International Advance Journal of Engineering Research (IAJER)
Volume 2, Issue 12 (December- 2019), PP 01-08
ISSN: 2360-819X
www.iajer.com

Face Detection and Recognition System

Mr. Tilak Pandya[1], Mr. Praveen Chauhan[2], Ms. Viplavi Panchal[3]
[1&3] MCA Students, CMPICA, CHARUSAT, Changa
[2]BCA Student, CMPICA, CHARUSAT, Changa

ABSTRACT:- Computer vision becomes more focused in last few decades because of its precision in decision making and identification. The computation power is increasing rapidly in computer systems over a period of time which results satisfactory results against the versatile demand of applications. Artificial intelligence, machine learning, deep learning, etc buzz words have change the life of people by incorporating automation and preciseness in their routine line. In this article, we have constructed an open application to detect human face once it is registered in our system. The system is "Open" in terms as it has enough capacity to get plugged and play with any application. The application is developed using python as a language and openCV library using PyCharm tool. The application can be incorporated with any of the application to improve robustness of it.

Keywords: Face Detection & Recognition, PyCharm, OpenCV

I. INTRODUCTION:

The goal of this article is to provide an easier human-machine interaction routine when user authentication is needed through face detection and recognition. With the aid of a regular web camera, a machine is able to detect and recognize a person's face; a custom login screen with the ability to filter user access based on the users' facial features will be developed. Face detection and recognition is a major area of research within biometric signal processing. Since most techniques assume the face images normalized in terms of scale and rotation, their performance depends upon the accuracy of the detected face position within the image. This makes face detection a crucial step-in the process of face recognition. Several face detection techniques have been proposed so far, including motion detection, skin color segmentation and neural network based methods. Skin tone detection does not perform equally well on different skin colors and is sensitive to changes in illumination. In this paper we present a model-based approach that works on grayscale still images.

Facebook, Amazon, Google and other tech companies have different implementations of it. Before they can recognize a face, their software must be able to detect it first. Amazon has developed a system of real time face detection and recognition using cameras. Facebook uses it mostly on photos that their users upload in order to suggest tagging friends.

A real time face recognition algorithm based on TensorFlow, OpenCV, MTCNN and Facenet. Face reading depends on OpenCV2, embedding faces is based on Facenet, detection has done with the help of MTCNN, and recognition with classifier. The main idea was inspired by OpenFace. However, the author has preferred Python for writing code. In This project, HAAR cascades are used for face detection and Eigenface, Fisherface and LBPH are used for face recognition.

Face detection: is the process of locating a face region in image. This step does not care who the person is, just that it is a human face.

Face recognition: is the process of putting a label to a known face. Just like humans learn to recognize their family, friends and celebrities just by seeing their face, there are many techniques for a computer to learn to recognize a known face

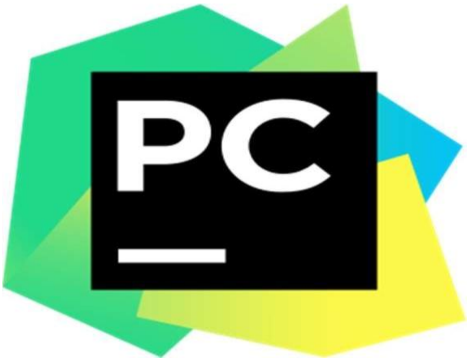
II. TECHNOLOGY USED:

The technology, which uses machine learning to detect, match and identify faces, is being used in a wide variety of ways, including entertainment and marketing. The Kinect motion gaming system, for example, uses facial recognition to differentiate among players.

