FACE DETECTION SYSTEM

**By**

**Praveen Chauhan (16BCA013)**

**Viplavi Panchal (16BCA032)**

**Tilak Pandya (16BCA034)**

**Under Guidence**

**Of**

**Internal Guide**

**Abhilash Shukla**

**Submitted to**

****

**Smt. Chandaben Mohanbhai Patel Institute of Computer Applications**

**CHARUSAT**

**May, 2019**

****

Accredited with Grade A by NAAC,

Accredited with Grade A by KCG

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

Changa

**Acknowledgement**

Knowledge in itself is a continuous process. At this moment of our substantial enhancement, We rarely find words to express our gratitude towards those who were constantly involved with us.

The completion of any inter disciplinary project depends upon co-ordination, co-operation and combined efforts of resources of knowledge, creativity, skill, energy and time. The work being accomplished now, we feel our most sincere urge to recall and knowledge through these lines, trying best to give full credit wherever it deserves.

We would like to thank our project guide **Mr. Abhilash Shukla** and Dean & Principal **Dr. Atul Patel** who advised and gave us moral support through the duration of our project. Without their constant encouragement we could not have been able to achieve what we have.

It’s our good fortune that we had support and well wishes of many. We are thankful to all and those names which have been forgotten to acknowledgement here but contributions have not gone unnoticed.

With Sincere Regards,

**Praveen Chauhan (16BCA013)**

**Viplavi Panchal (16BCA032)**

**Tilak Pandya (16BCA034)**

**Table Context**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | | | Index | Page No |
| 1. |  |  | Project Profile | 4 |
| 2. |  |  | Introduction of tools | 6 |
| 3. |  |  | System Study | 11 |
|  | 3.1 |  | Existing System | 12 |
|  | 3.2 |  | Proposed System | 12 |
| 4. |  |  | System Analysis  Diagrams   * Use Case Diagram * Activity Diagram * Class Diagram * Sequence Diagram | 13 |
| 5. |  |  | Screen Layouts | 19 |
| 6. |  |  | System Testing | 23 |
| 7. |  |  | Future Enhancement/Drawbacks | 25 |
| 8. |  |  | Bibliography/References | 27 |
| 9. |  |  | Reporting Report (to be collected from respective internal guide) | 30 |

Project Profile

* **Project Profile**

**Project Name:** Face Detection System

**Type of Application:** System Application

**Project Description:**Our purpose to detect the images and compare the images and give the appropriate result. It can capture images and compare with other images during reorganization. User has to register for accessing the file. For registration, Permission must be given by registered users. If matching criteria is less than or equal to 50 then your face or image is definitely matched. If matching criteria is greater 50 then your matching is invalid. So user should try again. If still it can’t match your face then you have to register yourself. After recognize successfully you have 4 options on screen.

1. Hide file
2. Unhide file
3. Open Dataset
4. Open History

Ultimately we’ve just focused on prototype of face detection that can be used in different applications or different products.

**Team Size:** 3

**Front End:**  Python

**Tools used:** Pycharm IDLE

**Libraries:** Opencv, Tkinter,os, numpy, Playsound, Image, datetime, Webbrowser, random

Introduction to tools

* **Front End Tool:**
* **Python**

download

#### 1)Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

#### 2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

#### 3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

#### 4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

#### 5) Free and Open Source

Python language is freely available at address. The source-code is also available. Therefore it is open source.

#### 6) Object-Oriented Language

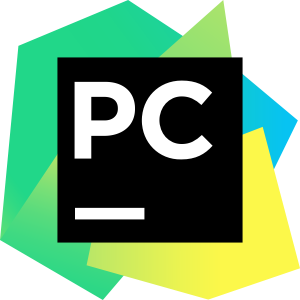
Python supports object oriented language and concepts of classes and objects come into existence.

#### 7) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

* **Tools Used**
* **Pycharm IDLE**

With PyCharm you can develop applications in Python. In addition, in the Professional edition, one can develop Django, Flask and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to **Settings | Plugins**or **PyCharm | Preferences | Plugins** for macOS users, to find out more or set them up during the first IDE launch).



PyCharm is a cross-platform IDE that works on Windows, macOS, and Linux.

