**FACE RECOGNITION SYSTEM**

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

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



**Submitted to:**

**Er.Sonam Khattar**

**Submitted By:**

**ATUL , ADITYA , AASHISH**

**Roll No**

**17BCS2268 , 17BCS2275 , 17BCS2270**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**ACKNOWLEDGEMENT**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of our project (Face recognition System) and who influenced our thinking, behaviour and acts during the course of study.

We express our sincere gratitude to our HOD (CSE Department) and all teachers for providing us an opportunity to work on this project and helps us to complete our work on time.

We are thankful to **Er.Sonam mam**, our support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings. Moreover, he provides his valuable suggestions and precious time in accomplishing our project report.

Lastly, we would like to thank the **Almighty** and my **Parents** for their moral support and our friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

**With Regards**

ATUL SHARMA (17BCS2268)

ADITYA MATHUR (17BCS2275)

AASHISH SIDDHARTH JHA (17BCS2270)

**ABSTRACT**

A facial recognition system is a technology capable of [identifying](https://en.wikipedia.org/wiki/Identification_of_human_individuals" \o "Identification of human individuals) or [verifying](https://en.wikipedia.org/wiki/Authentication" \o "Authentication) a person from a [digital image](https://en.wikipedia.org/wiki/Digital_image" \o "Digital image) or a [video frame](https://en.wikipedia.org/wiki/Film_frame" \o "Film frame) from a [video](https://en.wikipedia.org/wiki/Video" \o "Video) source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected [facial features](https://en.wikipedia.org/wiki/Face" \o "Face) from given image with faces within a [database](https://en.wikipedia.org/wiki/Database_management_system" \o "Database management system). It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analyzing patterns based on the person's facial textures and shape.

While initially a form of computer [application](https://en.wikipedia.org/wiki/Application_software" \o "Application software), it has seen wider uses in recent times on mobile platforms and in other forms of technology, such as robotics. It is typically used as access control in [security systems](https://en.wikipedia.org/wiki/Burglar_alarm" \o "Burglar alarm) and can be compared to other [biometrics](https://en.wikipedia.org/wiki/Biometrics" \o "Biometrics) such as [fingerprint](https://en.wikipedia.org/wiki/Fingerprint" \o "Fingerprint) or eye [iris recognition](https://en.wikipedia.org/wiki/Iris_recognition" \o "Iris recognition) systems.Although the accuracy of facial recognition system as a biometric technology is lower than [iris recognition](https://en.wikipedia.org/wiki/Iris_recognition" \o "Iris recognition) and [fingerprint recognition](https://en.wikipedia.org/wiki/Fingerprint" \o "Fingerprint), it is widely adopted due to its contact-less and non-invasive process. Recently, it has also become popular as a commercial identification and marketing tool. Other applications include advanced human-computer interaction, video surveillance, automatic indexing of images, and video database, among others.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **TOPIC NAME** | **PAGE NO.** |
| 1. | INTRODUCTION | 1-2 |
| 2. | HARDWARE REQUIRED | 3 |
| 3. | SOFTWARE REQUIRED | 4-6 |
| 4. | BLOCK DIAGRAM | 7 |
| 5. | SOFTWARE REQUIREMENT SPECIFICATION | 8-10 |
| 6. | ARCHITECTURE DIAGRAM |  |
| 7. | * FLOW CHART | 11 |
| 8. | * DATA FLOW DIAGRAM | 12 |
| 9. | PROJECT METHODOLOGY | 13 |
| 10 | SCREEN SHOTS | 14-15 |
| 11. | CONCLUSION | 16 |
| 12. | FUTURE SCOPE | 17-18 |
| 13. | REFERENCES | 19 |

**INTRODUCTION:-**

“Python based personal assistant” is an AI and ML based project which will be created using Python language, which provides suitable libraries to work upon the data set.

One of the goals of Artificial intelligence (AI) is the realization of natural dialogue between humans and machines. in recent years, the dialogue systems, also known as interactive conversational systems are the fastest growing area in AI. Many companies have used the dialogue systems technology to establish various kinds of Virtual Personal Assistants(VPAs) based on their applications and areas, such as Microsoft’s Cortana, Apple’s Siri, Amazon Alexa, Google Assistant, and Facebook’s M. However, in this proposal, we have used the multi-modal dialogue systems which process two or more combined user input modes.

Personal assistant refers to a better-quality assistant system that will give result to user that the user may be interested in. So, this system basically provides the results on the demand of the user.

We may take example of Google assistant, suppose a user want to listen the music or search on the internet than it can be done by the personal assistant. So, there must be some algorithms or techniques

So, this project will include the implementation of those algorithms and techniques of ML and AI for Virtual assistant systems.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Now a days, machine learning applications are promulgated in finance, health care, pattern recognition, networking, robotics, etc.

Computers have became a very important devices and as well as less expensive over time.  
The application works same like Siri/ Google Assistant etc. But the application deals with the computer itself mainly.  
The U.I of the application is self-explanatory and minimal.  
Currently it takes text as input as most of the people are not very good at speaking.

**TECHNOLOGY USED**

**Python Programing**

****

Python is one of those rare languages which can claim to be both *simple* and *powerful*. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in.

The official introduction to Python is:

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

I will discuss most of these features in more detail in the next section.

**HARDWARE REQUIRED**

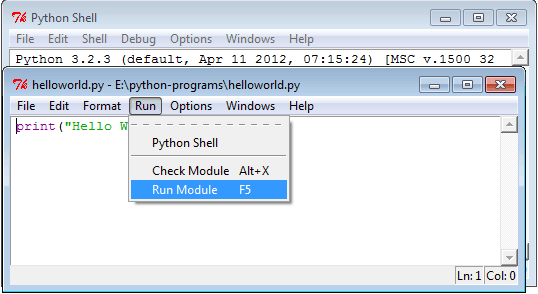
* PC(Personal Computer) / Laptop
* Minimum Ram :-4 GB
* Minimum Harddisk:-500 GB
* Windows 7/8/10
* Camera (Higher the Quality Better the Results)

**SOFTWARE REQUIRED**:-

**Python (3.7..)**

IDLE (short for integrated development environment or integrated development and learning environment) is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment" \o "Integrated development environment) for [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)), which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many [Linux distributions](https://en.wikipedia.org/wiki/Linux_distributions" \o "Linux distributions). It is completely written in Python and the [Tkinter](https://en.wikipedia.org/wiki/Tkinter" \o "Tkinter) GUI toolkit ([wrapper](https://en.wikipedia.org/wiki/Wrapper_function" \o "Wrapper function) functions for [Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl)/[Tk](https://en.wikipedia.org/wiki/Tk_(framework)" \o "Tk (framework))).

