MID TERM MINI PROJECT REPORT
(2020)

Securing data using Face Recognition System

Institute of Engineering & Technology

Team Members

Tanuj Johal
(171500356)
Anushka Soni
(171500059)

Supervised By
Mr. Neeraj Khanna
Technical Trainer
Department of Computer Engineering & Applications

ABSTRACT

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of face-dimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faceswas formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project is securing data using face recognition system. As the case of data breaches are increasing day by day, the best way to secure the data is by locking i with entity wich cant be easily replicate decreasing the chance of data leaking. In this project a user can lock a folder by making his/her face as the paswword for the folder.

CONTENTS

1. INTRODUCTION

1.1 Project Description

The project is based on the face recognition and unlocking system. In this project we are going to capture the picture of the client and then program will unlock the folder using the client photo as a password. In this project the client has an advantage to secure his/her folder using the face as a password which will be used to unlock the folder. We will be having multiple modules to encounter this task. There will be the frontend of the folder unlocking system which have two option in the starting. The first one is Register in this option the program will save the 100 photo of the client in the database and map it with his/her name that client will give at the time of capturing the photo. After pushing the data into the Database our model will be ready to recognize the particular photo and will tell you the name of the client according to the photo. Now, After setting up all the data into the database. You will have the option to setup the password on the folder. At the time of setting up the password on the folder. You will give the your face as an input and it will generate the database of the your photo accordingly. When you will try to unlock the folder you need to show your face it will recognize your face using the machine the algorithm CNN and after validating it with our database it will unlock your folder. The another thing that we have included in this project is that beside doing locking and unlocking you can also recognize your face. For that we have another option as well that is recognize which will try to recognize your picture if your picture exist in the database and it also be mapped according to your name. So, like Face recognition project it will also tell client his/her name as well.

1.2 About Face Recognition and Face Detection System

Face recognition is the task of identifying an already detected object as a known or unknown face.Often the problem of face recognition is confused with the problem of face detectionFaceRecognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

1.2.1 Face Recognition

DIFFERENT APPROACHES OF FACE RECOGNITION:

There are two predominant approaches to the face recognition problem: **Geometric** (feature based) and **photometric** (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

Recognition algorithms can be divided into two main approaches:

1. **Geometric :** Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features.

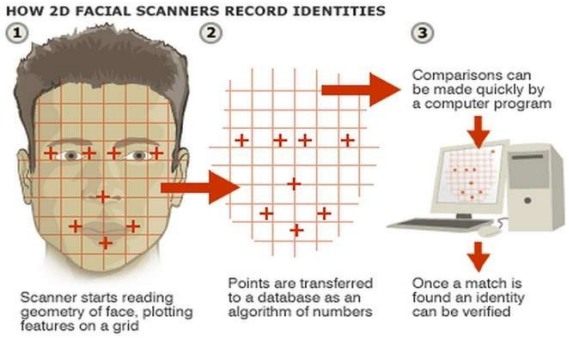


Fig 1.1 Geometric Facial Recognition

2. **Photometric stereo :** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006) (Figure 1.2.2)

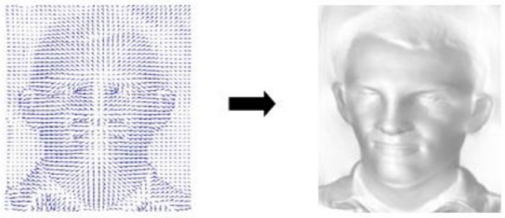


Fig 1.2 Photometric stereo image

Popular recognition algorithms include:

1. Principal Component Analysis using Eigenfaces, (PCA)
2. Linear Discriminate Analysis,
3. Elastic Bunch Graph Matching using the Fisherface algorith

1.2.2 Face Detection

Face detection involves separating image windows into two classes; one containing faces (tarning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin colour and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

Theory or face detection classifiers

A computer program that decides whether an image is a positive image (face image) or negative image (non-face image) is called a classifier. A classifier is trained on hundreds of thousands of face and non-face images to learn how to classify a new image correctly.

OpenCV provides us with two pre-trained and ready to be used for face detection classifiers:

1. Haar Classifier
2. LBP Classifier

Both of these classifiers process images in gray scales, basically because we don't need color information to decide if a picture has a face or not. As these are pre-trained in OpenCV, their learned knowledge files also come bundled with OpenCV opencv/data/.

To run a classifier, we need to load the knowledge files first, as if it had no knowledge.

Each file starts with the name of the classifier it belongs to. For example, a Haar cascade classifier starts off as haarcascade_frontalface_alt.xml.

These are the two types of classifiers we will be using to analyze Casper.

a) HAAR Classifier

The Haar Classifier is a machine learning based approach, an algorithm created by Paul Viola and Michael Jones; which are trained from many many positive images (with faces) and negatives images (without faces).

It starts by extracting Haar features from each image as shown by the windows below:

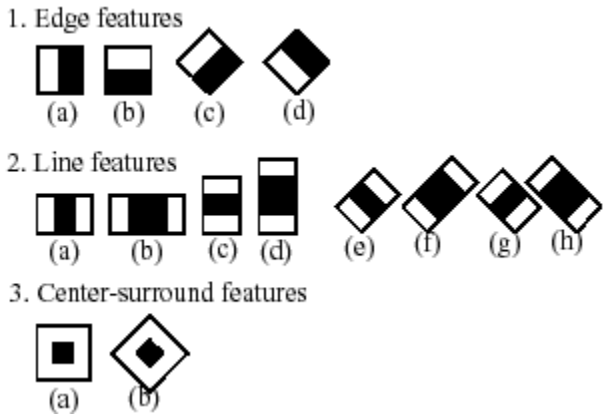


Fig 1.3 Extracting HAAR feature

Each window is placed on the picture to calculate a single feature. This feature is a single value obtained by subtracting the sum of pixels

under the white part of the window from the sum of the pixels under the black part of the window.

