© 2017 IJEDR | Volume 5, Issue 2 | ISSN: 2321-9939

Automatic Driving License Verification System Through Android Smartphones

[1]Megha J, [2]Suneetha Hiremath, [3]Divya A, [4]Kavya G S

Students,
Information Science & Engineering,
NIE Institute Of Technology, Mysuru, India

***Abstract*—The main objective of the project is automatic verification of DL of a person, pre-processing images and extract features, perform face recognition by using existing methods as well as implementing new techniques, handle camera setup, calibrate and capture still faces, Use of database of different registration of driving licenses makes better performance evaluation, to identify the person with high rate of success. To achieve the best result for face recognition we are going to employ GLCM filters and EigenFace Algorithm in PCA. Due to the large quantity of potential applications, over the past years many algorithms have been proposed, which can be separated into three categories: holistic, facial feature based and hybrid. Even though some algorithms have achieved a high accuracy, there is still the need for a significant improvement to achieve robustness in uncontrolled conditions while achieving a high computational efficiency. In our project we try to build an android application to verify the DL of the person. The end result of the application is that it provides the verified DL of the person. To verify the DL of the person we first capture the face image of the person whose DL is to be verified and then we pass that image to the face recognition tool build using the matlab software once the face get recognized than the DL of the person will be retrieved from the DL database where the DL of every persons will be stored. At the end our project is achieving quick verification of the DL and try to help the traffic police management in the verification of the driving license.**

***Index Terms*— Face Recognition, Gray level co-occurance matrix(GLCM) features, PCA algorithm, Eigenfaces, Text Matching, Optical Character Recognition(OCR) , Android, DL Verification**

I. Introduction

In this project, the goal is to find best match of an image captured by smart phone and the image that exist in the existing database. Using a pre-stored image database, the face recognition system should be able to identify the person in the scene. Before face recognition is performed, the system should determine whether or not there is a face in a given image. Once a face is detected, face region should be isolated from the scene for the face recognition. In face recognition system, if the isolated image get matched with image in the existing database, then the DL of the person is retrieved from the DL database.

This can be done by using various descriptors. The classification function is given by a metric learning algorithm, i.e. an algorithm which finds the best distance that separates the input data. In some cases we cannot get the accurate results so to overcome this drawback we use the text matching process which includes DL No, Name of the DL Holder, Validity date and Address of the DL holder.

II. Problem statement

Face Recognition is difficult problem due to pose or viewpoint changes, variations in the illumination conditions, the possibility of occlusions due to glasses and other objects, differences in expression, age, changes in hair and facial hair and image quality. Therefore the automatic decision of whether a pair of images depicts the same person or not in the taken image becomes a highly challenging problem.

The image quality is primary requirement of face recognition system is suspects good quality face image and a good quality image is one which is collected under expected conditions. For extracting the image features the image quality is important. Without the accurate computations of facial features the robustness of the approaches will also be lost. Thus even the best recognition algorithm deteriorate as the quality of the image declines. Same face appears differently due to change in lighting. Illumination can change the appearance of an object drastically. We must overcome irregular lighting. Following image processing can be used to provide a degree of lighting invariance.

The main factors that make this problem challenging are image degradation due to blur, and appearance variations due to illumination and pose. Rotation of face image (Rotation in tilt) induce very large changes in face appearance. Recognition rates fall drastically when images from two different poses of same person are matched.

III. Solution

To overcome the above problem we are using a special kind of filters called as the GLCM features, where each filters performs certain task. Here we are only concentrating on invariants that are present in the face, they are extracted and passed through eigen faces and it compares with the existing image present in the database and provides the output. To get an even accurate result we are using text matching which is useful for object recognition and matching applications.

IV. LITERATURE SURVEY

Mohamed Anouar Borgi et. al. [1] used two approach to extract facial features new multi-scale directional framework, called shearlet Network (SN) and they have applied a refinement of the Multi-Task Sparse Learning (MTSL) framework to exploit the relationships among multiple shared tasks generated by changing the regularization parameter during the recognition stage.

Yan Ouyang et. al. [2] They have applied a refinement of the Multi-Task Sparse Learning (MTSL) framework to exploit the relationships among multiple shared tasks generated by changing the regularization parameter during the recognition stage. Sparse Representation based Classification (SRC) is used due to its robustness to occlusions.

Yong Wu et. al. [3] Weber-face (WF) is an illumination insensitive face representation based on Weber' law. They develop a generalized Weber-face (GWF) which extracts the statistics of multi-scale information from face images.

Liton Chandra Paul et. al. [4] This paper mainly addresses the building of face recognition system by using Principal Component Analysis (PCA). PCA is a statistical approach used for reducing the number of variables in face recognition. Recognition is performed by projecting a test image onto the subspace spanned by the eigenfaces and then classification is done by measuring minimum Euclidean distance. A number of experiments were done to evaluate the performance of the face recognition system.

Zheng Zhang et. al. [5] Noise modeling framework to improve a representation based classification (NMFIRC) method for robust face recognition. The representation based classification method has evoked large repercussions in the field of face recognition. representation based classification method (RBCM) always first represents the test sample as a linear combination of the training samples, and then classifies the test sample by judging which class leads to a minimum reconstruction residual.