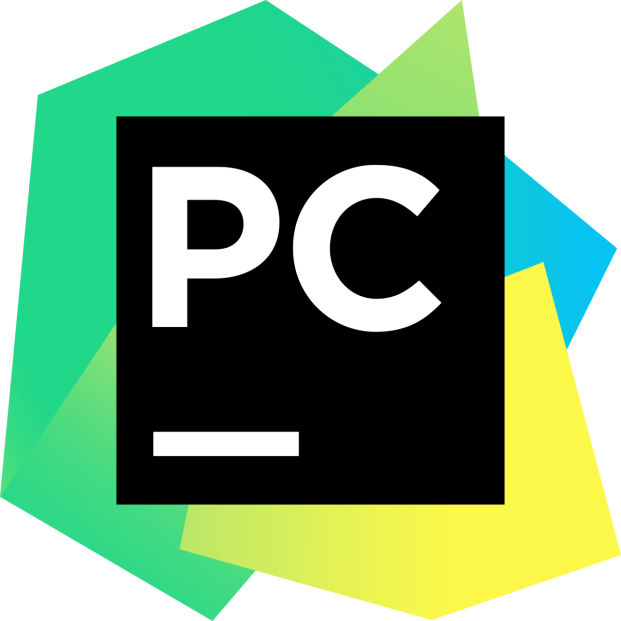
IDLE is intended to be a simple [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment" \o "Integrated development environment) and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

****

* **Pycharm**

PyCharm is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment" \o "Integrated development environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming" \o "Computer programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) language. It is developed by the Czech company [JetBrains](https://en.wikipedia.org/wiki/JetBrains" \o "JetBrains).[[2]](https://en.wikipedia.org/wiki/PyCharm" \l "cite_note-2) It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control systems](https://en.wikipedia.org/wiki/Revision_control" \o "Revision control) (VCSes), and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)" \o "Django (web framework)).

PyCharm is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \o "Cross-platform), with [Windows](https://en.wikipedia.org/wiki/Windows" \o "Windows), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux" \o "Linux) versions. The Community Edition is released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License" \o "Apache License),[[3]](https://en.wikipedia.org/wiki/PyCharm" \l "cite_note-community-3) and there is also Professional Edition released under a [proprietary license](https://en.wikipedia.org/wiki/Proprietary_software" \o "Proprietary software) - this has extra features.

****

* **OpenCV**

OpenCV (*Open source computer vision*) is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision" \o "Computer vision). Originally developed by [Intel](https://en.wikipedia.org/wiki/Intel_Corporation" \o "Intel Corporation), it was later supported by [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage" \o "Willow Garage) then Itseez (which was later acquired by Intel. The library is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \o "Cross-platform) and free for use under the [open-source](https://en.wikipedia.org/wiki/Open-source" \o "Open-source) [BSD license](https://en.wikipedia.org/wiki/BSD_license" \o "BSD license).

**History**

Officially launched in 1999, the OpenCV project was initially an [Intel Research](https://en.wikipedia.org/wiki/Intel_Research_Lablets" \o "Intel Research Lablets) initiative to advance [CPU](https://en.wikipedia.org/wiki/Central_processing_unit" \o "Central processing unit)-intensive applications, part of a series of projects including [real-time](https://en.wikipedia.org/wiki/Real-time_computing" \o "Real-time computing) [ray tracing](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)" \o "Ray tracing (graphics)) and [3D display](https://en.wikipedia.org/wiki/3D_Display" \o "3D Display) walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel’s Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

* Advance vision research by providing not only open but also [optimized code](https://en.wikipedia.org/wiki/Code_optimization" \o "Code optimization) for basic vision infrastructure. No more [reinventing the wheel](https://en.wikipedia.org/wiki/Reinventing_the_wheel" \o "Reinventing the wheel).
* Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
* Advance vision-based commercial applications by making [portable](https://en.wikipedia.org/wiki/Portability_(computer_science)" \o "Portability (computer science)), performance-optimized code available for free – with a license that did not require code to be open or free itself.

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" \o "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" \o "C++) 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[[6]](https://en.wikipedia.org/wiki/OpenCV" \l "cite_note-6) 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[[7]](https://en.wikipedia.org/wiki/OpenCV" \l "cite_note-7) and user site.[[8]](https://en.wikipedia.org/wiki/OpenCV" \l "cite_note-8)

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

* **Applications**

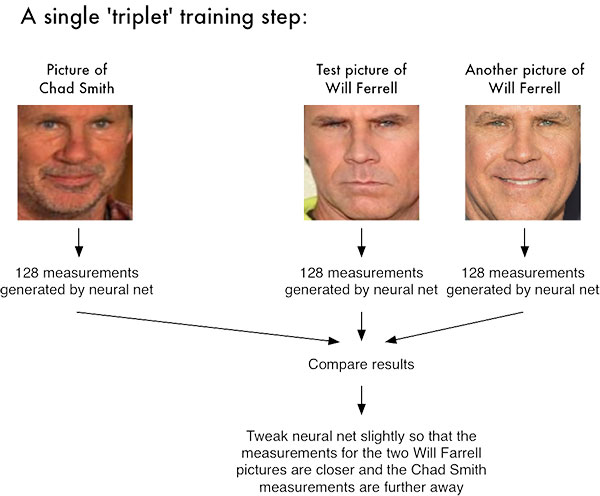
OpenCV's application areas include:

