VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Smart Attendance System using Face Recognition

CSE3013 - Artificial Intelligence J Component Review

Team Members

Abhinav Jaiswal	**20BCE2624**
Vanshit Kandoi	**20BCE2667**
Sneha Jayshri	**20BCE2673**

Review 3
Slot: B1+TB1

Under the guidance of:
Dr. ANNAPURNA JONNALAGADDA
Associate Professor Grade 2
School of Computer Science and Engineering
NOVEMBER 2022

DECLARATION

We hereby declare that the thesis entitled **"Team AVS"** submitted by us, for the award of the degree of B.Tech. Computer Science and Engineering, is a record of bonafide work carried out by us under the supervision of DR. ANNAPURNA JONNALAGADDA.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore **Signature of the Candidate**

Date: 10-11-22

CERTIFICATE

This is to certify that the thesis entitled **"Team AVS"** submitted by **Abhinav Jaiswal (20BCE2624), Vanshit Kandoi (20BCE2667) and Sneha Jayshri (20BCE2673)** for the award of the degree of B.Tech. Computer Science and Engineering, is a record of bonafide work carried out by them under our supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

The Project report fulfils the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of HOD**

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

It is our pleasure to express with deep sense of gratitude to DR. ANNAPURNA JONNALAGADDA., Associate Professor Grade 1, School of Computer Science and Engineering, Vellore Institute of Technology, for her constant guidance, continual encouragement, and understanding; more than all, she taught us patience in our endeavour. Our association with her is not confined to academics only, but it is a great opportunity on our part to work with an intellectual and expert in the field of Biometrics.

We would like to express our gratitude to G VISWANATHAN (Chancellor),

SEKAR VISWANATHAN (Vice President), DR. ANAND A. SAMUEL (Vice Chancellor), S NARAYANAN (Pro-Vice Chancellor), for providing an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood we express ingeniously our whole-hearted thanks to all teaching staff and members working as limbs of our university for their not-selfcentred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize our course study successfully. We would like to thank our parents for their support.

It is indeed a pleasure to thank our friends who persuaded and encouraged us to take up and complete this task. At last but not least, we express our gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date: 17-11-22 **Name of the student**

TABLE OF CONTENTS

1. Abstract

Humans can be identified by their distinct face features. A given face is compared to the faces kept in the database in order to identify the individual in the face recognition technique. The goal is to select a face in the database that is the most similar to the one being used.

Facial recognition technology is one of the fastest developing topics in biometrics. The growth in commercial interest and the development of practical technologies to enable the development of face recognition can be ascribed to the current interest in face recognition.

Biometrics, law enforcement and surveillance, human-computer interface, and multimedia management are all areas of economic interest. (For example, automatically labelling a certain person in a collection of digital images) Access control, smart cards, passport checks, criminal investigations.

The face is the most important portion of the human body, and its distinctive traits make it even more important for human identification. Various techniques and technologies are utilised across the world to improve the accuracy and reliability of face recognition.

Healthcare, security, defence, forensics, and transportation are all areas where this ever-expanding technology is being used, and greater precision is required. However, in developing facial recognition technology, some obstacles are universal, such as position, occlusion, emotions, ageing, and so on.

2. Problem statement

Attendance is an important part of daily classroom evaluation. The teacher often checks it at the start and end of class, although it's possible that they might miss someone or check some students' answers more than once. Face recognition technology based on high-definition monitor video and other information technology is used to solve the problem of recognizing faces for the purpose of taking attendance.

3. Introduction

Attendance Management keeps track of your employee or students present/absent details. It is the system to document the time your employees/students work and the time they take off.

In this digital era, face recognition system plays a vital role in almost every sector. Face recognition is one of the mostly used biometrics. It can used for security, authentication, identification.

This is an artificial intelligence based attendance management system with face recognition technology. The main objective of this AI based software solution is to update attendance with employees' face using computer vision.

This project can be divided into two main sub-systems. Attendance receiver module also called as face scanner for reading and updating attendance to the database, and management module for creating datasets of employees, training AI models with GUI, entering employees details and other management relented operations .

i. Motivation

We seek to provide a valuable attendance service for both teachers and students. Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.

ii. Significance

Facial recognition is becoming more prominent in our society. It has made major progress in the field of security. It is a very effective tool that can help low enforcers to recognize criminals and software companies are leveraging the technology to help users access the technology. This technology can be further developed to be used in other avenues such as ATMs, accessing confidential files, or other sensitive materials. This project servers as a foundation for future projects based on facial detection and recognition. This project also convers web development and database management with a user-friendly UI. Using this system any corporate offices, school and organization can replace their traditional way of maintaining attendance of the employees and can also generate their availability (presence) report throughout the month.

iii. Scope and applications

- Entering Attendances with time & Leaving Attendances with time.
- First time it will update the entering attendance.
- Second time, it will update leaving attendance.
- Do not need to touch the device, the attendance can be done with face.
- Training can be done with a single click.
- It has a dataset creation module, where system can collect images of new employee.

- An easy approach management panel.
- Unauthorized access detection and prevention.
- Voice outputs
- Eye should be blinked to confirm the identify, this prevents digital image faking attack.

iv. SWOC/SWOT analysis

- **Strength**
  a) Able to take complete attendance in one click.
  b) Effective and efficient attendance.
  c) Enhanced algorithms
  d) Time saving.
  e) Better way to maintain database.
- **Weakness**
  a) Inability to differentiate between live faces and photographic faces.
  b) Lack of enhancement in pre-existing algorithm.
  c) Model fails if it’s not able to differentiate between two persons correctly.
  d) Poor camera quality could lead to difficulties.
  e) Capturing multiple faces at a single go.
- **Opportunities**
  a) Pre-defined high quality Python Library.
  b) High configuration devices.
  c) Currently not used in great scale.
  d) Could be used in various other fields as well.
  e) More improvements can be done in near future.
- **Threats**
  a) Highly competitive environment.
  b) Presence of already implemented face recognition systems.
  c) If system is not trained well could lead to difficulties.
  d) If Live capturing is not done perfectly then it could lead to proxies.
  e) System should be trained to capture multiple faces correctly.

4. Literature Survey

a. Existing models

Various Facial Recognition models are there for:

- Anti-spoofing-based Models,
- Multi-Modal based models,
- Mobile Face Biometrics based models,
- Biometrics attendance system-based models,
- Touchless Fingerprint Recognition based on Hierarchical Clustering Models,
- Cloud Based Facial Recognition Models.

b. Gaps identified / Issues in the existing models

Various gaps identified in these models are:

•Anti-spoofing-based Models-There is lack of research as most databases are proprietary and not public.

•Multi-Modal based models-High Requirements are there, the constraints or hardware limitations of the method.

•Mobile Face Biometrics based models-Poor Image Quality Limits Facial Recognitions Effectiveness. High possibility of getting small image size making facial recognition more difficult.

•Biometrics attendance system based models-Budgetas well as Biometric features can be reconstructed from the template.Face recognition becomes more challenging in case when an image differs by surgical variations of faces for increasing beauty, aging, poses and many more.

•Touchless Fingerprint Recognition based on Hierarchical Clustering Models-Heavily relies on quality of pre-processed output image; Costlier; Needs more efficiency in the pre-processing techniques and quality-invariant features.

•Cloud Based Facial recognition Models-cloud-based APIs are highly dependent on network access with a costly subscription.

