

In this research, the classification process has been done using Euclidean distance and Changed Manhattan distance methods. These methods are used for measuring similarity between feature vectors [15].

2.4 Face detection

Face detection methods can be classified into four groups, which are knowledge based methods, feature invariant approaches and template matching methods [16].

Knowledge-based methods

These methods are called as rule based methods which encodes the human knowledge of what can be found in a face. There are some relationships between facial features. These relationships can be captured using rules which are mainly used for face localization [16].

Feature invariant approaches

These approaches are mainly used to find structural features, which can be identified even in various conditions of pose, viewpoint or lighting. These methods also mainly used for face localization [16].

Template matching methods

In these methods, several patterns of faces and facial features are stored in a database. Once an image is inputted to detect, the image is compared with stored images and return the detection output. These methods mainly used for face localization and detection [16].

Appearance-based methods

Set of images is trained for different facial appearance and the models are learned from the set. In these methods, learned models used to detect faces. These methods mainly used for face detection [16].

2.4.1 Face Detection Methods on Android

The face detection process can be done in an android application. For face detection, there is an API in Android SDK. The findFaces method of android.media.FaceDetector class can be used to detect faces in the image. By using this class can be found some other information such as eyesDistance, pose, and confidence [17].

2.4.2 Face Detection on OpenCV

OpenCV (Open Source Computer Vision Library) is a cross platform library which can be mainly used for image processing and it provides a number of functions for face detection [18].

2.4.2.1 Face Detection using Haar Cascades

Haar Cascades is a machine learning based approach, which can be used to detect objects in images. In order to detect faces, this algorithm needs to train the classifier with a lot of positive and negative images. Positive images in the sense images with faces and negative images in the sense images without faces. After training the classifier, features can be extracted from it. There are three types of common Haar features which can be used to extract features from images. Edge features, Line features and four rectangle features as shown in figure 2.11. [18]

Each of these features has a value which is computed by getting subtraction of the number of pixels in white rectangle from the number of pixels in the black rectangle. Such like that a large number of features can be calculated. But most of these features are irrelevant. To solve this, all the features are needed to apply for all the training images. Then feature with minimum error rate can be selected.

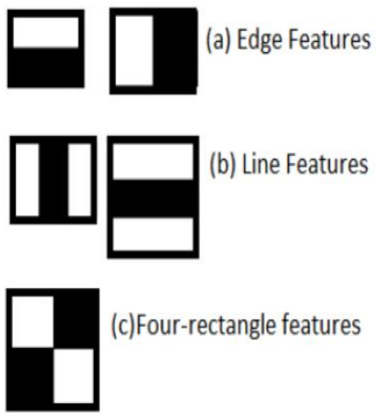


Figure 2.11: Common Haar Features *[18]*

To discard non face region, Cascade of Classifiers concept can be used and in this concept, all the features are grouped into different stages of classifiers. Then all the failing windows can be discarded [18].

2.4.3 Face Detection using DLib

Dlib is a library which is written in C++ and consists with machine learning algorithms and tools. There are two face detection models in Dlib library, which are Histogram of Oriented Gradients (HOG) and Convolutional Neural Network (CNN). HOG can be used to detect only frontal faces [19]. CNN is more accurate than HOG but to get expected speed it should be executed on a GPU, since it takes much more computational power [19].

With the Dlib, get_frontal_face_detector method can be directly used to detect front faces and also, Dlib library can be easily used to determine facial landmarks, since it has landmark detection method which produces 68 (x,y) coordinates as shown in figure 2.12 to map with each facial feature [20].

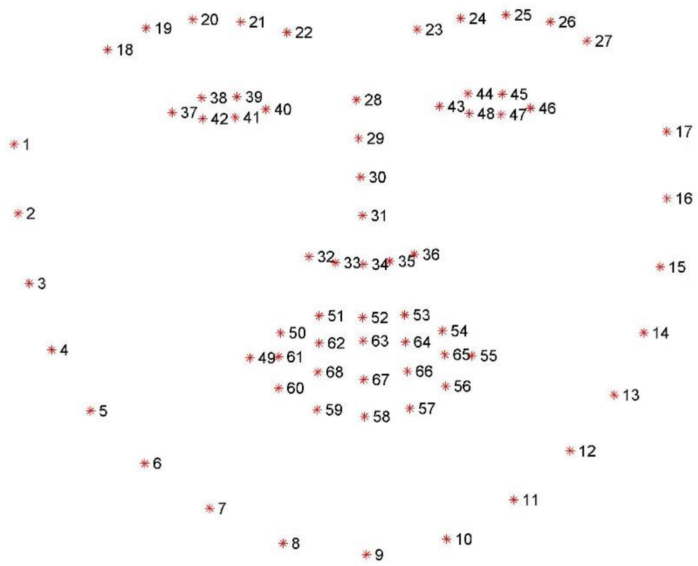


Figure 2.12: Dlib 68 facial coordinates *[21]*

Therefore, with the Dlib library facial regions can be easily accessed using these indexing.

2.5 Facial feature extraction

After the face detection process, the next step is to extract features from the face. In syndrome identification approach, facial feature extraction is a main task which is used to collect the set of features from an image [22]. There are several techniques for feature extraction. Some of them are Local Binary Pattern, Principal Component Analysis (PCA), Fisher Linear Discriminant Analysis and Gabor Wavelet Transform.

2.5.1 Local Binary Pattern

Local Binary Pattern (LBP) approach mainly used for texture classification, segmentation and image retrieval. LBP describes the shape and the texture of an image by dividing the image into small regions and extracting features as binary patterns. The extracted features are concatenated into single feature histogram and the binary number is generated based on the neighbor pixel value and the center pixel value. If neighbor pixel is greater than the center value, then it gets one, else it gets zero as shown figure 2.13.

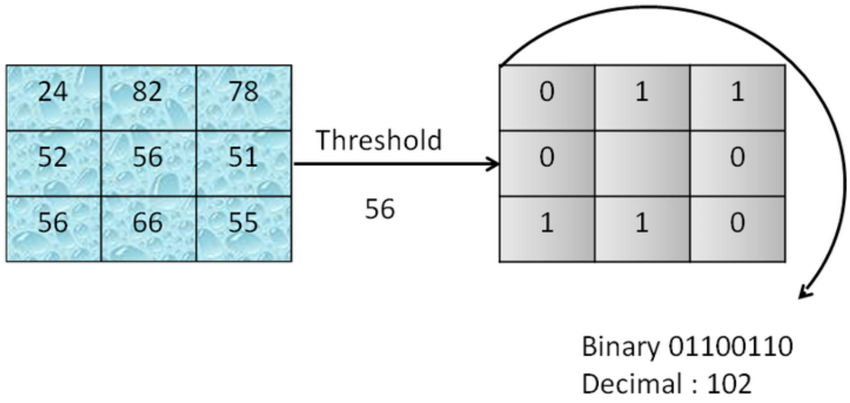


Figure 2.13: LBP calculation Example *[23]*

Basically LBP detects few main local textures, including spots, line ends, edges and corners as shown in figure 2.14. The texture of the image represents as a feature vector and it is basically a histogram of these patterns. These histograms can be used to measure similarity between images and for that distance between histograms need to be calculated.

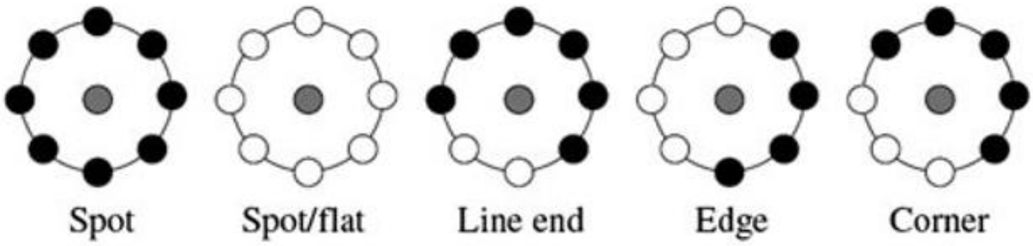


Figure 2.14: texture primitives of LBP *[24]*

2.5.2 Principal Component Analysis

PCA is a popular algorithm for facial feature extraction. This algorithm finds principal components of the training set by transforming the face images in to characterized set of feature images [25].

2.5.3 Fisher Linear Discriminant Analysis (FLDA)

This algorithm is mostly used to search for highlight data and it is also used to reduce the dimensions by removing noise. This method allows to develop a set of feature vectors from images of the same category [25].

2.5.4 Gabor Wavelet Transform

Gabor Wavelet functions extract local facial features from facial landmarks by optimizing resolution in time and frequency domains [14].

2.6 Face classification

In the classification process image features are categorized into several categories and analyze image features based on the several properties. This process mainly consists of two phases, they are training phase and testing phase [22]. There are several algorithms for classifications such as K-nearest neighbor, Support Vector Machine and Random forest. Some distances are very useful when classifying data.

2.6.1 Distances in classification

2.6.1.1 Euclidean distance

"Euclidean distance is the straight line distance between two points". If there are two points u=(x_1, y1) and v=(x_2, y2) then the Euclidean distance EU (u, v) can be computed as shown below [26].

$$EU\,(u, v) \quad = \quad \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \qquad (2\text{-}1)$$

Above case is two dimensional. If the points are multi-dimensional and two points are a = (x_1, x2 ...xn) and b = (y1, y2,…,y_n) the Euclidean distance is EU(a, b) can be computed as shown below [26].

$$EU\,(a, b) \quad = \quad \sqrt{(x1 - y1)^2 + (x2 - y2)^2 + \cdots \ldots + (xn - yn)^2} \qquad (2\text{-}2)$$

$$= \quad \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

2.6.1.2 Manhattan distance

Manhattan distance is the distance between two points measured along axes at right angles. If there are two points u=(x_1, y_1) and v=(x_2, y_2) then the Manhattan distance EU (u, v) can be computed as shown below [26].

$$MH\ (u, v) \quad = \quad |x1 - x2| + |y1 - y2| \qquad (2\text{-}3)$$

If the points are multi-dimensional and two points are a = (x_1, x2 ...xn) and b = (y1, y2,…,y_n) the Manhattan distance is MH(a, b) can be computed as shown below [26].

$$MH\ (a, b) \quad = \quad |x1 - y1| + |x2 - y2| + \ldots + |xn - y2| \qquad (2\text{-}4)$$

$$= \quad \sum_{i=1}^{n} |xi - yi|$$

2.6.2 Classifiers

2.6.2.1 K-nearest neighbor (kNN)

K-nearest neighbor algorithm can be used for classification and it classifies new instances based on similarity measure. This is a lazy learning algorithm and it stores training dataset. In this algorithm, k closest members of the training dataset are located for each predicting image. To do this, need to find the distances to each member of the test class and this algorithm uses a Euclidean Distance measure to find the distances. After finding K nearest neighbor, based on their class labels, majority voting is applied to determine the test image class label. Since, the K-NN algorithm compares the test data with all the training data, it is a slow classifier than the other classifiers [27].

2.6.2.2 Support Vector Machine (SVM)

SVM is also a mostly using algorithm for classification. In the linear SVM algorithm, the dataset divides into two classes using a hyper-plane. The hyper-plane should be created as it can separate the two classes better. After choosing the hyper-plane, from the hyper-plane to the nearest data in both classes is maximized [27].

2.6.2.3 Random forest (RF)

RF algorithm is a randomized decision tree based algorithm. For each decision tree, this algorithm randomly sampling a subset of feature set and a subset of the training dataset. Therefore, available data can be reduced and it makes data comparison process easier. It has several advantages, including an efficient performance with high or low dimensions, lightweight and allow to add new data without retraining the dataset [28].

2.7 Strengths and Difficulties Questionnaire (SDQ)

The SDQ is a globally accepted screening test which is used to identify mental and health problems of children between 4 to 17 years old. All the children's hospitals use this standard screening questionnaire to identify behavioral difficulties of children [29]. There are two main versions of SDQ, which are baseline version and follow up version. Each version consists at least one of the following components.

2.7.1 Psychological attributes

All versions of the SDQ includes this component which consists 25 items, which are divided into following five scales.

- Emotional symptoms
- Conduct problems
- Hyperactivity/inattention
- Peer relationship problems
- Pro social behavior

2.7.2 An impact supplement

This component assesses the impact of difficulties on the child's life. If there is an impact, it enquire further about chronicity, distress, social impairment, and burden to others [29].

2.7.3 Follow-up questions

This component includes in follow up version of the SDQ. There are only two questions in this component to identify whether there is any progress of the child's behavior or not. And to identify whether the therapies have helped in other ways [29].

Chapter 3

METHODOLOGY

3.1 Chapter overview

This chapter describes what the best approaches for each step (face detection, feature extraction and classification). Most suitable method among the methods which are described in the literature review, are selected for the implementation of the application.

3.2 Proposed Approach

The proposed approach is based on the client server architecture. Here, the client is the mobile application and on the server side, there is a combination of script and database. This approach mainly consists of three phases which are face detection, facial feature extraction and classification. The architecture of the proposed approach is shown in figure 3.1.

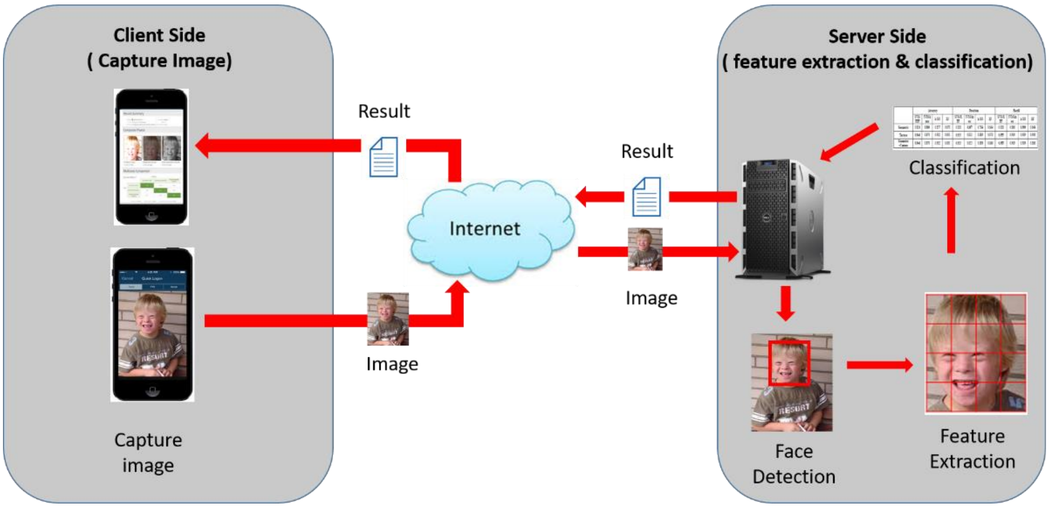


Figure 3.1: Proposed Approach

All the three main phases are done on the server side. For that a RESTFUL web service needs to be implemented. Since this is based on the image processing approach and python is one of the best languages for both image processing and web API developments, the web service can be developed with Python Flask. Only the image capturing and result representation parts handle on the client side. For that an HCI based mobile application needs to be implemented.

