**FACE RECOGNITION SYSTEM**

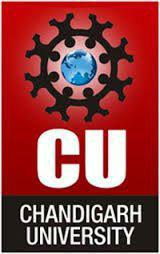
**INDUSTRIAL TRAINING REPORT**

Submitted in partial fulfilment of the requirements for the award of degree of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE ENGINEERING**

**

Submitted By:

RAHUL KUMAR

17BCS1548

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**MAY-JUNE 2019**

**ABOUT THE COMPANY**

EME GROUP was established in 2006 by dedicated team of faculty members who are contributing their knowledge in Research, Industry and Teaching. EME Group is a fastest growing group has founded various Institutes under its umbrella and consists:

• ENGINEERING MADE EASY: It provides coaching for Engineering Entrance Exams like: IES / GATE / PSU

• EME TECHNOLOGIES: It provides 6 Weeks / 6 Months Industrial Training to B-Tech / Diploma / MCA / BCA students

EME Group is software Development & Training Centre, managed by a team of highly qualified software & hardware professionals. They provide trusted and expert training for a few IT companies to their utmost satisfaction. Also provide coaching for exams such as GATE, IES and PSU, full study material and regular updates.

The institute’s objectives is to empower the future computer Professionals by providing them decent work atmosphere, individual attention, creating confidence in them by encouraging them take-up the Project on their own, right from selection of topic until its implementation, under the supervision and guidance of experienced and expert faculty.

**ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and teachers. I would like to extend my sincere thanks to all of them.

We are highly indebted to our Mentor Er. Sunil Bhutani sir for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project.

We would like to express our gratitude towards our parents & faculty of Eme Technology for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

**ABSTRACT**

A face recognition system is one of the biometric information processes, its applicability is easier and working range is larger than others, i.e.; fingerprint, iris scanning, signature, etc. A face recognition system is designed, implemented and tested at Chandigarh University, Computer Engineering Department. The system uses a combination of techniques in two topics; face detection and recognition. The face detection is performed on live acquired images without any application field in mind. Processes utilized in the system are white balance correction, skin like region segmentation, facial feature extraction and face image extraction on a face candidate.

Then a face classification method that uses Feed Forward Neural Network is integrated in the a

system. The system is tested with a database generated.The tested system has acceptable performance to recognize faces within intended limits. System is also capable of detecting and recognizing multiple faces in live acquired images.

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1. | ER diagram of Face Recognition System | 14-15 |
| 2. | Process Flow Diagram | 16-17 |
| 3. | Data Flow Diagram of Face Recognition System | 19-20 |

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
|  |  |  |
| 1. | Introduction | 7-9 |
|  |  |  |
| 2. | SRS | 10-13 |
|  |  |  |
| 3. | Architecture Diagram Like ( DFD, E-R Diagram) | 14-20 |
|  |  |  |
| 4. | Project methodology | 21-30 |
| 5. | Screen shots | 31-35 |
| 6. | Conclusion and Future Scope | 36 |
| 7. | References | 37 |

**Chapter 1**

**Introduction**

What is face recognition? Or what is recognition? When you look at an apple fruit, your mind immediately tells you that this is an apple fruit. This process, your mind telling you that this is an apple fruit is recognition in simple words. So what is face recognition then? I am sure you have guessed it right. When you look at your friend walking down the street or a picture of him, you recognize that he is your friend Paulo. Interestingly when you look at your friend or a picture of him you look at his face first before looking at anything else. Ever wondered why you do that? This is so that you can recognize him by looking at his face. Well, this is you doing face recognition.

But the real question is how does face recognition works? It is quite simple and intuitive. Take a real life example, when you meet someone first time in your life you don't recognize him, right? While he talks or shakes hands with you, you look at his face, eyes, nose, mouth, color and overall look. This is your mind learning or training for the face recognition of that person by gathering face data. Then he tells you that his name is Paulo. At this point your mind knows that the face data it just learned belongs to Paulo. Now your mind is trained and ready to do face recognition on Paulo's face. Next time when you will see Paulo or his face in a picture you will immediately recognize him. This is how face recognition work. The more you will meet Paulo, the more data your mind will collect about Paulo and especially his face and the better you will become at recognizing him.

Face recognition is an important part of the capability of human perception system and is a routine task for human while building a similar computational model of face recognition. The computation model not only contribute to

theoretical insights but also to many practical like automated crowd surveillance, access control, design of human computer interface (HCI), content-based image database management, criminal identification. The earliest work face recognition can be traced back least to the 1950s in psychology, and to the 1960s in the engineering field of the earliest studies include work on facial expression are emotions by Darwin. But research on automatic machine recognition of faces started in the 1970s and after that the seminal work of Kaneda. In 1995, are view paper gave a thorough survey of face recognition technology at that time. At that time, video-based face recognition was still a nascent stage. During the past decades, a face recognition received increased attention and has advanced technically. Many commercial systems for still face recognition are recently significant research efforts have been focused on video based face modelling/tracking, recognition and from integration. New databases have been created and evaluate the recognition techniques using these databases have been carried out. Now, the face recognition has become one of the most active applications of pattern recognition, image analysis and understanding the easiest ways to distinguish the main individual identity of each other. Face recognition is of a personal identification system that uses personal and for characteristics of a person to identify the person's identity.

**Facial Recognition using Python Libraries**

The most popular and probably the simplest way to detect faces using Python is by using the OpenCV package. Originally written in C/C++, OpenCV now provides bindings for Python. It uses machine learning algorithms to search for faces within a picture. Faces are very complicated, made of thousands of small patterns and features that must be matched. The face recognition algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve, known as classifiers.

A face may have 5000 or more classifiers, all of which must match for a face to be detected. Since there are at least 5,000 or more tests per block, you might have millions of calculations to do, which makes it a difficult process. To solve this, OpenCV uses cascades. The OpenCV cascade breaks the problem of detecting faces into multiple stages. It performs a detailed test for each block. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The cascades are a bunch of XML files that contain OpenCV data used to detect objects. You initialize your code with the cascade you want, and then it does the work for you. Since face detection is such a common case, OpenCV comes with a number of built-in cascades for detecting everything from faces to eyes to hands to legs.

You may use other alternatives to OpenCV, like dlib – that come with Deep Learning based Detection and Recognition models.

**Face Recognition using Python Algorithm**

Face Recognition using Python and OpenCV follows a well-defined pattern. When you meet someone for the first time in your life, you look at his/her face, eyes, nose, mouth, color, and overall features. This is your mind learning or training for the face recognition of that person by gathering face data. Then the person tells you his/her name. At this point, your mind knows that the face data it just learned belongs to the person. Now, your mind is trained and ready to do face recognition. Next time when you will see the person or his/her face in a picture you will immediately recognize. This is how Face Recognition works. The more you will meet, the more data your mind will collect about the person and the better you will become at recognizing him/her.

The coding steps for face recognition are the same as we discussed it in real life example above.

**Training Data Gathering**: Gather face data (face images in this case) of the persons you want to recognize

**Training of Recognizer**: Feed that face data (and respective names of each face) to the face recognizer so that it can learn.

**Recognition:** Feed new faces of the persons and see if the face recognizer you just trained recognizes them.

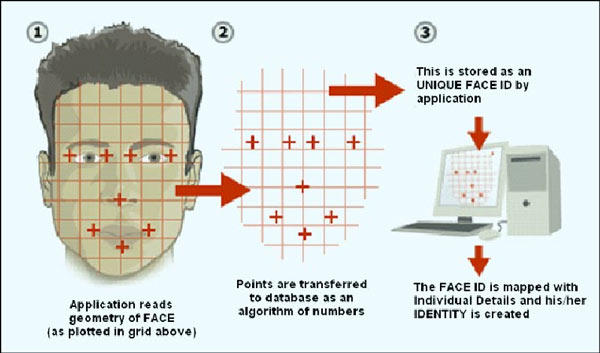


Fig 1.1

**Chapter 2:**

**SRS**

**AIM:** - To make a project based on facial recognition system.

**Introduction**: -Facial recognition system is that system in which the user’s face have been detected by the computer system and after detecting the user logged in into the website.

**PROJECT PROBLEM INVESTIGATION**

**OBJECTIVE OF THE PROJECT-** The main purpose of choosing this project is basically the protection of the user information from the hackers and the unauthorized peoples who are trying to access your system or website without your permission. This project is made with the help of the python and several tools that we are using in this project are as given below. We are also using some concepts of Deep learning it is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

**PROJECT PROBLEM**

**ANALYSE THE PROJECT**:- The main function performed by the user when he/she will logged into the system. For sign in they have to fill their email and the password and after filling all this they have to focus on the camera by doing all these they have to clicked the login and if the id, password and the face of the user matches meet the requirements he/she will successfully login.

If the face doesn’t match the system doesn’t meet all the requirements then they will not able to logged in into the system.