Ref No	***Paper Title***	***Journal Name and Publication***	***Advantage***	***Technique Used***	***Gaps Identified***
1	Biometric Anti Spoofing Methods: A Survey in Face Recognition	IEEE Access [Biometric Antispoofing Methods: A Survey in Face Recognition | IEEE Journals & Magazine | IEEE Xplore] [J. Galbally, S. Marcel and J. Fierrez Vol. 2, pp. 1530-1552, 2014]	Anti-spoofing countermeasures used against most modern face spoofing methods	Facial Recognition	Lack of research as most databases are proprietary and not public.
2	An Efficient Multi-Modal Biometric Verification System Using FKP and Iris	IOSR Journal of Engineering (IOSRJEN) http://www.iosrjen.org/Papers/vol8_issue4/Version-1/E0804012835.pdf [Sukhdev Singh, Chander Kant Vol. 08, Issue 4,PP 2835,2018]	It is clear from that the performance of proposed multimodal method (FKP+Iris) is better than individual biometric i.e. FKP and Iris.	FKP, Feature Extraction, Fusion, Iris, SIFT(Scale Invariant Feature Transform), SURF(Speeded Up Robust Features).	The constraints or hardware limitations of the methods and biometrics had not been mentioned.
3	Multibiometric authentication system using finger vein and iris in cloud computing	Cluster Computing https://link.springer.com/article/10.1007%2Fs10586-018-1824-9#citeas, https://dblp.org/rec/journals/cluster/IlankumaranC19.html	Finger vein, Iris, Cloud computing	Finger vein and iris authentication technology is contactless and requires no maintenanc	1. Requires at least Intel i5 core processor with 2.2 GHz, 6 GB RAM, 64 bit operating system for

		[S. Ilankumaran, C. Deisy Clust. Comput. 22 (Suppl 1): 103-117,2018]		e on a regular basis.	processing the data. 2. The postures of the fingers should not vary much.
4	Biometrics and Face Recognition Techniques	Biometrics and Face Recognition Techniques http://winteknologi.com/img/product/pdf/ede8225c99f6e1883d4ae14c66fb20191117.pdf [Renu Bhatia 2277 128X,2013]	This paper deals with the introduction of face recognition and the procedure of face recognition technique. Face Recognition is considered fast, passive and a non-intrusive system to verify and identify people. Face recognition technique is used to identify terrorists, criminals,	Biometric techniques Face Recognition techniques.	Face recognition is affected by change in lighting, the person’s hair, age, and if the people wear glasses, low resolution images. It requires camera as equipment for user identification; thus, it is doubtful to become popular.

			and other types of persons for law enforcement purposes.		
5	Techniques and Challenges of Face Recognition : A Critical Review	Techniques and Challenges of Face Recognition: A Critical Review https://www.sciencedirect.com/science/article/pii/S1877050918321252 [Shilpi Singha S.V.A.V.Prasad ICACC,2018]	This article deals with different types of face recognition. It also informs about the different challenges faced in the face recognition technique like aging factor, facial features, expressions , pose variations, thermal image, occlusion and illumination .	Face Recognition and various techniques.	Face recognition becomes more challenging in case when an image differs by surgical variations of faces for increasing beauty, aging, poses and many more. Unless and until these problems are solved face recognition can't be regarded as the most convenient biometric techniques.
6	Mobile Face Biometrics	A Survey Of Mobile Face Biometrics https://scholar.google.com/scholar?hl=en&as_sdt=0%2	1) Easy To Integrate 2) Automated	Face Recognition .	Poor Image Quality Limits Facial Recognitions

		C5&q=face+biometrics&oq=face+biomet#d=gs_qabs&u=%23p%3DC1OfUORVmycJ [Ajita Rattani, Reza Derakhshani Vol. 72,2018]	Identification		Effectiveness. High possibility of getting small image size making facial recognition more difficult.
7	Biometrics attendance system	International Conference on Complex Medical Engineering https://www.researchgate.net/publication/269962935_Biometric_Attendance_System [Imran Anwar Ujan,International islamic university malaysia and Dr. Imdad Ali Ismaili,2015]	1.Computerize the daily attendance system 2. More authentic 3. Protect the proxy which is doing daily attendance	1.C#.net set of reliable techniques for fingerprint 2.Image enhancement and minutiae extraction. 3.In combination with these development techniques, statistical experiments can then be performed on the fingerprint data set. The results from these experiments can be used to	1.Budget 2. Biometric features can be reconstructed from the template.

				help us better understand what is involved in determining the statistical uniqueness of fingerprint minutiae	
8	Touchless Fingerprint Recognition based on Hierarchical Clustering	ICMVA 2021: 2021 International Conference on Machine Vision and Applications https://dl.acm.org/doi/10.1145/3459066.3459068 [Rudresh Dwivedi, Varun Pandya, Parthiv Shah (School of technology, Pandit Deendayal Petroleum University),2021]	1. No direct contact between the finger and the sensor 2. High quality images 3. Prevents stealing of fingerprints 4. Reduced error rate and computaion time	1. Contrast Limited Adaptive Histogram Equalization (CLAHE) 2. Hierarchical Agglomerative Clustering	1. Heavily relies on quality of preprocessed output image 2. Higher costs 3. Needs more efficiency in the preprocessing techniques and quality-invariant features. 4. High possibility of getting low quality images for comparision.

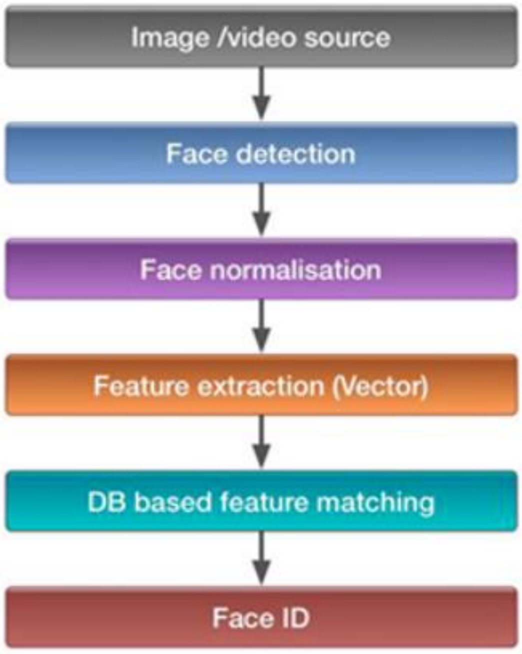
9	FACIAL RECOGNITION	Cloud Security Using Face Recognition https://www.researchgate.net/publication/319482102_Cloud_Security_Using_Face_Recognition [1. Santosh Kumar 2. Debanjan Sadhya 3. Durgesh Singh 4. Sanjay Kumar Singh ,Nov 2015]	1. Face recognition is the most common biometric and easiest to access in comparison to others like fingerprint and iris people ususally openly display their faces and thus it can be easily used as an identifier	1. Face recognition using principal component analysis 2. Enrollemnt of User to cloud server and face recognition system	1.Requires a large data set to rule out slight similarities
10	Cloud Based Facial recognition	Evaluation of Face Recognition using APIs and Libraries [Philip Masek and Magnus Thulin, June 2014]	The Prioritization of software quality characteristics when adopting face recognition software and identifying how a quality evaluation	In-depth interviews- data collection & analysis; Experiments- using OpenCV, OpenBR, Face++ and Recognition on Sample Test Data: Labelled Faces in the Wild (LFW)	It was found that reliability, functional suitability, and maintainability are the three most important quality characteristics when adopting the technology.

			can be applied. The evaluation typically assesses the accuracy of identifying and recognizing faces in different circumstances such as pose, illumination and expression.	[6]; a database consisting of face images of people published by the University of Massachusetts.	The study has found that cloud-based face-recognition software will allow organizations &developers a quick to market implementation Although cloud-based APIs are highly dependent on network access with a costly subscription, they offer high computational accuracy.

5. Implementation

i. Framework/ Architecture/ Flow chart

Proposed Model



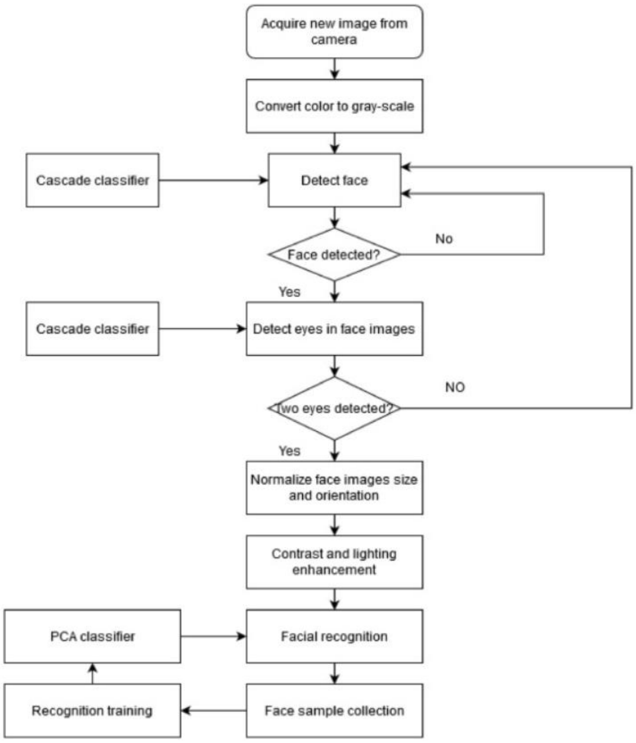
Architecture / Framework

HAAR Cascade Classifier:

Haar value calculation:

Pixel value = (Sum of the Darkpixels / Number of Darkpixels) – (Sum of the Light pixels / Number of Light pixels)

Haar Classifier is an object detection algorithm. In order to detect the object and to identify what it is; the features will be extracted from the image.