3.2.1 Face detection phase

This phase consists of few steps as shown in figure 3.2, initially, the captured image needs to convert into a grayscale version to reduce lighting effects and then resize the image to speed up the processing time [30]. Next, need to detect the face and determine facial landmarks from it.

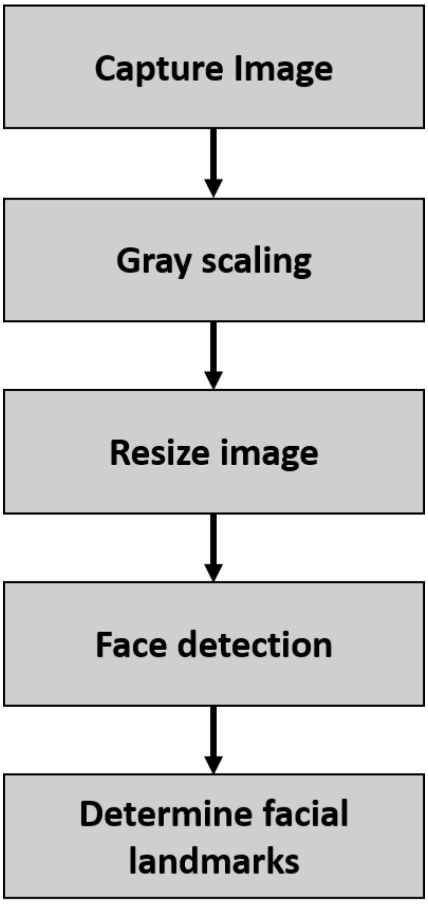


Figure 3.2: Face detection phase

All the above steps can be done using most suitable techniques. According to the proposed approach, among the above steps, image capturing part handle in the client side.

3.2.1.1 Capture Image

From the client application, two options can be provided to add the image of the child. The first option is capture an image using the mobile device and the second option is to browse an image from the gallery. When capturing images to delegate the action, an Intent can be invoked and the Intent automatically starts an external activity. Then the captured image needs to get from the back in order to process it. This can be done inside the onActivityResult() and the image can be got as a small Bitmap by using the key “data”. Then the image can be got as a byte array in order to encode it to Base64. For that the bitmap should be compressed using ByteArrayOutputStream. By using that byte array, the bitmap can be encoded into base64 and then it should be encoded in JSON format. This JSON encoded data can be sent to the server side as an HTTP request. Once the server receives requests from the client side, the JSON string should be de serialized to a Python object. This string (in utf-8) can be encoded into bytes and then base64 bytes can be decoded to image bytes. Finally, the image can be loaded from the byte string.

3.2.1.2 Gray Scaling

In this scenario, color information is not important. Therefore, gray scaling can be done to reduce noise as well as to eliminate lighting effects on the image. Gray scaling can be done using a color-space conversion method. OpenCV contains more than 150 color-space conversion methods [31] [32]. Among all the methods most widely used and accurate method can be used to convert the image to grayscale. Therefore, cv2.cvtColor function can be used. Along with the input image, CV_RGB2GRAY can be used as the tag.

3.2.1.3 Resize Image

Some images in the sample dataset vary in size. Therefore, a base size can be established to speed up the processing time by producing low data size. For that OpenCV resize function can be used. But when resizing images, should be considered about the aspect ratio of the original image. It should be maintained after resizing the image. Therefore, resize function of imutils can be used to resize the image and the aspect ratio of the original image also can be maintained from it. imutils is a series of convenience functions which can be used for basic image processing functionalities [33].

3.2.1.4 Face detection and facial landmark detection

As explained in the literature review, there are several algorithms for face detection. According to the proposed approach, face detection part handles on the server side. Therefore, among the OpenCV Haar Cascades algorithm and DLib library, Dlib library can be selected based on the publicly available benchmark [34] test results. There are two models for face detection and facial landmark detection in Dlib. Which are Histogram of Oriented Gradients (HOG) and Convolutional neural network (CNN). Since, frontal faces are always used with this scenario and by considering the computational power, DLib Histogram of Oriented Gradients (HOG) can be used to detect faces and determine facial landmarks. HOG is the fastest method on CPU and it works well with frontal faces. Furthermore, it is a lightweight model. Dlib get_frontal_face_detector can be used for face detection and Dlib shape_predictor can be used to estimate the pose of the face and to take form of 68 landmarks. In this scenario, by considering the facial features of the children with Down syndrome, seven landmarks needs to determine which includes the jaw, left eye, right eye, nose, mouth, left eyebrow and right eyebrow. After the determination, these landmarks can be saved separately in a processed image path.

3.2.2 Facial feature extraction phase

Facial feature extraction phase is the most important phase in this approach. As explained in the literature review, there are a variety of techniques for feature extraction. Among these techniques; Local Binary Pattern (LBP) can be used based on the existing research studies and surveys [35]. In the facial image analysis domain, Local Binary Pattern has proven good demonstration of use and better performance when compared with other techniques [36].

As mentioned in the above phase, in order to extract features, detected landmarks should save in a path. Feature extraction can be done separately for each landmark and finally, can be concatenated all the histograms to get final histogram. With LBP, feature extraction is simply a texture extraction. That means, in LBP, features are extracted through texture analysis process.

When calculating LBP of each image, each pixel should be compared with its eight neighbors and subtract the center pixel value. If the value is negative, then encode it to 0 otherwise encode it to 1 [37]. A binary number can be obtained by concatenating all the binary values in clockwise direction. Then LBP values of each pixel of the image are stored in a matrix. Likewise, for each landmark (left eye, right eye, jaw, nose, left eyebrow and right eyebrow), texture extraction should be done. Figure 3.3 shows the texture feature extraction for a right eye.

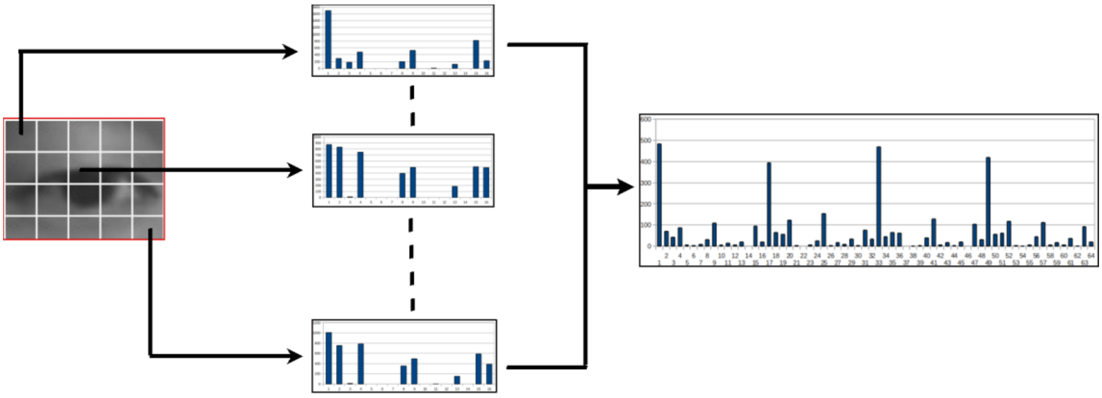


Figure 3.3: Texture extraction for each landmark *[38]*

After determining LBP values for each landmark, these histograms should be concatenated to get the final LBP value as shown in Figure 3.4.

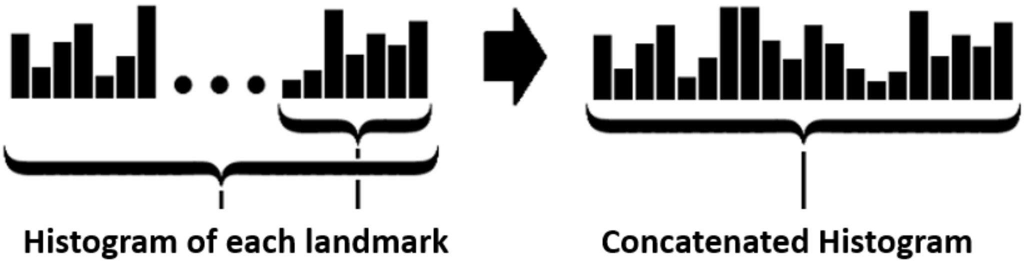


Figure 3.4: Concatenate histograms

3.2.3 Classification phase

After the feature extraction phase, classification process needs to be done. Among the classifiers which are mentioned in the literature review, K-nearest neighbor, Support Vector Machine and Random forest, the most suitable classifier should be chose for the classification. By considering the project domain and based on the existing research studies, Support Vector Machine (SVM) can be selected as the most suitable classifier. Since the training dataset contains a limited set of data, rather than K-nearest neighbor and Random forest, Support Vector Machine is most suitable for this.

Different SVM algorithms have different kernel functions. Therefore, among Linear, Poly, Rbf and Sigmoid classifiers, the most suitable classifier should be selected for the classification. This should be selected based on the accuracy level, precision, recall and f_score. After selecting the classifier, following steps can be followed for the classification.

3.2.3.1 Get trained data

To do the classification, there should be a trained dataset. For that, sample images should be found for both Down syndrome and normal. The accuracy of the data model will be increased with increase in the number of images in sample dataset. By using these sample data, two datasets can be generated for Down syndrome and normal as each dataset have 10 data for each facial landmark (jaw, left eye, right eye, nose, left eyebrow, right eyebrow). When creating datasets, these steps should be followed. Convert image to grayscale, resize image, face detection using DLib library, determine facial landmarks using DLib library and save them separately. After creating the dataset, this dataset should be trained using LBP and saved to a .csv file.

3.2.3.2 Set classifier

Among Linear, Poly, Rbf and Sigmoid classifiers, the selected classifier should be set as the classifier.

3.2.3.3 Train the model

Then the model should be trained with trained labels and trained data.

3.2.3.4 Get probabilities for both normal and Syndrome

Next, as explained in the feature extraction phase, the histogram should be taken from the input image and predicted probabilities should be taken which is the average of the k-folds. For both normal and Down syndrome, the probabilities should be predicted based on the trained dataset.

3.2.3.5 Return results based on highest probability

Finally, the result can be returned based on the highest probability. If the probability of Down syndrome is higher than the probability of normal, then the result should be down syndrome and vice versa.

3.3 SDQ scoring scheme

As explained in the introduction, the SDQ is the globally accepted screening test to identify mental and health problems of children between 4 to 17 years old. There are two main versions in SDQ, which are the baseline version and the follow-up version. Overall, there are 42 questions in the SDQ and these questions are grouped into few groups. Table 3 1 shows the item details in each version.

Table 3-1: SDQ items in each version

	Informant	**Parent**	
	Age range	**4-17**	
	Application	**Baseline**	**Follow-up**
	Rating period	**6 months**	**1 month**
Items	**Item Content**		
1-25	**Symptoms**	✓	✓
26	**Overall**	✓	✓
27	**Duration**	✓	✗
28-33	**Impact**	✓	✓
34-35	**Follow up progress**	✗	✓
36-38	**Cross-Informant information**	✓	✗
39-42	**Cross-Informant information**	✗	✗

After scoring the SDQ questionnaire, as the result 7 scores can be found. Which are total difficulties score, emotional symptom scale, hyperactivity scale, peer problem scale, pro social scale and the impact score. These scores can be calculated based on standard scorning scheme. SDQ items and scores are shown in table 3-2. Also SDQ impact items and scores are shown in table 3-3.