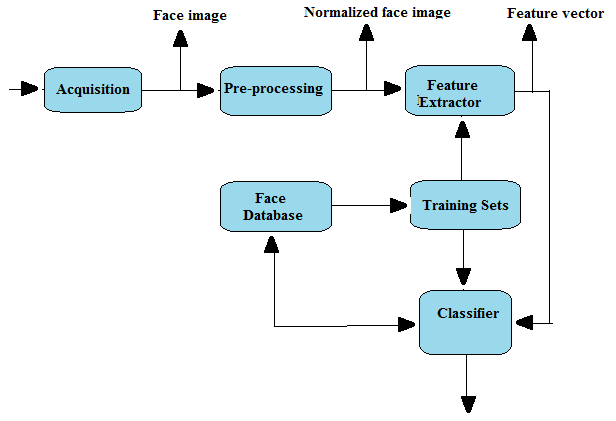
* 2D and 3D feature toolkits
* [Egomotion](https://en.wikipedia.org/wiki/Egomotion" \o "Egomotion) estimation
* [Facial recognition system](https://en.wikipedia.org/wiki/Facial_recognition_system" \o "Facial recognition system)
* [Gesture recognition](https://en.wikipedia.org/wiki/Gesture_recognition" \o "Gesture recognition)
* [Human–computer interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction" \o "Human–computer interaction) (HCI)
* [Mobile robotics](https://en.wikipedia.org/wiki/Mobile_robotics" \o "Mobile robotics)
* Motion understanding
* Object identification
* [Segmentation](https://en.wikipedia.org/wiki/Segmentation_(image_processing)" \o "Segmentation (image processing)) and recognition
* [Stereopsis](https://en.wikipedia.org/wiki/Stereopsis" \o "Stereopsis) stereo vision: depth perception from 2 cameras
* [Structure from motion](https://en.wikipedia.org/wiki/Structure_from_motion" \o "Structure from motion) (SFM)
* [Motion tracking](https://en.wikipedia.org/wiki/Video_tracking" \o "Video tracking)
* [Augmented reality](https://en.wikipedia.org/wiki/Augmented_reality" \o "Augmented reality)

To support some of the above areas, OpenCV includes a statistical [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) library that contains:

* [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm)" \o "Boosting (meta-algorithm))
* [Decision tree learning](https://en.wikipedia.org/wiki/Decision_tree_learning" \o "Decision tree learning)
* [Gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting" \o "Gradient boosting) trees
* [Expectation-maximization algorithm](https://en.wikipedia.org/wiki/Expectation-maximization_algorithm" \o "Expectation-maximization algorithm)
* [k-nearest neighbor algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm" \o "K-nearest neighbor algorithm)
* [Naive Bayes classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier" \o "Naive Bayes classifier)
* [Artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network" \o "Artificial neural network)
* [Random forest](https://en.wikipedia.org/wiki/Random_forest" \o "Random forest)
* [Support vector machine](https://en.wikipedia.org/wiki/Support_vector_machine" \o "Support vector machine) (SVM)
* [Deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_network" \o "Deep neural network) (DNN) [[9]](https://en.wikipedia.org/wiki/OpenCV" \l "cite_note-9)

**BLOCK DIAGRAM:-**



** SOFTWARE REQUIREMENTS SPECIFICATIO**

* **Purpose**

Facial recognition is mostly used for security purposes, though there is increasing interest in other areas of use. In fact, facial recognition technology has received significant attention as it has potential for a wide range of application related to law enforcement as well as other enterprises.

* **Scope of Project**

A face recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation. Recognition algorithms can be divided into two main approaches, geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances.

### Performance The System Requirements

The system will accept faces for datasets and check the other faces for the similarity

### User Interfaces

The user interface will run on Windows as pythin executable files

### Hardware Interfaces

The facial recognition system requires a good camera for a good dataset . The camera requires necessary driver installed within the operating system.. The external camera requires 1 USB port on the PC.

### Software Interfaces

Python and its libraries mentioned above

### Communication Interfaces

There are no external communications interface requirements

### Security

The computer that runs the program will have its own security. Only the System Admin will log in to the system with his/her username and password. The person whose face and the iris are recognized will access to view the output.

### Maintainability

As a tool to obtain the ease of maintainability UML will be used in the development process.

### Portability

To ensure portability, the application will be developed in PYTHON language.

**Technology Enablers**

**Smart phones – CPU, memory, storage and gesture interface** -Smart phones in the market today offer powerful computing environment in terms of CPU, memory, and file storage that can be used to run complex software applications.

Touch screen and video gesture interface being offered by these phones enables a more user friendly interface that can make the interaction with the smart agent easier as well as reduce the friction of using such an interface.

**Face recognition** - Recognition software enables identify any face in the view of the camera by using ” haar cascades” and “open CV” libraries it makes a rectangle around the detected face to crop the screenshot along side the rectangle.

* OpenCV
* Python
* Deep learning

**High Quality Face Recognition with Deep Metric Learning**

**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 OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

**Pillow Library**

**Python Imaging Library** (abbreviated as **PIL**) (in newer versions known as Pillow) is a [free](https://en.wikipedia.org/wiki/Free_and_open_source_software" \o "Free and open source software) [library](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux. The latest version of PIL is 1.1.7, was released in September 2009 and supports Python 1.5.2–2.7, with [Python 3](https://en.wikipedia.org/wiki/Python_3" \o "Python 3) support to be released "later".