Methodology

Face detection

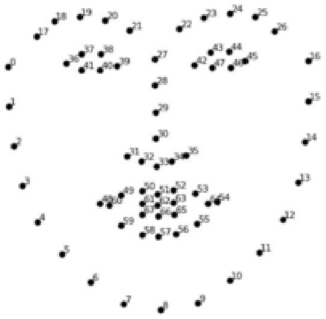
In the field of technology Face detection is treated as the demanding and practically applied approach. The identification of each face present in an image is the major task of the face detection [11, 1314]. Here the implementation is done using OpenCV.

i. Loading the input images.
ii. Converting the input images into gray scale images.
iii. Applying the Haar cascade and LBP classifier.
iv. Comparing both classifier based on the accuracy and time.
   a. Importing the required libraries
   b. Taking the images which are captured by the camera.
   c. To process the image through the classifiers it is converted into gray scale image.
   d. Image will be loaded using OpenCV
   e. By default, image will be loaded into BGR color space

DLIB:

Dlib is a versatile and well-diffused facial recognition library, with perhaps an ideal balance of resource usage, accuracy and latency, suited for real-time face recognition.

For face recognition, this algorithm uses face landmark estimation. The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face:



The face recognition algorithm in DLIB library, the neural network generates 128 measurements instead of training the network to recognize pictures.

Face detection with dlib (HOG and CNN)

We discovered dlib’s two face detection functions, one for a HOG + Linear SVM face detector and another for the MMOD CNN face detector.

From there, we configured and reviewed our project directory structure.

We then had two choices to implement two Python scripts:

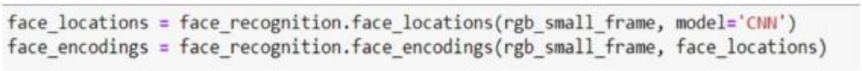
hog_face_detection.py: Applies dlib’s HOG + Linear SVM face detector.

cnn_face_detection.py: Utilizes dlib’s MMOD CNN face detector.

We can then run these face detectors on a set of images and examine the results, noting when to use each face detector in a given situation.

Also, A HOG + Linear SVM face detector that is accurate and computationally efficient.

We have used CNN in our project:



```
face_locations = face_recognition.face_locations(rgb_small_frame, model='CNN')
face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
```

Evaluation metrics

Accuracy calculation

True positive (TP): It is an actual object of interest that is correctly identified. The correctly classified faces can be calculated as [5]: True positives rate (TPR) = TP/(TP+FP)

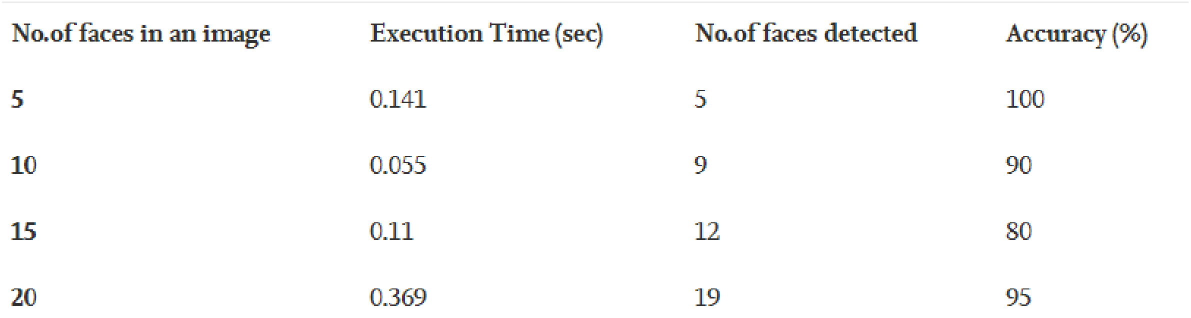
False-positives (FP): It is a non-object of interest which is falselyidentified as the true object.

False-negatives (FN): It is an actual object of interest falsely identified as negative. False negatives rate (FNR)=FN/(FN+TP)

Accuracy=(TP+TN)/(TP+TN+FP+FN)

Accuracy is obtained for the Haar cascade is 96.24% and for LBP classifier 94.74%.

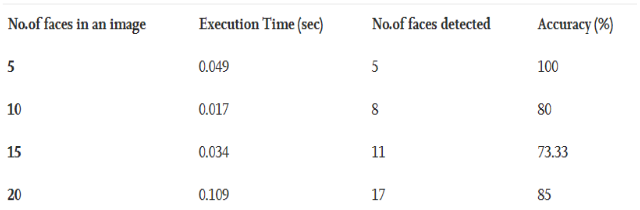
Table 1. Haar Cascade classifier.



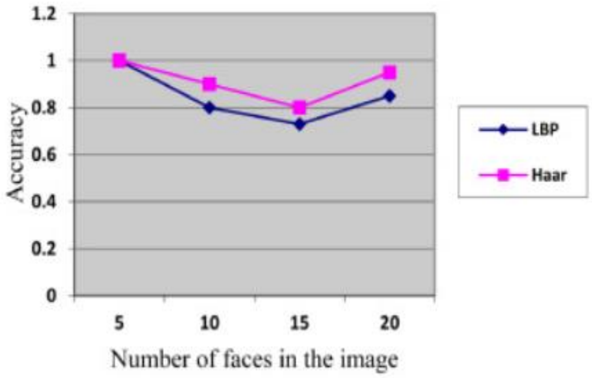
No.of faces in an image	Execution Time (sec)	No.of faces detected	Accuracy (%)
5	0.141	5	100
10	0.055	9	90
15	0.11	12	80
20	0.369	19	95

Model Not Used: **LBP Classifier**

Table 2. LBP classifier.



No.of faces in an image	Execution Time (sec)	No.of faces detected	Accuracy (%)
5	0.049	5	100
10	0.017	8	80
15	0.034	11	73.33
20	0.109	17	85



DLIB:

Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

ii. Algorithm

FACE RECOGNITION:

A prominent computer vision library is Open CV (Open Source Computer Vision). The Eigenfaces technique approaches face recognition holistically: a facial picture is a point in a high-dimensional image space, and a lower-dimensional representation is discovered, making classification simple.

HAAR CASCADING:

: It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001. Facial detection using Haar feature-based Cascade classifier is an effective object detection method.

DLIB IN FACE RECOGNITION:

It's a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face. The dlib library provides two functions for face detection. The first one is a HOG + Linear SVM face detector, and the other is a deep learning MMOD CNN face detector (image source). The dlib library provides two functions that can be used for face detection: HOG + Linear SVM: dlib.