Martin Grafmüller et. al. [6] In IT applications optical character recognition with smart cameras becomes more and more popular. A method is proposed that is based on the Bayes theorem to take account of prior knowledge for line and character segmentation.

V. EXISTING SYSTEM

There are 3 existing systems for face recognition they are holistic method, model based method and template based method. Holistic approaches attempt to identify faces using global representations i.e description based on the entire image than on local features of the face, Model based methods which employ shape and texture of the face, along with 3D depth information, and finally the Template based or feature extraction method the input image to identify and extract distinctive facial features such as the eyes, mouth, nose etc. compute the geometric relationships among those facial points, thus reducing the input facial image to vector of geometric feature.

Disadvantage

The disadvantage of these existing system is, if the lightning effects and the position of the face with respect to camera is varied greatly then accuracy will affect. A noisy image or partially occluded face causes recognition performance to degrade gracefully.

VI. PROPOSED SYSTEM

GLCM features and Eigen face based face representation presented promising results in face recognition applications due to its robustness against illumination and facial expression changes. The power of Gabor lays in its properties like the computation of local structure corresponding to different spatial frequency(scale), spatial localization, orientations and inessentiality of manual annotations. GLCM features overcome the disadvantage of existing system i.e the noise will be eliminated, illumination problem is removed and any kind of images can be detected. Text matching features is useful for object recognition and matching applications. It describes the region surrounding a key point, in a specific scale and orientation.

Advantages

Automation of DL verification, Improve the success rate of face recognition algorithm, A portable time efficient tool building, There will be some variations in the image like noise, lighting changes, expression etc., some algorithm can be improved by studying on these challenges, Use of database of different registration of driving licenses makes better performance evaluation.

VII. DESIGN

System Architecture

Figure 1 explains the High level software design, also called as software architecture, is the first step to analyze and consider all requirements for a software and attempt to define a structure which is able to full fill them. For this also, non-functional requirements have to be considered such as scalability and maintainability. This first design step has to be more or less independent of the programming language

First an image of the face is acquired. This acquisition can be accomplished by digitally scanning an existing photograph or by using an electro-optical camera to acquire a live picture of a subject. Second, software is employed to detect the location of any faces in the acquired image. This task is difficult, and often generalized patterns of what a face "looks like" (two eyes and a mouth set in an oval shape) are employed to pick out the faces. The third step is to preprocess the captured image. This can be done by using GLCM features so that the noise, contrast, homogeneity, correlation are to be filtered. Feature extraction being the fourth step is important towards classification task. Different vendors use different methods to extract the identifying features of a face. The next step is to compare the features generated in step three with those in a database of known faces. Instead of capturing the image and then detecting face in that image, other steps can be performed on an image in the given database. Just as the human perception system uses both local features and the whole face region to recognize a face, a machine recognition system should use both. After recognizing the face, the next step involves Text matching which uses optical character recognition(OCR) technique and compare the result with data present in the database. If it matches then the image of the DL is displayed which is stored in the database.

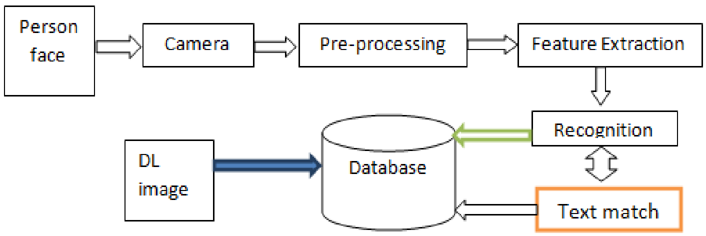

Fig. 1: Face Recognition System Architecture

Data Flow diagram

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data.

The **Figure 2** explains how the overall system works both from the client end and the sever end. The user gets his user credential from the admin that is user name and the password with this details he can login to the application once the user is authenticated he can use the application to verify the DL. First the face image needs to be captured and then face detected and then face recognition is performed and after recognition the face if there is match the output will be generated.