Table 3-2: SDQ Items and scores

		Not True	Some-what True	Certainly True	
Standard Values for Data Entry ========→		**0**	**1**	**2**	
Data element	**SDQ Item number and description**	**Item Score**			**Summary Score**
Emotional Symptoms Scale					0-10
Item 03	Often complains of headaches, ...	0	1	2	
Item 08	Many worries or often seems worried	0	1	2	
Item 13	Often unhappy, depressed or tearful	0	1	2	
Item 16	Nervous or clingy in new situations	0	1	2	
Item 24	Many fears, easily scared	0	1	2	
Conduct Problem Scale					0-10
Item 05	Often loses temper	0	1	2	
Item 07	**Generally well behaved ......**	**2**	**1**	**0**	
Item 12	Often fights with other children	0	1	2	
Item 18	Often lies or cheats	0	1	2	
Item 22	Steals from home, school.......	0	1	2	
Hyperactivity Scale					0-10
Item 02	Restless, overactive....	0	1	2	
Item 10	Constantly fidgeting ...	0	1	2	
Item 15	Easily distracted	0	1	2	
Item 21	**Thinks things out before acting**	**2**	**1**	**0**	
Item 25	**Good attention span, ...**	**2**	**1**	**0**	
Peer Problem Scale					0-10
Item 06	Rather solitary, prefers to play alone	0	1	2	
Item 11	**Has at least one good friend**	**2**	**1**	**0**	
Item 14	**Generally liked by other children**	**2**	**1**	**0**	
Item 19	Picked on or bullied....	0	1	2	
Item 23	Gets along better with adults ...	0	1	2	
Prosocial Scale					0-10
Item 01	Considerate of other people's feelings	0	1	2	
Item 04	Shares readily with other children, ...	0	1	2	
Item 09	Helpful if someone is hurt....	0	1	2	
Item 17	Kind to younger children	0	1	2	
Item 20	Often volunteers to help others ...	0	1	2	
SDQ Total Difficulties Score = Sum of Scales below					0-40
	Emotional Symptoms Scale	0-10			
	Conduct Problem Scale	0-10			
	Hyperactivity Scale	0-10			
	Peer Problem Scale	0-10			

Table 3-3: SDQ Impact items and scores

		Item Responses				
		Not at all	A little	A medium amount	A great deal	
Standard Value for Data Entry ========→		**0**	**1**	**2**	**3**	
Data element	**SDQ Item number and description**	**Item Score**				**Summary score**
Item 28	Difficulties upset or distress child	0	0	1	2	
Item 29	Interfere with HOME LIFE	0	0	1	2	
Item 30	Interfere with FRIENDSHIP	0	0	1	2	
Item 31	Interfere with CLASSROOM LEARNING	0	0	1	2	
Item 32	Interfere with LEISURE ACTIVITIES	0	0	1	2	
SDQ IMPACT SCORE						0-10

When scoring the SDQ through the client application for each question relevant score should be given based on the answers given by the parent. According to the scores given for each item, emotional symptom scale, conduct problem scale, hyperactivity scale, peer problem scale, pro social scale, total difficulties score and the impact score can be calculated respectively, using equation 3-1, equation 3-2, equation 3-3, equation 3-4, equation 3-5 and equation 3-6.

$$\text{Emotional symptom scale} = Item\ 3\ score + Item\ 8\ scor + Item\ 13\ score + Item\ 16\ score + Item\ 24\ score \tag{3-1}$$

$$\text{Conduct problem scale} = Item\ 5\ score + Item\ 7\ score + Item\ 12 score + Item\ 18\ score + Item\ 22\ score \tag{3-2}$$

$$\text{Hyperactivity scale} = Item\ 2\ score + Item\ 10\ score + Item\ 15\ score + Item\ 21\ score + Item\ 25\ score \tag{ 3-3}$$

$$\text{Peer problem scale} = Item\ 6\ score + Item\ 11\ score + Item\ 14\ score + Item\ 19\ score + Item\ 23\ score \tag{ 3-4}$$

$$\text{Total difficulties score} = \text{Emotional Symptom Scale} + \text{Conduct Problem Scale} + \text{Hyperactivity Scale} + \text{Peer Problem Scale} \quad (3\text{-}5)$$

$$\text{Impact score} = Item\ 28\ score + Item\ 29\ score + Item\ 30\ score + Item\ 31\ score + Item\ 32\ score \quad (3\text{-}6)$$

After calculating these scales by applying above equations, based on the range of each scale, it can be examined whether it is normal, abnormal or borderline. This can be categorized according to the table 3-4.

Table 3-4 : Categorizing SDQ scores

Score	**Normal**	**Borderline**	**Abnormal**
Total difficulties score	0-13	14-16	17-40
Emotional symptom scale	0-3	4	5-10
Conduct problem scale	0-2	3	4-10
Hyperactivity scale	0-5	6	7-10
Peer problem scale	0-2	3	4-10
Pro social scale	6-10	5	0-4
Impact score	0	1	2-10

Chapter 4

IMPLEMENTATION

4.1 Chapter overview

In the previous chapter, the proposed approach and the methodology has been explained with the reasons for selecting specific technique for each step. This chapter describes the implementation process phase by phase.

4.2 Dataset Creation

In order to create the dataset, 20 face samples were gathered, including 10 Down syndrome face samples and 10 healthy face samples. Because of the difficulty to find Down syndrome samples with the approval, this sample set was limited to 20. Unless, it's better to use a large sample set to create the dataset.

By using this face sample, two datasets were created for Down syndrome and none syndrome. Each dataset have 10 data for each facial landmark, which includes, jaw, left eye, right eye, nose, left eyebrow and right eyebrow. Datasets for Down syndrome and none Down syndrome are displayed respectively in figure 4.1 and figure 4.2. Dataset creation steps are given below.

Step 1: Convert image to grayscale

In this scenario color information is not important, since facial feature extraction is the main concern. Therefore, gray scaling technology has been used to reduce noise and to eliminate the lightening effects of the image. Gray scaling contains only shades of white and black, therefore, it is ideal to use gray scaling such that it is easy to conduct analytical processes. Besides that, in the face region, gray information can be used easily to extract some features such as eyebrows, pupils and lips appear since these regions are darker that surrounding facial regions [39].

Step 2: Resize image

Some images of the sample dataset vary in size, therefore a base size has been established for all images to speed up the processing time by producing a lower data size [30]. When resizing images, should be consider about the aspect ratio of the original image and should be maintained the same aspect ratio after resizing the image.

Step 3: Face detection using DLib library

As explained in the Literature review there are several ways to detect faces from images. Among the OpenCV Haar Cascades Classifier and DLib, Dlib library has been selected for face detection based on the publicly available benchmark [34] test results. In DLib library there are two models, HOG and CNN. Even though CNN is accurate than HOG, by considering the computational power HOG model has been used for face detection [19].

Step 4: Determine facial landmarks using DLib library and save them separately

In this scenario, always inputs are frontal faces, therefore, the HOG model of the Dlib library has been used to determine facial landmarks [40]. By considering the distinct facial features of children with Down syndrome, seven landmarks have been detected in each image, these are jaw, left eye, right eye, nose, mouth, left eyebrow and right eyebrow.

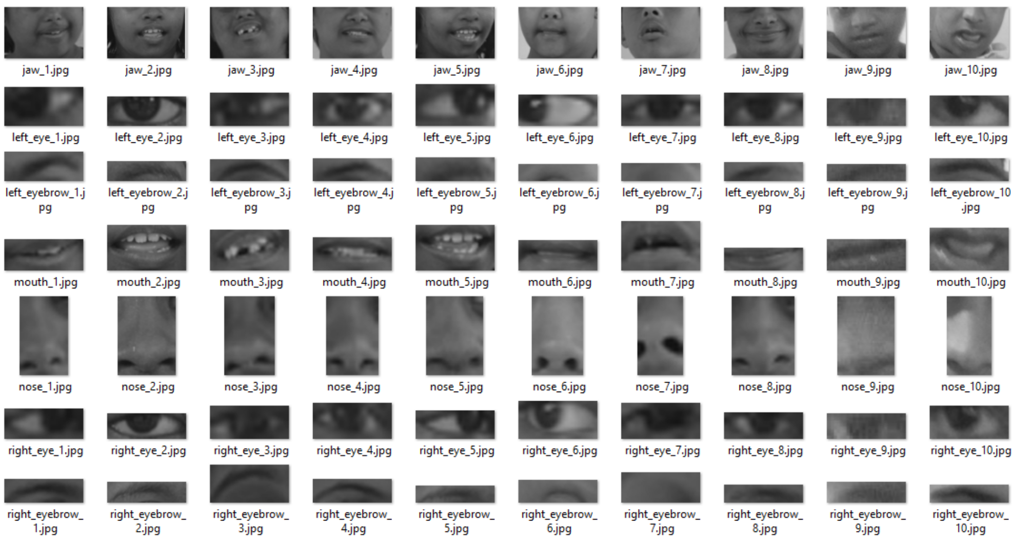


Figure 4.1: Dataset Down syndrome

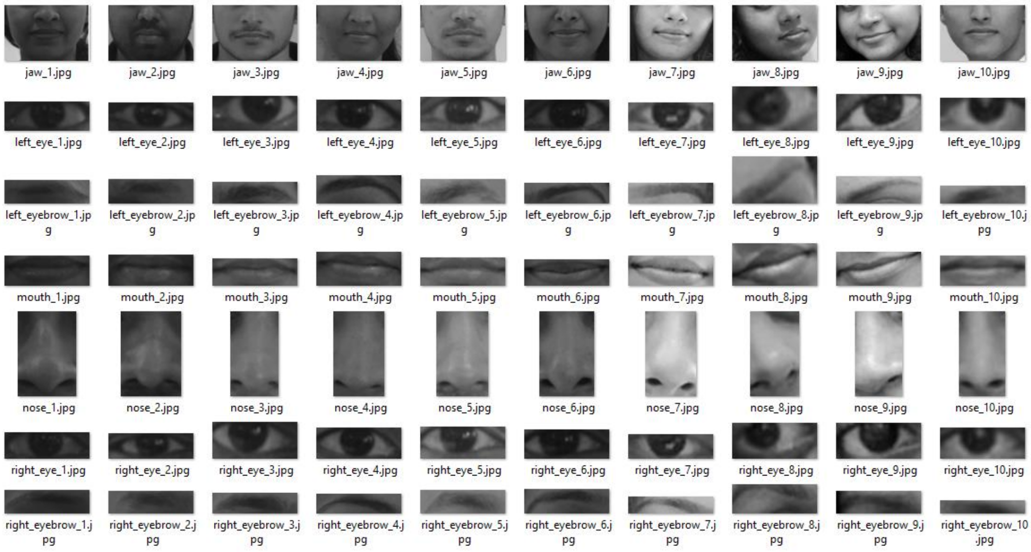


Figure 4.2: Dataset none Down syndrome

4.3 Train Dataset

After creating the dataset, this dataset was trained using LBP. This process includes following steps are as follows:

Step 1: Get images of each landmark separately

Step 2: Calculate local binary pattern (LBP) of each image

Step 3: Normalize LBP values

Step 4: Calculate histograms of each LBP array

Step 5: Concatenate 7 histograms

Step 6: Write these values to a .csv file

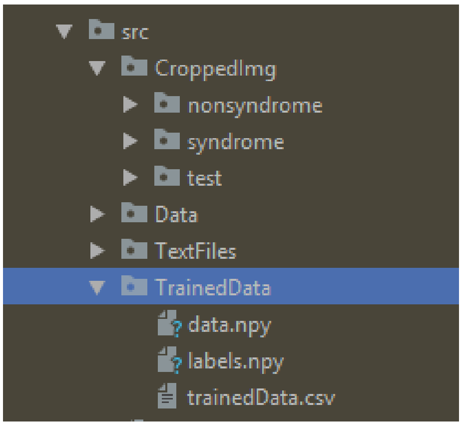


Figure 4.3: Trained dataset

4.4 Implementation of the web service

The web service has been developed using Python Flask as a REST API along with the OpenCV, Dlib library and etc. OpenCV library has a large number of optimized machine learning algorithms [41]. Dlib is also a modem library which contains good machine learning algorithms [42].

The main task of this API is to analyze given images and return result. Clients can interact with the API through a HTTP POST request. This main task (Down syndrome analysis) consists with three sub tasks. Which are face detection, facial feature extraction, classification.

4.4.1 Face detection and facial landmarks determination

Face detection function was implemented according to following sequence of steps.

- Convert the image to grayscale

Gray scaling has been done using a color-space conversion method. OpenCV contains more than 150 color-space conversion methods [31] [32]. Among all the methods most widely used and accurate method has been used to convert the image to grayscale. Along with the input image, CV_RGB2GRAY has been used as the tag. Figure 4.4 shows the relevant source code.

```
# **********Gray scale the image*************
def gray_scale_image(img):
    gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    return gray_image
```

Figure 4.4: Gray scaling code snippet

- Resize the image

The image has been resized to speed up the processing time by producing a lower data size. This can be done using the OpenCV resize function. But as mentioned in the above sections, when resizing images, should be consider about the aspect ratio of the original image and should be maintained the same aspect ratio after resizing the image.

Therefore, resize function of imutils has been used to maintain the aspect ratio of the original image. Figure 4.5 shows the relevant code snippet.

```
# *********Resize the image************
def resize_image(img):
    img_resize = imutils.resize(img, width=500)

    return img_resize
```

Figure 4.5: Image resizing code snippet

- Detect face and determine facial landmarks using Dlib library

get_frontal_face_detector method has been directly used to detect front faces. Also, when using Dlib library, the implemented facial land mark detector can be directly used to determine facial features and these features are mapped with 68 (x, y) coordinates as shown in Figure 4.6.

```
#For dlib's 68-point facial landmark detector:
FACIAL_LANDMARKS_68_IDXS = OrderedDict([
    ("mouth", (48, 68)),
    ("right_eyebrow", (17, 22)),
    ("left_eyebrow", (22, 27)),
    ("right_eye", (36, 42)),
    ("left_eye", (42, 48)),
    ("nose", (27, 36)),
    ("jaw", (0, 17))
])
```

Figure 4.6: Dlib's 68 points

After the determination, landmarks are saved separately in the processed image path as shown in Figure 4.7.

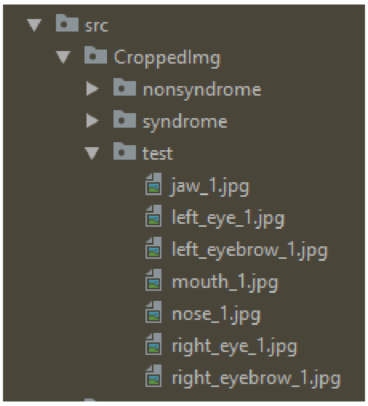


Figure 4.7: Detected landmarks

4.4.2 Facial feature extraction

Facial feature extraction function has been implemented based on the following sequence.