**Face Recognition approaches**:Face as a biometric feature is less reliable due to variations in illumination conditions, poses, and expressions. 3D facial recognition methods resolved the reliability issues like pose change and lighting. Image based face recognition approaches can be classified as under:

* **Holistic Approach:-**In this approach, the complete face is consider as a single feature for detection and recognition. It compares the similarities of whole face, ignoring individual features like eyes, mouth, nose etc.
* **Model Based Approach: -**Model based facial recognition is another approach. 3D facial model can be acquired using both active and passive means [51]. Extensively used active 3D image acquisition technique is infrared input which project laser beam onto an object and records its reflection [52] resulting best and accurate 3D models recognition. Stereo Imaging is the passive technique for the acquisition of 3D model [53] in which two or more cameras simultaneously capture a scene from different angles.

**ALGORITHMS: -**The most traditional and extensively used algorithms in the research of face recognition are PCA [12; 13] and Linear Discriminant Analysis (LDA) [14; 15]. A.M. Martinez and A.C. Kak [16] Shows that PCA is better than LDA with small dataset.

* **Principle Component Analysis (PCA):-**PCA is a most commonly used method in the f when solving single sample per person problems, instead of using general features. PCA is also called as Eigen faces. Eigen faces are set of orthogonal vectors used for human face recognition. Eigen vectors are also referred to as Eigen faces.
* **Linear Discriminant Analysis (LDA):-**LDA is most commonly used for dimensionality reduction and feature extraction [21; 22]. Its variation, 2D LDA has A.
* **TOOLS**

**OPEN-CV: -**OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCV, it can take advantage of the hardware acceleration of the underlying heterogeneous computer platform.

The first alpha version of OpenCV was released to the public at the [IEEE Conference on Computer Vision and Pattern Recognition](https://en.wikipedia.org/wiki/Conference_on_Computer_Vision_and_Pattern_Recognition) in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.

The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the [C++](https://en.wikipedia.org/wiki/C%2B%2B) interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations.

In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.

On May 2016, Intel signed an agreement to acquire Itseez, a leading developer of OpenCV

**Computer languages used in this projects for front end are:**

* PYTHON.

**Computer languages, database and framework used in this project for back end are:**

* File datbase

**OPERATING ENVIRONMENT:**

* WINDOWS 7
* WINDOWS 8,8.1
* WINDOWS 10.

**Software:**

* Spyder.

**ASSUMPTIONS and DEPENDENCIES:**

a) Only the user who had saved his data can access the system.

b) Face recognition system can directly interact with the system

c) Deep learning have to use for train the machine.

**COMMUNICATION INTERFACE:**

**a)** Browser,

**b)** Google Chrome,

**c)**Mozilla,

**d)** Browser supported in windows XP or higher.

**SAFETY REQUIREMENTS:**

BACKUP

Multiple Copies of Data.

**SECURITY REQUIREMENTS:**

In the program we have used the different type of algorithm for some security issues these

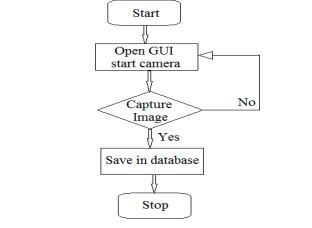
will be safe with us and no one will take any advantage of the personal information as the

data will be end to end encrypted. Your password, personal data will be encrypted.