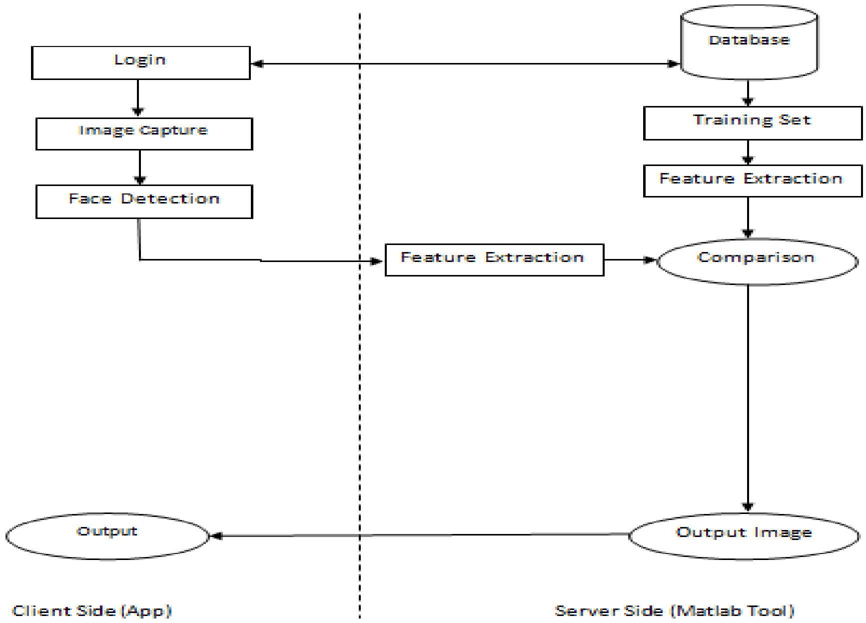

Fig. 2: Data flow diagram for whole face recognition system

Flow chart for DL System

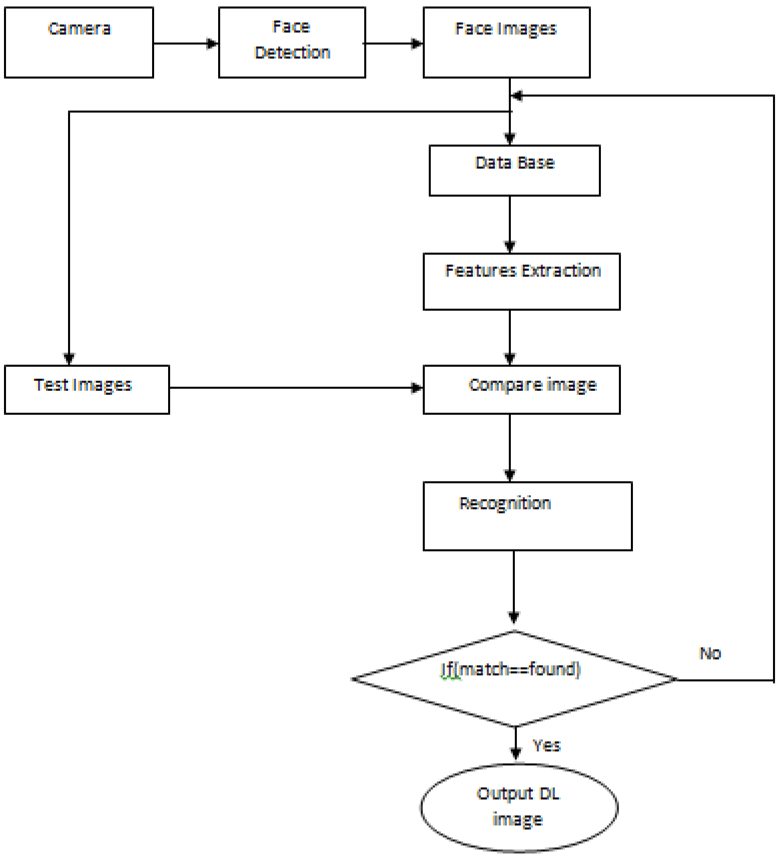

Fig. 3: Flow chart for DL system

Algorithms

Eigenface algorithm

Let a face image Γ(x, y) be a two dimensional M by N array of intensity values. In this thesis, I used a set of image by 200 × 149 pixels. An image may also be considered as a vector of dimension M × N, so that a typical image of size 200 × 149 becomes a vector of dimension 29,800 or equivalently a point in a 29,800 dimensional space.

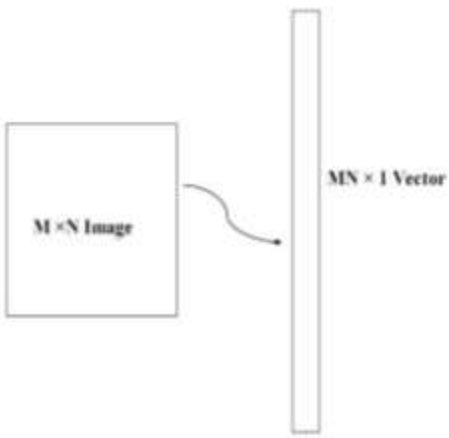

Fig. 4: Conversion of M × N image into MN ×1 vector

Step1: prepare the training faces Obtain face images I1, I2, I3, I4 , ... IM (training faces). The face images must be centered and of the same size.

Step 2: Prepare the data set Each face image Ii in the database is transformed into a vector and placed into a training set S.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \dots \dots, \Gamma_M\}$$

In My example M = 34. Each image is transformed into a vector of size MN × 1 and placed into the set. For simplicity, the face images are assumed to be of size N × N resulting in a point in dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Step 3: compute the average face vector The average face vector (Ψ) has to be calculated by using the following formula:

$$\Psi = \frac{1}{M}\sum_{n=1}^{N} \Gamma_n$$

Step 4: Subtract the average face vector The average face vector is subtracted from the original faces and the result stored in the variable ,

$$\Phi_i = \Gamma_i - \Psi$$

Step 5: Calculate the covariance matrix We obtain the covariance matrix C in the following manner,

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T$$

$$= AA^T \qquad (N^2 \times N^2 \ matrix \quad \text{Where,}$$

$$A=[\Phi_1, \Phi_2, \Phi_3, \Phi_4, \dots \dots \dots \Phi_M$$

$$(N^2 \times M \ matrix)$$

Step 6: Calculate the eigenvectors and eigenvalues of the covariance matrix The covariance matrix C in step 5 has a dimensionality of $N^2 X N^2$, so one would have N^2 eigenface and eigenvalues. For a 256 × 256 image that means that on must compute a 65, 536 × 65, 536 matrix and calculate 65,536 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for our task. In general, PCA is used to describe a large dimensional space with a relative small set of vectors [3]. Compute the eigenvectors U_i of AA^T The matrix AA^T is very large **- -** -> not practical!!!