iii. Complexity analysis

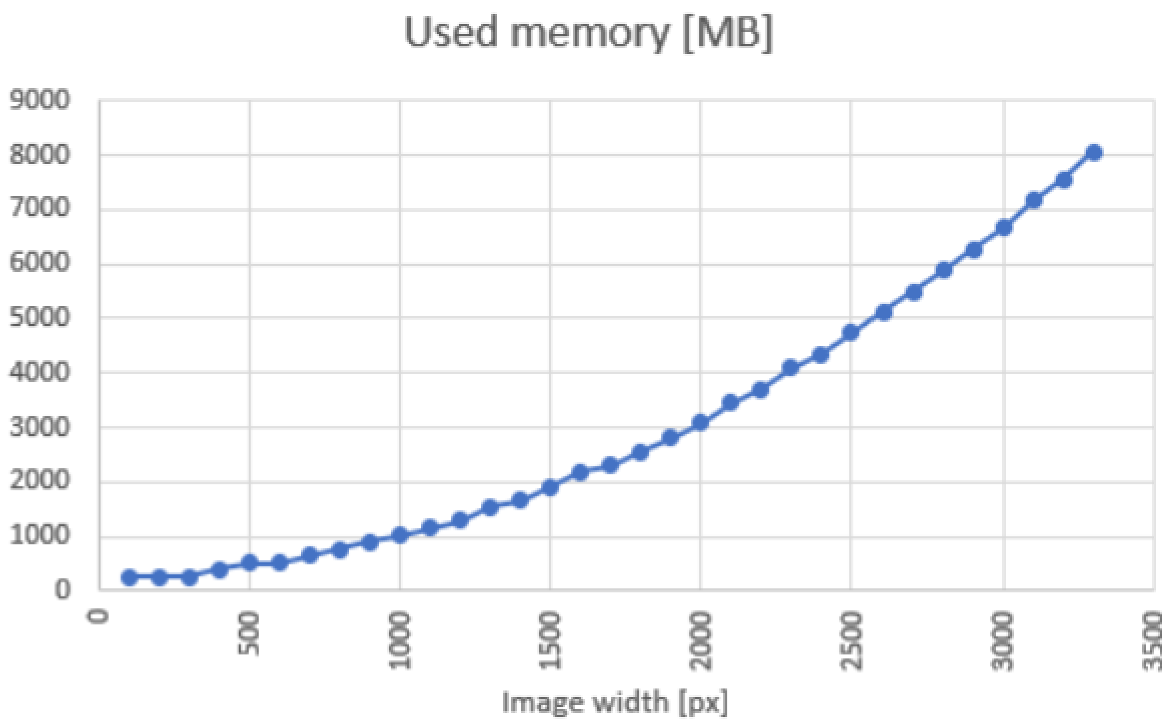
We have tested on several dimensions:

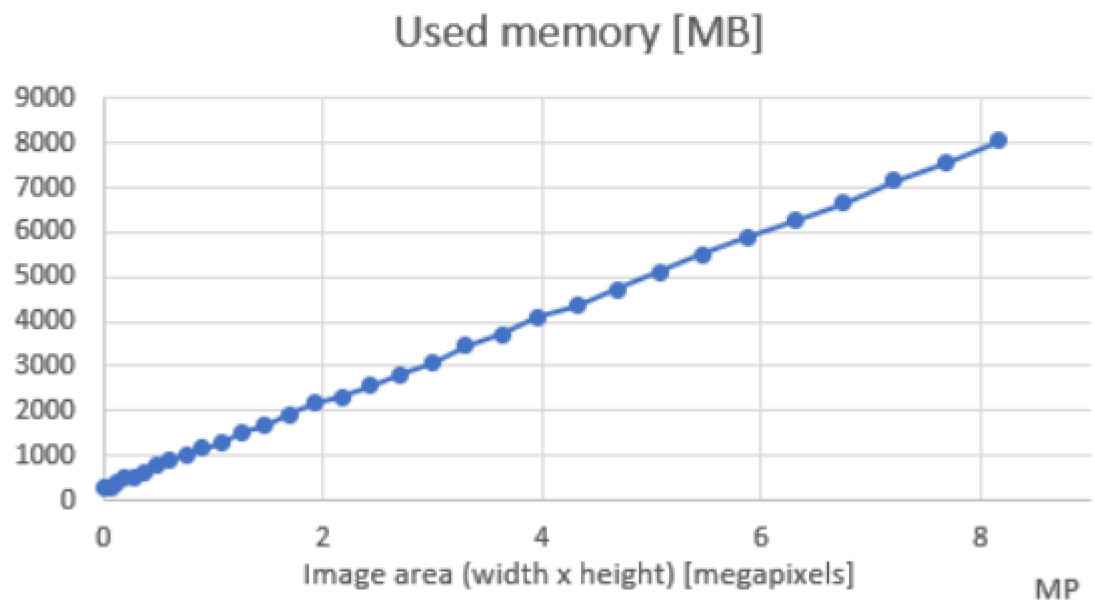
Image size (width:height was 4:3, and width scaled from 100px to 3800px in 100px increments. 3800px was maximum width my setup machine, with 12GB of RAM, could handle)

Three image types, as I wanted to see does image content make any difference for DLib:

- Completely empty (black) image.
- Image with random data (white noise).
- Actual image with lot of people on it.

With and without upsampling in DLib directly (we are going to scale down image anyway before feeding it to DLib, so this was to see what is a difference in performance and quality of detection with and without upsampling for equivalent sizes, e.g. is it better to scale down 4000×4000 to 1000×1000 and have 1 upsample in DLib, or just scale down directly to 2000×2000 with 0 upsamples in DLib).


Used memory [MB]
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
500
1000
1500
2000
2500
3000
3500
Image width [px]


Used memory [MB]
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
Image area (width x height) [megapixels]
MP

v. Discussion on implementation aspects

1. Tools/packages/languages

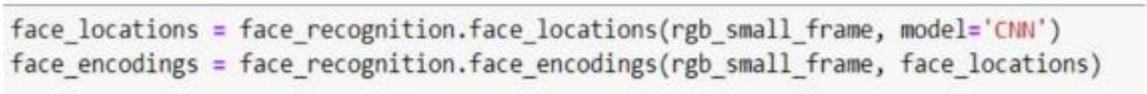
- Python 3.6 - Main programming language.
- OpenCV - for computer vision.
- Face_Recognition module.
- mysql module - for database management.

2. Implementation aspects in the literature

Literature reviews helps us in many ways, but mainly in deciding our model.

We discovered dlib’s two face detection functions, one for a HOG + Linear SVM face detector and another for the MMOD CNN face detector. From there, we configured and reviewed our project directory structure. We then had two choices to implement two Python scripts: hog_face_detection.py: Applies dlib’s HOG + Linear SVM face detector. cnn_face_detection.py: Utilizes dlib’s MMOD CNN face detector. We can then run these face detectors on a set of images and examine the results, noting when to use each face detector in a given situation. Also, A HOG + Linear SVM face detector that is accurate and computationally efficient.

We have used CNN in our project:



```
face_locations = face_recognition.face_locations(rgb_small_frame, model='CNN')
face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
```

3. Data sets

The dataset was collected from Kaggle and GitHub. After going through a lot of datasets they decided to choose Celebrity faces dataset to train our model. Since our project was based on face recognition so Celebrity faces was best suited for this purpose. We had also tested the model and made improvements in the code after implementing the different datasets. We have also done the encoding of the images. For each person we trained the model by a took a single picture.

4. Metrics for evaluation

PERFORMANCE EVALUATION OF THE SYSTEM

Performance Evaluation percentage	percentage
Students Recognition Rate (Live video)	77%
false-positive rate (Students)	28%
Unknown person Recognition Rate (existing model)	60%
Unknown person false-positive rate (exisitng model)	30%
Unknown person Recognition Rate (proposed model)	60%
Unknown person false-positive rate (proposed model)	14%

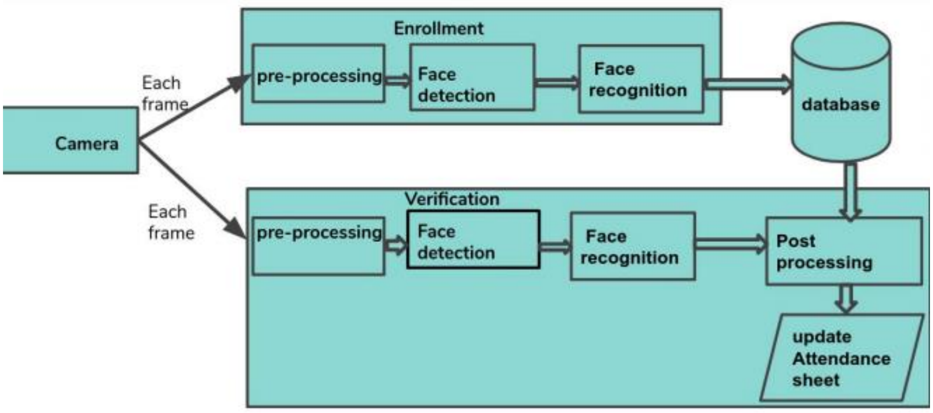
5. Summary of your observations

Face recognition rate of students is 77% and its false-positive rate is 28%. This system is recognizing students even when students are wearing glasses or grown a beard. Face Recognition of unknown persons for both existing and proposed models is 60%. This happened mostly due to detecting random objects in the background as the face of a person by face detection algorithm. Its falsepositive rate is 14% and 30% for the proposed and existing model respectively. The threshold value only affected the false positive rate of an unknown person. In the existing system, it is observed due to when the person in the video turned his head greater slightly then confidence value for that frame may get greater than favourable filter value then the person in the frame is considered as an unknown person, favourable filter value considered as 50. But, in the proposed system, if confidence is greater than 50 and 95 then only a person is considered as an unknown person and that person's image is saved as an unknown person.