**Num-Py Library**

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

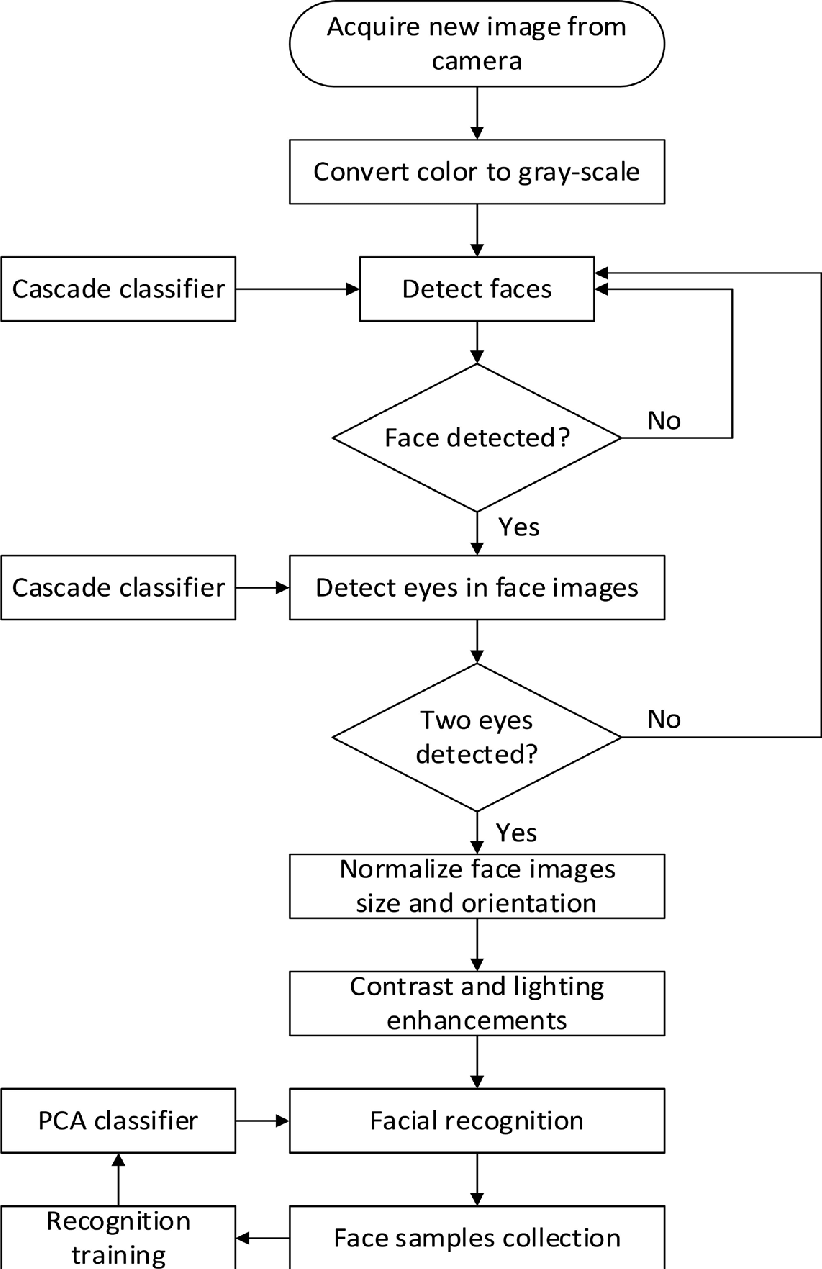
Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**Flow Chart: -**

A flowchart is a common type of [chart](http://en.wikipedia.org/wiki/Chart" \o "Chart), that represents an [algorithm](http://en.wikipedia.org/wiki/Algorithm" \o "Algorithm) or a [process](http://en.wikipedia.org/wiki/Process_%28general%29" \o "Process (general)), showing the steps as boxes of various kinds, and their order by connecting these with arrows. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

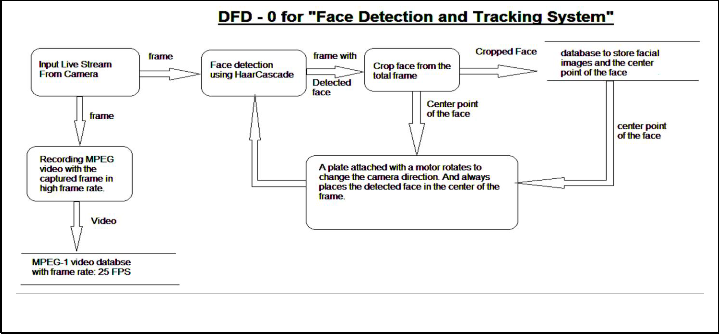
Flowcharts are used in designing and documenting complex processes. Like other types of diagram, they help visualize what is going on and thereby help the viewer to understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

* A processing step, usually called activity, and denoted as a rectangular box
* A decision usually denoted as a diamond.

****

**DATA FLOW DIAGRAM**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A **DFD** is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.



**PROJECT METHODOLOGY**

**Set up Hardware and Camera Accessories:**

1. Connect the keyboard, mouse, and a monitor to the computer.

2. Make sure you are equipped with a high quality webcam if not , connect an external one

**Install the Python Libraries for the code:**

Follow these instructions to install the SDK and sample code on your device. Run all of the commands on this page in a terminal on the device (either directly or via an SSH connection).

