Abstract—There are millions of visually impaired people who need assistance. This paper presents an implementation of a visual detection and recognition system as a Smartphone application which can recognize persons being captured by the camera in real-time. The developed system employs the Viola Jones face detection and a Discrete Cosine Transform (DCT)-coefficient based facial recognition algorithm. The system was implemented as a client-server application, the client being an Android application incorporating the face and a Java Servlet incorporating our implementation of the DCT-based face detection algorithm. The system was extensively tested and achieve the best recognition rate of 98% for persons looking directly at the camera.

Keywords—Android, Face recognition, Face detection, Smartphone Application

I. INTRODUCTION

An important sense for social interaction in humans is the sense of sight. This sense makes it possible to recognize other humans which we know, even if we could have met the person long time ago. However, there are 285 million people worldwide who are visually impaired: 39 million are blind and 246 million have low vision [1]. The portability of Smartphones compared with the increasing computing power they pack makes it an attractive platform to develop a system which can help the visually impaired people interact better with the world around them.

This paper presents the implementation of a system where a Smartphone can be used by visually impaired users to identify people in their surroundings. The proposed system was tested using different face databases in order to determine the performance factors. This work can be easily extended to provide user authentication services which rely on autonomous face recognition to log the user in.

The rest of this paper is organised as follows: Section 2 gives an overview of face detection and recognition algorithms and describes a system similar to the one being developed. Section 3 describes the design of the system while looking at the algorithms chosen in detail. In Section 4 the implementation details of the developed system are discussed. Testing results and evaluation are given in Section 5 and conclusions and future work are presented in Section 6.

II. BACKGROUND AND RELATED WORK

A. Face Detection and Recognition

Face detection systems are usually used in biometric systems to identify the person’s identity. These systems have been used extensively in airports and other working environments. The first step towards automatic face recognition is to detect the set of pixels which contain the face to be recognized. The Viola and Jones [2] was used in this work since it is capable to detect faces in real-time. There are other face detection systems that can be found in literature [3] - [5]. However, the method presented by Viola and Jones is found to be accurate enough and requires limited computational resources, which is ideal for Smartphone implementations.

A typical face recognition system, shown in Figure 1, goes through two main phases: training and classification. In training, a database of known faces is used to select features of the faces which will be used as a reference template for the known persons. During classification, an unknown image is fed to the system to be recognised. Features from the unknown face are compared with the features of the known faces in the database which were identified and selected during training. The unknown image is classified as the person whose features closely match the features of the unknown image.

Early research in face recognition has focused on detecting facial features and subsequently form a model based on the relationships between these facial features [7]. The authors in [8] and [9] describe automated and semi-automated techniques which use normalized distances and ratios between these features to classify faces. However, these early techniques were deemed not to be robust enough for real-world use as they were easily breakable [10]. More recent work in the field explored the use of DCT coefficients as basis for feature selection, with promises for high recognition rates [6].
B. Smartphones

A Smartphone can be described as a portable device which apart from having the normal phone functions, allows installation of software which provide other features such as web browsing, sending and receiving email, edit and share documents pictures and videos. The two major Smartphone platforms are iOS, which powers the iPhone and iPad devices from Apple, and Android, which powers mobile devices from various other manufacturers.

III. DESIGN

The system is designed to make use of a Smartphone camera, which captures video in real time. The frames are supplied to the face detection module. During the training phase, the face regions from the face detector are tagged by the user with the person’s name and fed to the feature selector. The feature selector selects the features to be used to represent the particular person, taking into account any face regions of the same person which were already processed and stored in the database. These features are stored or updated in the database.

During face recognition the proposed system employs the detected face regions as input to the face classifier. The classifier uses features stored in the database to find a close match for the unknown list of faces representing the known users. The name of the classified person is then read out to the user. Figure 2 shows the basic modules making up the system being developed in this project.

The face detection method chosen for this project is the DCT based method which is presented by Fabrizia M. de S. Matos et al. in [6]. This algorithm was chosen as previous work by Fenech K. in [11] shows that this method produces high recognition accuracy when compared to the Eigenface-based (which uses PCA) methods proposed by in Turk and Pentland in [7]. Furthermore, this method does not require a normalisation phase and is relatively simple to implement while having low computational cost.