In the end, the algorithm considers the fact that generally: most of the region in an image is a non-face region. Considering this, it’s a better idea to have a simple method to check if a window is a non-face region, and if it's not, discard it right away and don’t process it again. So we can focus mostly on the area where a face is.

b) LBP Cascade Classifier

As any other classifier, the Local Binary Patterns, or LBP in short, also needs to be trained on hundreds of images. LBP is a visual/texture descriptor, and our faces are also composed of micro visual patterns. LBP features are extracted to form a feature vector that classifies a face from a non-face

Each training image is divided into some blocks as shown in the picture below.

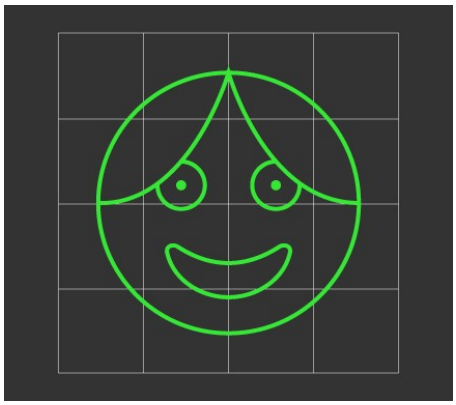
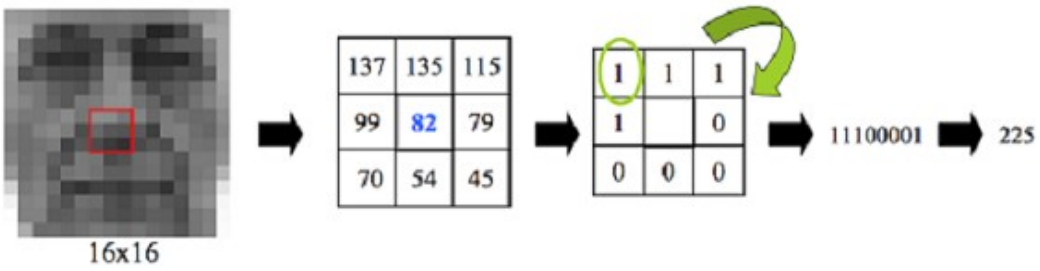


Fig 1.4 Division of training image into blocks

LBP Windows (disregard the first grader drawing)

For each block, LBP looks at 9 pixels (3×3 window) at a time, and with a particular interest in the pixel located in the center of the window.Then, it compares the central pixel value with every neighbor's pixel value under the 3×3 window. For each neighbor pixel that is greater than or equal to the center pixel, it sets its value to 1, and for the others, it sets them to 0.

After that, it reads the updated pixel values (which can be either 0 or 1) in a clockwise order and



forms a binary number. Next, it converts the binary number into a decimal number, and that decimal number is the new value of the center pixel. We do this for every pixel in a block.

Each OpenCV face detection classifier has its pros and cons, but the major differences are in accuracy and speed.

So, in case more accurate detections are required, Haar classifier is the way to go. This is more suitable in technology such as security systems or high-end stalking.

But the LBP classifier is faster, therefore, should be used in mobile applications or embedded systems.

2. METHODOLOGY

Below are step how the flow of project will go to unlock folder using face recogniton system:

1. Interfacing of camera module to capture live face image.
2. Creating a database of authorized person.
3. Capturing current face, saving it and comparing with database in real-time.
4. Interface relay as output module depend on wether to unlock folder on given key or not.

The project will be done in two phase. In Phase 1 the face will be detected and database of user will be created and stored using face detection system and in Phase 2 the data stord will be retieved and compared with real time frame and will predict the name and will unlock the folsder.

2.1 PHASE 1 :

1. Saving the face portion
2. Take frame
3. Detect face
4. Extract the face
5. Resize the extracted image
6. Savethe extracted image

Finally the faces of person will be saved in prove folder with the name of the persons

those faces will then be used to recognize face

2.2 PHASE 2:

1. get faces from training folder
2. compute model
3. get real time frame
4. detect face from reat-time

5. mach that face with stored database
6. predict name according to the database stored
7. compare the key of locked folder
8. unlock the folder

3. TOOLS/ENVIRONMENT USED

3.1 Hardware Requirements:

- Processor : intel i3 core or higher
- RAM : Minimum 4GB primary memory
- Hard disk : Minimum 1GB hard disk space
- Monitor : Preferably color monitor (16 bit color) and above
- Webcamera
- Compact Disk drive
- A keyboard and a mouse

3.2 Software Requirements:

- Operating System : Windows operating system
- Language : Python, Powershell
- Front-end tool : Python Tkinter
- Python Version : Python 3.6 and above
- Jupyter Notebook

4. CONTRIBUTION SUMMARY

- Face Detection module developed by Tanuj Johal in Python
- Locker.bat file for locking folder developed by Anushka soni in Powershell.

5. WORK DETAIL

Work done till now:

1. Face Detection Module
2. Locker Module

Work left:

1. Face Recognition System
2. Creating database of user during real-time
3. Connection of locker module and face recognition module.