*pip install opencv-python*

*pip install Pillow*

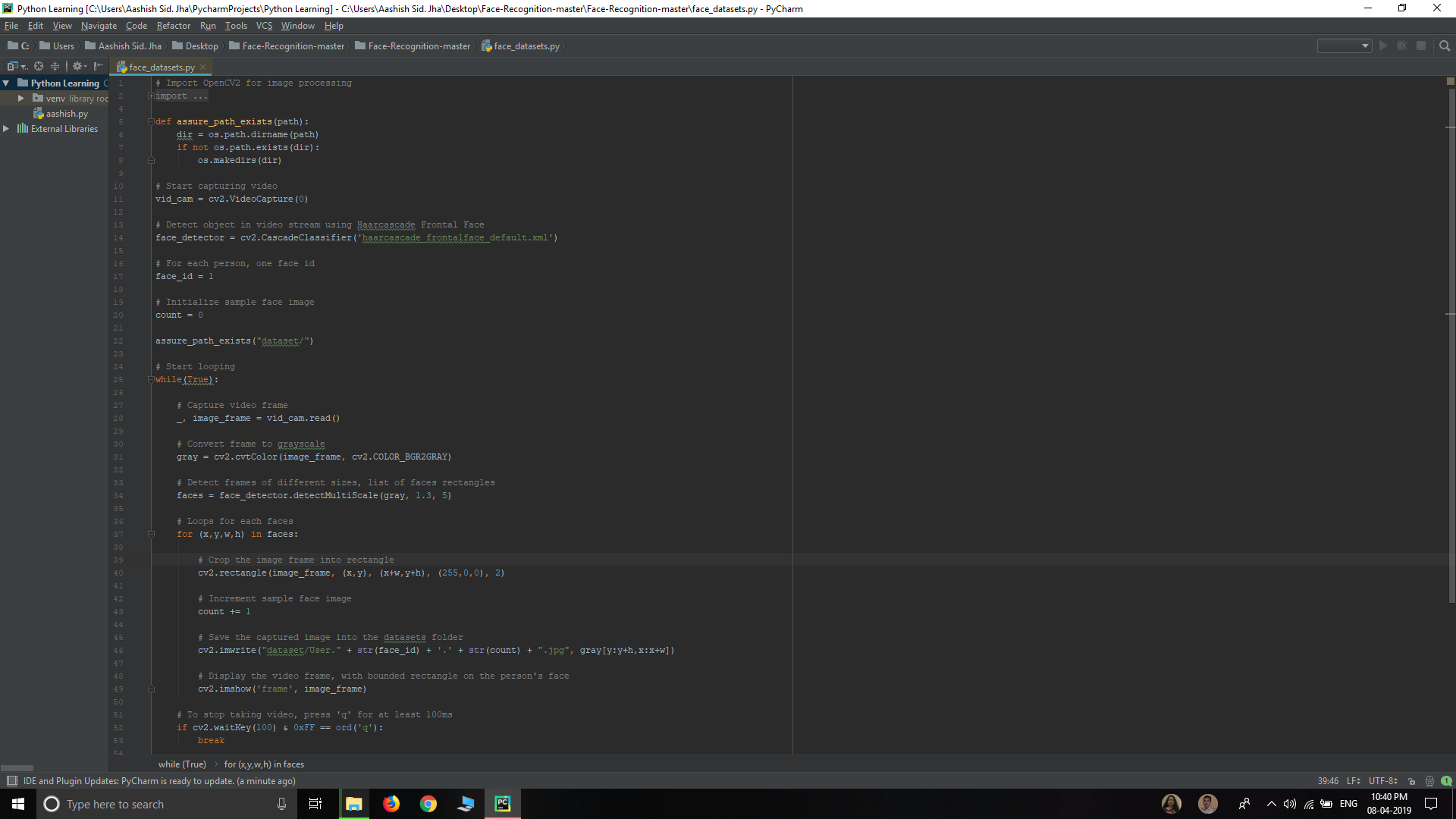
*pip intall opencv-contrib-python*

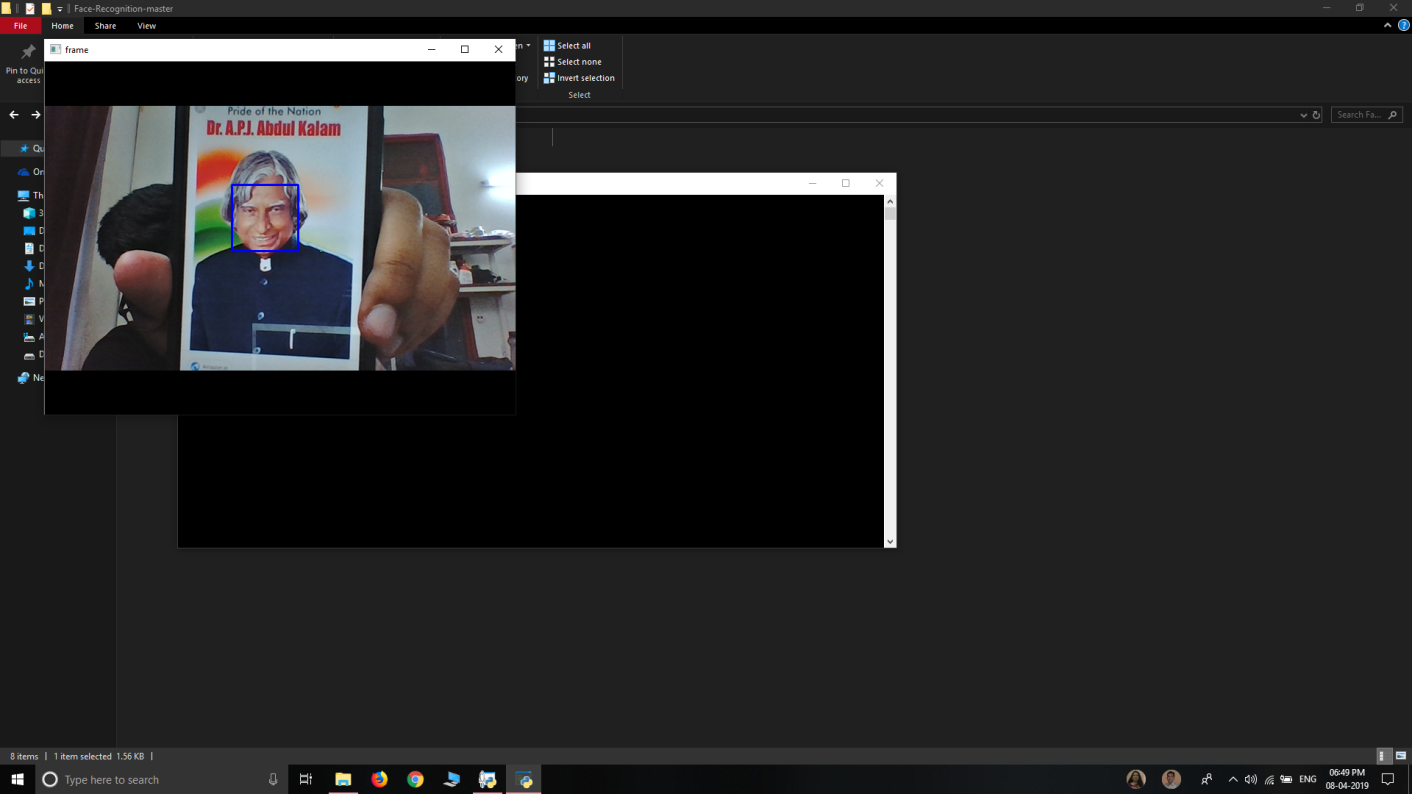
**Run the sample code:**

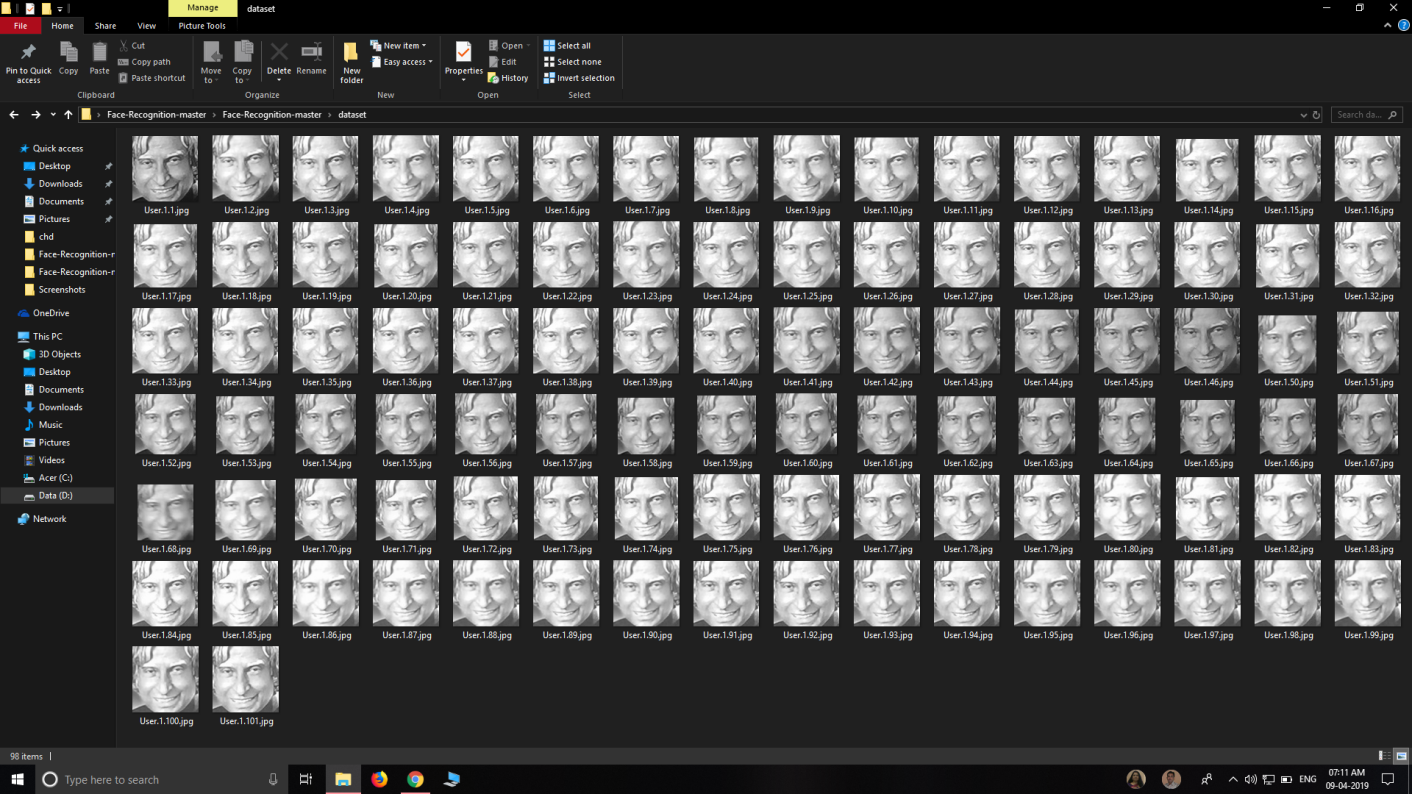
At this point, you are ready to run the sample

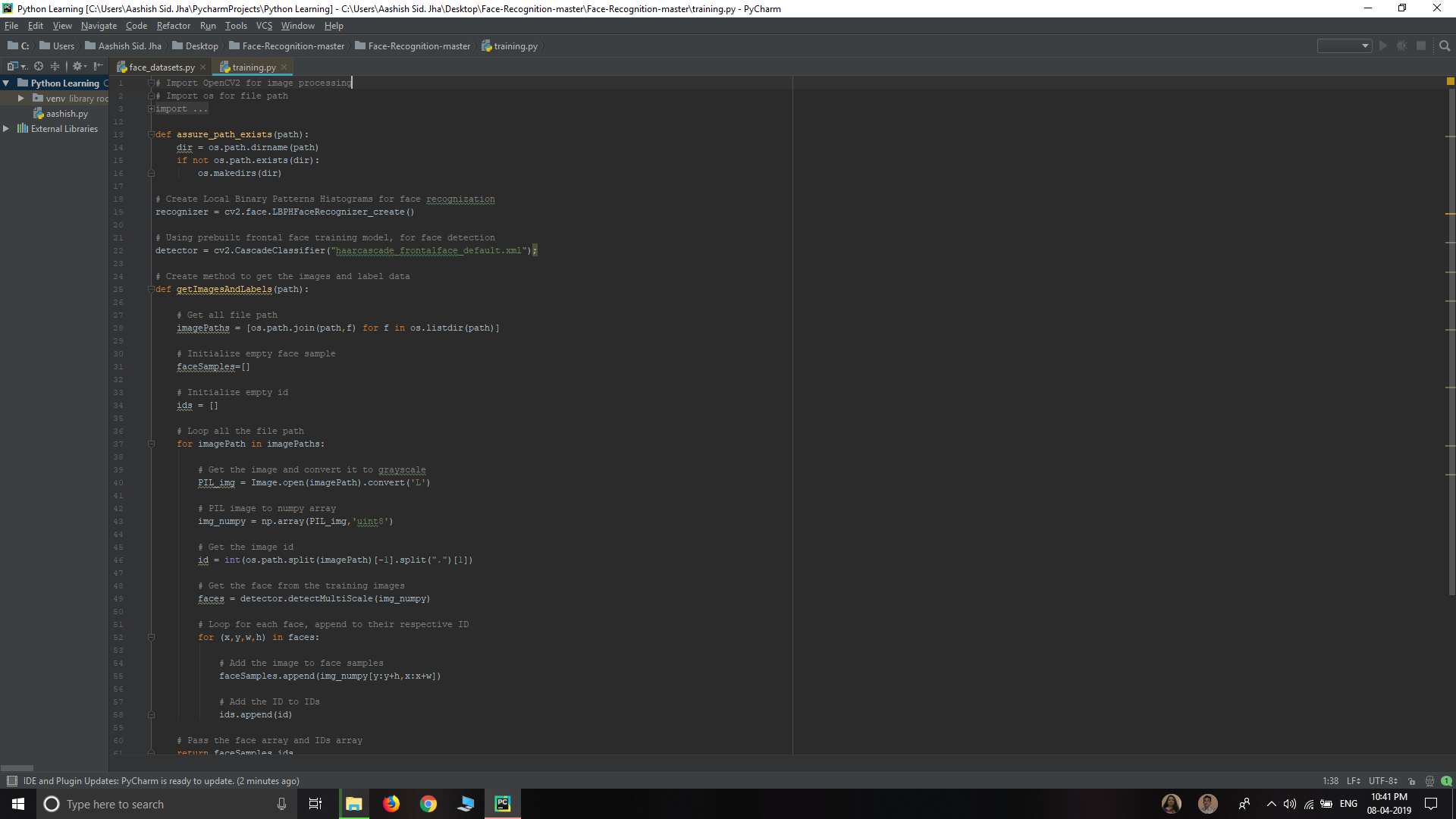
* In the following command:
* Run File “ face\_datasets.py ” the camera wil pop-up and face will be detected with rectngle around it . It wil take 101 photographs of the face so be patient .
* Check the dataset folder to check the quality of the dataset. If you are not satisfied with the dataset created then , repeat the above step.
* Now, run the “training.py ” file to train the program with the created dataset , it will create a “ trainer.yml” file by the dataset.
* Now run , “face\_recognition.py “ to start recogninsing faces it will detect the face the view of the camera and show the percentage of similarity between the detected face and the face in the data set.