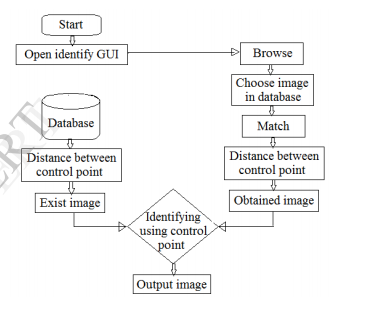
**Chapter 3**

**Architecture Diagram Like (DFD, E-R Diagram)**

**E-R Diagram:**

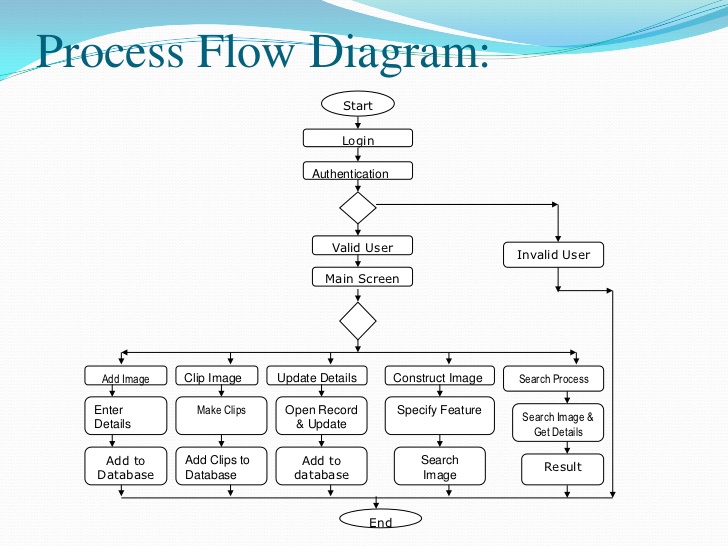
****

**Fig 3.1**

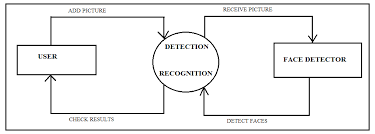
****

**Fig 3.2**

**Process Flow Diagram:**

****

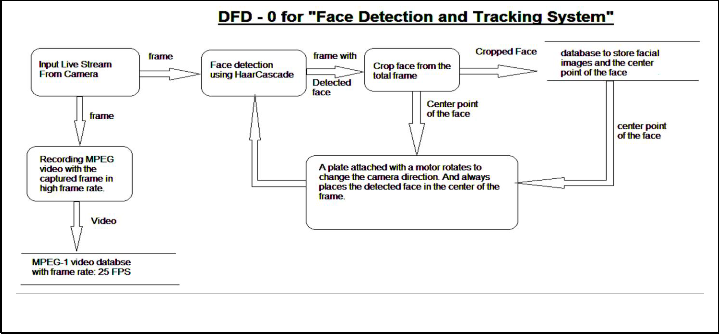
**Fig 3.3**

****

**Fig 3.4**

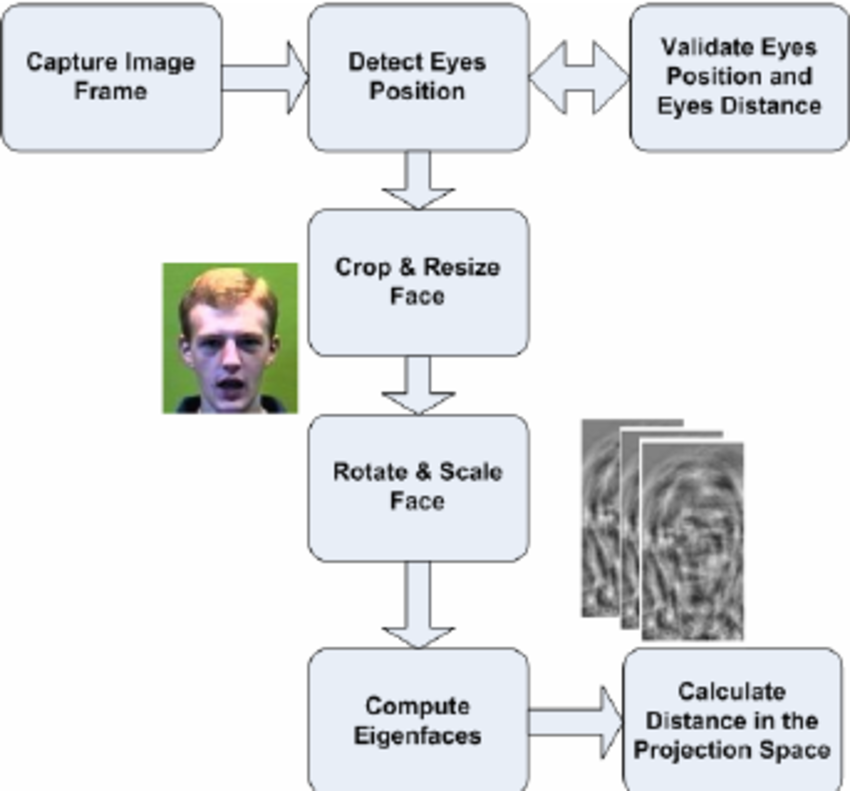
**Data Flow Diagram:**

**Level 0 :**

****

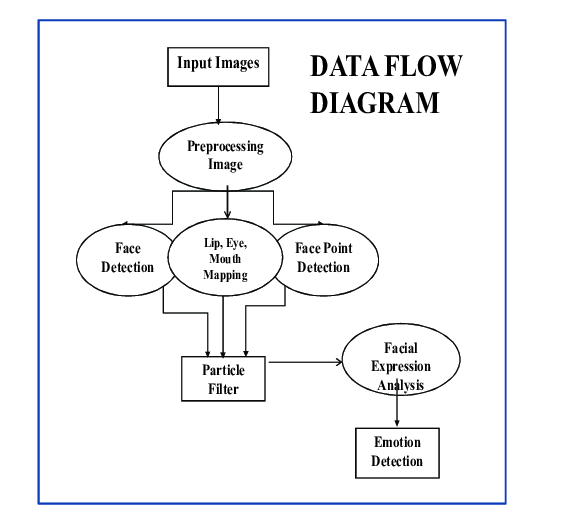
**Fig 3.5**

**Level 1:**

****

**Fig 3.6**

**Level 2 :**

****

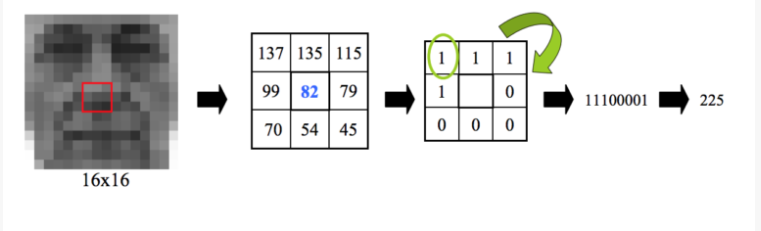
**Fig 3.7**

**Chapter 4**

**Project methodology**

We had used the LBHP face recognize because our project is based on this recognizer only.given below you are read in details that how this method is works.

**Local Binary Patterns Histograms (LBPH) Face Recognizer** Both Eigenfaces and Fisherfaces are affected by light and in real life, perfect light conditions are not always available. LBPH face recognizer is an improvement to overcome this drawback.LBPH algorithm tries to find the local structure of an image and it does that by comparing each pixel with its neighboring pixels. With so much just on the horizon, it will be interesting to see where this rise in Facial Recognition technology takes us.



**Fig 4.1**

Face recognition is fast gaining importance in the various fields. We have entered an age when Facial Recognition technologies will soon be part of everyday life. China, for example, monitors by CCTV or by police wearing special glasses and then logs onto a database that checks on the habitual behaviour of the people, their social credit and even their friends. Cameras and facial recognition are increasingly being used in public and private buildings.

Some schools in the United States are now installing facial recognition systems, to prevent gun attacks by students, given that most rampages are carried out by students whose faces will already be on a database and have full access to the premises. This has led to increased demand for coders and developers with knowledge of Face Recognition algorithms; Python and OpenCV, in particular.

Face Recognition with Python takes just a few lines of code to have a fully working face recognition application and you have the option of switching between face recognizers with a single line of code change. You can come up with detailed codes with a simple approach, and what more, a much better outcome.

Mastering Python for face recognition or otherwise will prepare you better for a rewarding career in Python. Tremendous growth, enormous learning, and lucrative salary are some of the well-known perks of a promising career in Python. Add to that the magic touch of a Data Analytics course, and you are ready to rock!

Python career also offers diversity in terms of career choices. One can start off as a developer or programmer and later switch to the role of a data scientist. With a substantial amount of experience and Python online course certification, one can also become a certified trainer in Python or an entrepreneur. But the bottom line remains the same.

**2.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.

**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 normal (Zhao and Chellappa, 2006)



**Fig 4.2**

**b) 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).

**FEATURE BASE APPROCH:**

We have used the different approaches to enhance the compatibility of the project so that the user can use it in the easier manner of the view that can enable them to show their

Compatibility of the done work not to give the required based and as explained below as:

1)One advantage of 3D face recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view.

2) Three-dimensional data points from a face vastly improve the precision of face recognition. 3D research is enhanced by the development of sophisticated sensors that do a better job of capturing 3D face imagery.

3) The sensors work by projecting structured light onto the face. Up to a dozen or more of these image sensors can be placed on the same CMOS chip—each sensor captures a different part of the spectrum.

4) Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the [Technion](https://en.wikipedia.org/wiki/Technion) applied

tools from [metric geometry](https://en.wikipedia.org/wiki/Metric_geometry) to treat expressions as [isometries](https://en.wikipedia.org/wiki/Isometries).

5) Skin texture analysis: - Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called Skin Texture Analysis, turns the unique lines, patterns, and spots apparent in a person’s skin into a mathematical space.

**Working Model of the Project**

Connected region analysis

Morphological processing

Skin color

segmentation

Input image

Templating matching

Face coordination

**Fig 4.3**

**Skin Color Segmentation: -**

We chose the HSV (Hue, Saturation, Value) color space for segmentation since it decouples the chrominance information from the luminance information. Thus, we can only focus on the hue and the saturation component. The faces in each training image were extracted using the ground truth data and a histogram was plotted for their H and S colorcomponent.The histograms reveal that the H and S color components for faces are nicely clustered. This information was used to define appropriate thresholds for H and S space that correspond to faces. The threshold values were embedded into the color segmentation routine.

**Morphological Processing**: - skin color segmentation did a good job of rejecting non-skin colors from the input image. However, the resulting image has quite a bit of noise and clutter. A series of morphological operations are performed to clean up the image, as shown in. The goal is to end up with a mask image that can be applied to the input image to yield skin color regions without noise and clutter.

**Total numbers of modules**

There are total 5 modules in the project.

**(a)Module 1(Installing the libraries OpenCV, pillow,numpy)**

**b) Module 2(creating the Dataset creator).**

**c)Module 3(train the machine).**

**d)Module 4(connect with database placed in folder).**

**e )Module 5(detector).**

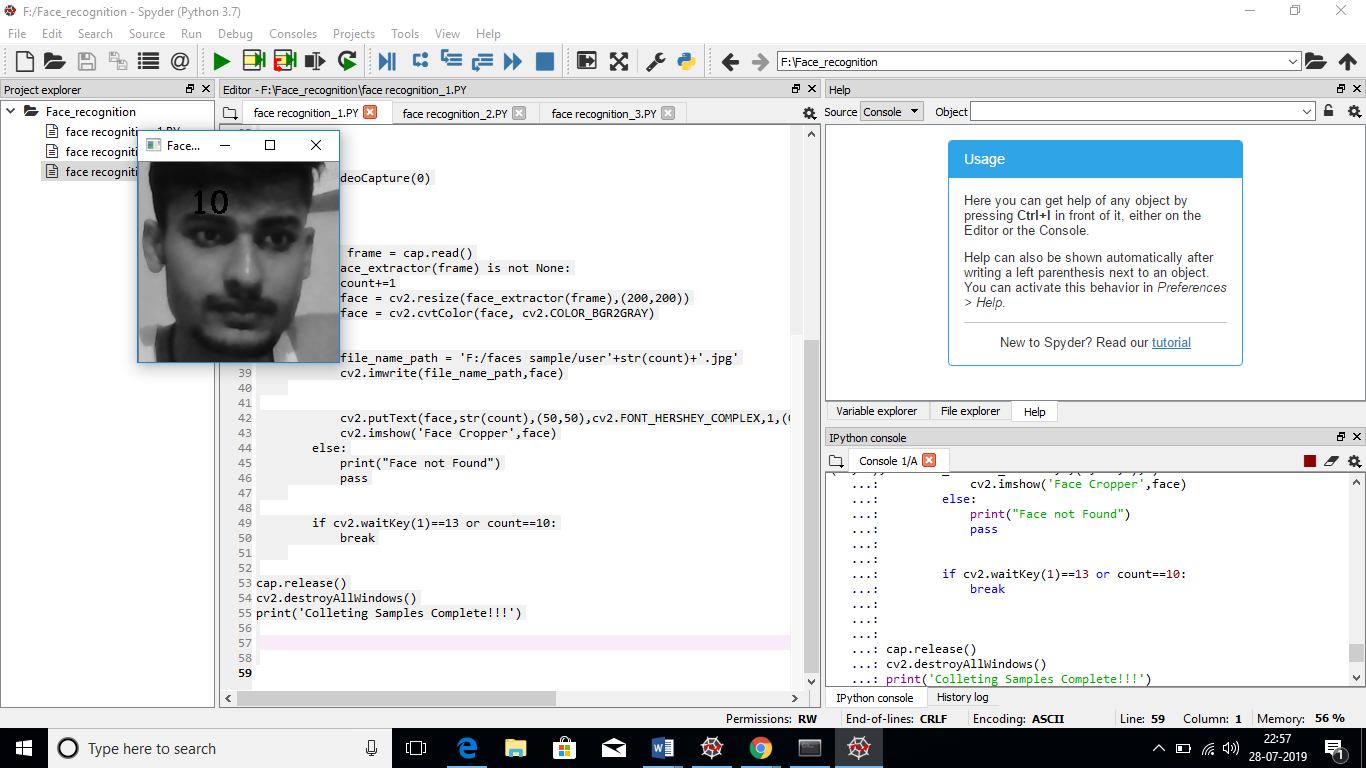
**Description of the covered modules.**

1. **Module 1(Installing the libraries OpenCV, NumPy).:** - For the creation of this project we have installed the many libraries like OpenCV, pillow and the NumPy. This will help for running the different modules so as to create the new format of the project in the basic way to enhance the compatibility of the camera and the system that is used for recognizing the images.