The first step is to convert each person’s q training faces image into the frequency domain using 2D DCT-II. Assuming that a face image is composed of a by b pixels, it can be treated as a matrix \( f(x, y) \) of dimensions \( N \times M \). Its DCT transform, \( F(u, v) \):

\[
F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \cos \left( \frac{(2x + 1)\pi u}{2N} \right) \cos \left( \frac{(2y + 1)\pi v}{2M} \right)
\]

where

\[
\sigma(u, v) = \begin{cases} 
\frac{1}{\sqrt{N}} & \text{if } x = 0 \\
\frac{2}{N} & \text{otherwise}
\end{cases}
\]

After each pose q of person j has been converted into their DCT coefficients, m coefficients to be used for comparison with person j are selected according to some criteria. In [6], three different criteria are proposed: however, in previous work in [11], it was shown that the average amplitude method gives the highest recognition rate while being the simplest to implement and compute. Based on these findings, this criterion was chosen as the feature selector for this project.

By using this criterion, the average of the \( i \)-th coefficient of the \( j \)-th person is given by:

\[
\bar{\mu}_i = \frac{1}{q} \sum_{k=1}^{q} |x_{ij,k}|
\]

where, \( i \) represents the \( i \)-th coefficient value, ranging from 1 to \( N \), with each image of dimension \( a \times b \) having a total of \( a \times b = N \) coefficients, \( j \) represents the \( j \)-th person in the training database, \( k \) represents the \( k \)-th pose of the set of poses belonging to a particular person in the database and \( q \) represents the number of poses of each particular person in the training database.

For each person \( j \), a set of \( N \) coefficients with average amplitudes calculated using (3) is generated. Out of this set, the top \( m \) coefficients with the highest average values are selected to represent person \( j \).

In [6], an unknown face image is classified based on the calculation of a measure of similarity between the test face and all the training faces in the database. Going back to the list of \( m \) coefficients for each person in the training database, the \( m \) selected coefficients for person \( j \) are \( \{y_{1,j}, y_{2,j}, ..., y_{m,j}\} \). Using the same positions that the \( m \) selected coefficients refer to, the amplitudes of the training coefficients for person \( j \) in pose \( k \) are \( \{y_{1,k}, y_{2,k}, ..., y_{m,k}\} \).

First, the test image \( f \) is converted to its DCT coefficients using (1). This operation generates a list of \( N \) DCT coefficients amplitudes which positions will correspond to the coefficients of each pose in the training database. The \( m \) selected DCT coefficients for test image \( f \) are \( \{v_{1,f}, v_{2,f}, ..., v_{m,f}\} \) where the position of coefficient \( v_{lf} \) is in the same position as \( y_{ij,l} \).

The following equation gives the distance between the test face \( f \) and person \( j \) in pose \( k \):

\[
\sum_{l=1}^{m} (v_{lf} - y_{lf})^2
\]

Figure 2: Block diagram of our face recognitions system
In order to classify the test face \( f \) as person \( j \), (4) is repeated for person \( j = 1, 2, \ldots, p \) and pose \( k = 1, 2, \ldots, q \). Test face \( f \) is classified as the person whose one of the poses has the smallest distance to the test face, as shown in (5):

\[
Dist_{ij,k} \leq Dist_{ij,q}, \forall j \neq q, \forall k \neq i
\]

IV. IMPLEMENTATION

In order to cope with the limited processing power and low levels of storage provided by Smartphones, while at the same time be battery-friendly, it was decided to implement the application as a client-server application. The client part runs on the Smartphone and takes care of getting pictures from the Smartphone camera and selects the faces regions using face detection. These regions are then sent to the server for performing the actual recognition. The client and server communicate using the HTTP protocol. Figure 3 shows how the various components of the system and how data flows between the components of the system.

The server was implemented as a Java Servlet. This makes it possible to share code between the client and the server-side portion of the application. The Servlet acts as a simple, stateless web service and uses JSON as the data encapsulation method. Communication between the client and server is done using the standard HTTP protocol.