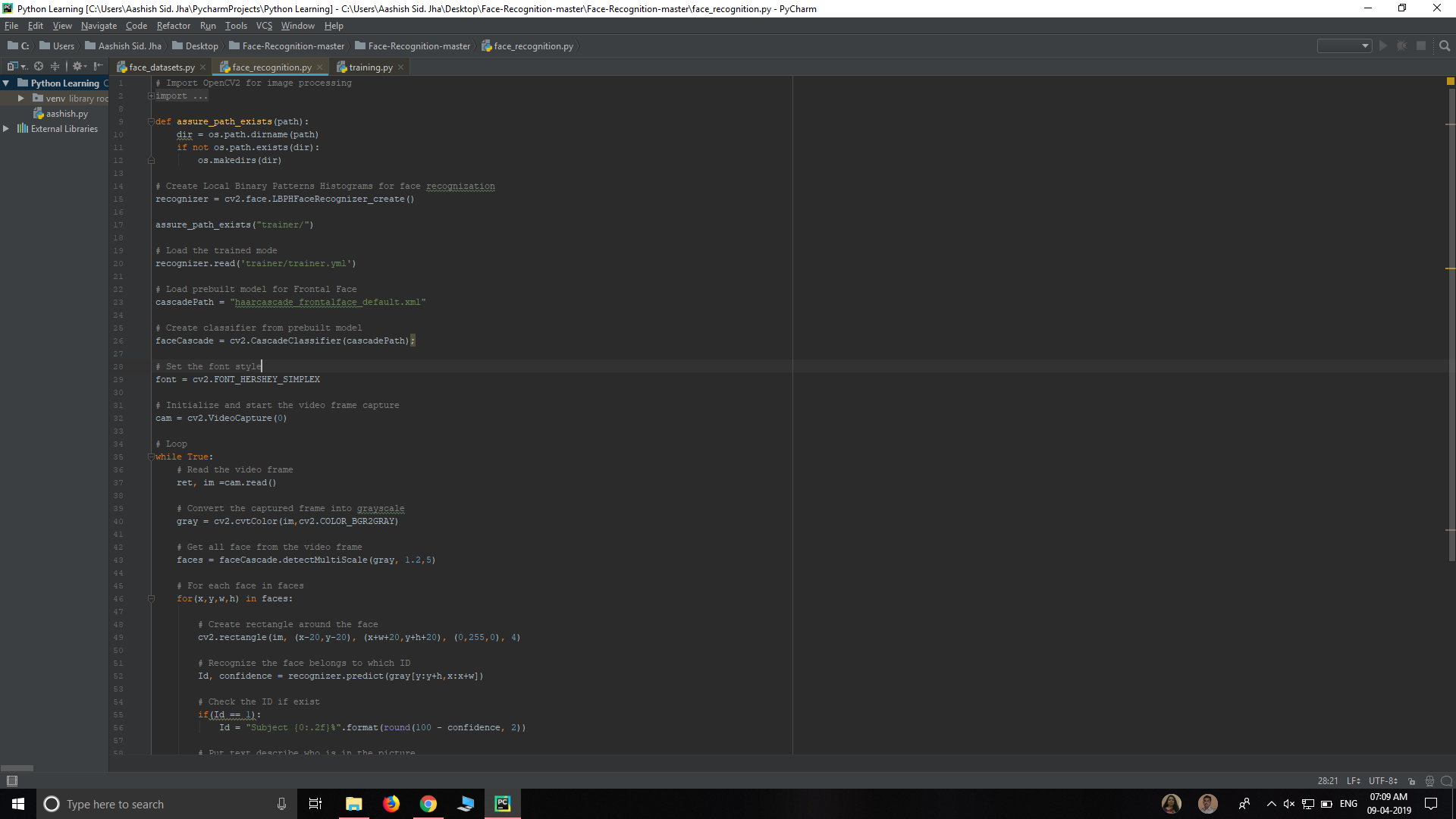
**SCREEN SHOTS**

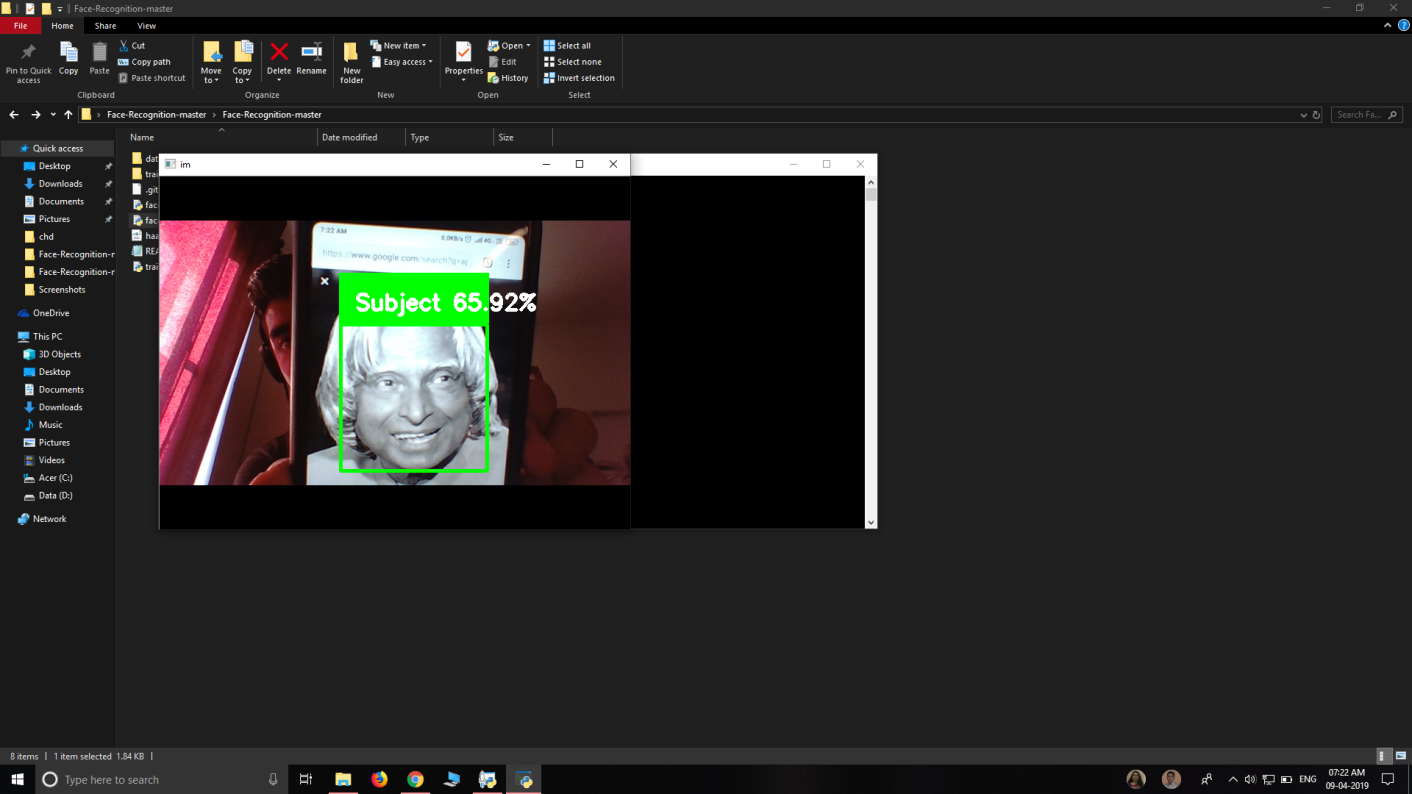


****

****

****

****

****

**CONCLUSION:-**

The Facial recognition System helps us to identify the faces in field of view of the camera and make the data set of the respective faces and train itself with the data set of the the faces in data set. By training it means making a vector of the face in data set. It makes a file named training of the format ” yml “ . By running the module of the Facial Recognition it firstly detects the face in the cameras’s view. It compares the detected face with the vectors made from the faces in data set . Roughly if the Percentage of the similarity is somewhat between the around 50’s , 40’s or 60’s it means the face of the same person of that of the data set according to the tests we have ran on it.By adding multiple data-sets uniquely by adding the ID to the respective data sets of the different faces , and adding the IF condition on the percentage of the match , we can display the face ID on the top of the face detection Square.

The facial expression recognition system presented in this research work contributes a resilient face recognition model based on the mapping of behavioural characteristics with the physiological biometric characteristics. The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise and disgust are associated with geometrical structures which restored as base matching template for the recognition system. The behavioural aspect of this system relates the attitude behind different expressions as property base. The property bases are alienated as exposed and hidden category in genetic algorithmic genes. The gene training set evaluates the expressional uniqueness of individual faces and provide a resilient expressional recognition model in the field of biometric security. The design of a novel asymmetric cryptosystem based on biometrics having features like hierarchical group security eliminates the use of passwords and smart cards as opposed to earlier cryptosystems. It requires a special hardware support like all other biometrics system. This research work promises a new direction of research in the field of asymmetric biometric cryptosystems which is highly desirable in order to get rid of passwords and smart cards completely. Experimental analysis and study show that the hierarchical security structures are effective in geometric shape identification for physiological traits.