Here the problem is the dataset is small. In future, an effort could be made to build a better dataset that might practically give a more accurate result. We can improve Haar cascade classifiers through the synthesis of new training examples which can improve the recognition rate of unknown persons. A system alert (voice and visual) can be included if an intruder is detected in the class.

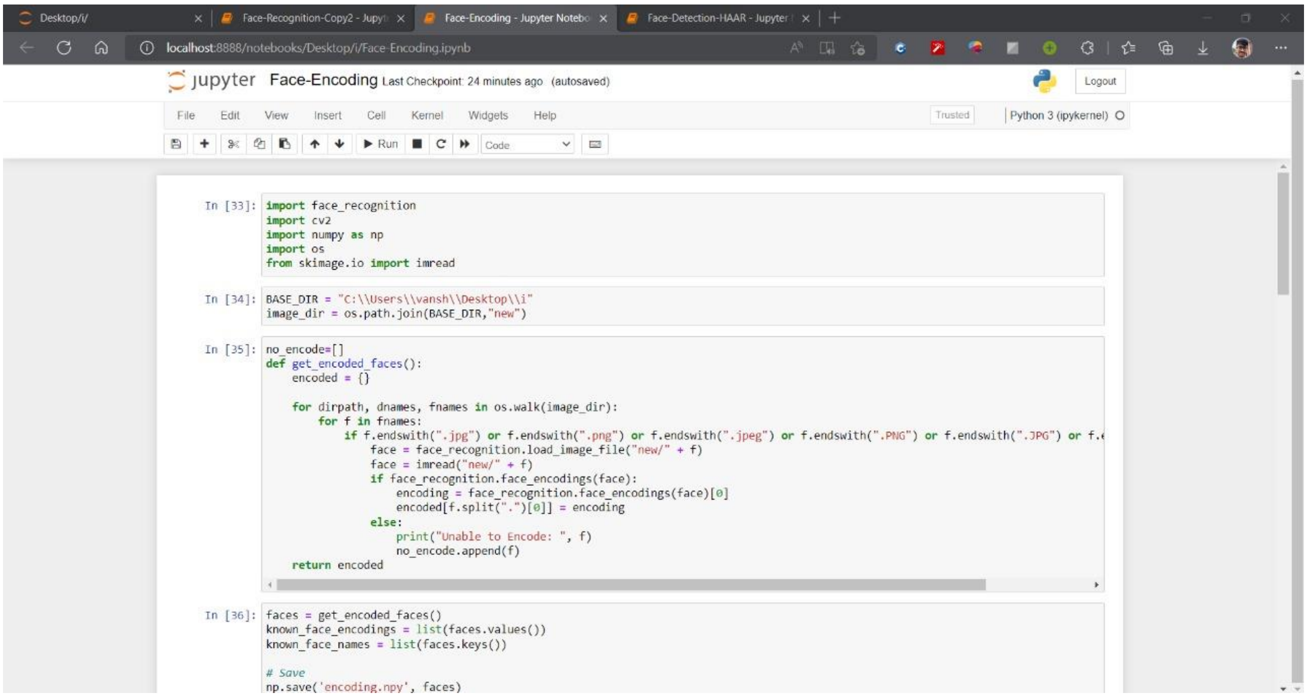
6. Result analysis (analytics and visualizations)

- This system will include a way for the lecturer or teaching assistant to track the attendance of students throughout a lecture, section, or lab. It will save time and effort, especially if there are a lot of pupils in the lecture.
- The aim of the automated attendance system is to minimize the shortcomings of the conventional (manual) approach. The application of image processing techniques in the classroom will be demonstrated via this attendance system.
- This approach will enhance an institution's reputation in addition to simply assisting with the attendance system.
- *The Proposed System Architecture.*

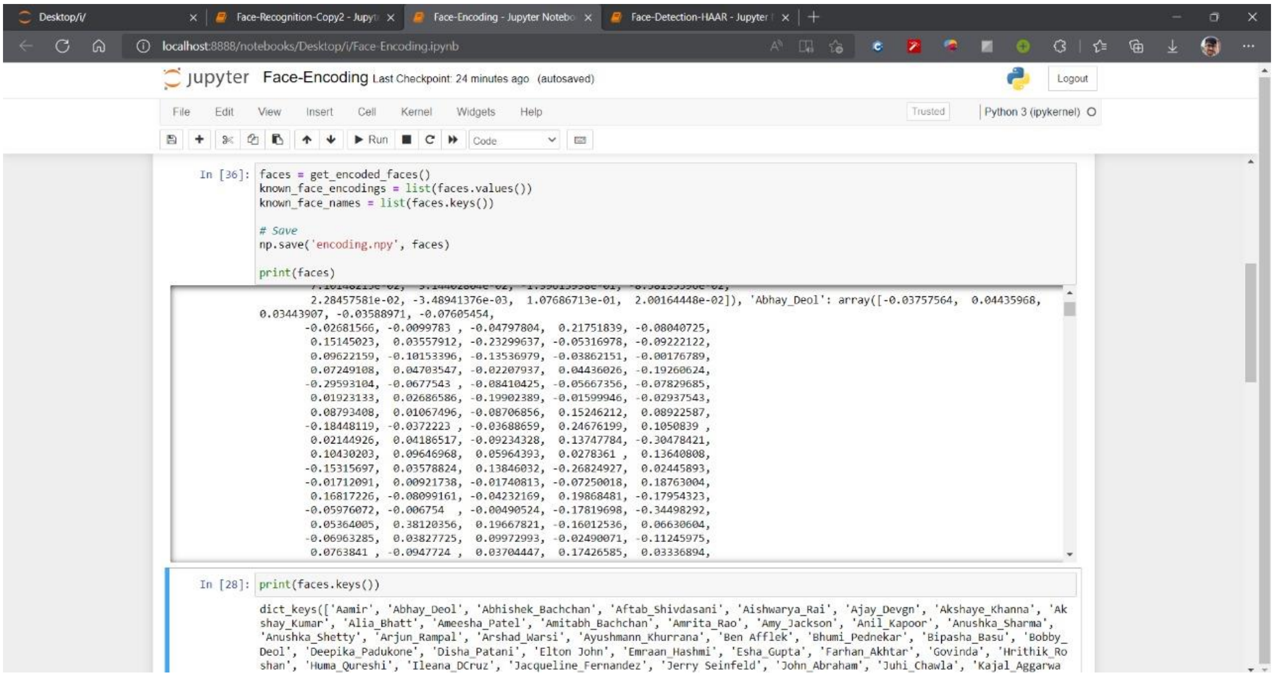


- This system provides functionalities such as taking images of students along with their details for the database, training the images in the database and on the camera and start tracking people entering the class. When students enter the classroom this system detects the faces of students who are entering the classroom from the camera and pre-processed for further processing. The implementation of each stage is mentioned in detail in the next section.