- Get images of each landmark separately

Since, detected landmarks have been saved separately in a specific path, each landmark can accessed as shown Figure 4.8.

```
# path for image of each landmark
path = img_path[0] + img_path[1] + "/" + array[i] + "_" + img_path[2]

if os.path.isfile(path):
    image = cv2.imread(path,cv2.IMREAD_GRAYSCALE)
    height, width = image.shape
    img_lbp = np.zeros((height - 2, width - 2), np.uint64)
    max_lbp = 0
```

Figure 4.8: Get each landmark code

- Calculate local binary pattern (LBP) of each image

When calculating LBP of each image, each pixel has been compared with its eight neighbors and subtract the center pixel value. If the value is negative, then encode it with 0 otherwise encode it with 1 [37]. A binary number can be obtained by concatenating all the binary values in clockwise direction. Then LBP values of each pixel of the image are stored in 62x62 matrix.

```
# calculate lbp values of image
for i in range(0, height - 2):
    for j in range(0, width - 2):
        img_lbp[i, j] = lbp_calculated_pixel_8_1(image, i + 1, j + 1)

        if max_lbp < img_lbp[i][j]:
            max_lbp = img_lbp[i][j]

img_lbp_normalized = np.zeros((height - 2, width - 2), np.float32)
```

Figure 4.8: Calculate LBP code snippet

- Normalize LBP values

These LBP values are normalized by dividing them with the maximum value.

```python
# normalize lbp values
for i in range(height - 2):
    for j in range(width - 2):
        if (max_lbp != 0):
            img_lbp_normalized[i][j] = img_lbp[i][j] / float(max_lbp)
        else:
            img_lbp_normalized[i][j] = img_lbp[i][j]
```

Figure 4.9: Normalize LBP values

- Calculate histograms of each LBP array

Finally, local 59-bin histogram is calculated using this LBP matrix. These histograms give an overall idea about the distribution of the image [43]. cv2.calcHist() function of OpenCV has been used to calculate histograms. Figure 4.11 shows the relevant code.

```python
# calculate histogram
hist = cv2.calcHist([img_lbp_normalized], [0], None, [59], [0, 1])
hist = hist.flatten()
```

Figure 4.10: Calculate histogram

- Concatenate histograms

Finally, 7 histograms have been concatenated to get the final histogram in order to the classification process.

```python
# concatenate histograms of each landmarks
if i == 0:
    feature_hist = hist.tolist()
else:
    feature_hist = feature_hist + hist.tolist()
loop = loop + 1
```

Figure 4.11: Concatenate histograms

4.4.3 Classification

Classification part has been implemented using the SVM classifier. Among Linear, Poly, Rbf and Sigmoid classifiers, Linear SVM classifier has been selected based on the precision, recall and f1-score. Classification report is shown in Figure 4.13.

```
############ Linear ##############

Confusion Matrix:
[[14  1]
 [ 2 13]]

Classification Report:
               precision    recall  f1-score   support

           0       0.88      0.93      0.90        15
           1       0.93      0.87      0.90        15

   micro avg       0.90      0.90      0.90        30
   macro avg       0.90      0.90      0.90        30
weighted avg       0.90      0.90      0.90        30
```

```
############ Poly ##############

Confusion Matrix:
[[15  0]
 [ 6  9]]

Classification Report:
               precision    recall  f1-score   support

           0       0.71      1.00      0.83        15
           1       1.00      0.60      0.75        15

   micro avg       0.80      0.80      0.80        30
   macro avg       0.86      0.80      0.79        30
weighted avg       0.86      0.80      0.79        30
```

```
############ Rbf ##############

Confusion Matrix:
[[ 0 15]
 [ 0 15]]

Classification Report:
               precision    recall  f1-score   support

           0       0.00      0.00      0.00        15
           1       0.50      1.00      0.67        15

   micro avg       0.50      0.50      0.50        30
   macro avg       0.25      0.50      0.33        30
weighted avg       0.25      0.50      0.33        30
```

```
############ Sigmoid ##############

Confusion Matrix:
[[ 0 15]
 [ 0 15]]

Classification Report:
               precision    recall  f1-score   support

           0       0.00      0.00      0.00        15
           1       0.50      1.00      0.67        15

   micro avg       0.50      0.50      0.50        30
   macro avg       0.25      0.50      0.33        30
weighted avg       0.25      0.50      0.33        30
```

Figure 4.12: Classifier selection report

When implementing the classifier, initially the selected classifier has been trained with the trained data and labels. Then the probabilities have been predicted using the histogram of the input image and this has been done using cross validation. Then the probabilities for both Normal and Syndrome have been taken using the predicted probability. The final result has been generated based on the highest probability. If the probability of normal is greater than the probability of syndrome then the result has been generated as normal.

The steps of this process can be listed down as below.

- Get trained data

```
# get trained data
file_train_data = os.path.realpath("venv\\src\\TrainedData\\data.npy")
file_train_labels = os.path.realpath("venv\\src\\TrainedData\\labels.npy")
```

Figure 4.13: Get trained data

- Set classifier

```
# classifier
svclassifier = SVC(kernel='linear',probability=True,C=100) # Linear Kernal
```

Figure 4.14: Set classifier

- Train the model

```
# train the model
svclassifier.fit(X, y)
```

Figure 4.15: Train the model

- Get histogram of the input image to test

```
# histogram of given image to test
hist = np.asarray(hist)
hist = hist.reshape(1, -1)
```

Figure 4.16: Get histogram of the input image

- Predict probabilities which is the average of the k-folds [44]

```
# predicted probabilities which is the average of the k-folds
y_pred = svclassifier.predict(hist)
prob = svclassifier.predict_proba(hist)
```

Figure 4.17: Predict probabilities

- Get the probabilities for both Normal and Syndrome

```
normal_prob = prob[0][0]
syndrome_prob = prob[0][1]
```

Figure 4.18: Get probabilities of normal and syndrome

- Return the result based on the highest probability

```
if normal_prob >= syndrome_prob:
    label = 'Normal'
    percentage = float("{:.2f}".format(normal_prob * 100))
else:
    label = 'Syndrome'
    percentage = float("{:.2f}".format(syndrome_prob * 100))

print("normal={:.2f}%, syndrome={:.2f}%".format(normal_prob * 100, syndrome_prob * 100))

output = {'label': label,
          'percentage': percentage}


return output
```

Figure 4.19: Result based on max probability

4.5 Implementation of the client application

The client mobile application has been developed as an Android application. This application consists with three main components, these are Down syndrome detection test, SDQ test and progress evaluation based on SDQ results. From these components, Down syndrome detection test interacts with the python web service. This communication happens through Hypertext Transfer Protocol (HTTP). In addition to that, all the components interact with the database on the web server. And here, PHP and the SQL used to retrieve data from the server. For making connection to the PHP script, HTTP protocol has been used from the android application.

4.5.1 Children registration and Login

Parents can register with the application by providing the child's information.

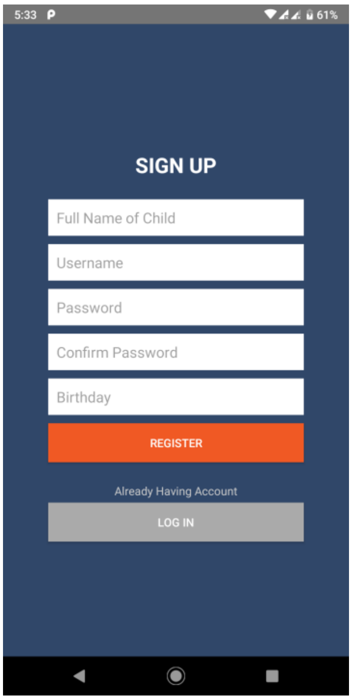


Figure 4.21: Register Interface

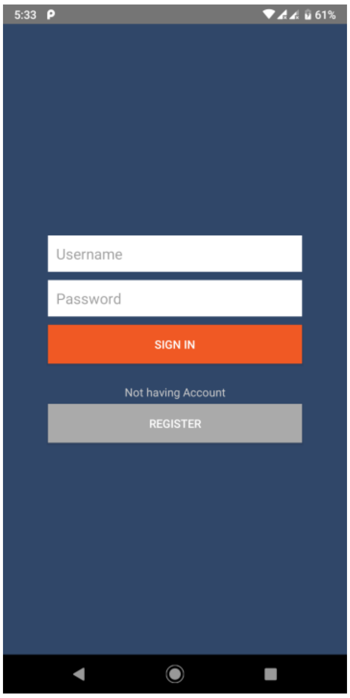


Figure 4.20: Login Interface

4.5.2 Down syndrome detection test

Figure 4.23: Detector test Interface

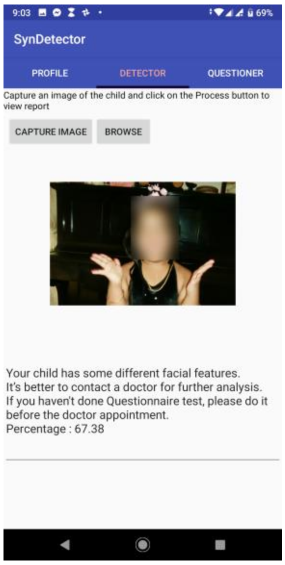


Figure 4.22: Display results of the Detector

This is the main functionality of this application. Parents can capture or browse an image of the child, then the application encodes the image into JSON format and passes it to the web service as a HTTP request and gets the response from the web service.

4.5.3 Strengths and Difficulties Questionnaire

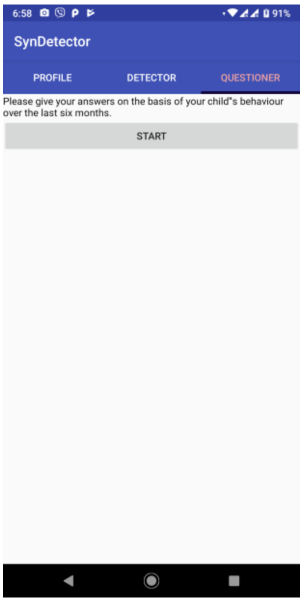


Figure 4.24: Questionnaire Interface

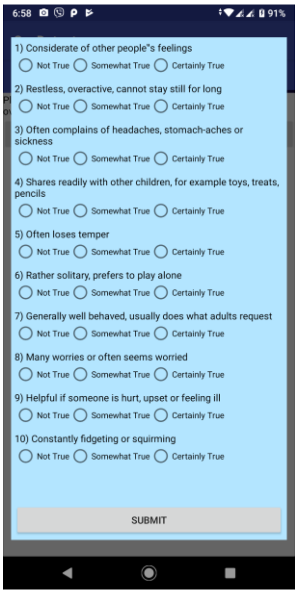


Figure 4.26: SDQ

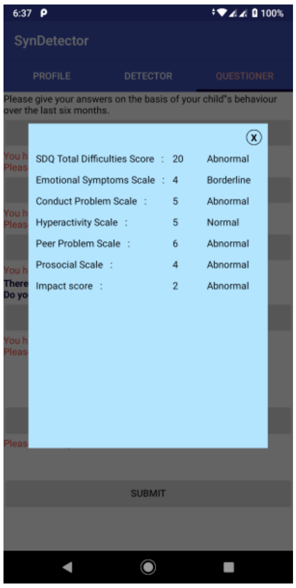


Figure 4.25: SDQ Results

This is the globally accepted screening test to identify mental and health problems of children between 4 to 17 years old. Initially parents can give answers based on the child's behavior over the last six months. In the first attempt, the application gives the basic version of the SDQ. From the second attempt, the application gives the follow up version of the SDQ. After complete the questionnaire, the application shows the result which includes total difficulties score, emotional symptom scale, hyperactivity scale, peer problem scale, pro social scale and the impact score by examining whether the scores are normal, borderline or abnormal based on the standard scoring scheme as shown in Figure 4.25.

4.5.4 Child profile and progress evaluation

When login to the application parents always can view the child profile which includes all the historical records from the registration date.

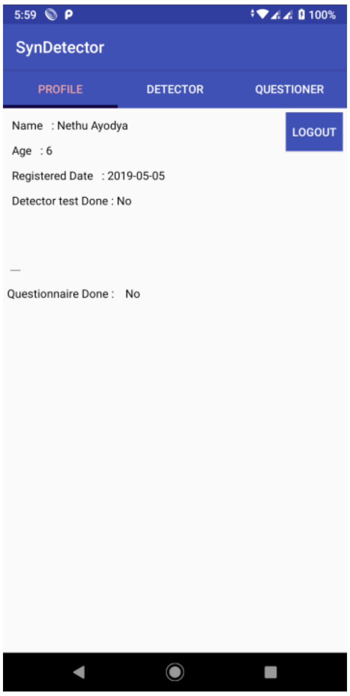


Figure 4.27: Profile before doing detector test and SDQ

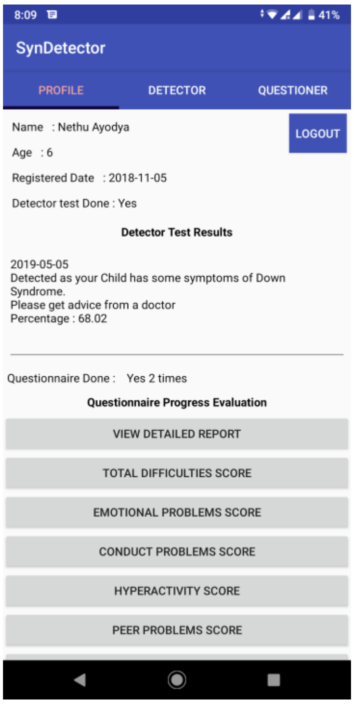


Figure 4.28: Profile after doing detector test and SDQ

From the second attempt of the SDQ, there is a progress evaluation tool. In that tool there is a graphical representation of the SDQ result and also parents and doctors can view a detail report of the child.

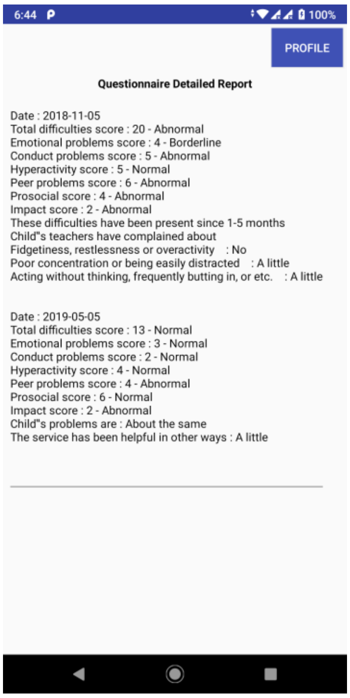


Figure 4.30: SDQ detailed report

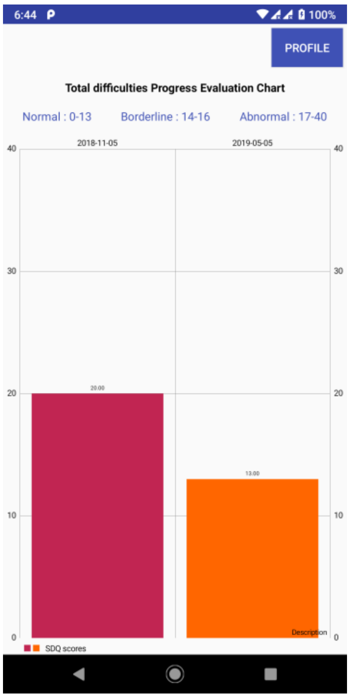


Figure 4.29: Progress evaluation chart - Total difficulties

Chapter 5

EVALUATION

5.1 Chapter overview

The Implemented mobile application is evaluated in this chapter. Since there are three main functionalities in this application, each functionality has been evaluated separately.