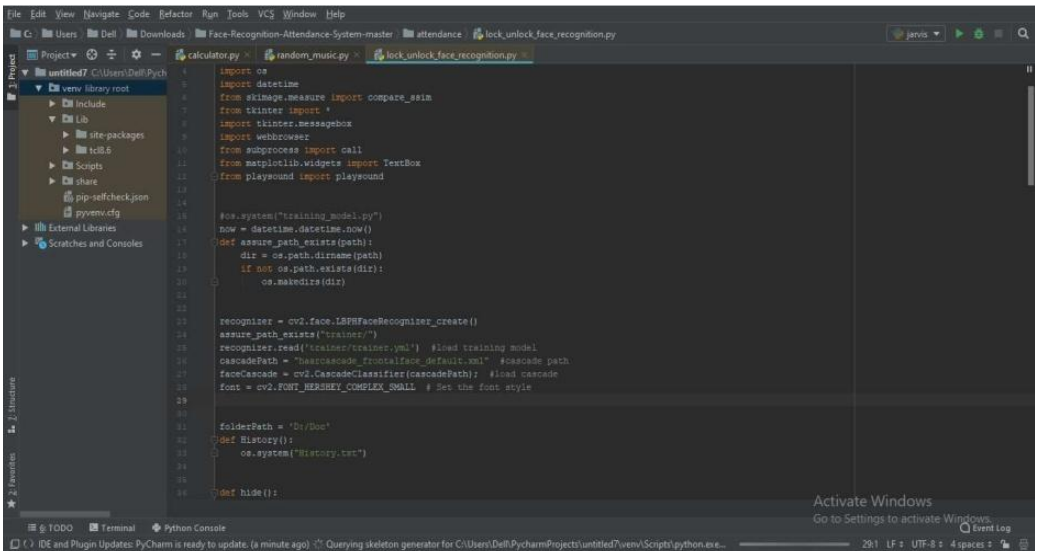
I. Software (Integrated Development Environments)

- 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).


File Edit View Navigate Code Refactor Run Tools VCS Window Help
C: › Users › Dell › Downloads › Face-Recognition-Attendance-System-master › attendance › lock_unlock_face_recognition.py
jarvis
Project
calculator.py
random_music.py
lock_unlock_face_recognition.py
untitled7 C:\Users\Dell\Pych
venv library root
Include
Lib
site-packages
tcl8.6
Scripts
share
pip-selfcheck.json
pyvenv.cfg
External Libraries
Scratches and Consoles
1: Project
2: Structure
2: Favorites
6: TODO
Terminal
Python Console
Event Log
Activate Windows
Go to Settings to activate Windows.

```
import os
import datetime
from skimage.measure import compare_ssim
from tkinter import *
import tkinter.messagebox
import webbrowser
from subprocess import call
from matplotlib.widgets import TextBox
from playsound import playsound


#os.system("training_model.py")
now = datetime.datetime.now()
def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)


recognizer = cv2.face.LBPHFaceRecognizer_create()
assure_path_exists("trainer/")
recognizer.read('trainer/trainer.yml')  #load training model
cascadePath = "haarcascade_frontalface_default.xml"  #cascade path
faceCascade = cv2.CascadeClassifier(cascadePath);  #load cascade
font = cv2.FONT_HERSHEY_COMPLEX_SMALL  # Set the font style


folderPath = 'D:/Doc'
def History():
    os.system("History.txt")


def hide():
```

IDE and Plugin Updates: PyCharm is ready to update. (a minute ago) Querying skeleton generator for C:\Users\Dell\PycharmProjects\untitled7\venv\Scripts\python.exe... 29:1 LF UTF-8 4 spaces

II. Language used:

- **Python:**

a) Easy to Learn and Use: python is easy to learn and use. It is developer-friendly and high level programming language.

b) Expressive Language: Python language is more expressive means that it is more understandable and readable.

c) 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.

d) 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.

e) Free and Open Source: Python language is freely available at address. The source-code is also available. Therefore it is open source.

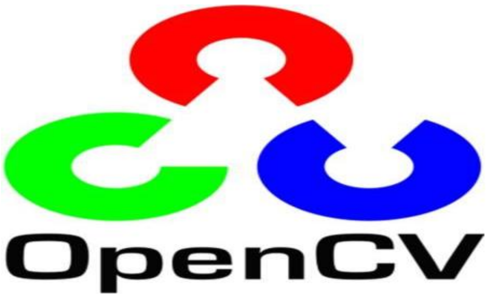
f) Object-Oriented Language: Python supports object oriented language and concepts of classes and objects come into existence.

g) Large Standard Library: Python has a large and broad library and provides rich set of module and functions for rapid application development.

III. Library/Framework:

- 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 Open CV 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.

III. PROBLEMS / DRAWBACKS:

Most face recognition algorithms are extremely sensitive to lighting conditions, so that if it was trained to recognize a person when they are in a dark room, it probably won't recognize them in a bright room, etc. This problem is referred to as "lumination dependent", and there are also many other issues, such as the face should also be in a very consistent position within the images (such as the eyes being in the same pixel coordinates), consistent size, rotation angle, hair and makeup, emotion (smiling, angry, etc), position of lights (to the left or above, etc).