Step 6.1: consider the matrix

$$L = A^T A \ (M \times M \ matrix)$$

Step 6.2: compute eigenvectors of L= A^TA

$$A^T A v_i = \mu_i v_i$$

What is the relationship between U_i and V_i ?

$$A^T A v_i = \mu_i v_i$$

$$A A^T A v_i = \mu_i A v_i \quad CA v_i = \mu_i A v_i \quad [\text{since } C = A A^T]$$

$$C u_i = \mu_i A v_i$$ where, $u_i = A v_i$ Thus, $C = A A^T$ and $L = A^T A$ have the same eigenvalues and their eigenvectors are related as follows:

$$u_i = A v_i$$

Note 1: C= AA^T can have upto eigenvalues and eigenvectors.

Note 2: L= A^TA can have upto M eigenvalues and eigenvectors.

Note 3: The M eigenvalues of C= AA^T (along with their corresponding eigenvectors) correspond to the M largest eigenvalues of L= A^TA (along with their corresponding eigenvectors). Where V_i is an eigenvector of L = A^TA . From this simple proof we can see that A V_i is an eigenvector of C= AA^T . The M eigenvectors of L = A^TA are used to find the M eigenvectors of C that form our eigenface basis:

$$u_i = \sum_{i=1}^{M} v_i \Phi_i$$

Where, U_i are the Eigenvectors i.e. Eigenfaces.

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues) Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of Characteristic features of the faces.

Gray level co-occurance matrix (GLCM)

Gray Level Co-Occurrence Matrix (GLCM) has proved to be a popular statistical method of extracting textural feature from images. According to co-occurrence matrix, Haralick defines fourteen textural features measured from the probability matrix to extract the characteristics of texture statistics of remote sensing images. In this paper four important features, Angular Second Moment (energy), (inertia moment), Correlation, Entropy, and the Inverse Difference Moment are selected for implementation using Xilinx ISE 13.4.

1. Angular Second Moment : Angular Second Moment is also known as Uniformity or Energy. It is the sum of squares of entries in the GLCMAngular Second Moment measures the image homogeneity. Angular Second Moment is high when image has very good homogeneity or when pixels are very similar

$$ASM = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P_{ij}^2 \quad \textbf{... 1}$$

Where i, j are the spatial coordinates of the function p (i, j), Ng is gray tone.

2 Inverse Difference Moment : Inverse Difference Moment (IDM) is the local homogeneity. It is high when local gray level is uniform and inverse GLCM is high.

$$IDM = \frac{\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P_{ij}}{1+(i-j)^2} \quad ...2$$

IDM weight value is the inverse of the Contrast weight.

3.Entropy : Entropy shows the amount of information of the image that is needed for the image compression. Entropy measures the loss of information or message in a transmitted signal and also measures the image information.

$$ENTROPY = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} -P_{ij} * logP_{ij} \quad ...3$$

4. Correlation : Correlation measures the linear dependency of grey levels of neighboring pixels.Digital Image Correlation is an optical method that employs tracking & image registration techniques for accurate 2D and 3D measurements of changes in images. This is often used to measure deformation, displacement, strain and optical flow, but it is widely applied in many areas of science and engineering. One very common application is for measuring the motion of an optical mouse.

$$Correlation = \frac{\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} (i,j)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad ...4$$

The formulation and extraction of the four given image features are extracted using matlab for calculating GLCM as image cannot be directly given as input to implement using FPGA.Image feature extraction method used in this paper is given in **Figure 5**. All the texture features are real numbers. Real numbers cannot be displayed using waveforms which show only bits as outputs.

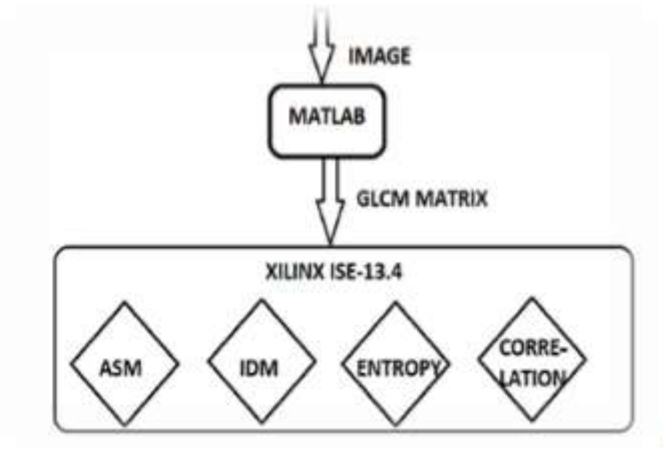

Fig.5: Extraction of image features.