Firstly, we had written the python code by importing this library so that the machine is able to recognize the face.

**Elaborate this with an example**

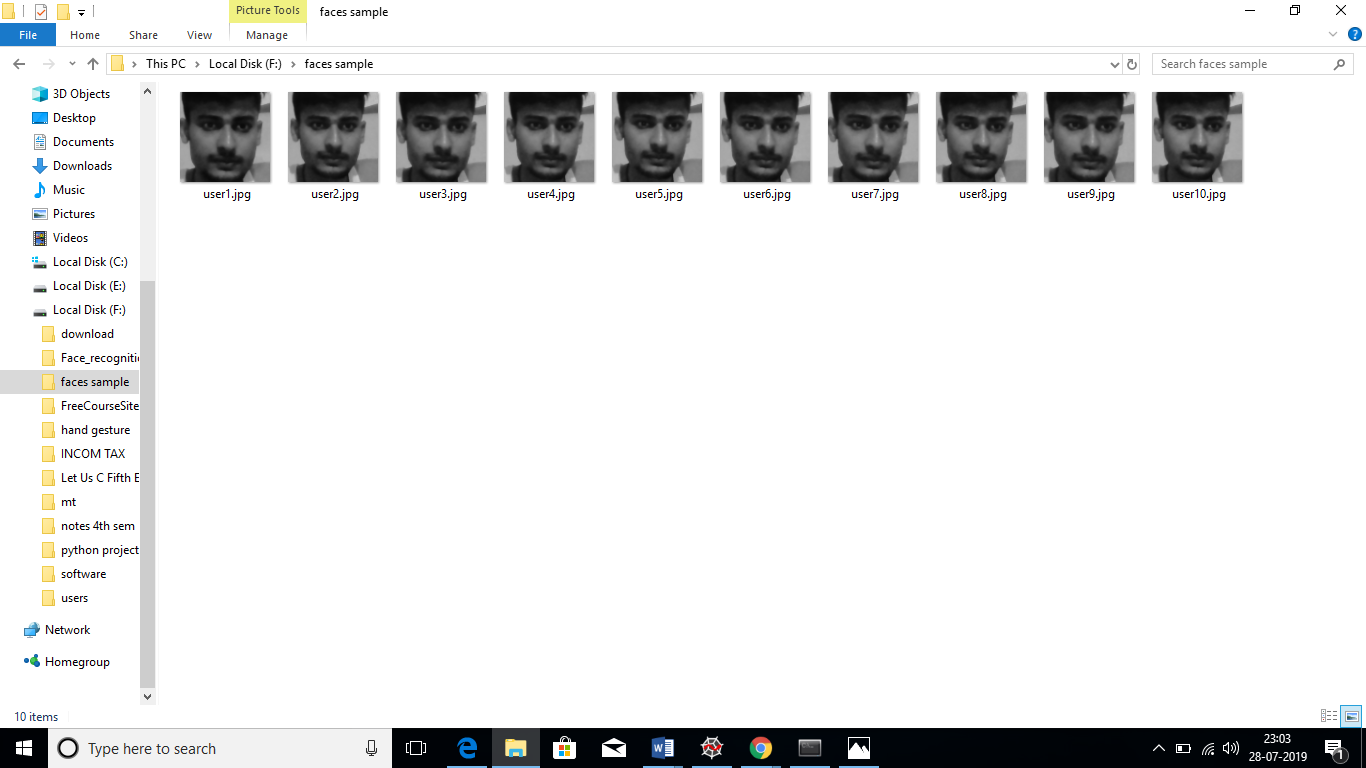
As you will see that the system is now finally recognizing the image and able to detect the face.



**Fig 4.4**

**b) Module 2(creating the Dataset ).** For the recognition of the images we have to create the dataset so that the machine is able to recognize the image and identify of the person properly.

This can be done in that manner that firstly we have to run the code while execution we will observe the camera will open and it start capturing the images of the person who look at the camera and make the dataset at the background. Hence, this process is required if we don’t do the camera is not able to take the dataset of the persons properly and thus this has its own pros and cons. The persons has to look straight at the camera in order to make the perfection if the person watch here and there so it is difficult for the computer to detect the person properly.



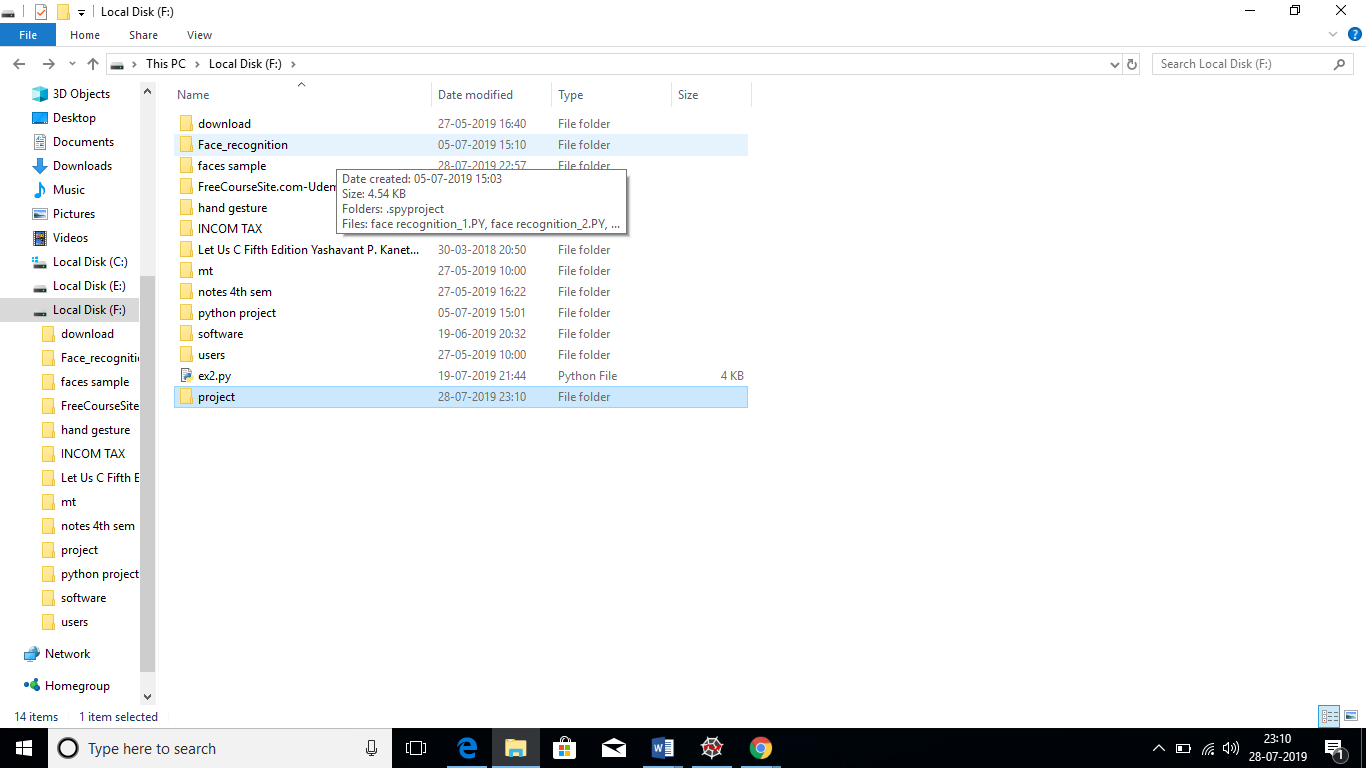
**Fig 4.5**

**c)Module 3(trainer): -**After doing so we will move forward towards the trainer so that the machine will save the dataset at the backend. Whenever the coder wants to check it whether it working or not. Now it save the dataset that we had created in the Module 2 for storing the images.