5.2 Evaluation of the Web Service

The web service has been evaluated using 30 face samples which include 15 Down syndrome face samples and 15 normal face samples. By using these face samples, a test dataset has been generated. Finally, using the implemented data model an evaluation report has been taken for the test dataset values, which includes all the performance measurements such as confusion matrix, accuracy, precision, recall and f-score. The screenshot of the evaluation report given in the Figure 5.1 and these performance measurements are described in the following sections.

```
################## Evaluation Report #####################

Confusion Matrix:
[[14  1]
 [ 2 13]]

Accuracy Score:
0.9

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.93      0.90        15
           1       0.93      0.87      0.90        15

   micro avg       0.90      0.90      0.90        30
   macro avg       0.90      0.90      0.90        30
weighted avg       0.90      0.90      0.90        30
```

Figure 5.1: Web service evaluation report

5.2.1 Confusion matrix

In the evaluation report of the web service, confusion matrix has been taken to identify the number of true positives, true negatives, false positives and false negatives. The confusion matrix consists with four different combinations of actual values and predicted values related to two classes Down syndrome and normal [45] [46]. Table 5-1 shows the confusion matrix.

Table 5-1: Confusion Matrix

	Predicted Value		
Actual Value		Positive (Down Syndrome)	Negative (Normal)
	Positive (Down Syndrome)	14	1
	Negative (Normal)	2	13

According to the above confusion matrix following information can be gathered.

- True Positives (TP) – 14

This is the value, which the web service predicted as Down syndrome and actually these faces are Down syndrome. There were 15 Down syndrome face samples in the test dataset, among these samples 14 faces were identified as Down syndrome and 1 face was identified as normal. Therefore the number of true positives is 14.

- True Negatives (TN) – 13

This is the value, which the web service predicted as Normal and actually these faces are normal. There were 15 normal face samples in the test dataset, among these samples 13 faces were identified as normal and 2 faces were identified as Down syndrome. Therefore the number of true negatives is 13.

- False Positives (FP) – 2

This is the value, which the web service predicted as Down syndrome, but actually these faces are normal. As given above, there were 15 normal syndrome face samples in the test dataset, among these samples 2 face were identified as Down syndrome. Therefore the number of false positives is 2.

- False Negatives (FN) – 1

This is the value, which the web service predicted as Normal but actually these faces are Down syndrome. As given above, there were 15 Down syndrome face samples in the test dataset, among these samples 1 face was identified as normal. Therefore, the number of false negatives is 1.

5.2.2 Accuracy

Accuracy of the web service has been taken, since it is the most important performance measurement of this type of implementations. It is the ratio of the number of correct predictions to the total number of test data. Accuracy of the web service can be evaluated as follows [45], based on the results of the confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5-1}$$

$$= \frac{14 + 13}{14 + 13 + 2 + 1} = 0.9$$

$$= 90\%$$

It has proven that as a percentage, the accuracy of the web service is 90%. Which means, the web service has 90% of correct predictions [47]. It is better, if the web service can achieve 100% or closer value. But, 90% is also a considerable value.

5.2.3 Precision

The precision of the web service has been taken, since it is an important performance measurement of this type of implementations. It is the ratio of the number of correct positive (Down syndrome) predictions to the total number of positive (Down syndrome) predictions. For a good data model it should be high as possible, since if the precision is high, it causes to low false positives. The precision of the web service can be evaluated as follows, based on the results of the confusion matrix.

$$Precision = \frac{TP}{TP + FP} \tag{5-2}$$

$$= \frac{14}{14 + 2} = 0.875$$

$$= 87.5\%$$

It has proven that as a percentage the precision of the web service is 87.5%. Which means, if the web service predicted as Down syndrome, there is an 87.5% probability, the prediction is correct [47]. This is a considerably good value for the precision.

5.2.4 Recall

Recall is also an important performance measurement since it is the ratio of the number of correct positive (Down syndrome) predictions to the total number of actual positive (Down syndrome) data in the test dataset. For a good data model it also should be high as possible. Recall of the web service can be evaluated as follows, based on the results of the confusion matrix.

$$Recall = \frac{TP}{TP + FN} \tag{5-3}$$

$$= \frac{14}{14 + 1} = 0.933$$

$$= 93.3\%$$

It has proven that as a percentage the recall of the web service is 93.3%. Which means, if the actual value is Down syndrome, there is a 93.3% probability, the web service predicts it correctly as Down syndrome [47]. This is a good value for recall.

5.2.5 F1-score

"F1-score is the weighted average of the precision and recall " [48]. The best value for f1-score is 1 and worst value is 0. F-score of the web service can be evaluated as follows, based on the results of the confusion matrix.

$$F_{score} = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5-4}$$

$$\frac{2 \times 0.933 \times 0.875}{0.933 + 0.875} = 0.903$$

$$= 90.3\%$$

It has proven that as a percentage the F-score of the web service is 90.3%. This is a good value for F1-score.

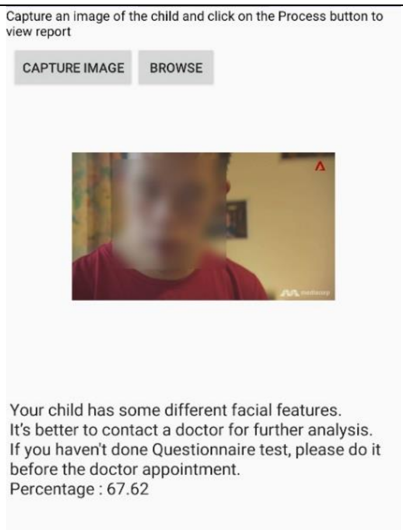
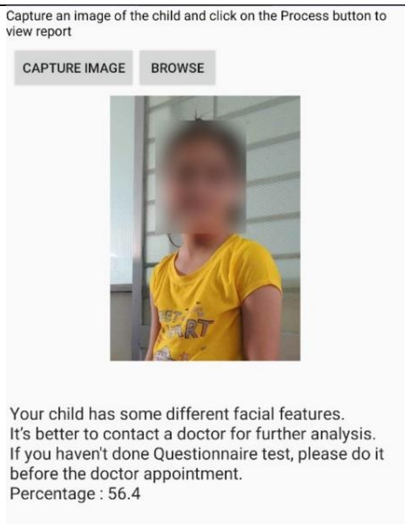
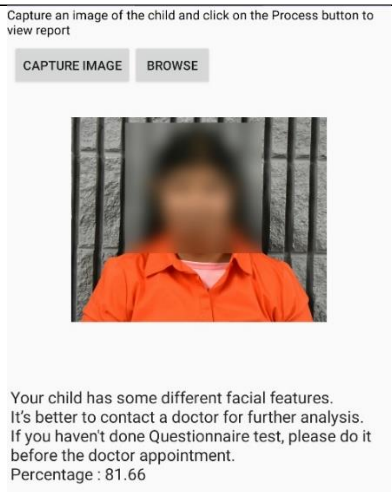
5.3 Evaluation of the Mobile Application

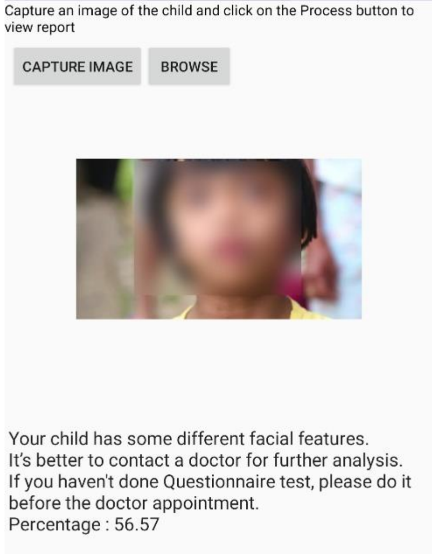
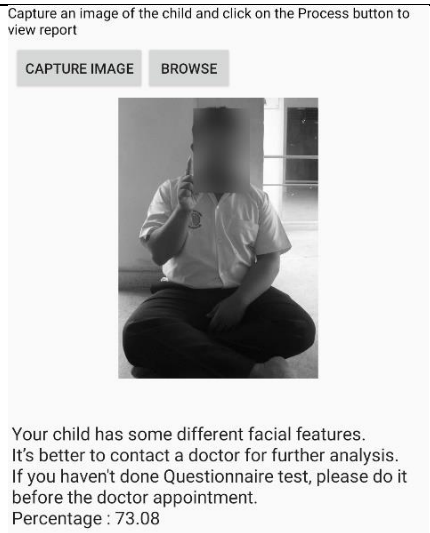
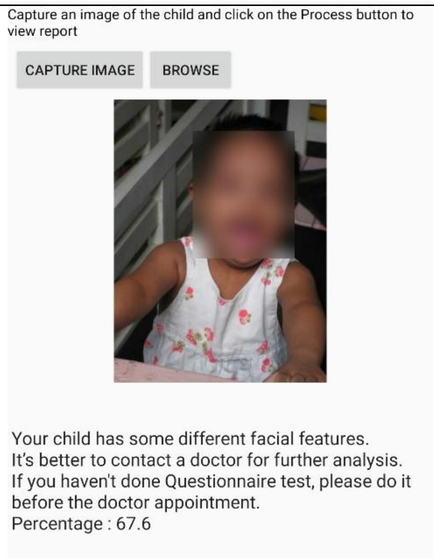
There are three main functionalities in the mobile application which are Down syndrome detection test, SDQ and progress evaluation. These functionalities are evaluated as below.

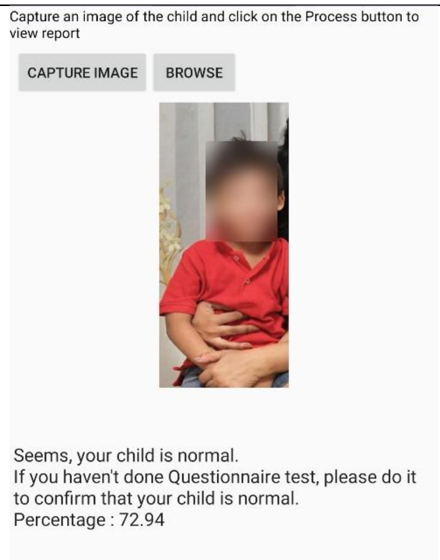
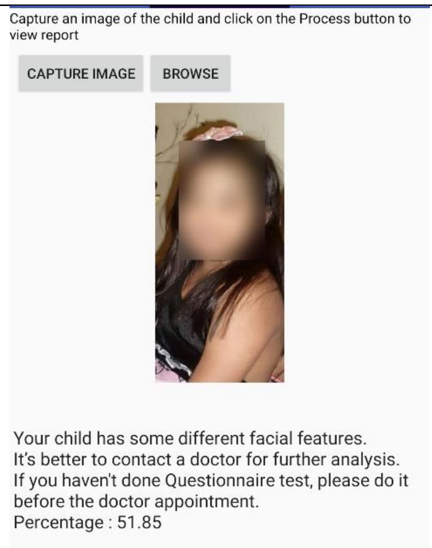
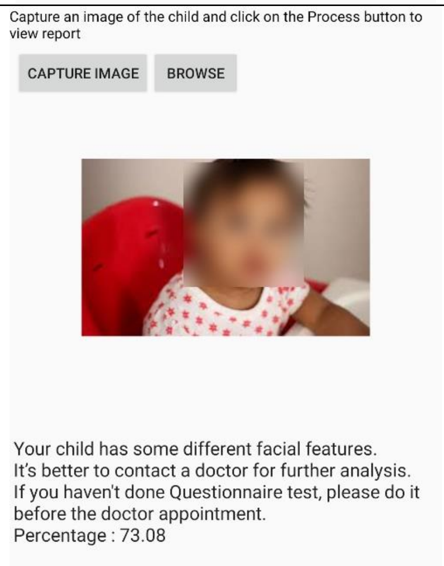
5.3.1 Evaluation of Down syndrome detection test

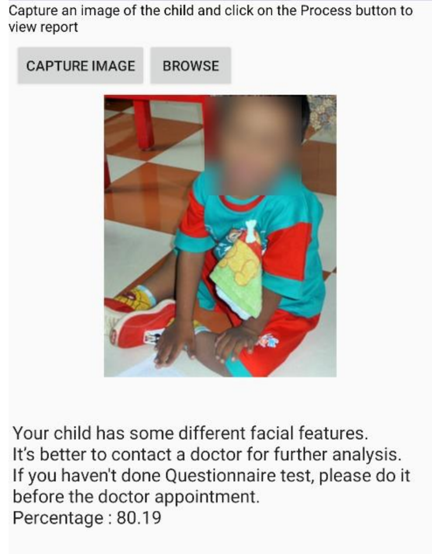
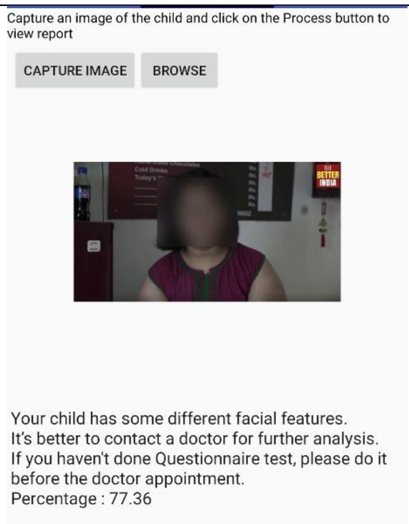
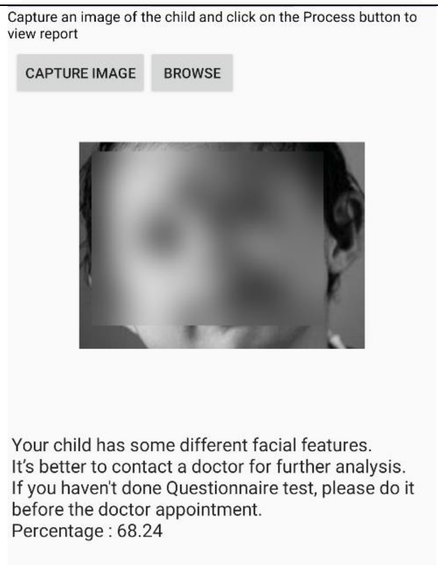
Down syndrome detection test has been evaluated using 30 face images which are not included in the training dataset. This image set includes 15 Down syndrome images and 15 normal images.