VIII. IMPLEMENTATION

DL Verification system is broken into different modules, with a certain amount of dependency among them. The system has the following modules:

1. Login.
2. Capture image and text details.
3. Local Server.
4. Face Recognition.

Login

In this module the main aim is to authenticate the user by entering the user name and password which is preexisting in the database.

The below **Figure 6** explains the steps involved in authentication process.

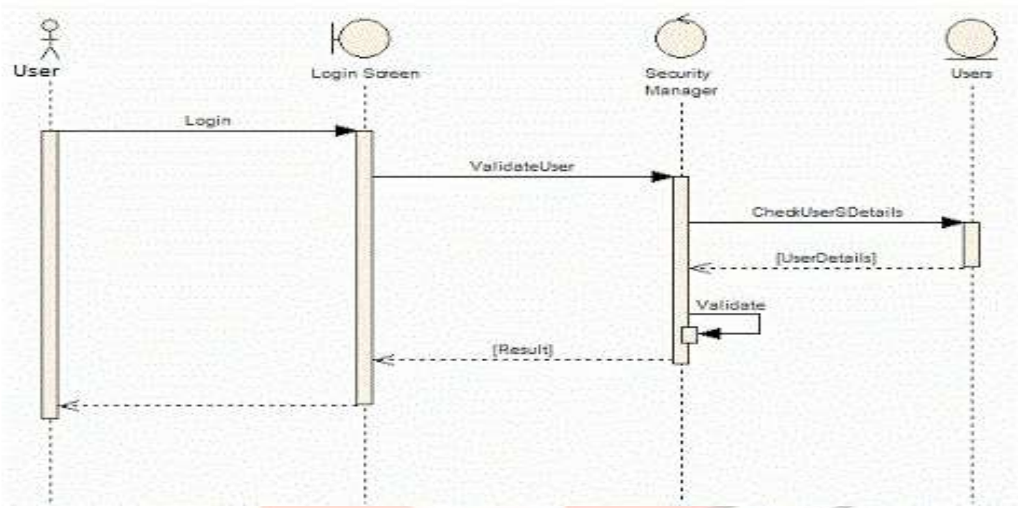

Fig. 6: Authentication Process

Capture image and text details

In this module the face of the person will be captured using smart phone camera and the details like name of the person, date of birth and DL no will be entered and sent to the local server.
The below **Figure 7** shows the steps performed in the capturing of an image.

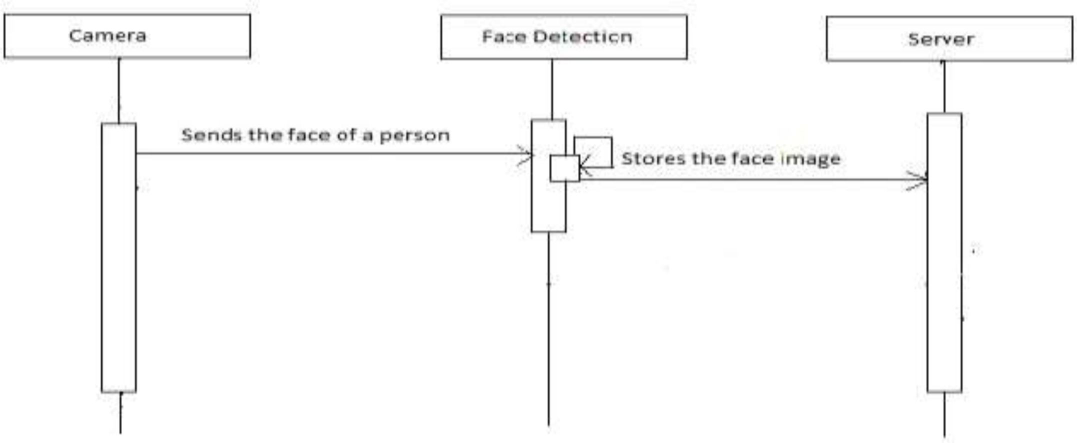

Fig. 7: Camera Module

Local Server

In this module the image and the details will be retrieved and sends to the tool which is created using matlab for the face recognition.

Face Recognition

In this module the image sent from the camera is preprocessed and the features of the face is extracted by dividing the image into blocks to get the accurate face features using GLCM features. Once the features are extracted it is compared with the face database and if there is a match the particular dl image associated with the face of the person is retrieved and sent to the android application. The below **Figure 8** explains the steps in the face recognition.