IV. IMPLIMENTATION:

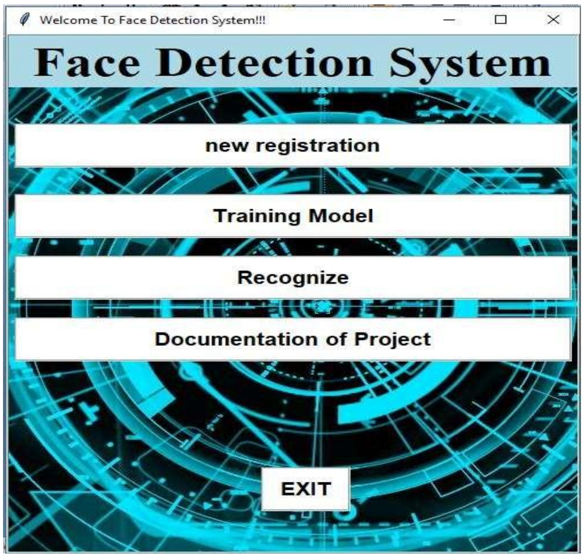
a) Splash Screen:

b) Graphical User Interface**:**

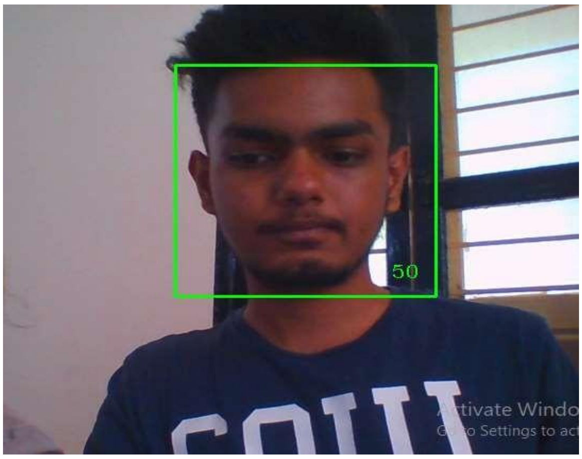
For new registration user should take permission of authenticated user.

Train model enables you to train images of new users that you want the model to recognize and to get a prediction score.

By Recognize**,** user’s web cam open and start searching for faces.



c) Webcam Access:

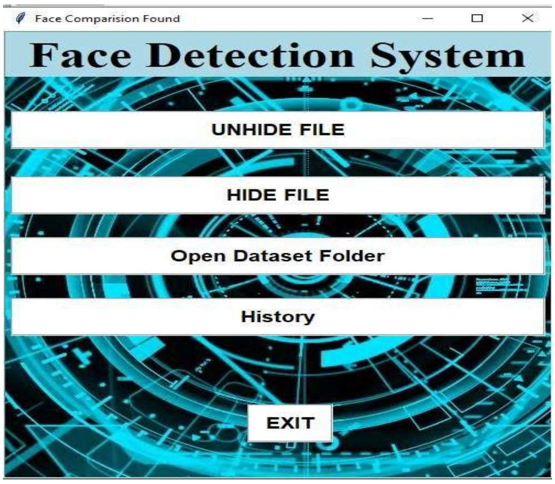


d) Recognize UI:

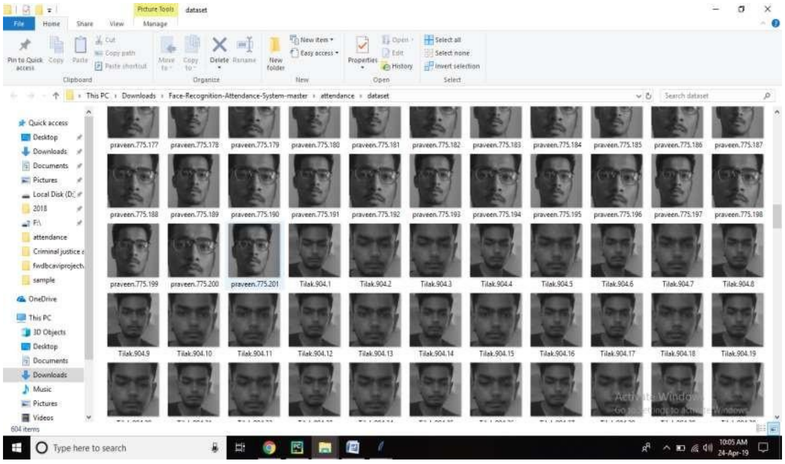
After recognized the authenticated user's face, Another UI opened. Unhide option unhide and open the file which is hidden before.

Hide option use to hide the files which is visible at any particular dictionary. Open dataset opens the folder where all images of user's faces are stored.

History is text file which is stored History of user's authentication activity.