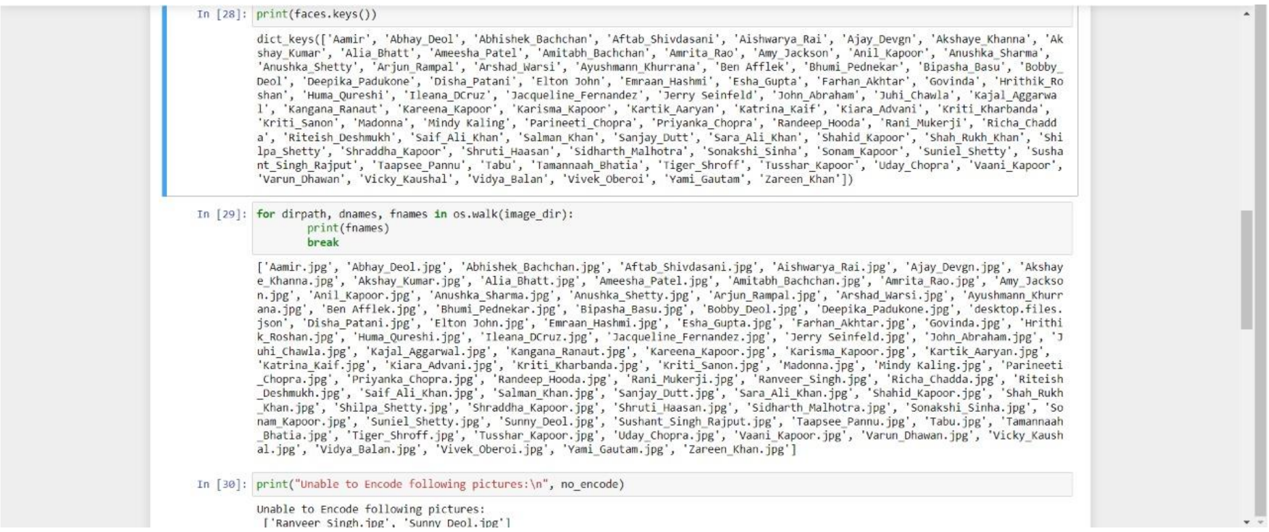
FACE ENCODING:



```
In [33]: import face_recognition
         import cv2
         import numpy as np
         import os
         from skimage.io import imread

In [34]: BASE_DIR = "C:\\Users\\vansh\\Desktop\\i"
         image_dir = os.path.join(BASE_DIR,"new")

In [35]: no_encode=[]
         def get_encoded_faces():
             encoded = {}

             for dirpath, dnames, fnames in os.walk(image_dir):
                 for f in fnames:
                     if f.endswith(".jpg") or f.endswith(".png") or f.endswith(".jpeg") or f.endswith(".PNG") or f.endswith(".JPG") or f.
                         face = face_recognition.load_image_file("new/" + f)
                         face = imread("new/" + f)
                         if face_recognition.face_encodings(face):
                             encoding = face_recognition.face_encodings(face)[0]
                             encoded[f.split(".")[0]] = encoding
                         else:
                             print("Unable to Encode: ", f)
                             no_encode.append(f)
             return encoded

In [36]: faces = get_encoded_faces()
         known_face_encodings = list(faces.values())
         known_face_names = list(faces.keys())

         # Save
         np.save('encoding.npy', faces)
```

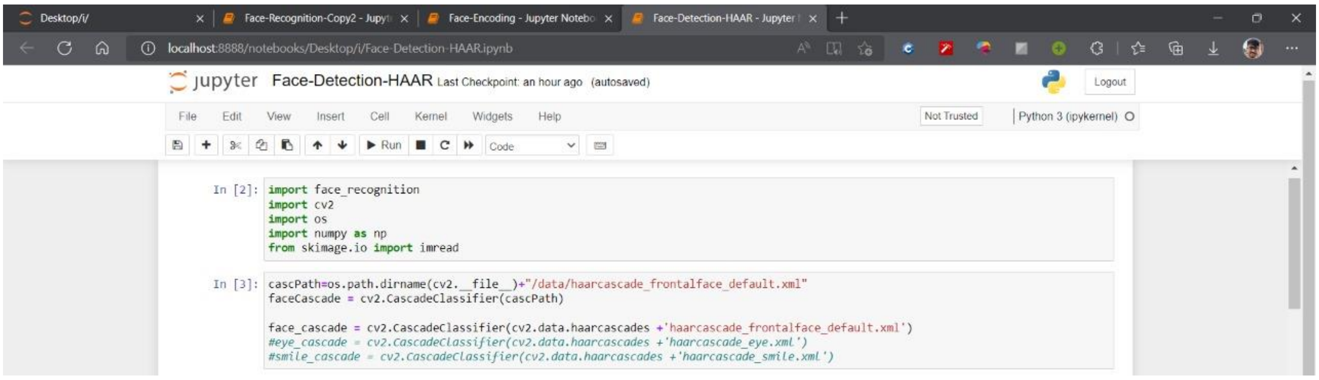


```
In [36]: faces = get_encoded_faces()
         known_face_encodings = list(faces.values())
         known_face_names = list(faces.keys())

         # Save
         np.save('encoding.npy', faces)

         print(faces)

                2.28457581e-02, -3.48941376e-03,  1.07686713e-01,  2.00164448e-02]), 'Abhay_Deol': array([-0.03757564,  0.04435968,
         0.03443907, -0.03588971, -0.07605454,
               -0.02681566, -0.0099783 , -0.04797804,  0.21751839, -0.08040725,
                0.15145023,  0.03557912, -0.23299637, -0.05316978, -0.09222122,
                0.09622159, -0.10153396, -0.13536979, -0.03862151, -0.00176789,
                0.07249108,  0.04703547, -0.02207937,  0.04436026, -0.19260624,
               -0.29593104, -0.0677543 , -0.08410425, -0.05667356, -0.07829685,
                0.01923133,  0.02686586, -0.19902389, -0.01599946, -0.02937543,
                0.08793408,  0.01067496, -0.08706856,  0.15246212,  0.08922587,
               -0.18448119, -0.0372223 , -0.03688659,  0.24676199,  0.1050839 ,
                0.02144926,  0.04186517, -0.09234328,  0.13747784, -0.30478421,
                0.10430203,  0.09646968,  0.05964393,  0.0278361 ,  0.13640808,
               -0.15315697,  0.03578824,  0.13846032, -0.26824927,  0.02445893,
               -0.01712091,  0.00921738, -0.01740813, -0.07250018,  0.18763004,
                0.16817226, -0.08099161, -0.04232169,  0.19868481, -0.17954323,
               -0.05976072, -0.006754  , -0.00490524, -0.17819698, -0.34498292,
                0.05364005,  0.38120356,  0.19667821, -0.16012536,  0.06630604,
               -0.06963285,  0.03827725,  0.09972993, -0.02490071, -0.11245975,
                0.0763841 , -0.0947724 ,  0.03704447,  0.17426585,  0.03336894,

In [28]: print(faces.keys())

         dict_keys(['Aamir', 'Abhay_Deol', 'Abhishek_Bachchan', 'Aftab_Shivdasani', 'Aishwarya_Rai', 'Ajay_Devgn', 'Akshaye_Khanna', 'Ak
         shay_Kumar', 'Alia_Bhatt', 'Ameesha_Patel', 'Amitabh_Bachchan', 'Amrita_Rao', 'Amy_Jackson', 'Anil_Kapoor', 'Anushka_Sharma',
         'Anushka_Shetty', 'Arjun_Rampal', 'Arshad_Warsi', 'Ayushmann_Khurrana', 'Ben Afflek', 'Bhumi_Pednekar', 'Bipasha_Basu', 'Bobby_
         Deol', 'Deepika_Padukone', 'Disha_Patani', 'Elton John', 'Emraan_Hashmi', 'Esha_Gupta', 'Farhan_Akhtar', 'Govinda', 'Hrithik_Ro
         shan', 'Huma_Qureshi', 'Ileana_DCruz', 'Jacqueline_Fernandez', 'Jerry Seinfeld', 'John_Abraham', 'Juhi_Chawla', 'Kajal_Aggarwa
```