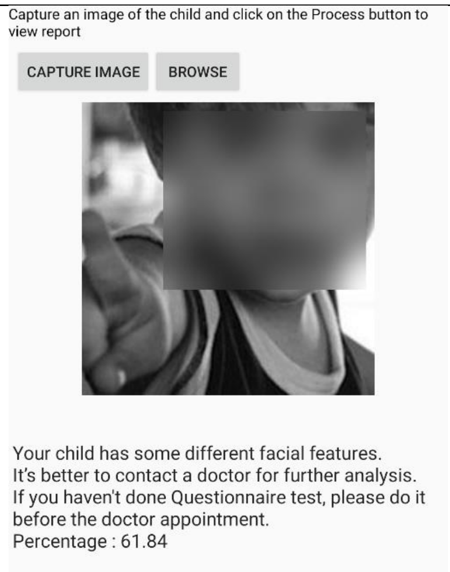
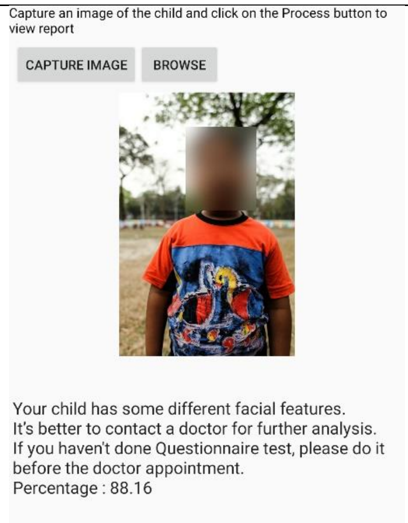
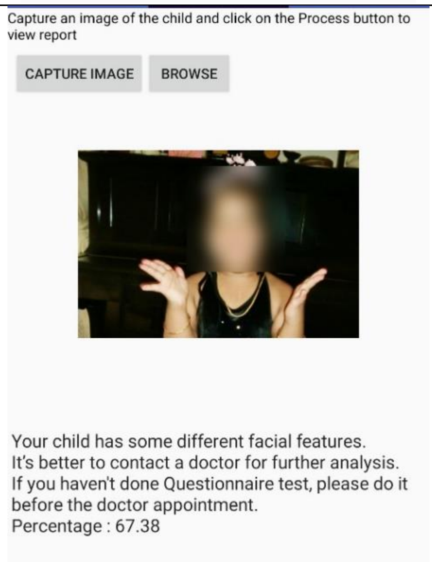
Table 5-2: Detection test results

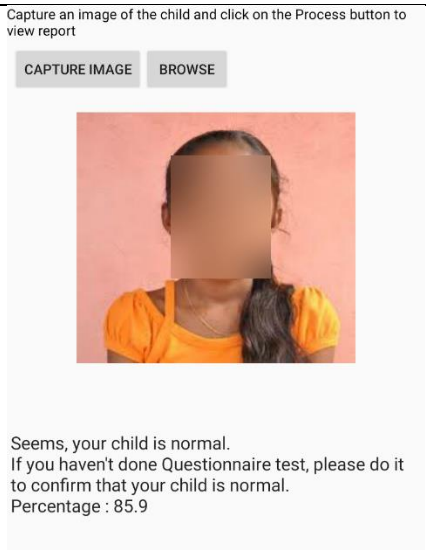
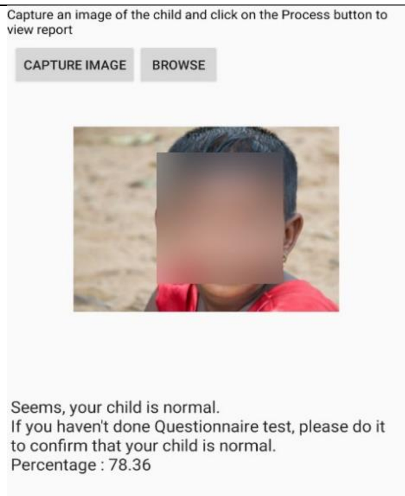
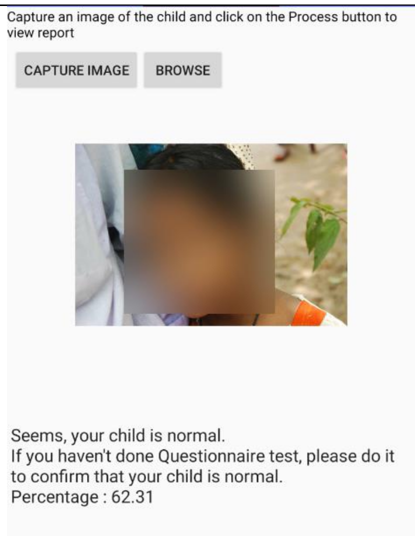
Image ID	Actual status	App Results	Percentage	Screenshot
1	Down Syndrome	Down Syndrome	67.62	
2	Down Syndrome	Down Syndrome	56.4	
3	Down Syndrome	Down Syndrome	81.66	

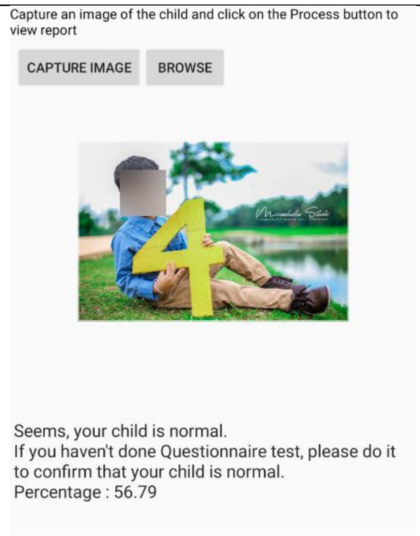
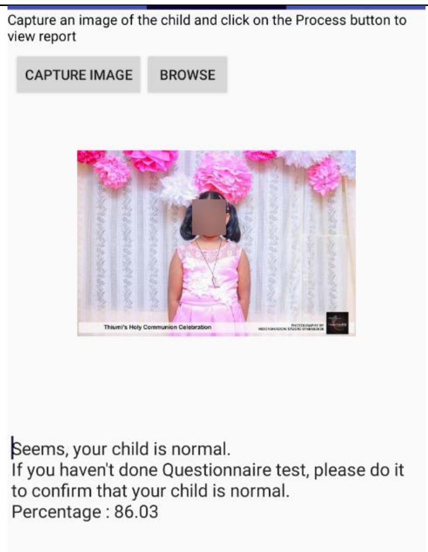
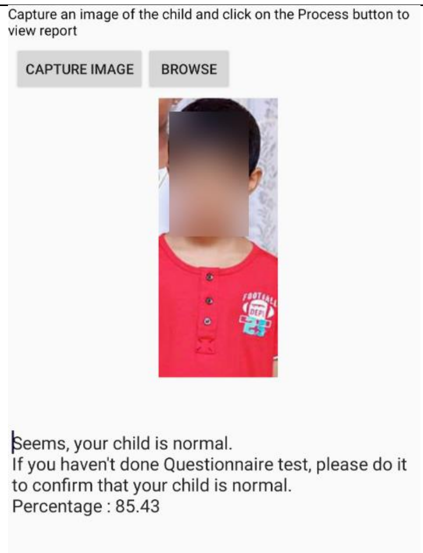
4	Down Syndrome	Down Syndrome	56.57	
5	Down Syndrome	Down Syndrome	73.08	
6	Down Syndrome	Down Syndrome	67.6	

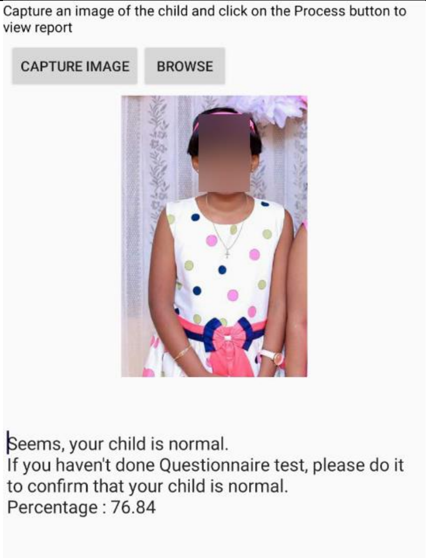
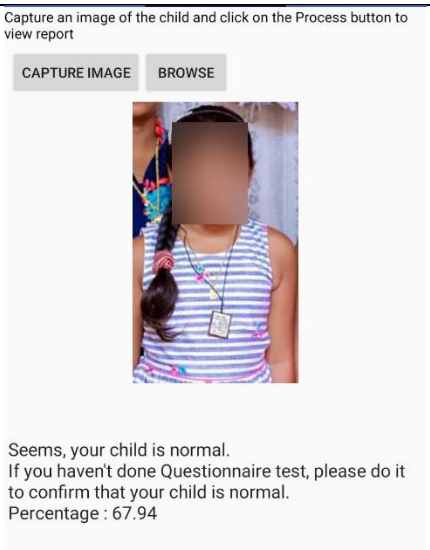
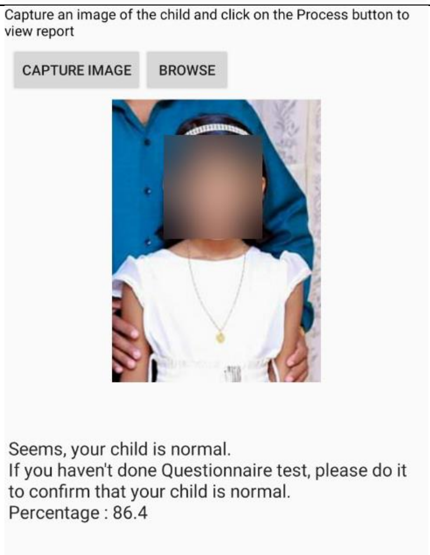
7	Down Syndrome	Normal	72.94	
8	Down Syndrome	Down Syndrome	51.85	
9	Down Syndrome	Down Syndrome	73.08	

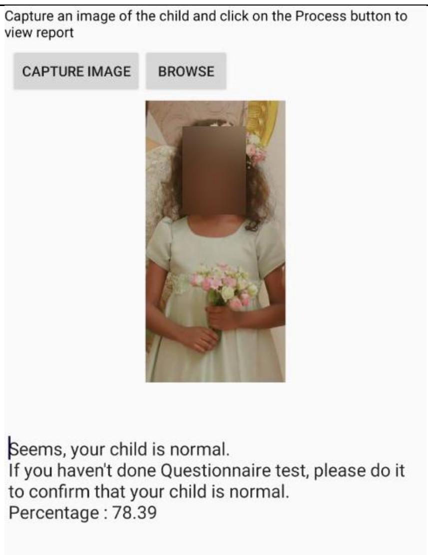
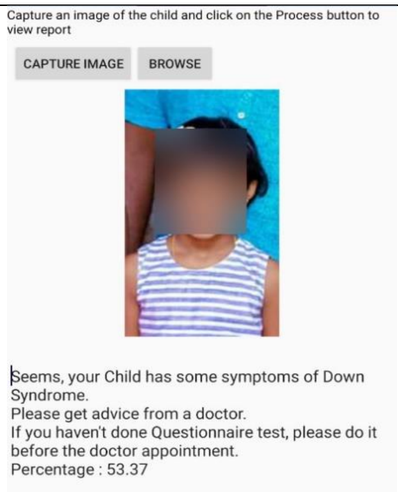
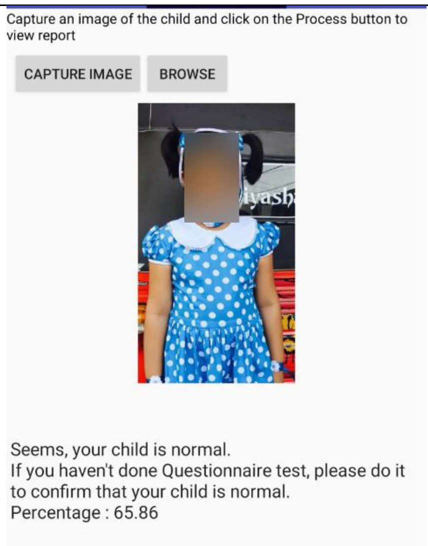
10	Down Syndrome	Down Syndrome	80.19	
11	Down Syndrome	Down Syndrome	77.36	
12	Down Syndrome	Down Syndrome	68.24	

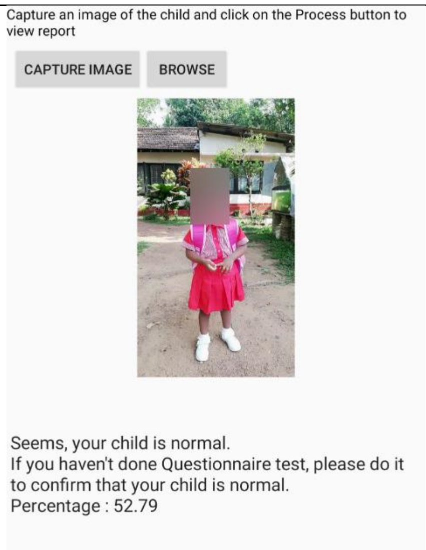
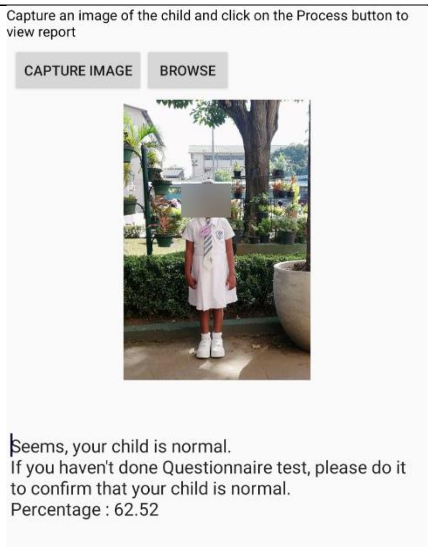
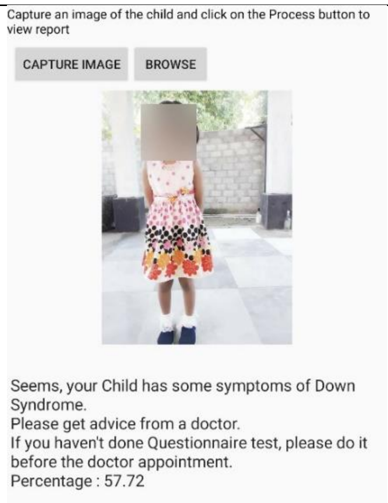
13	Down Syndrome	Down Syndrome	61.84	
14	Down Syndrome	Down Syndrome	88.16	
15	Down Syndrome	Down Syndrome	67.38	

16	Normal	Normal	85.9	
17	Normal	Normal	78.36	
18	Normal	Normal	62.31	

19	Normal	Normal	56.79	
20	Normal	Normal	86.03	
21	Normal	Normal	85.43	

22	Normal	Normal	76.84	
23	Normal	Normal	67.94	
24	Normal	Normal	86.4	

<table>
<tr><td>25</td><td>Normal</td><td>Normal</td><td>78.39</td><td>
</td></tr>
<tr><td>26</td><td>Normal</td><td>Down Syndrome</td><td>53.37</td><td>
</td></tr>
<tr><td>27</td><td>Normal</td><td>Normal</td><td>65.86</td><td>
</td></tr>
</table>

<table>
<tr><td>28</td><td>Normal</td><td>Normal</td><td>52.79</td><td>
</td></tr>
<tr><td>29</td><td>Normal</td><td>Normal</td><td>62.52</td><td>
</td></tr>
<tr><td>30</td><td>Normal</td><td>Down Syndrome</td><td>57.72</td><td>
</td></tr>
</table>

According to the results given in Table 5-2, evaluation is shown in Table 5-3.