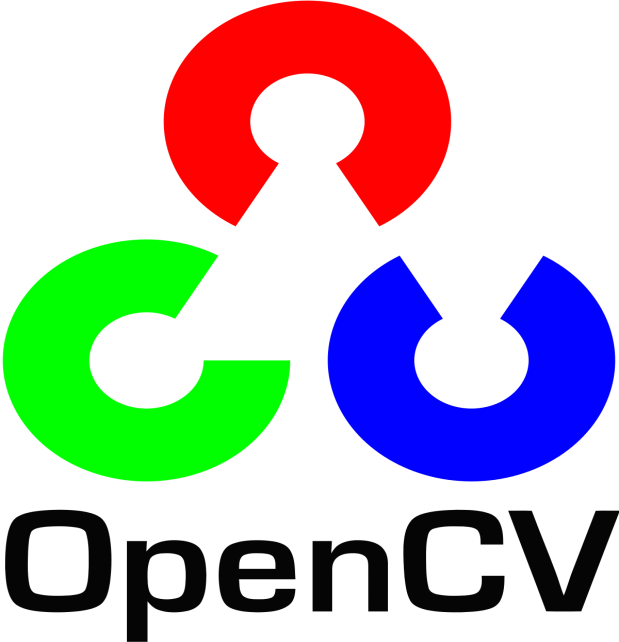
If you need assistance installing PyCharm, see the installation instructions: Requirements, Installation and Launching.

In general to start developing in Python with PyCharm you need to download, install and start PyCharm (depending on your platform).

* **Librares:**

1. **OpenCV :**

OpenCV (Open Source Computer Vision) is a library of programming functions for real-time computer vision. The face detection part of the project was made using an OpenCV Library for Scala. The reason was that most Face APIs are restricted to doing detection on pictures only, whereas the project was required to have face detection done on a live video footage to speed up the process of checking student attendance and prevent queues before lectures.



The OpenCV library proved to be flexible enough for the project as it can accurately detect a face in real time and highlight it by drawing a rectangle around the faces of the students passing by. This all happens in a window separate from the face recognition so the lecturer can keep track of both students passing by while having their faces detected and the feedback from the recognition part of the system. While faces are being detected, the application takes a snapshot of the live footage every second and then sends it to the recognition system.

1. **Tkinter:**Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete [Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl) interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter).There are several popular GUI library alternatives available, such as [wxPython](https://en.wikipedia.org/wiki/WxPython" \o "WxPython), [PyQt](https://en.wikipedia.org/wiki/PyQt" \o "PyQt) ([PySide](https://en.wikipedia.org/wiki/PySide" \o "PySide)), [Pygame](https://en.wikipedia.org/wiki/Pygame" \o "Pygame), [Pyglet](https://en.wikipedia.org/wiki/Pyglet" \o "Pyglet), and [PyGTK](https://en.wikipedia.org/wiki/PyGTK" \o "PyGTK).

System Study

* **Existing System :**

In the last twenty years, the computer-based facial recognition field has expanded rapidly. Several algorithms have been introduced and improved to the point where computers can rival humans in accuracy of facial recognition. In order to develop our product, we need to understand how we identify faces, and to understand and evaluate the different existing facial recognition algorithms and examine existing applications of this technology. Sinha, Balas, Ostrovsky, and Russell outline nineteen basic results regarding human facial recognition, including many of the methods that humans use to identify faces. &hey show that the study of human processes involved in facial recognition and the artificial algorithms being used for facial recognition systems are inextricably linked together.

* **Proposed System :**
  + - * Our purpose to detect the images and compare the images and give the appropriate result.
      * It can capture images two times and compare that images.
      * If matching criteria is greater 50 then your face or image is not matched or less than 50 then it’ll allow you to access further details. If still it can’t match your face then you have to register yourself.
      * After recognize successfully you have 4 options on screen.

1. Hide file
2. Unhide file
3. Open Dataset
4. Open History
   * + - Ultimately we’ve just created one prototype that can be used in different applications or different products.