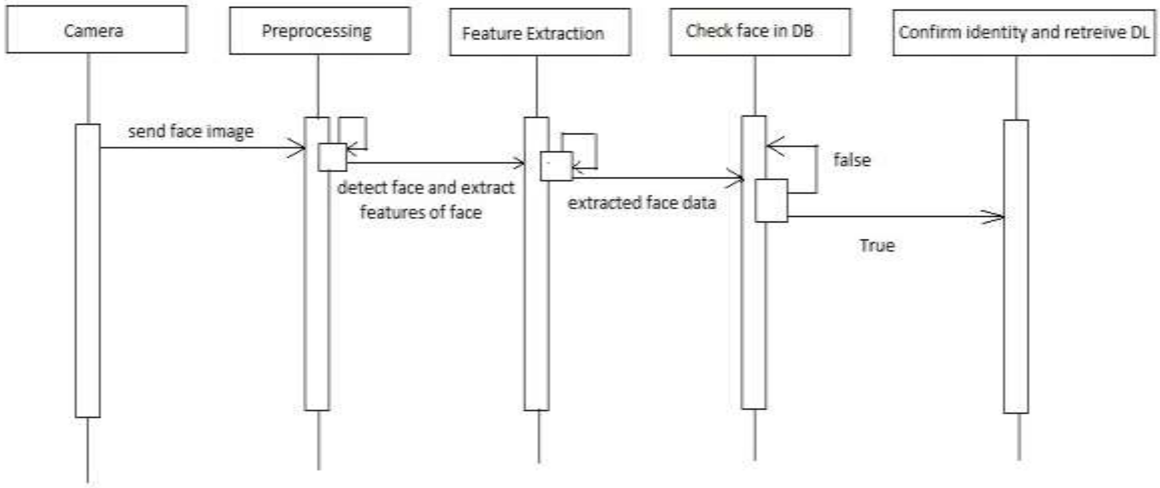

Fig. 8: Face Recognition

IX. SNAPSHOTS AND RESULT DISCUSSION

Web Application

Fig. 9: Matlab GUI

The above **Figure 9** describes the gui created using the matlab software which contains the buttons like create database, start process and exit button. Firstly we need to create the database to do that we should click on the create database button the creation of the database is shown in the next page, once you create the database next to initiate the process you need to click on the start process.

Creation of Database

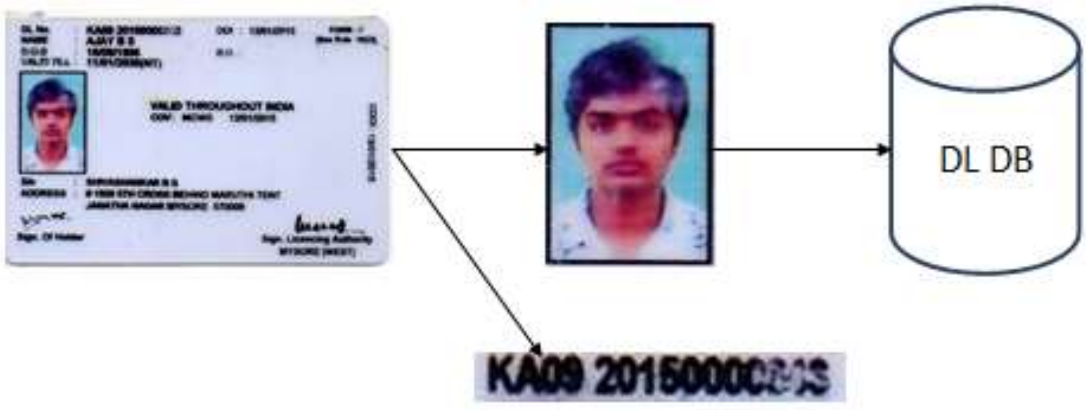

Fig. 10: DB creation

The above image explains the creation of database. The first step in the creation of database is storing the original DL image. The second step is to extract the face and details of the image which is stored in the database. The third step is to extract the characters from the DL number image and store it a text file.

Login Page

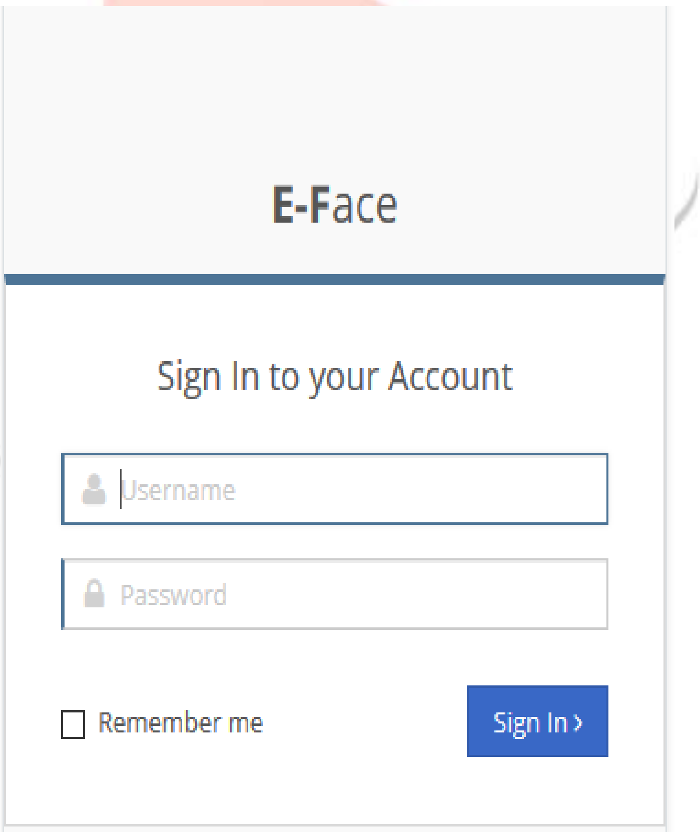

Fig. 11: Login Page

This is the login page of the web app which contains user name, password and sign in button, which is used to authenticate the user. The user name and password are predefined in the database. Once the user is authenticated the user is allowed to move to the next page.