Table 5-3: Detector test evaluation

	Number of Images	Number of correct result	Correct percentage	Number of incorrect result	Incorrect percentage
Down Syndrome	15	14	93.33%	1	6.67%
Normal	15	13	86.67%	2	13.33%
Overall	30	27	90%	3	10%

5.3.2 Evaluation of SDQ Scoring mechanism

The implemented SDQ scoring mechanism has been evaluated using 30 SDQ samples which have been scored according to the existing manual process. Results of the manual process have been compared with results of the application. If there is a difference between two results, correct result has been decided based on a doctor approval.

5.3.2.1 Evaluate emotional problems score

Table 5-4: Emotional problems score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	4	4	-	Correct
2	3	3	-	Correct
3	5	5	-	Correct
4	4	4	-	Correct
5	4	4	-	Correct
6	6	5	5	Correct
7	3	3	-	Correct
8	4	4	-	Correct
9	4	4	-	Correct
10	5	5	-	Correct
11	5	5	-	Correct
12	6	6	-	Correct

13	6	6	-	Correct
14	2	2	-	Correct
15	3	3	-	Correct
16	3	3	-	Correct
17	4	4	-	Correct
18	5	5	-	Correct
19	3	3	-	Correct
20	4	4	-	Correct
21	4	4	-	Correct
22	6	6	-	Correct
23	5	5	-	Correct
24	2	2	-	Correct
25	2	2	-	Correct
26	4	4	-	Correct
27	5	5	-	Correct
28	4	4	-	Correct
29	6	6	-	Correct
30	3	3	-	Correct

According to the table 5-4, accuracy of the emotional problem score can be calculated as follows.

$$Accuracy\ of\ emotional\ problem\ score = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100\%$$
(5-5)

$$Accuracy\ of\ emotional\ problem\ score = 100\%$$

5.3.2.2 Evaluate conduct problem score

Table 5-5: Conduct problem score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	3	3	-	Correct
2	3	3	-	Correct
3	3	3	-	Correct
4	2	2	-	Correct
5	5	5	-	Correct
6	3	3	-	Correct
7	4	4	-	Correct
8	5	5	-	Correct
9	3	3	-	Correct
10	5	5	-	Correct
11	4	4	-	Correct
12	2	2	-	Correct
13	3	3	-	Correct
14	2	2	-	Correct
15	4	4	-	Correct
16	3	3	-	Correct
17	3	3	-	Correct
18	4	4	-	Correct
19	3	3	-	Correct
20	5	5	-	Correct
21	2	2	-	Correct
22	2	2	-	Correct
23	3	3	-	Correct
24	5	5	-	Correct
25	5	5	-	Correct
26	3	3	-	Correct

27	6	6	-	Correct
28	2	2	-	Correct
29	2	2	-	Correct
30	3	3	-	Correct

According to the table 5-5, accuracy of the conduct problem score can be calculated as follows.

$$Accuracy\ of\ conduct\ problem\ score = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100\% \quad (5\text{-}6)$$

$$Accuracy\ of\ conduct\ problem\ score = 100\%$$

5.3.2.3 Evaluate hyperactivity score

Table 5-6: hyperactivity score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	7	7	-	Correct
2	5	5	-	Correct
3	4	4	-	Correct
4	6	6	-	Correct
5	6	6	-	Correct
6	5	5	-	Correct
7	8	8	-	Correct
8	2	3	3	Correct
9	6	6	-	Correct
10	3	3	-	Correct
11	8	8	-	Correct
12	5	5	-	Correct
13	9	9	-	Correct
14	6	6	-	Correct

15	5	5	-	Correct
16	6	6	-	Correct
17	4	4	-	Correct
18	5	5	-	Correct
19	5	6	6	Correct
20	4	4	-	Correct
21	6	6	-	Correct
22	8	8	-	Correct
23	3	3	-	Correct
24	6	6	-	Correct
25	6	6	-	Correct
26	5	5	-	Correct
27	3	3	-	Correct
28	6	6	-	Correct
29	5	5	-	Correct
30	6	6	-	Correct

According to the table 5-6, accuracy of the hyperactivity score can be calculated as follows.

$$Accuracy\ of\ hyperactivity\ scale = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100\% \qquad (5\text{-}7)$$

$$Accuracy\ of\ hyperactivity\ scale = 100\%$$

5.3.2.4 Evaluate peer problem score

Table 5-7: Peer problem score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	3	3	-	Correct
2	5	5	-	Correct
3	3	3	-	Correct
4	2	2	-	Correct
5	2	2	-	Correct
6	2	2	-	Correct
7	5	5	-	Correct
8	2	2	-	Correct
9	3	3	-	Correct
10	4	4	-	Correct
11	3	3	-	Correct
12	2	2	-	Correct
13	5	5	-	Correct
14	4	4	-	Correct
15	6	6	-	Correct
16	3	3	-	Correct
17	2	2	-	Correct
18	2	2	-	Correct
19	3	3	-	Correct
20	2	2	-	Correct
21	3	3	-	Correct
22	2	2	-	Correct
23	4	4	-	Correct
24	3	3	-	Correct
25	3	3	-	Correct
26	3	3	-	Correct

27	3	3	-	Correct
28	2	2	-	Correct
29	2	2	-	Correct
30	3	3	-	Correct

According to the table 5-7, accuracy of the peer problem score can be calculated as follows.

$$Accuracy\ of\ peer\ problem\ scale = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100\% \qquad (5\text{-}8)$$

$$Accuracy\ of\ peer\ problem\ scale = 100\%$$

5.3.2.5 Evaluate pro social score

Table 5-8: Pro social score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	6	6	-	Correct
2	7	7	-	Correct
3	5	5	-	Correct
4	5	5	-	Correct
5	6	6	-	Correct
6	3	3	-	Correct
7	5	5	-	Correct
8	6	6	-	Correct
9	7	7	-	Correct
10	4	4	-	Correct
11	6	6	-	Correct
12	6	6	-	Correct
13	6	6	-	Correct
14	5	5	-	Correct

15	4	4	-	Correct
16	6	6	-	Correct
17	5	5	-	Correct
18	5	5	-	Correct
19	5	5	-	Correct
20	7	7	-	Correct
21	7	7	-	Correct
22	7	7	-	Correct
23	5	5	-	Correct
24	8	8	-	Correct
25	6	6	-	Correct
26	4	4	-	Correct
27	5	5	-	Correct
28	7	7	-	Correct
29	6	6	-	Correct
30	2	2	-	Correct

According to the table 5-8, accuracy of the pro social score can be calculated as follows.

$$Accuracy\ of\ pro\ social\ scale = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100\% \qquad (5\text{-}9)$$

$$Accuracy\ of\ ppro\ social\ scale = 100\%$$

5.3.2.6 Evaluate total difficulties score

Table 5-9: Total difficulties score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	17	17	-	Correct
2	16	16	-	Correct
3	15	15	-	Correct
4	14	14	-	Correct
5	17	17	-	Correct
6	16	15	15	Correct
7	20	20	-	Correct
8	13	14	14	Correct
9	16	16	-	Correct
10	17	17	-	Correct
11	20	20	-	Correct
12	15	15	-	Correct
13	23	23	-	Correct
14	14	14	-	Correct
15	18	18	-	Correct
16	15	15	-	Correct
17	13	13	-	Correct
18	16	16	-	Correct
19	14	15	15	Correct
20	15	15	-	Correct
21	15	15	-	Correct
22	18	18	-	Correct
23	15	15	-	Correct
24	16	16	-	Correct
25	16	16	-	Correct
26	15	15	-	Correct

27	17	17	-	Correct
28	14	14	-	Correct
29	15	15	-	Correct
30	15	15	-	Correct

According to the table 5-9, accuracy of the total difficulties score can be calculated as follows.

$$Accuracy\ of\ total\ difficulties\ score = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100\% \ldots\ldots\ldots\ldots\ldots\ldots(5\text{-}10)$$

Accuracy of $total\ difficulties\ score = 100\%$

5.3.2.7 Evaluate impact score

Table 5-10: Impact score evaluation

SDQ ID	Result -Manual	Result - Application	If difference, correct result	App result Correct/Wrong
1	0	0	-	Correct
2	1	1	-	Correct
3	5	5	-	Correct
4	3	3	-	Correct
5	7	7	-	Correct
6	0	0	-	Correct
7	8	8	-	Correct
8	1	1	-	Correct
9	0	0	-	Correct
10	0	0	-	Correct
11	0	0	-	Correct
12	0	0	-	Correct
13	0	0	-	Correct
14	0	0	-	Correct
15	5	5	-	Correct

16	6	6	-	Correct
17	3	3	-	Correct
18	1	1	-	Correct
19	3	3	-	Correct
20	1	1	-	Correct
21	0	0	-	Correct
22	1	1	-	Correct
23	0	0	-	Correct
24	1	1	-	Correct
25	0	0	-	Correct
26	0	0	-	Correct
27	5	5	-	Correct
28	8	8	-	Correct
29	1	1	-	Correct
30	0	0	-	Correct

According to the table 5-10, accuracy of the impact score can be calculated as follows.

$$Accuracy\ of\ impact\ score\ = \frac{Number\ of\ correct\ results}{Total\ number\ of\ results} \times 100 \qquad (5\text{-}11)$$

Accuracy of $impact\ score\ = 100\%$

5.3.3 Evaluation of SDQ and Progress evaluation functionalities based on feedbacks

SDQ and progress evaluation functionalities are evaluated through questionnaires given to two parties, doctors and parents. There are two separate questionnaires for two parties and for each party five people were selected.

5.3.3.1 Feedbacks from Doctors

There are six MCQ questions in this questionnaire and each question has following five answers.

1. Strongly agreed
2. Agreed
3. Slightly agreed
4. Slightly disagreed
5. Disagreed

The questions are listed below.

Q1 : The SDQ Scoring mechanism is accurate and it is aligned with the standard SDQ scoring scheme.

Q2 : This application is more convenient to use, rather than the existing manual process.

Q3 : Progress evaluation graphs and detailed report helpful, because it displays the summary of the patient’s history.

Q4 : Do you recommend this application for parents of children under care?

Q5 : The interfaces of the application are user friendly.

Q6 : Does it need any improvements in future? If agreed, comment your suggestions.

Table 5-11: Feedbacks from doctors

User	Q1	Q2	Q3	Q4	Q5	Q6
1	1	2	1	1	2	2
2	2	2	2	2	3	2
3	1	1	1	2	2	2
4	1	3	3	3	4	2
5	1	1	2	2	3	2
Average Value	1.2	1.8	1.8	2	2.8	2

Graphical representation of the average value with respect to the question number shown in Figure 5.2.

According to the Figure 5.2, it can be said that SDQ scoring mechanism is accurate and it has aligned with the standard SDQ scoring scheme. Also, seems this application is more efficient than the existing manual process and also the progress evaluation functionality is useful for their process to keep track of history of patients and to analyze the progress of patients. In addition to the MCQ answers, some comments have been written by doctors. As a summary, most of them suggested to add SDQ in Sinhala in addition to English.

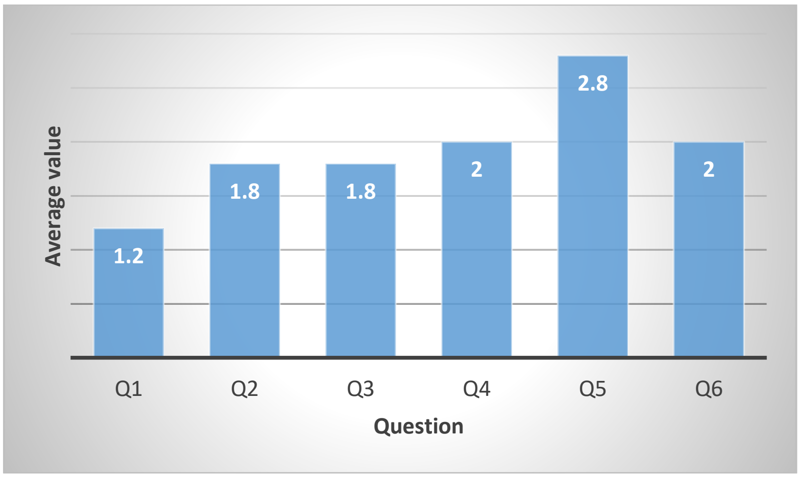


Figure 5.2: Evaluation based on feedbacks from doctors

5.3.3.2 Feedbacks from Parents

There are five MCQ questions in this questionnaire and each question has following five answers.

1. Strongly agreed
2. Agreed
3. Slightly agreed
4. Slightly disagreed
5. Disagreed

The questions are listed below.

Q1 : The interfaces of the application are user friendly.