The facial expression based face recognition system is made efficient with genetic algorithm invariants of the facial surface resulting to a recognition rate of 95.4%. The illustration of this model is given in this research work to build expressional representations using the concept of hierarchy based embedding approach. The facial representation model is deployed in laptop for biometric authentication process. The impact of the embedding space choice on the metric (distortion) concludes that spaces with spherical geometry are more favorable for representation of facial surfaces.

**FUTURE SCOPE**

The use of spherical canonical images allows us to perform matching in the spherical harmonic transform domain, which does not require preliminary alignment of the images. The errors introduced by embedding into an expressional space with some predefined geometry are avoided. In this facial expression recognition setup, end-to-end processing comprises the face surface acquisition and reconstruction, smoothening, sub sampling to approximately 2500 points. Facial surface cropping measurement of large positions of distances between all the points using a parallelized parametric version is utilized.

The general experimental evaluation of the face expressional system guarantees better face recognition rates. Having examined techniques to cope with expression variation, in future it may be investigated in more depth about the face classification problem and optimal fusion of color and depth information. Further study can be laid down in the direction of allele of gene matching to the geometric factors of the facial expressions. The genetic property evolution framework for facial expressional system can be studied to suit the requirement of different security models such as criminal detection, governmental confidential security breaches etc.

**REFERENCES**

1**.** [https://www.](https://www.python.org/)tutorialspoint.com.

2.<https://www.python.org>.

3.Automate the boring stuff with Python by AI Sweigart (Recommendation and review by Dom Watkins).

4.Learning Python by Mark Lutz and David Ascher (Recommended and review by Greg Pittman).

5. <https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>

6.Lecture Notes:

Python, computer Science Engineering, Chandigarh University.

1. <https://www.pyimagesearch.com/opencv-tutorials-resources-guides/>
2. <http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html>
3. <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>