User Credential Page

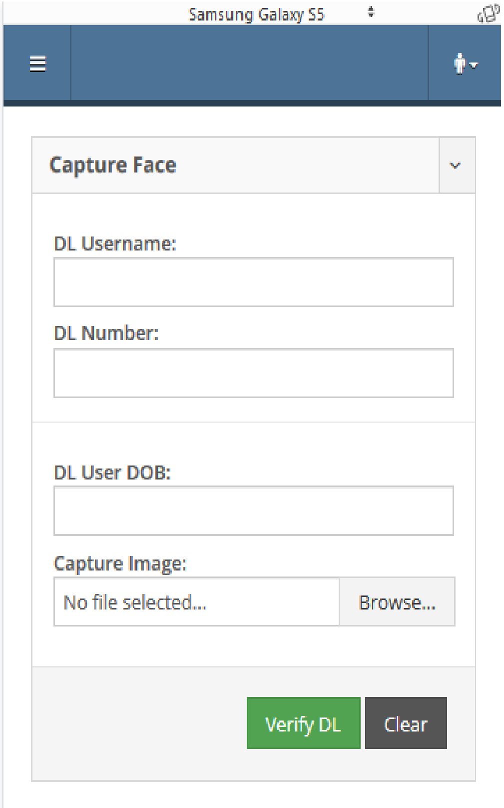

Fig . 12: User Credential Page

This is the second page of the app which describes the functionalities like capturing an face viewing an image and entering the details of the user such as DL name and DL number of the user. We have a verify DL button which is used to send the details to the database on the local server.

Dash Board

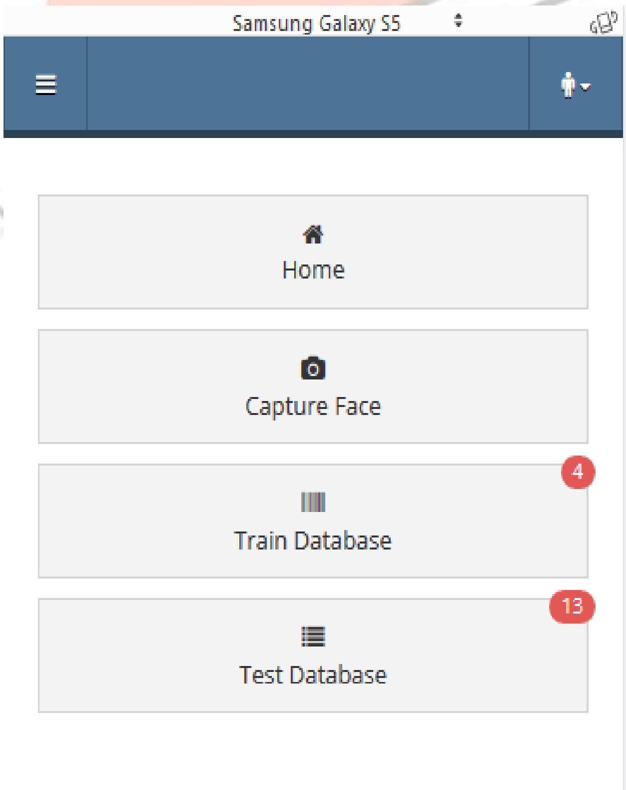

Fig. 13: Dash Board

This dash board shows the default home page of the web application which has options like home, capture face, train database and test database. when you click on the camera option you can see a camera will be opened ready to capture the image. When click the train and test databases you get see what are the train and test images that stored in the respective databases. The number 3 marked on the test and train databases indicates that there are 3 images present those databases.

Face Recognition and Text Matching

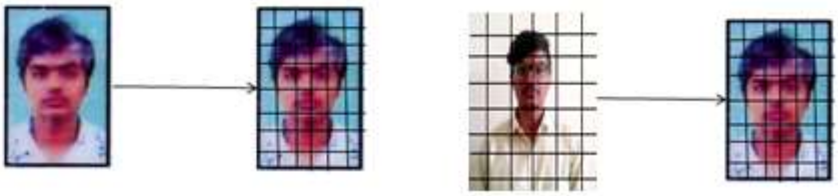

Fig. 14: Face Matching

In face recognition part the first step is to divide both the test image and train image into 64 parts and features of the face are extracted and compared between the images. The highest compared features are calculated and the output is given.