Q2 : This application is more convenient to use, rather than the existing manual process.

Q3 : This application is helpful, because it keeps all the history records of the child.

Q4 : It takes much time to complete the SDQ when comparing with the existing manual process.

Q5 : Does it need any improvements in future? If agreed, comment your suggestions.

Table 5-12: Feedbacks from doctors

User	Q1	Q2	Q3	Q4	Q5
1	3	2	1	4	2
2	2	2	1	3	3
3	1	1	1	5	2
4	3	3	2	4	2
5	2	1	2	4	1
Average Value	2.2	1.8	1.4	4	2

Graphical representation of the average value with respect to the question number shown in Figure 5.3.

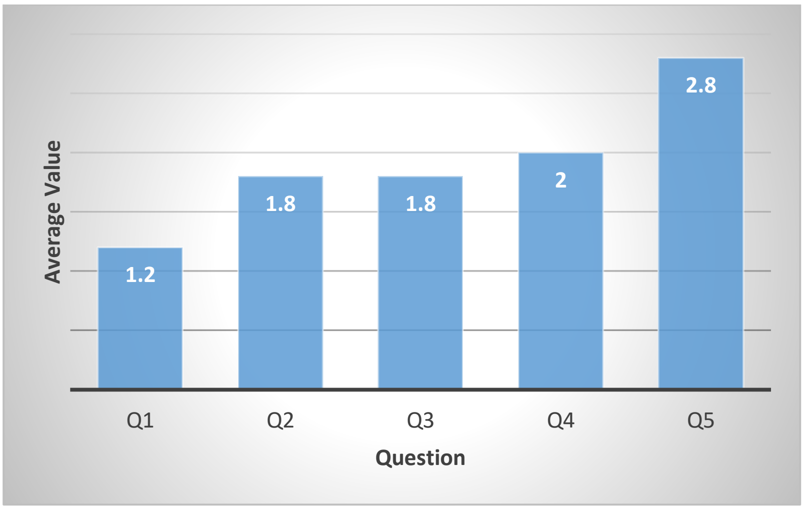


Figure 5.3: Evaluation based on feedbacks from doctors

According to the above chart, it has been proved that this application is more efficient than the existing manual process and also helpful since it keeps all the history records of the child. According to feedbacks, most of them suggested to add SDQ in Sinhala in addition to English.

Chapter 6

CONCLUSION

6.1 Chapter overview

This chapter describes the conclusion of the research which includes research contribution, limitations and future works.

Today, everyone has access to a mobile device and a variety of aspects can be represented using mobile applications. Medical and healthcare is a key field where we can use IT effectively and efficiently. There are several genetic disorders and Down syndrome is most common genetic disorder. Even this disease cannot be cured, earlier identification and earlier treatments are very important, since, it can help these children to grow more normally. This project is to address this problem and it is an identification and monitoring system for therapeutic intervention for children under care.

6.2 Research Contribution

The purpose of this project was to implement an identification and monitoring system for therapeutic intervention for children under care. Among the children under care, most of the children have been diagnosed as Down syndrome. Therefore, by considering the scope of the project, the author has mainly focused on children with Down syndrome since it is the most commonly occurring genetic disorder. Initially, the author has researched about the symptoms of Down syndrome, especially about distinct facial features than others. Based on these distinct facial features a Down syndrome detection supporting tool has been implemented using image processing and machine learning techniques. In addition to that two more functionalities have been implemented as requested by doctors. These are the SDQ functionality and the progress evaluation functionality. The SDQ is a globally accepted screening test to identify mental and health problems. This application provides the SDQ and score it according to the standard scoring scheme and find required scores. Implementation process has been described in detail in the chapter 4, and chapter 5 which includes the evaluation of the implemented application. According to the evaluation, the web service has 90% accuracy level, though it is a considerably good value, it is better to achieve 100% or a closer value. The evaluation shows that, the web service has 87.5% precision, it is ideal to have high precision as much as possible, but 0.875 is a considerably good value. Based on the evaluation, the web service has 93.3% Recall value, it is also a good value and it has a 90.3% F-score value which is the weighted average of the precision and recall. And, the accuracy of the SDQ scoring mechanism is also good. As per the feedbacks, both doctors and parents can obtain benefits from this application and seems it is useful and effective than existing manual process.

6.3 Research Limitations

Even though there are several genetic disorders, this application only focuses on children with Down syndrome. Down syndrome identification supporting tool has been developed using image processing techniques, based on their distinct facial features. Since facial appearances vary with the nationality, this implementation only considers about the Sri Lankan domain. And also there are few more factors to consider such as age and gender. Because of the difficulty to find images of children with Down syndrome with the approval, sample dataset were limited to 20 and could not able to consider about above factors. Otherwise, it’s better to use a large sample set to create the dataset. If the sample dataset has more images, more accuracy level can be achieved. According to the evaluation, the web service has 90% of accuracy level, 87.5% precision, 93.3% recall and 90.3% f-score. All these values are considerably good, but to achieve best performance, these values should reach best value for each measurement which is 100%. To reach the best values for above performance measurements, the training dataset should have a large number of images with high resolution. Therefore, the number of images in the training dataset can be considered as a limitation.

When considering the SDQ functionality, it has only the English version of the SDQ. It is better to have Sinhala version as well, since most of the parents convenient with the Sinhala language.

6.4 Future Works

The implemented application is an Android application. In the future, it can be implemented as a cross platform mobile application. In this scenario, therapy scheduling part is very important. Therefore, the implemented application can be enhanced with therapy scheduling functionality. And also as mentioned above, for the SDQ part, Sinhala version can be added in addition to the English version. Then parents can select their option based on their choice. .

REFERENCES

[1] P. I. Tomayess Issa, "Usability and Human Computer Interaction (HCI)," in *Sustainable Design*, London, Springer.

[2] A. Holzinger, "Human–Computer Interaction for Medicine and Health Care (HCI4MED): Towards making Information usable," *International Journal of Human-Computer Studies,* vol. 68, no. 6, pp. 325-327.

[3] "Askabiologist.asu.edu," [Online]. Available: https://askabiologist.asu.edu/explore/building-blocks-life. [Accessed 25 11 2017].

[4] G. Reference, "Genetics Home Reference," [Online]. Available: https://ghr.nlm.nih.gov/. [Accessed 24 11 2017].

[5] Q. Zhao, K. Rosenbaum, K. Okada, D. J. Zand, R. Sze, M. Summar and M. G. Linguraru, "Automated Down Syndrome Detection using Facial Photographs," in *35th Annual International Conference of the IEEE EMBS*, Osaka, Japan, 2013.

[6] L. Shaffer, "Disorders caused by chromosome abnormalities," *The Application of Clinical Genetics,* p. 159, 2010.

[7] J. H. Deena T Kochunni, "Types of Chromosomes - Autosomes and Sex chromosomes," Biologyexams4u.com, [Online]. Available: https://www.biologyexams4u.com/2013/02/types-of-chromosomes.html. [Accessed 29 5 2019].

[8] "Learn.genetics.utah.edu," 2017. [Online]. Available: http://learn.genetics.utah.edu/content/disorders/chromosomal/. [Accessed 25 11 2017].

[9] "NDSS," NDSS, [Online]. Available: http://www.ndss.org/resources/understanding-a-diagnosis-of-down-syndrome/. [Accessed 26 11 2017].

[10] "St. Elizabeth Healthcare - Condition," Stelizabeth.com, 2019. [Online]. Available: https://www.stelizabeth.com/HealthLibrary/Condition/down-syndrome. [Accessed 29 5 2019].

[11] "Nichd.nih.gov," US Department of Health and Human Services, [Online]. Available: https://www.nichd.nih.gov/health/topics/down/Pages/default.aspx. [Accessed 27 11 2017].

[12] E. Stumbo, "A closer look at the physical characteristics of Down syndrome - Ellen Stumbo," [Online]. Available: https://www.ellenstumbo.com/closer-look-physical-characteristics-down-syndrome/. [Accessed 28 11 2017].

[13] "Play.google.com," [Online]. Available: https://play.google.com/store/apps/details?id=com.fdna.face2gene&hl=en. [Accessed 5 12 2017].

[14] Ş. Saraydemir, N. Taşpınar, O. Eroğul, H. Kayserili and N. Dinçkan, "Down Syndrome Diagnosis Based on Gabor Wavelet Transform," *Journal of Medical Systems*, vol. 36, no. 5, pp. 3205-3213, 2011.

[15] K. Burçin and N. V. Vasif, "Down syndrome recognition using local binary patterns and statistical evaluation of the system," *Expert Systems with Applications,* vol. 38, no. 7, pp. 8690-8695, 2011.

[16] M.-H. Yang, D. J. Kriegman and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 34 - 58, 2002.

[17] "FaceDetector.Face | Android Developers," Developer.android.com, [Online]. Available: https://developer.android.com/reference/android/media/FaceDetector.Face.html. [Accessed 28 11 2017].

[18] "Docs.opencv.org," [Online]. Available: https://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0. [Accessed 04 12 2017].

[19] "dlib C++ Library," [Online]. Available: http://dlib.net/face_detector.py.html. [Accessed 14 04 2019].

[20] "Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python," 27 12 218. [Online]. Available: https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/. [Accessed 14 04 2019].

[21] A. Rosebrock, "Facial landmarks with dlib, OpenCV, and Python - PyImageSearch," PyImageSearch, [Online]. Available: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/. [Accessed 29 5 2019].

[22] T. Patel and B. Shah, "A Survey on Facial Feature Extraction Techniques for automatic face annotation," in *Innovative Mechanisms for Industry Applications (ICIMIA)*, Bangalore, India, 2017.

[23] R. Hebbalaguppe, A computer vision based approach for reducing false alarms caused by spiders and cobwebs in surveillance camera networks, 2014.

[24] " Local Representation of Facial Features (Face Image Modeling and Representation) (Face Recognition) Part 1," What-when-how.com, [Online]. Available: http://what-when-how.com/face-recognition/local-representation-of-facial-features-face-image-modeling-and-representation-face-recognition-part-1/. [Accessed 29 5 2019].

[25] R. H. Abiyev, "FACIAL FEATURE EXTRACTION TECHNIQUES FOR FACE," *Journal of Computer Science*, vol. 10, no. 12, pp. 2360-2365, 2014.

[26] M. D. Malkauthekar, "Analysis of euclidean distance and Manhattan Distance measure in face recognition," *International Journal of Computer Science and Engineering (IJCSE)* , vol. 3, no. 4, pp. 89-98, 2014.

[27] F. R. u. M. Classifiers, P. Parveen and B. Thuraisingham, "Face Recognition using Multiple Classifiers," in *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference*, Arlington, 2006.

[28] A. I. Salhi, M. Kardouchi and N. Belacel, "Fast and efficient face recognition system using random forest and histograms of oriented gradients," in *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG*, Darmstadt, Germany, 2012.

[29] "Sdqinfo.com," 2019. [Online]. Available: http://www.sdqinfo.com/a0.html. [Accessed 12 05 2019].

[30] "Improve Face Recognition Rate Using Different Image Pre-Processing Techniques," *American Journal of Engineering Research (AJER)*, vol. 5, no. 4, pp. 46-53, 2016.

[31] "Docs.opencv.org," 2019. [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html. [Accessed 16 05 2019].

[32] "OpenCV: Color conversions," Docs.opencv.org, 2019. [Online]. Available: https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html. [Accessed 16 5 2019].

[33] "imutils," PyPI, [Online]. Available: https://pypi.org/project/imutils/.

[34] "Performance Showdown of Publicly Available Face Detection Model," Medium, 30 04 2018. [Online]. Available: https://medium.com/nodeflux/performance-showdown-of-publicly-available-face-detection-model-7c725747094a. [Accessed 14 04 2019].

[35] D. Huang, C. Shan, M. Ardabilian , Y. Wang and L. Chen, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 41, no. 6, pp. 765 - 781, 2011.

[36] B. Pathya and S. Nainan, "Performance Evaluation of Face Recognition using LBP, PCA and SVM," *SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE),* vol. 3, no. 4, pp. 85-88, 2016.

[37] T. Ahonen, A. Hadid and M. Pietikäinen, "Face Recognition with Local Binary Patterns," in *ECCV 2004*, 2004.

[38] "Klios," Klios.fr, [Online]. Available: http://klios.fr/blogs/blog1.php. [Accessed 29 05 2019].

[39] H. Hatem, Z. Beiji and R. Majeed, "A Survey of Feature Base Methods for Human Face Detection," *International Journal of Control and Automation,* vol. 8, no. 5, pp. 61-78, 2015.

[40] "CNN based face detector from dlib," Towards Data Science, [Online]. Available: https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e01c. [Accessed 16 05 2019].

[41] "About," Opencv.org, 16 05 2019. [Online]. Available: https://opencv.org/about/. [Accessed 2019].

[42] "dlib C++ Library," Dlib.net, 2019. [Online]. Available: http://dlib.net/. [Accessed 16 05 209].

[43] "OpenCV: Histograms - 1 : Find, Plot, Analyze !!!," Docs.opencv.org, 2019. [Online]. Available: https://docs.opencv.org/3.1.0/d1/db7/tutorial_py_histogram_begins.html. [Accessed 16 5 2019].

[44] "sklearn.svm.SVC — scikit-learn 0.21.1 documentation," Scikit-learn.org, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. [Accessed 16 5 2019].

[45] M. Sokolova, N. Japkowicz and S. Szpakowicz, "Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation," in *Australasian Joint Conference on Artificial Intelligence*, 2006.

[46] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," in *Advances in Information Retrieval*.

[47] "Evaluating a Classification Model," ritchieng.github.io, 2019. [Online]. Available: https://www.ritchieng.com/machine-learning-evaluate-classification-model/. [Accessed 19 5 2019].

[48] "F1 Score," ritchieng.github.io, 2019. [Online]. Available: https://www.ritchieng.com/machinelearning-f1-score/. [Accessed 19 5 2019].

[49] Q. Zhao, K. Rosenbaum, K. Okada, D. J. Zand, R. Sze, M. Summar and M. G. Linguraru, "Using facial images for the diagnosis of genetic syndromes: A survey," in *Communications, Signal Processing, and their Applications (ICCSPA), 2015 International Conference*, Sharjah, United Arab Emirates, 2015.