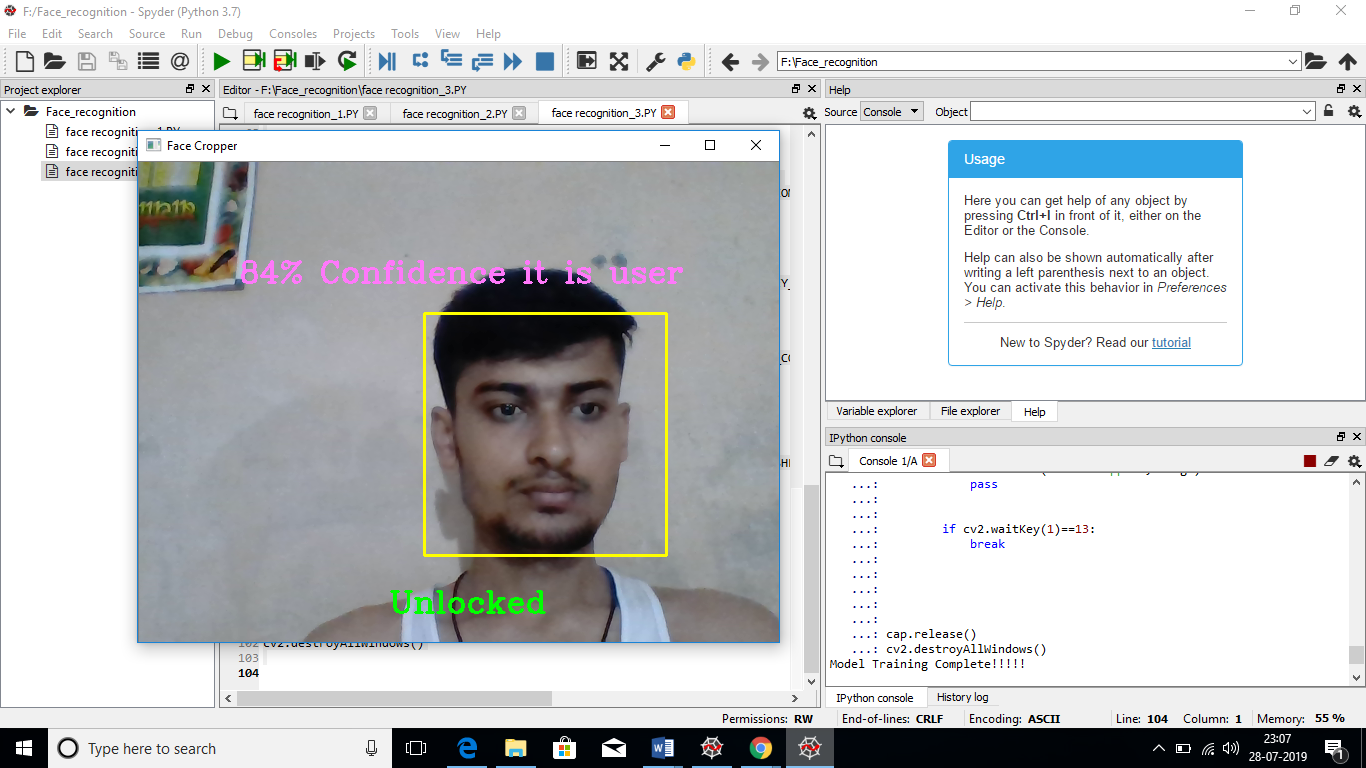
Each image will be trained 10 images per person. With doing this we are able to train our machine properly. After taking this the data will be stored in the database so that it will permanently be stored it if the computer is powered off

**d) Module 4(connect with the file database):**After moving towards the detector part first we have to store the data so that whenever we will run our python program it is able to recognize the image and thus this will be done we we have the database at the backend.

This is the backend of the project and here we had stored our data in order to know very well about the connectivity of the python with the database . After this we will move forward to the result i.e. toward the next module to check whether it is working or not.

****

**e)Module 5 (detector):** In the last we are left with the module 5 in which the face recognition is tested so that the project Is working properly or not.It is clear that the recognizer is able to detect the face properly and show the confidence limit of a face. Hence we are successfully make the project and all the things work according to the work only.