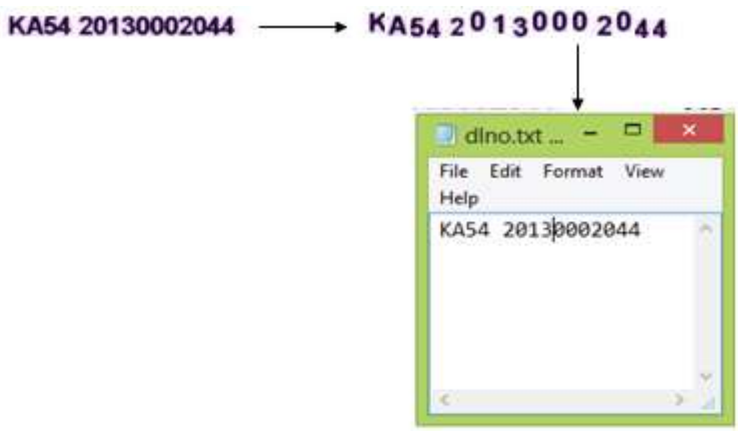

Fig. 15: Text Matching

The above figure explains the working of the OCR where the DL number image is given as input and the characters are extracted and stored in a text file for the comparison of DL number.

Final overview of the Face Recognition Process

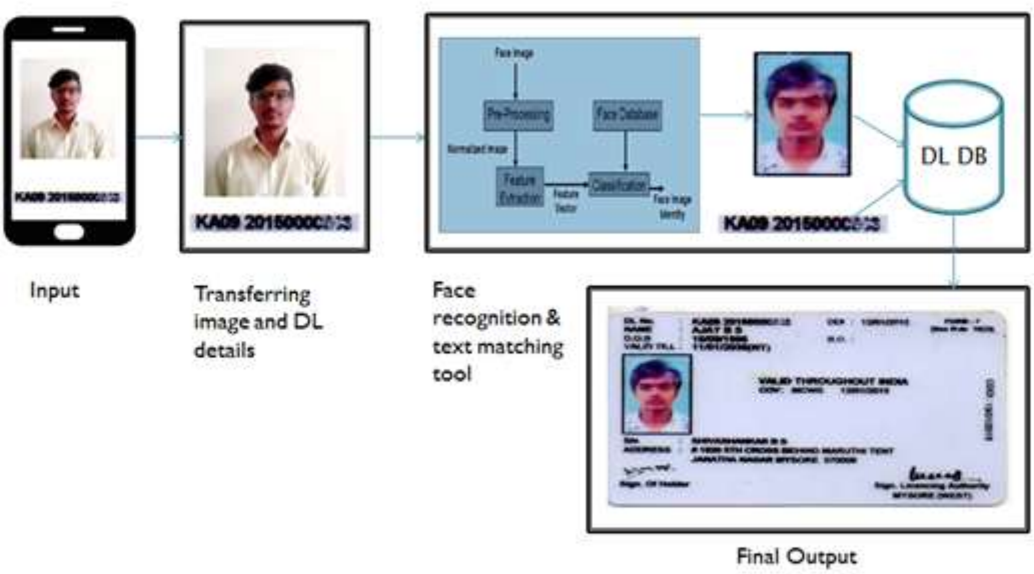

Fig. 16: Final demonstration of our Project

Firstly the face image is captured via the smart phone and then the DL number is entered and it is transferred to backend processing where the face recognition and text matching will be performed

The database consist of the test and the train images when we capture the face image via smart phone it comes and search in the database once when it find the captured image in the database it retrieves the DL of that particular image from the database as shown in the above image.

X. CONCLUSION AND FUTURE WORK

In our project we have used eigen algorithm for face recognition by extracting the facial features from the face image and then comparing it with the test images to obtain a match. the eigen face matching algorithm is used because this algorithm provides good matching rates and have good efficiency. Text matching is also performed where the DL no are matched to retrieve the DL from the database .An DL verification system has been built using the matlab software which verifies the DL. GUI has been developed for the user interaction where use can communicate with the application.

In the future there is scope for the improvement of detecting and recognizing for multiple faces in the same image and pose above 90 deg. Further experiment can be evaluated using robust algorithm for higher accuracy and recognition results.

XI. REFERENCES

[1] Mohamed Anouar Borgi, Demetrio Labate, Maher El Arbi, Chokri Ben Amar, “Sparse Multi-Stage Regularized Feature Learning for robust Face Recognition”, Research Groups on Intelligent Machines,University of Sfax,BP 1173, Sfax 3038, Tunisia,2015, pp 269-279.

[2] Yan Ouyang, Nong Sang , Rui Huang ,” Accurate and robust facial expressions recognition by fusing multiple sparse representation based classifiers”, Science and Technology on Multi-spectral Information Processing Laboratory, School of Automation, Huazhong University of Science and Technology ,February 2015, Volume 149, ISSN 2013.

[3] Yong Wu, Yinyan Jiang, Yicong Zhou, ”Generalized Weber-face for illumination-robust face recognition”, University of Macau,, China, Jan 2014,pp 262-267.

[4] Liton Chandra Paul1 , Abdulla Al Sumam, ‘’ Face Recognition Using Principal Component Analysis Method “, Volume 1, Issue 9, November 2012 , ISSN: 2278 – 1323.

[5] Zheng Zhang, LongWang, QiZhu, ZhonghuaLiu, YanChen*,”* Noise modelling and representation based classification methods for face recognition”, *YanChenPublished by Elsevier B.V., 11 August 2014*

[6] Martin Grafmüller, Jürgen Beyerer,” Performance improvement of character recognition in industrial applications using prior knowledge for more reliable segmentation”, BeyererPublished by Elsevier B.V., 2013