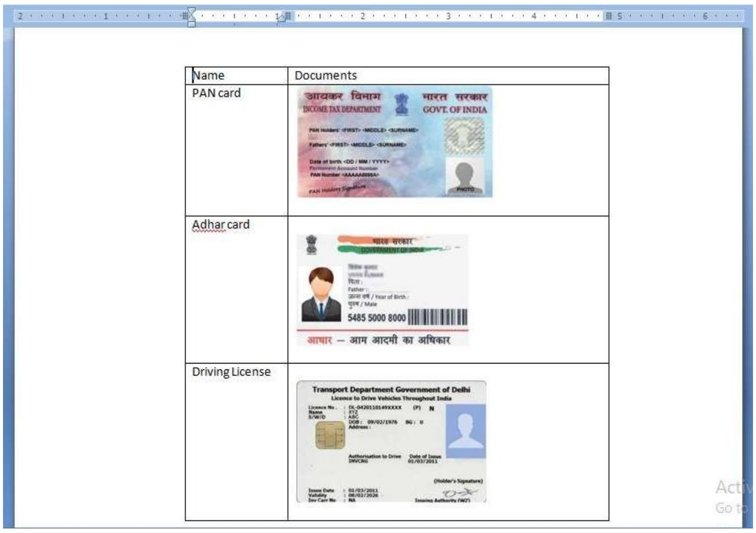


e) Dataset:

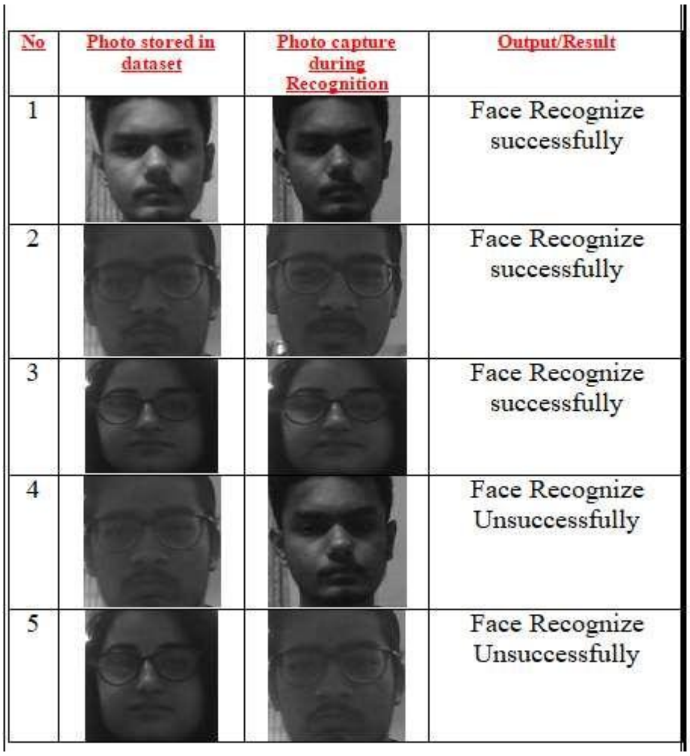


f) Hide or Unhide File:

Person can secure some private data in any file and put path of that file inside the code. It helps to hide or unhide the file.



Name	Documents
PAN card	
Adhar card	
Driving License	



No	Photo stored in dataset	Photo capture during Recognition	Output/Result
1			Face Recognize successfully
2			Face Recognize successfully
3			Face Recognize successfully
4			Face Recognize Unsuccessfully
5			Face Recognize Unsuccessfully

- Here you can see the table that describes different faces with their matches.
- Firstly images of detected faces are stored in dataset.
- The images of faces that was stored in dataset, will compare with images of faces while recognizing the authorize person's face.
- If the comparison is successfully matched then person can access specific system or else the user should try again or register him/her.

V. PROPOSER USE & APPLICATION:

Face recognition is also useful in human computer interaction, virtual reality, database recovery, multimedia, computer entertainment, information security e.g. operating system, medical records, online banking, Biometric e.g. Personal Identification - Passports, driver licenses , Automated identity verification - border controls , Law enforcement e.g. video surveillances , investigation , Personal Security - driver monitoring system, home video surveillance system. Applications & examples:

I. Face Identification: Face recognition systems identify people by their face images. Face recognition systems establish the presence of an authorized person rather than just checking whether a valid identification (ID) or key is being used or whether the user knows the secret personal identification numbers (Pins) or passwords. The following are example. To eliminate duplicates in a nationwide voter registration system because there are cases where the same person was assigned more than one identification number. The face recognition system directly compares the face images of the voters and does not use ID numbers to differentiate one from the others.