The database backend chosen for this project is MySQL. The main reason for selecting this database system is because it is a very popular database which is available with almost any web hosting service. For computation of the DCT, a multithreaded Java library called JTransforms [12] by Piotr Wendykier was used.

The client and server communicate over the HTTP protocol using a simple API. Parameters are sent from client to server using the GET HTTP method. This API provides functions such as the creation of new persons in the database, addition of training poses and classification of unknown faces. Other functions to aid testing and debugging were also added to the API.

The platform chosen for the Smartphone client, based on market share statistics [13], was the Android platform. The developed Android application acts as a simple, stateless client, using JSON as the data encapsulation method. In this example, the mobile phone camera is being pointed to a person’s face. The screen shows that the face detector has performed face detection on the image and has returned the detected face regions, while also drawing a box around the face in the image. Fig. 4 shows the application interface during face recognition mode.

For face detection, the open source computer vision, OpenCV [14], library was selected, as it implements an object detector based on the Viola and Jones classifier and is free for both academic and commercial use.

The Android application has two main functions. The main function is to perform face detection on faces contained in frames coming in from the Smartphone camera, which is used to mimic the user’s field of vision. The detected faces are then sent to the server for classification.

![Figure 3: Overview of developed system](image)

![Figure 4: Screenshot of the Android application](image)

The second function of the application is to perform training. As before, frames from the camera are processed by the face detector. The user is then asked to choose one of the detected face regions, followed by tagging the selected face region as a specific person. The face region and selected person ID are then sent to the server for training. Android’s built-in Text-To-Speech (TTS) speech synthesiser is used to read the names of the recognised persons to the user.

V. SIMULATION AND EVALUATION

A number of face databases were used to test the two main modules of the application. A ‘home made’ database made up of 30 random images from the web was used to test the face detector from the OpenCV library. The precision obtained from these tests was 0.96, while the recall rate was 0.81.

The face recognition algorithm was tested using three face databases. Using the AT&T face database, experiments to
quantify the ideal amount of coefficients to be used were carried out and the best recognition rate obtained was 98% using 75 coefficients.

Other experiments were done in order to determine the Minimum number of training poses required to obtain an acceptable facial recognition performance. From the results obtained, it was concluded that acceptable results (over 90% accuracy) can be obtained using only 4 or 5 training images.

More tests were carried out using the LFW face database [15] to see how the system performs when considering more difficult scenarios. From these tests, it was observed that while equalizing the image histograms does improve recognition by a small margin, large variations between the poses, such as head rotation, tilt of the face, expressions, eyewear and other facial features, contribute to a low recognition rate.

Following results obtained using the LFW database, it was necessary to carry out further testing with controlled variation between images. For this purpose, the FERET database was chosen, as this database contains images which are tagged according to the pose angle. FERET (Fa, Fb) in Figure 5 shows that a recognition rate of 98% is obtained when there is no rotation. The recognition performance is observed to decrease with increased rotation, as demonstrated in FERET(Fa, Fb, Rb, Rc) and FERET(Fa, Fb, Ra, Rb, Rc, Rd). These results confirm that the recognition rate decreases with increased head rotation angle.

![Recognition Rates Obtained](image)

**Figure 5:** Summary of the recognition rates obtained

**COMMENTS AND CONCLUSION**

The A summary of the recognition rates obtained in the tests carried out is shown in **Figure 5**. Based on the results obtained, it is recommended that future work explores possibilities of pre-processing the images in such a way as to minimise the differences in tilt and rotation of the head, facial expressions and lighting. An additional face recognition algorithm using different techniques can be added to the system, where results from the two algorithms are combined based on some confidence measure.

3D based techniques and methods such as the one proposed in [16] retrieve shape and texture parameters from a single 2D face image and use these parameters to compute 3D models which can then be used for recognition without the influence of pose orientation and illumination. In addition, as smartphones become even more powerful with larger storage, the client server application might be combined as one single Smartphone application, removing the need for an active data connection.

**REFERENCES**