System Analysis

**Use Case Diagram:**

**Face Detection System**

**USER**

**Activity Diagram:**

**Open User Interface**

**Access WebCam**

**Enter name**

**Capture Image**

**Recognize**

**Training Model**

**if (<50%) allowed**

**()**

**if (>50%) Not allowed**

**History**

**Open Dataset**

**Unhide File**

**Hide File**

**Class Diagram:**

**Image**

**Tempid**

**Path**

**register()**

**trained()**

**getimage()**

**recognize()**

**File**

**Hide**

**Unhide**

**hide()**

**unhide ()**

1\*

\*1

\*1

**Functionality**

**History**

**Dataset**

**Documentation**

**show ()**

**Sequence Diagram:**

:User Interface

:Image

:User

Open User Interface

Open()

Capture()

Capture()

Captured()

Captured successfully

Trained ()

Trained Successful

Trained ()

if(<50) allowed

OpenDataset()

Successfully Unhide

Successfully OpenDataset

Recognize ()

allowed

hide()

hide()

Successfully hide

Unhide()

Unhide()

OpenDataset()

History ()

History ()

Successfully History

Successfully History

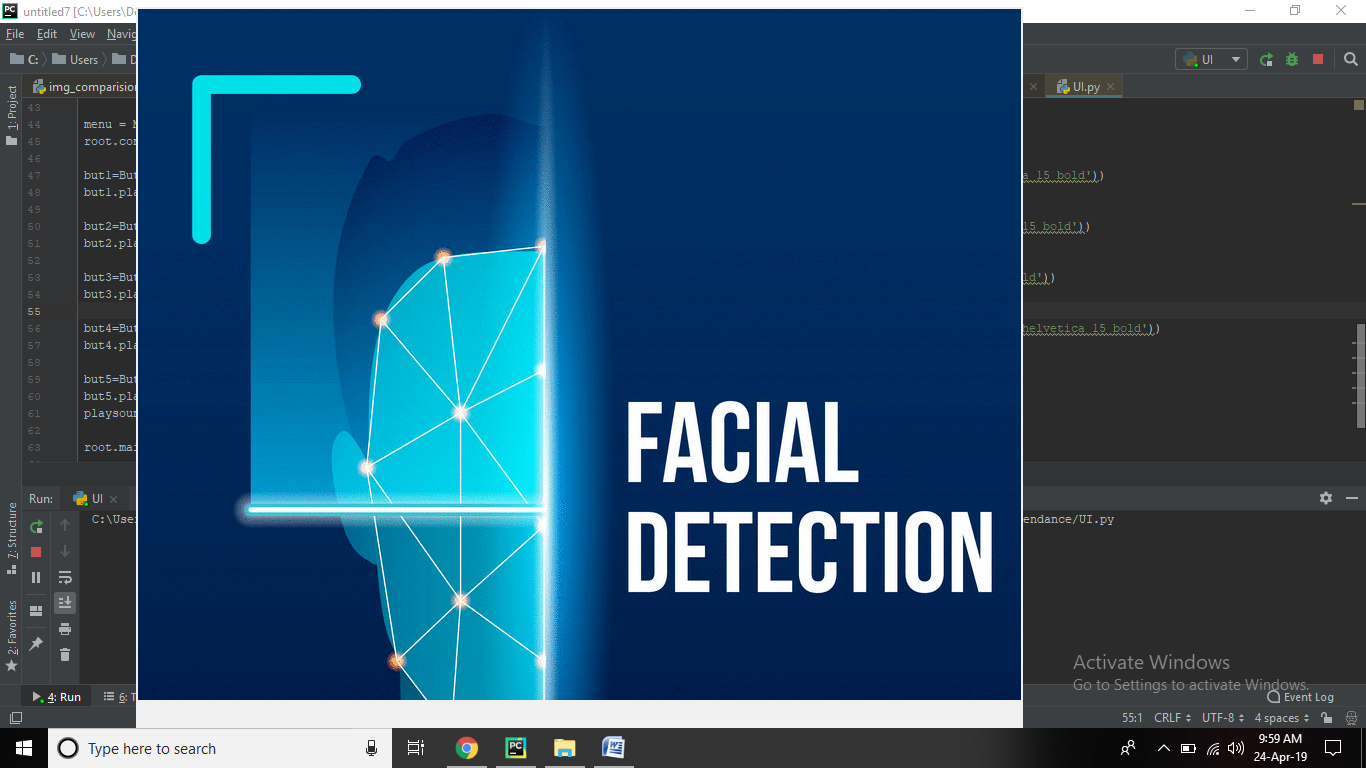
else (>50) Not allowed

Not allowed

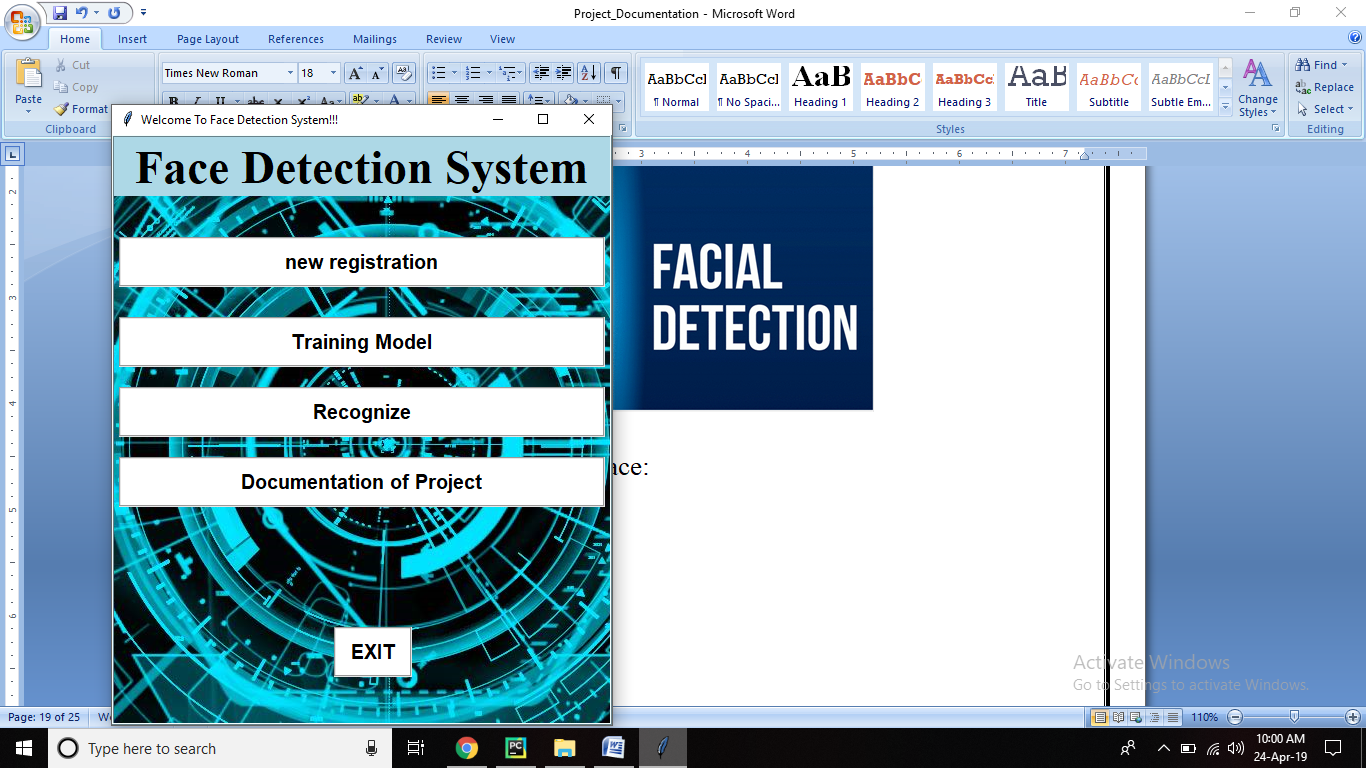
Not allowed Try again later

Screen Layouts

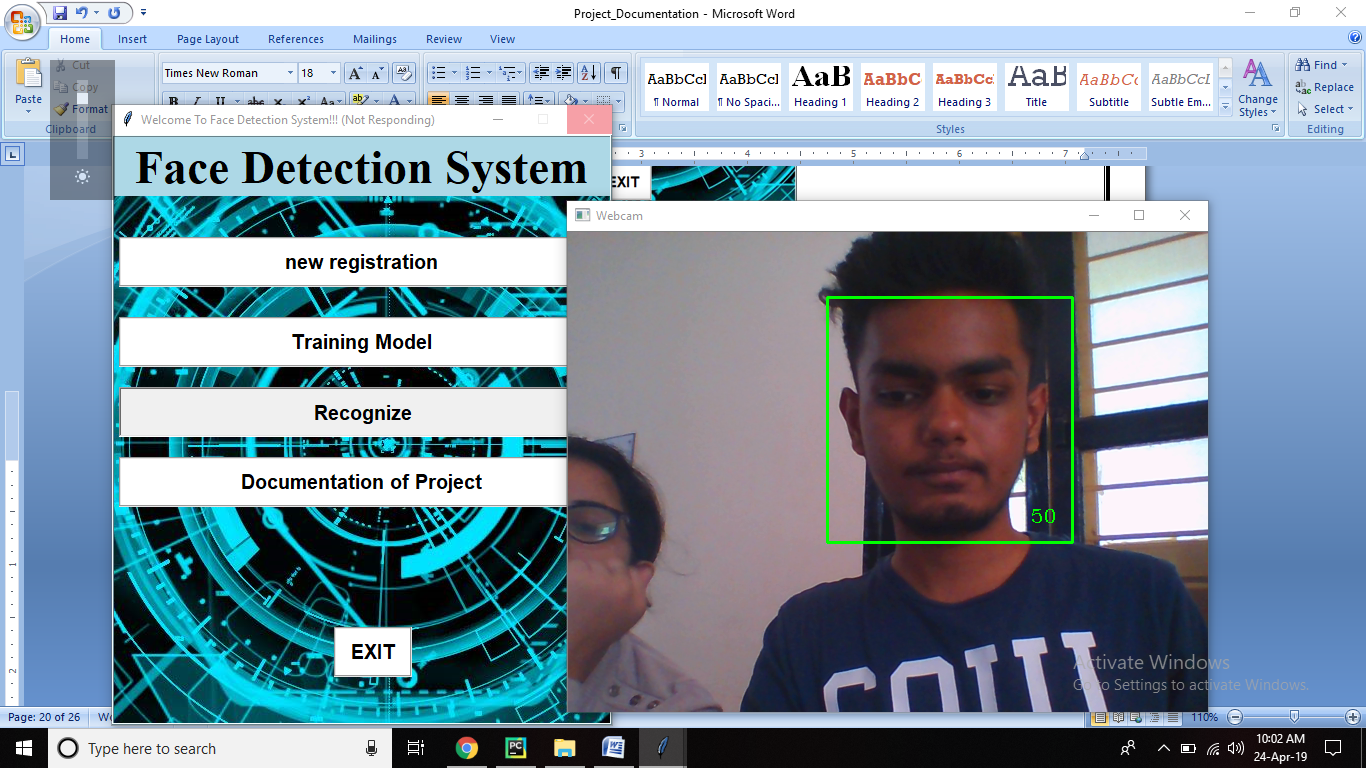
* Splash Screen:



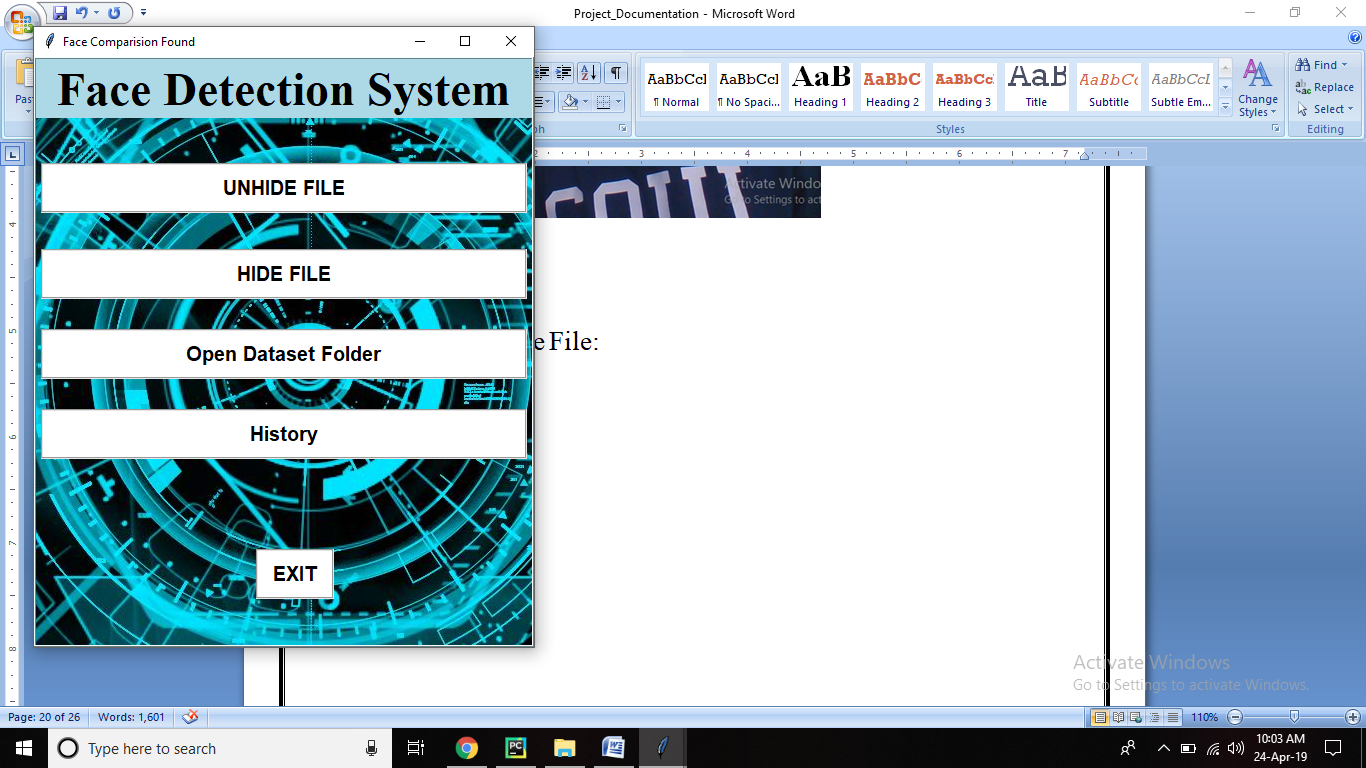
* Graphical User Interface:



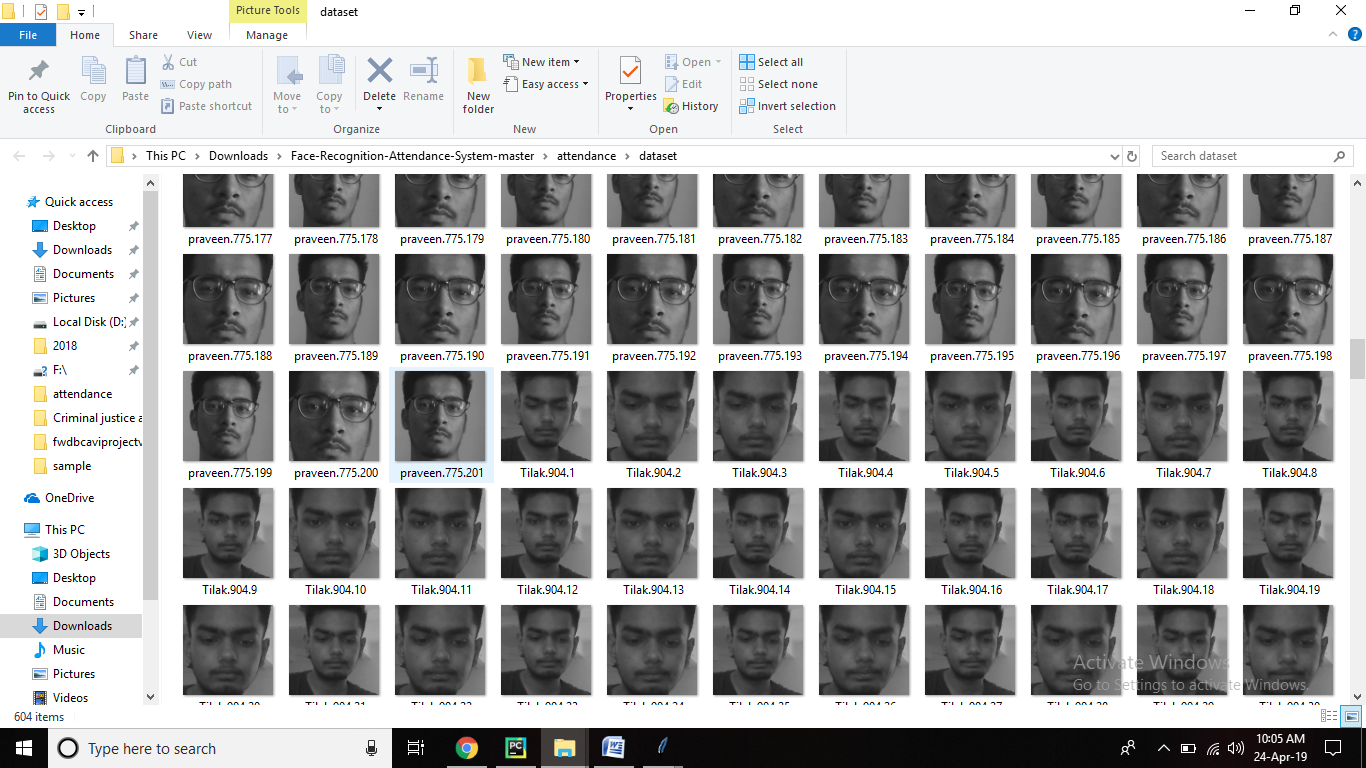
* Webcam Access:



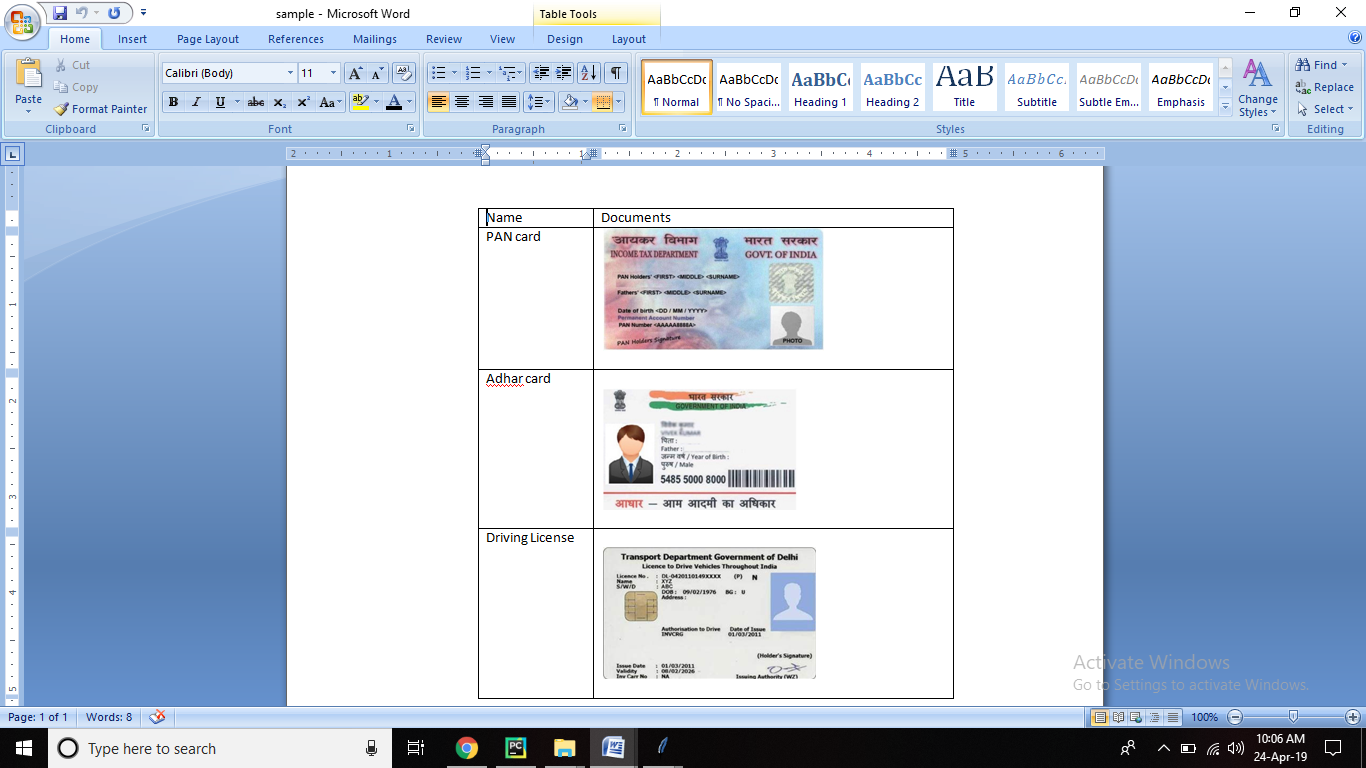
* Recognize UI:



* Dataset:



* Hide or Unhide File:



System Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Photo capture during Registration** | **Photo capture during**  **Comparision** | **Output/Result** |
| 1 | **User1.jpg** | User1.jpg | Face Recognize successfully |
| 2 | **User2.jpg** | C:\Users\Dell\Downloads\Face-Recognition-Attendance-System-master\attendance\dataset\User.1.11.jpg | Face Recognize successfully |
| 3 | **User3.jpg** | C:\Users\Dell\Downloads\Face-Recognition-Attendance-System-master\attendance\dataset\User.3.34.jpg | Face Recognize successfully |
| 4 | **User2.jpg** | User1.jpg | Face Recognize Unsuccessfully |
| 5 | **User3.jpg** | User2.jpg | Face Recognize Unsuccessfully |

Future enhancement

* Our prototype can be use in almost all Computer visionary actions or activities.
* Face Detection systems used today work very well under constrained conditions, although all systems work much better with frontal images and constant lighting.
* Next generation person Detection systems will need to recognize people in real-time and in much less constrained situations.
* Cameras and microphones today are very small, light-weight and have been successfully integrated with wearable systems. Audio and video based recognition systems have the critical advantage that they use the modalities humans use for recognition.
* Researchers are beginning to demonstrate that unobtrusive audio-and-video based person identification systems can achieve high recognition rates without requiring the user to be in highly controlled environments.
* According to Oracle's Hotel 2025 report, there is growing enthusiasm for facial recognition systems that could be used to identify and even interact with guests, with [72%](https://www.forbes.com/sites/oracle/2017/04/03/hotel-operators-guests-want-more-technology-survey/#247958012216) of hotel operators expecting to deploy such technology in the next four years.

**List of the Application that uses our Prototype:**

## PREVENT RETAIL CRIME

## UNLOCK PHONES

## SMARTER ADVERTISING

## FIND MISSING PERSONS

## HELP THE BLIND

## PROTECT LAW ENFORCEMENT

## AID FORENSIC INVESTIGATIONS

## RECOGNIZE VIPS AT SPORTING EVENTS

## TRACK SCHOOL ATTENDANCE

## FACILITATE SECURE TRANSACTIONS

## VALIDATE IDENTITY AT ATMS

## System DrAWBACKS

## Data Processing and Storage Can Limit

## Different Face Angles Can Throw Off

## Poor Image Quality Limits

## Small Image Sizes Make More Difficult

* Accuracy Issue

Bibliography/

References

* Python

<https://www.learnpython.org/>

* OpenCV

<https://docs.opencv.org/master/d9/df8/tutorial_root.html>

* Software Engineering

<https://www.wisdomjobs.com/e-university/software-engineering-tutorial-338.html>

* System Analysis & Designing

<https://www.tutorialspoint.com/system_analysis_and_design/>

Thank You