II. Access Control: In many of the access control applications, such as office access or computer logon, the size of the group of people that need to be recognized is relatively small. The face pictures are also caught under natural conditions, such as frontal faces and indoor illumination. The face recognition system of this application can achieve high accuracy without much co-operation from user. The following are the example. Face recognition technology is used to monitor continuously who is in front of a computer terminal. It allows the user to leave the terminal without closing files and logging out. When the user leaves for a predetermined time, a screen saver covers up the work and disables the mouse & keyboard. When the user comes back and is recognized, the screen saver clears and the previous session appears as it was left. Any other user who tries to logon without authorization is denied.

III. Security: Today more than ever, security is a primary concern at airports and for airline staff office and passengers. Airport protection systems that use face recognition technology have been implemented at many airports around the world. The following are the two examples. In October, 2001, Fresno Yosemite International (FYI) airport in California deployed Visage's face recognition technology for airport security purposes. The system is designed to alert FYl's airport public safety officers whenever an individual matching the appearance of a known terrorist suspect enters the airport's security checkpoint. Anyone recognized by the system would have further investigative processes by public safety officers.

IV. Image database investigations: Searching image databases of licensed drivers, benefit recipients, missing children, immigrants and police bookings.

V. General identity verification: Electoral registration, banking, electronic commerce, identifying newborns, national IDs, passports, employee IDs.

VI. Surveillance: Like security applications in public places, surveillance by face recognition systems has a low user satisfaction level, if not lower. Free lighting conditions, face orientations and other divisors all make the deployment of face recognition systems for large scale surveillance a challenging task. The following are some example of face based surveillance. To enhance town center surveillance in Newham Borough of London, this has 300 cameras linked to the closed circuit TV (CCTV) controller room. The city council claims that the technology has helped to achieve a 34% drop in crime since its facility. Similar systems are in place in Birmingham, England. In 1999 Visionics was awarded a contract from National Institute of Justice to develop smart CCTV technology.

VI. CONCLUSION:

As a concluding note, we recommend that the system is vital where implicit identity is must. User’s password or pattern can be cracked as it is explicit identity module but it is near to impossible to crack such implicit identity of the user. The system is designed and developed in such a way that it can be incorporated with any existing project or product. The detection accuracy we achieved is 80% approximately but it can be improved to 95% by using more sophisticated approached and hardware. We have used in built camera of laptop for computing the image for registration and recognition as well. More sophisticated and high resolution camera may improve the performance of the system. The system was developed as a part of our curriculum and we shall try to continue it with better precision and performance.

VII. REFERENCES:

[1]. Introduction to OpenCV: https://opencv.org/

[2]. Face Recognition Homepage: http://www.face-rec.org/algorithms/

[3]. Computer Vision Papers: http://www.cvpapers.com/

[4]. Devendra GondolePG student, Electronics and Telecommunications, Mumbai University, Mumbai, Maharashtra, India, Prof.

[5]. P. A. SalunkheProfessor, Electronics and Telecommunications, Mumbai University, Mumbai, Maharashtra, India “Face Recognition Based On Local Binary Pattern”.

[6]. Divyarajsinh N. Parmar Dr. Subhash Technical Campus Junagadh, Brijesh B. Mehta “Face Recognition Methods & Applications”.

[7]. Shervin EMAMI Senior Systems Software Engineer Mobile Computer Vision Team, NVIDIA AUSTRALIA , Valentin Petruț SUCIU 2 IT&C Security Master Department of Economic Informatics and Cybernetics Bucharest University of Economic Studies ROMANIA “Facial Recognition using OpenCV”.

[8]. Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz BioID AG, Berlin, Germany “Robust Face Detection Using the Hausdorff Distance”.

[9]. Chung-Hua Chu* and Yu-Kai Feng† “Study of Eye Blinking to Improve Face Recognition for Screen Unlock onMobile Devices”.

[10]. Sandeep Kumar, Sukhwinder Singh, Jjagdish Kumar “A Study on Face Recognition Techniques with Age and Gender Classification”.

[11]. Zafeiriou, Stefanos, Cha Zhang, and Zhengyou Zhang. "A survey on face detection in the wild: past, present and future." Computer Vision and Image Understanding, vol.138 pp.1 - 24, 2015.

[12]. “FACE DETECTION & FACE RECOGNITION USING OPEN COMPUTER VISION CLASSIFIRES” , LAHIRU DINALANKARA, August 4, 2017