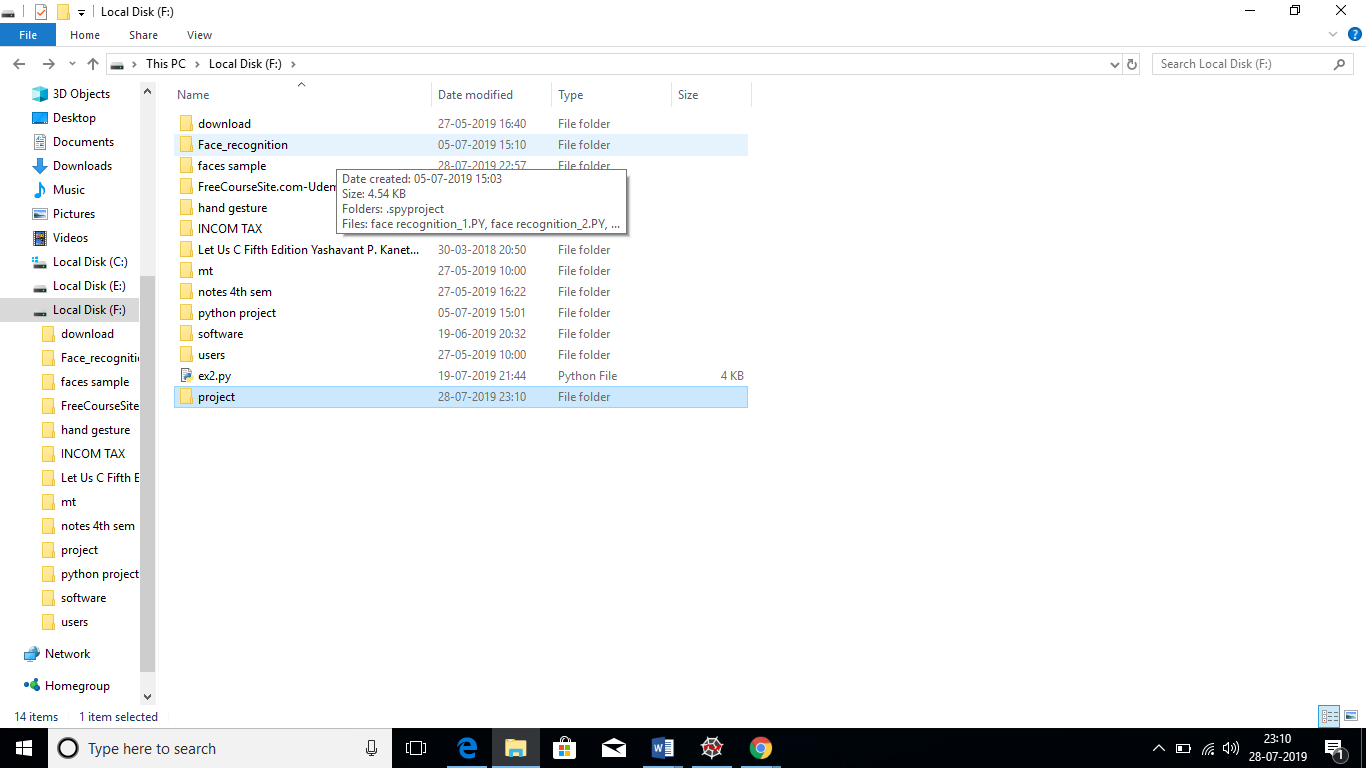


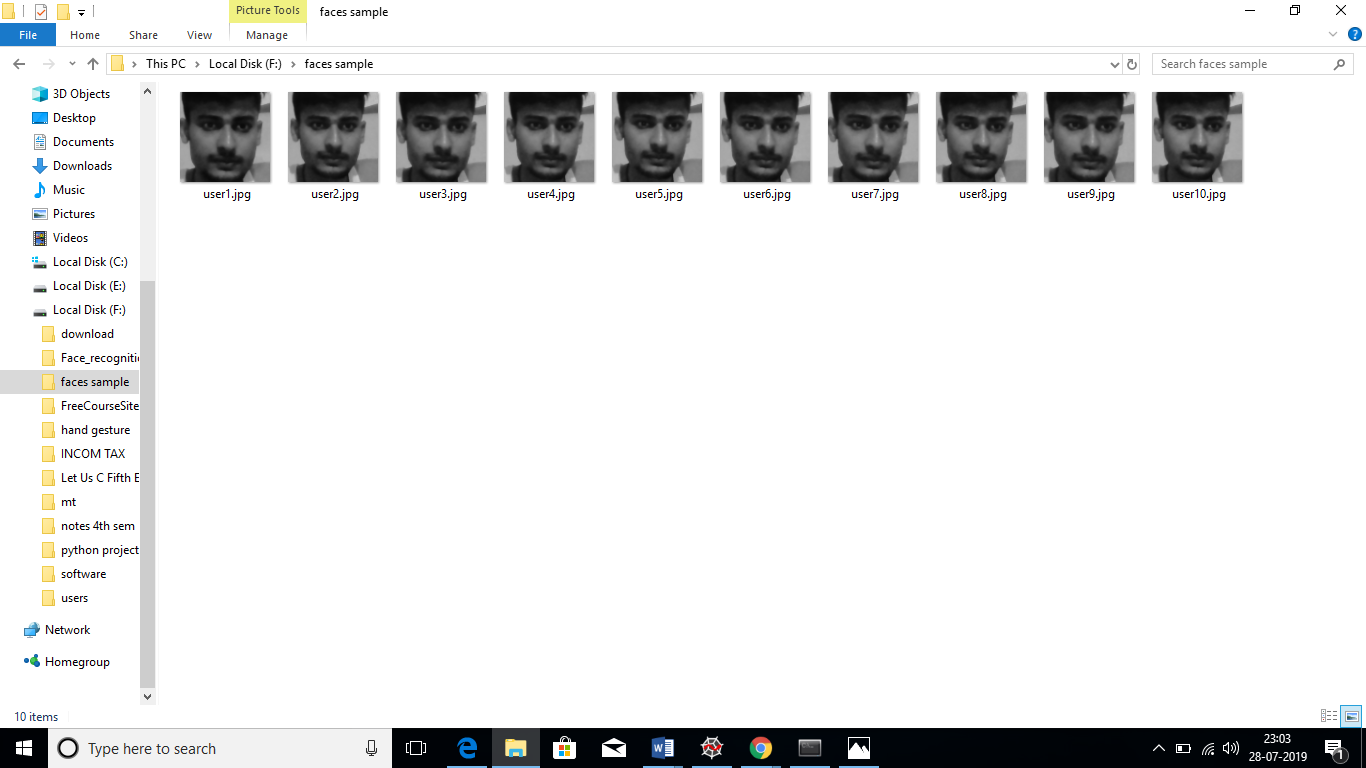
**Fig 4.7**

**Chapter 5**

**SCREENSHOTS**

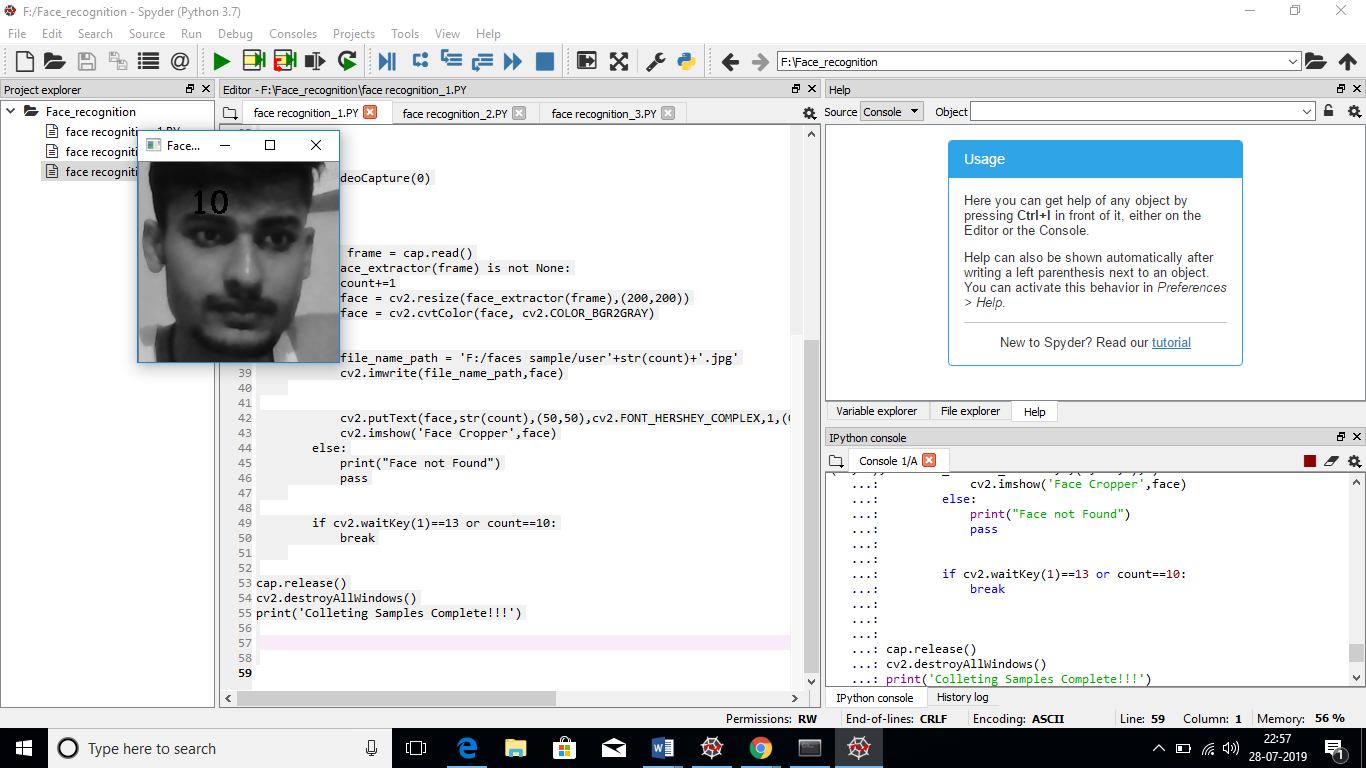
**Database folder:**

** Fig 5.1**

**Datasets: **

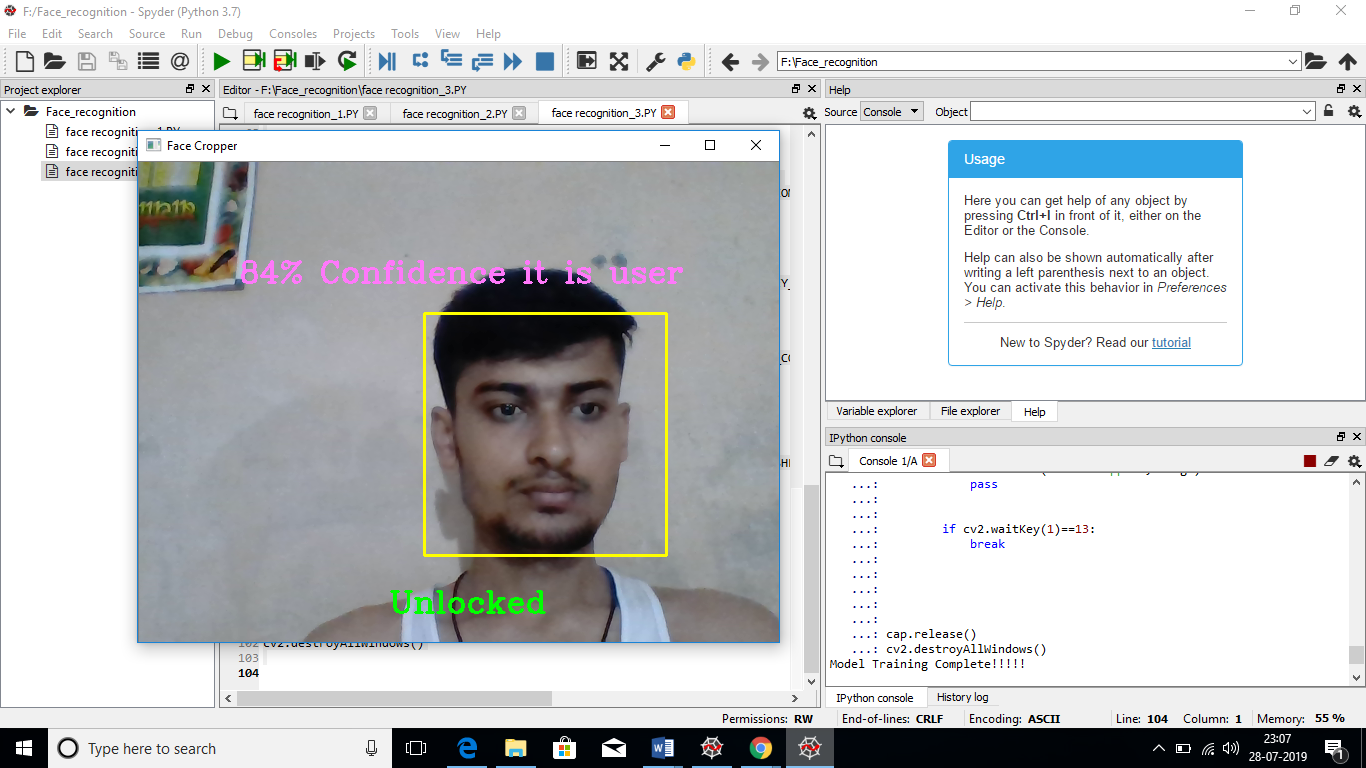
**Fig.5.2**

**Recognizing the face:**

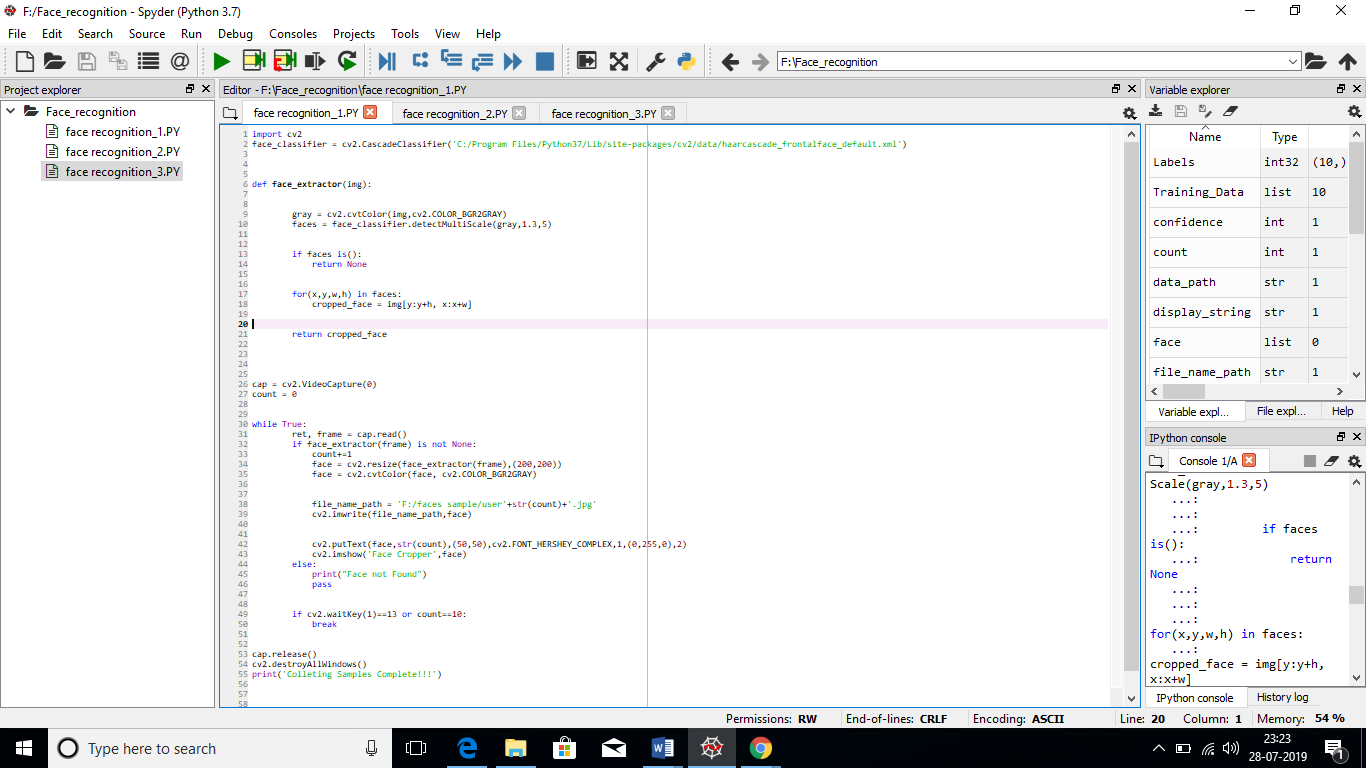


**Fig 5.3**

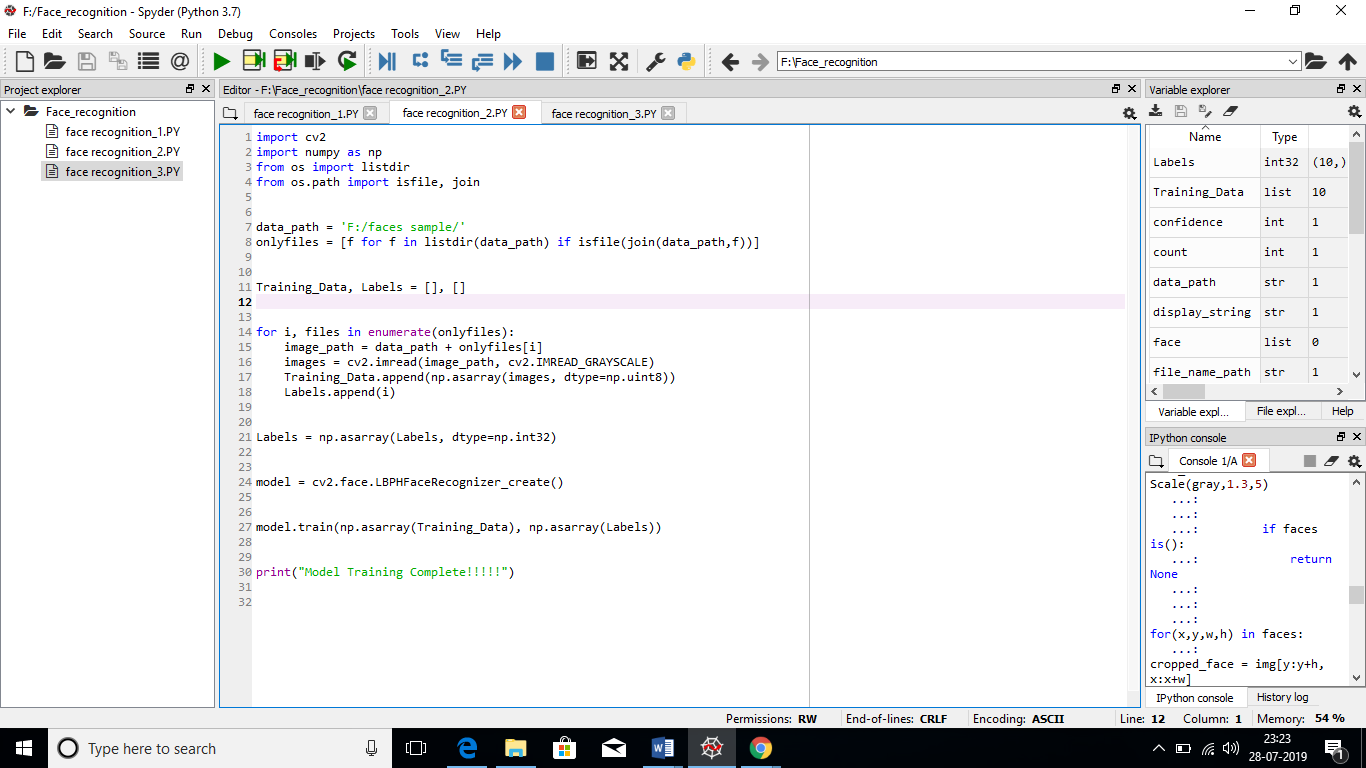
**Detector:**

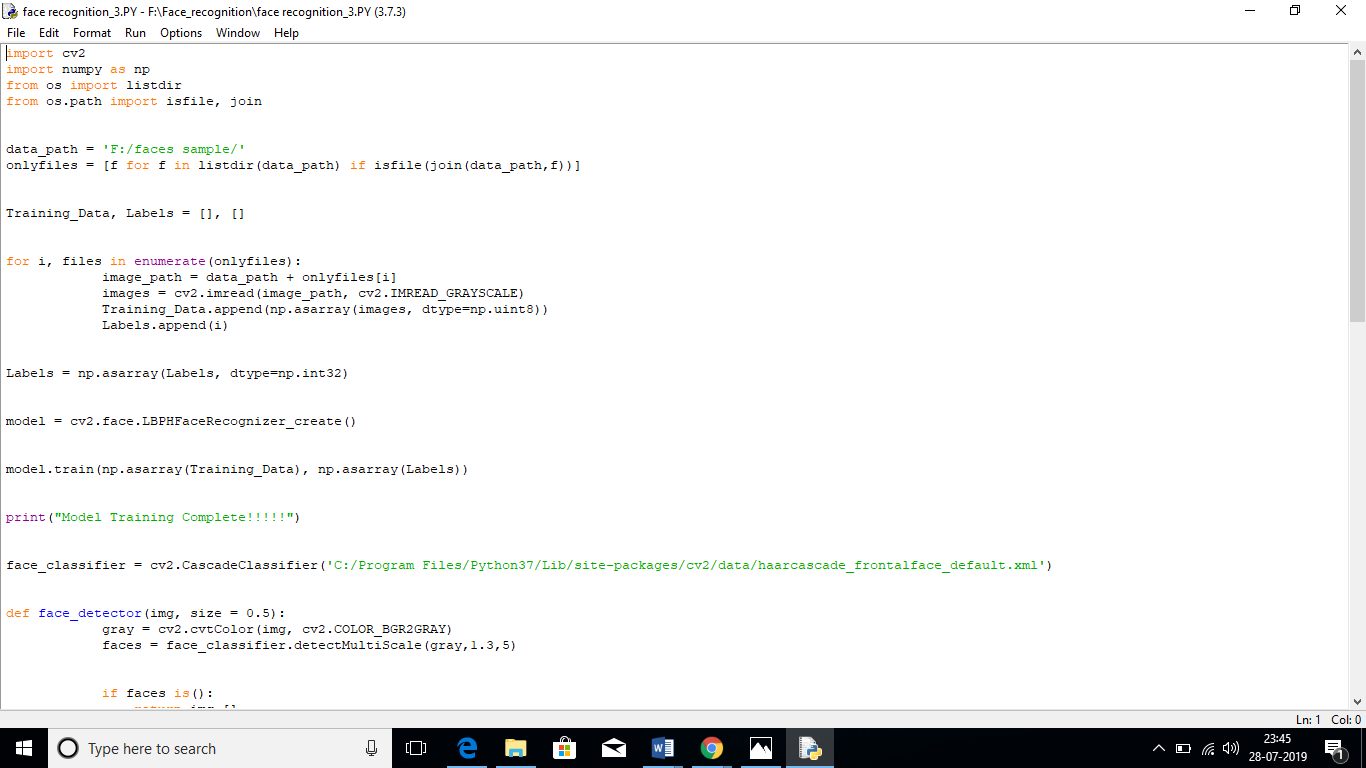
****

**Fig 5.4**

****

**Fig.5.5**

 **Fig 5.6**



**Fig 5.7**

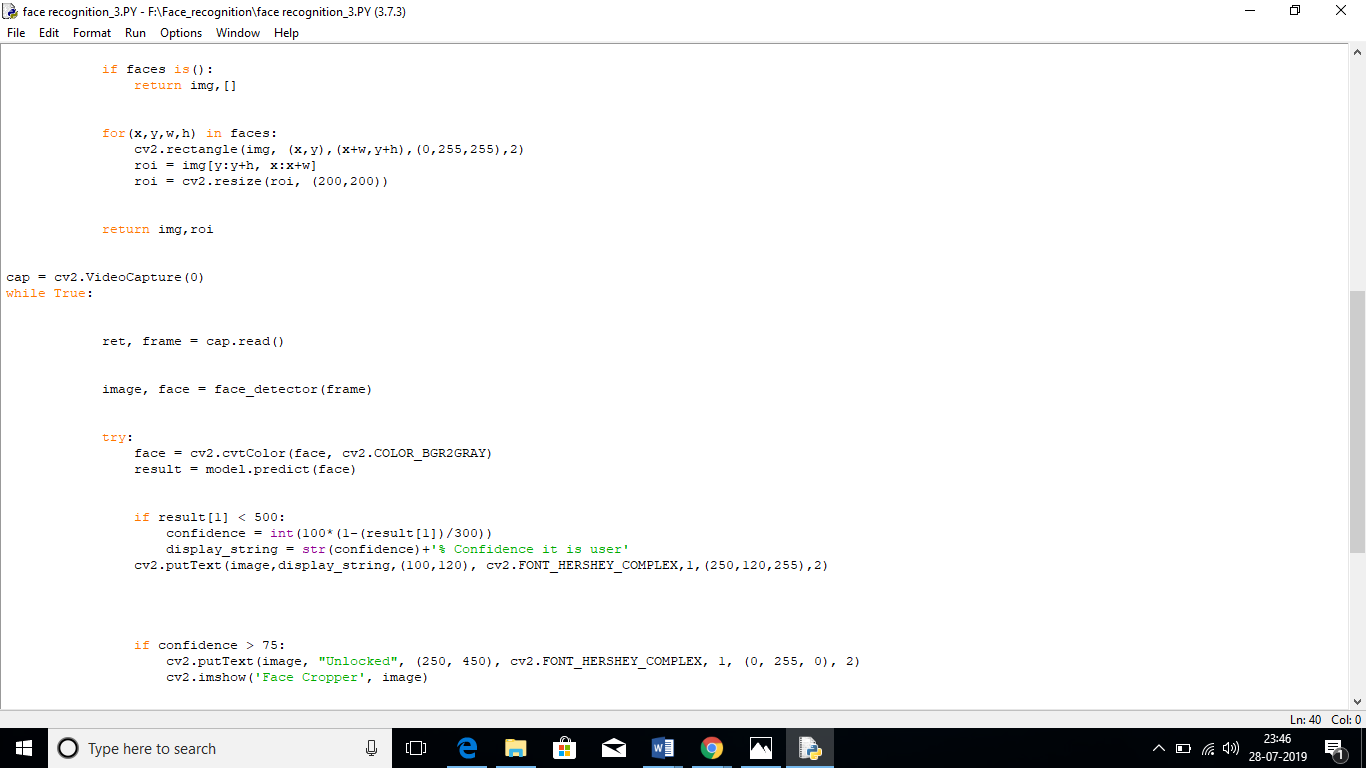


Fig.5.8

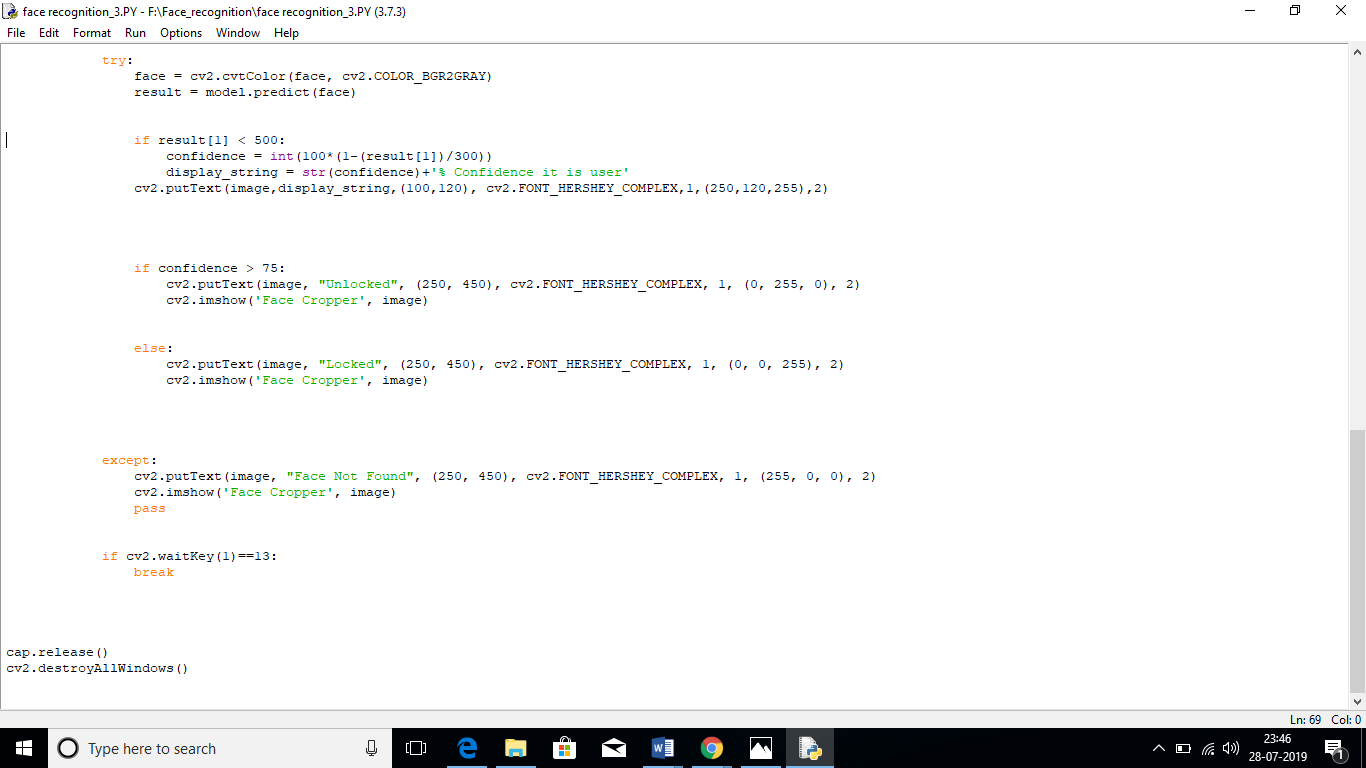


Fig.5.9

**Chapter 6**

**Conclusion and Future Scope**

We are currently extending the system to deal with a range of aspects (other than full frontal views) by defining a small number of face classes for each known person corresponding to characteristic views. Because of the speed of the recognition, the system has many chances within a few seconds to attempt to recognize many slightly different views, at least one of which is likely to fall close to one of the characteristic views. An intelligent system should also have an ability to adapt over time. Reasoning about images in face space provides a means to learn and subsequently recognize new faces in an unsupervised manner.

When an image is sufficiently close to face space (i.e., it is face-like) but is not classified as one of the familiar faces, it is initially labelled as “unknown”. The computer stores the pattern vector and the corresponding unknown image. If a collection of “unknown” pattern vectors cluster in the pattern space, the presence of a new but unidentified face is postulated.

A noisy image or partially occluded face should cause recognition performance to degrade gracefully. since the system essentially implements an auto associative memory for the known faces. This is evidenced by the projection of the occluded face image.

**References**

1. <http://www.google.com/>
2. <http://www.youtube.com/>
3. wikipedia
4. Davies, Ellis, and Shepherd (eds.), Perceiving

and Remembering Faces, Academic Press, London,

1. T. Kanade, “Picture processing system by computer

complex and recognition of human faces,”

Dept. of Information Science, Kyoto University,

Nov. 1973.