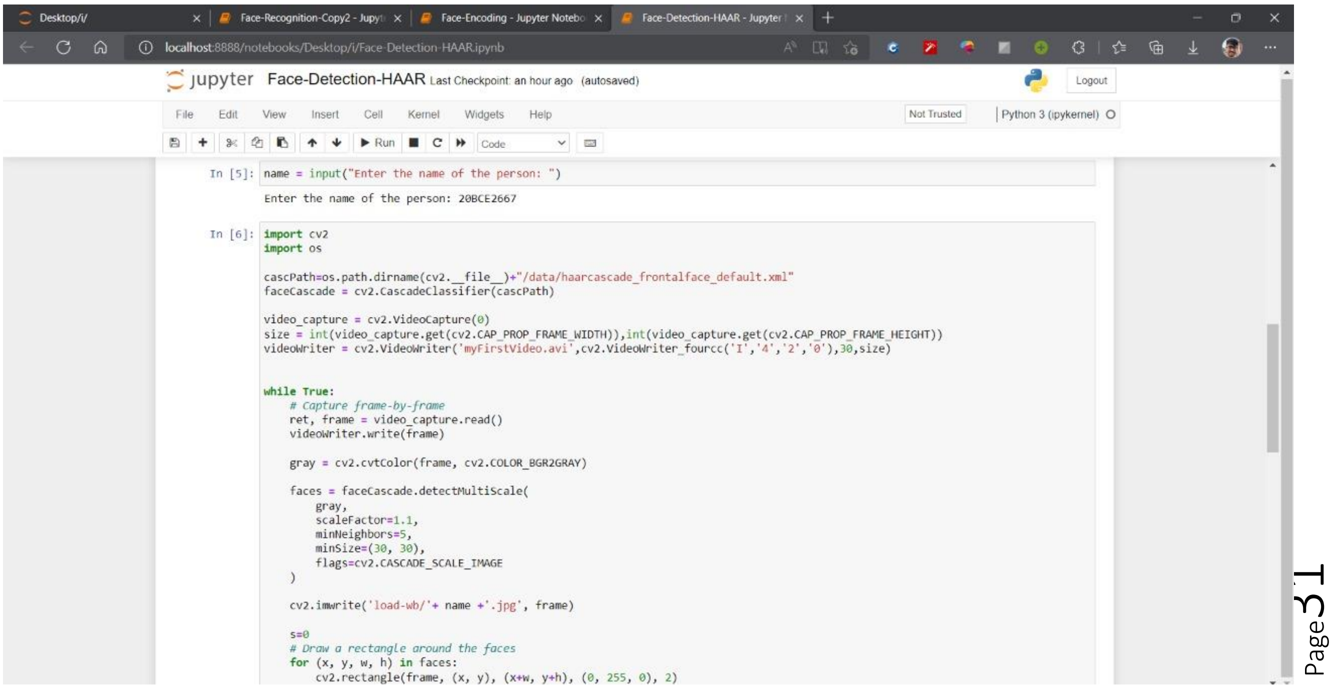
```
In [28]: print(faces.keys())

dict_keys(['Aamir', 'Abhay_Deol', 'Abhishek_Bachchan', 'Aftab_Shivdasani', 'Aishwarya_Rai', 'Ajay_Devgn', 'Akshaye_Khanna', 'Ak
shay_Kumar', 'Alia_Bhatt', 'Ameesha_Patel', 'Amitabh_Bachchan', 'Amrita_Rao', 'Amy_Jackson', 'Anil_Kapoor', 'Anushka_Sharma',
'Anushka_Shetty', 'Arjun_Rampal', 'Arshad_Warsi', 'Ayushmann_Khurrana', 'Ben Afflek', 'Bhumi_Pednekar', 'Bipasha_Basu', 'Bobby_
Deol', 'Deepika_Padukone', 'Disha_Patani', 'Elton John', 'Emraan_Hashmi', 'Esha_Gupta', 'Farhan_Akhtar', 'Govinda', 'Hrithik_Ro
shan', 'Huma_Qureshi', 'Ileana_DCruz', 'Jacqueline_Fernandez', 'Jerry Seinfeld', 'John_Abraham', 'Juhi_Chawla', 'Kajal_Aggarwa
l', 'Kangana_Ranaut', 'Kareena_Kapoor', 'Karisma_Kapoor', 'Kartik_Aaryan', 'Katrina_Kaif', 'Kiara_Advani', 'Kriti_Kharbanda',
'Kriti_Sanon', 'Madonna', 'Mindy Kaling', 'Parineeti_Chopra', 'Priyanka_Chopra', 'Randeep_Hooda', 'Rani_Mukerji', 'Richa_Chadd
a', 'Riteish_Deshmukh', 'Saif_Ali_Khan', 'Salman_Khan', 'Sanjay_Dutt', 'Sara_Ali_Khan', 'Shahid_Kapoor', 'Shah_Rukh_Khan', 'Shi
lpa_Shetty', 'Shraddha_Kapoor', 'Shruti_Haasan', 'Sidharth_Malhotra', 'Sonakshi_Sinha', 'Sonam_Kapoor', 'Suniel_Shetty', 'Susha
nt_Singh_Rajput', 'Taapsee_Pannu', 'Tabu', 'Tamannaah_Bhatia', 'Tiger_Shroff', 'Tusshar_Kapoor', 'Uday_Chopra', 'Vaani_Kapoor',
'Varun_Dhawan', 'Vicky_Kaushal', 'Vidya_Balan', 'Vivek_Oberoi', 'Yami_Gautam', 'Zareen_Khan'])

In [29]: for dirpath, dnames, fnames in os.walk(image_dir):
                 print(fnames)
                 break

['Aamir.jpg', 'Abhay_Deol.jpg', 'Abhishek_Bachchan.jpg', 'Aftab_Shivdasani.jpg', 'Aishwarya_Rai.jpg', 'Ajay_Devgn.jpg', 'Akshay
e_Khanna.jpg', 'Akshay_Kumar.jpg', 'Alia_Bhatt.jpg', 'Ameesha_Patel.jpg', 'Amitabh_Bachchan.jpg', 'Amrita_Rao.jpg', 'Amy_Jackso
n.jpg', 'Anil_Kapoor.jpg', 'Anushka_Sharma.jpg', 'Anushka_Shetty.jpg', 'Arjun_Rampal.jpg', 'Arshad_Warsi.jpg', 'Ayushmann_Khurr
ana.jpg', 'Ben Afflek.jpg', 'Bhumi_Pednekar.jpg', 'Bipasha_Basu.jpg', 'Bobby_Deol.jpg', 'Deepika_Padukone.jpg', 'desktop.files.
json', 'Disha_Patani.jpg', 'Elton John.jpg', 'Emraan_Hashmi.jpg', 'Esha_Gupta.jpg', 'Farhan_Akhtar.jpg', 'Govinda.jpg', 'Hrithi
k_Roshan.jpg', 'Huma_Qureshi.jpg', 'Ileana_DCruz.jpg', 'Jacqueline_Fernandez.jpg', 'Jerry Seinfeld.jpg', 'John_Abraham.jpg', 'J
uhi_Chawla.jpg', 'Kajal_Aggarwal.jpg', 'Kangana_Ranaut.jpg', 'Kareena_Kapoor.jpg', 'Karisma_Kapoor.jpg', 'Kartik_Aaryan.jpg',
'Katrina_Kaif.jpg', 'Kiara_Advani.jpg', 'Kriti_Kharbanda.jpg', 'Kriti_Sanon.jpg', 'Madonna.jpg', 'Mindy Kaling.jpg', 'Parineeti
_Chopra.jpg', 'Priyanka_Chopra.jpg', 'Randeep_Hooda.jpg', 'Rani_Mukerji.jpg', 'Ranveer_Singh.jpg', 'Richa_Chadda.jpg', 'Riteish
_Deshmukh.jpg', 'Saif_Ali_Khan.jpg', 'Salman_Khan.jpg', 'Sanjay_Dutt.jpg', 'Sara_Ali_Khan.jpg', 'Shahid_Kapoor.jpg', 'Shah_Rukh
_Khan.jpg', 'Shilpa_Shetty.jpg', 'Shraddha_Kapoor.jpg', 'Shruti_Haasan.jpg', 'Sidharth_Malhotra.jpg', 'Sonakshi_Sinha.jpg', 'So
nam_Kapoor.jpg', 'Suniel_Shetty.jpg', 'Sunny_Deol.jpg', 'Sushant_Singh_Rajput.jpg', 'Taapsee_Pannu.jpg', 'Tabu.jpg', 'Tamannaah
_Bhatia.jpg', 'Tiger_Shroff.jpg', 'Tusshar_Kapoor.jpg', 'Uday_Chopra.jpg', 'Vaani_Kapoor.jpg', 'Varun_Dhawan.jpg', 'Vicky_Kaush
al.jpg', 'Vidya_Balan.jpg', 'Vivek_Oberoi.jpg', 'Yami_Gautam.jpg', 'Zareen_Khan.jpg']

In [30]: print("Unable to Encode following pictures:\n", no_encode)

Unable to Encode following pictures:
 ['Ranveer_Singh.jpg', 'Sunny_Deol.jpg']
```

FACE –DETECTION-HAAR:



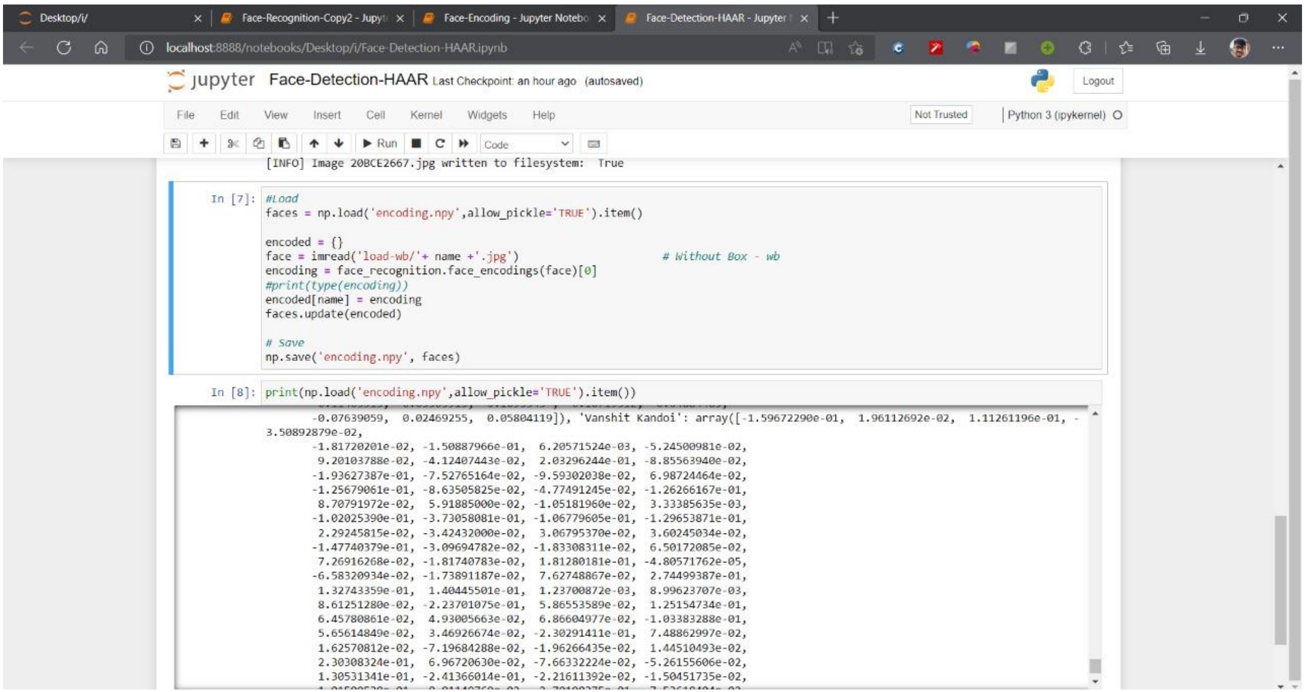
Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Detection-HAAR.ipynb
jupyter
Face-Detection-HAAR Last Checkpoint: an hour ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [2]: import face_recognition
        import cv2
        import os
        import numpy as np
        from skimage.io import imread

In [3]: cascPath=os.path.dirname(cv2.__file__)+"/data/haarcascade_frontalface_default.xml"
        faceCascade = cv2.CascadeClassifier(cascPath)

        face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +'haarcascade_frontalface_default.xml')
        #eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +'haarcascade_eye.xml')
        #smile_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +'haarcascade_smile.xml')
```



Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Detection-HAAR.ipynb
jupyter
Face-Detection-HAAR Last Checkpoint: an hour ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [5]: name = input("Enter the name of the person: ")

        Enter the name of the person: 20BCE2667

In [6]: import cv2
        import os

        cascPath=os.path.dirname(cv2.__file__)+"/data/haarcascade_frontalface_default.xml"
        faceCascade = cv2.CascadeClassifier(cascPath)

        video_capture = cv2.VideoCapture(0)
        size = int(video_capture.get(cv2.CAP_PROP_FRAME_WIDTH)),int(video_capture.get(cv2.CAP_PROP_FRAME_HEIGHT))
        videoWriter = cv2.VideoWriter('myFirstVideo.avi',cv2.VideoWriter_fourcc('I','4','2','0'),30,size)


        while True:
            # Capture frame-by-frame
            ret, frame = video_capture.read()
            videoWriter.write(frame)

            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

            faces = faceCascade.detectMultiScale(
                gray,
                scaleFactor=1.1,
                minNeighbors=5,
                minSize=(30, 30),
                flags=cv2.CASCADE_SCALE_IMAGE
            )

            cv2.imwrite('load-wb/'+ name +'.jpg', frame)

            s=0
            # Draw a rectangle around the faces
            for (x, y, w, h) in faces:
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

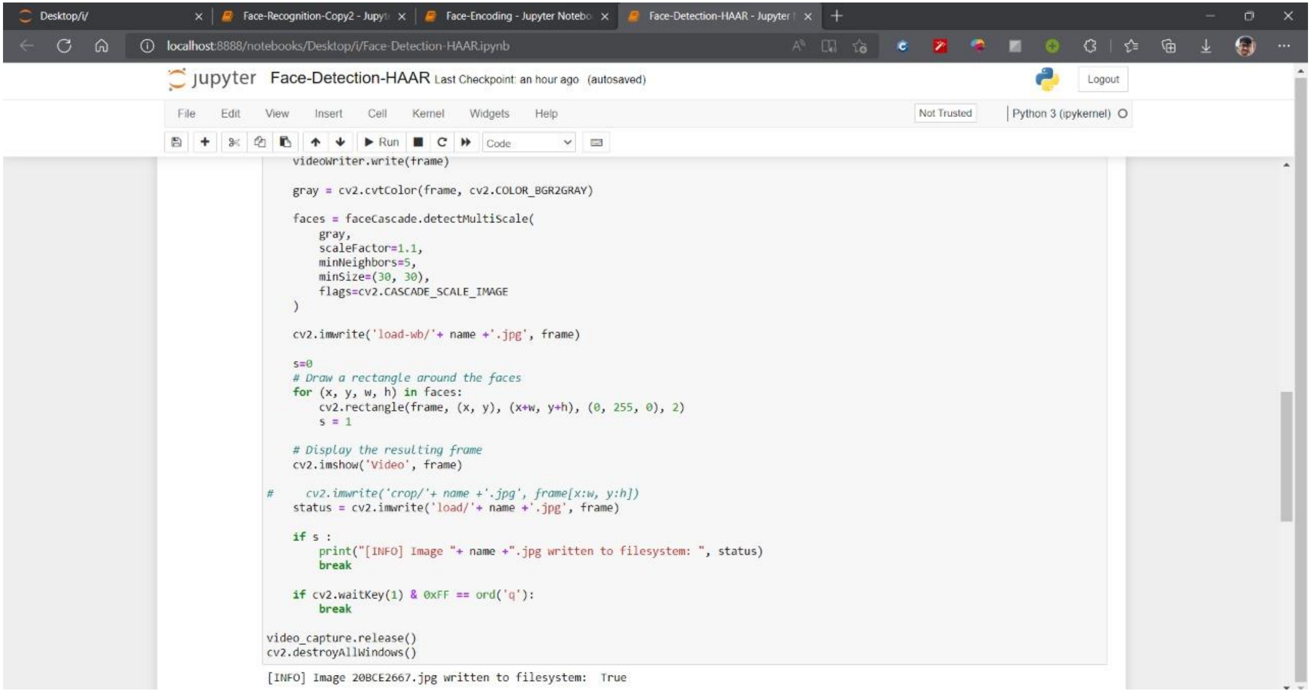

Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Detection-HAAR.ipynb
jupyter Face-Detection-HAAR Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
[INFO] Image 20BCE2667.jpg written to filesystem:  True
```

```
In [7]: #Load
        faces = np.load('encoding.npy',allow_pickle='TRUE').item()

        encoded = {}
        face = imread('load-wb/'+ name +'.jpg')                    # Without Box - wb
        encoding = face_recognition.face_encodings(face)[0]
        #print(type(encoding))
        encoded[name] = encoding
        faces.update(encoded)

        # Save
        np.save('encoding.npy', faces)
```

```
In [8]: print(np.load('encoding.npy',allow_pickle='TRUE').item())
```

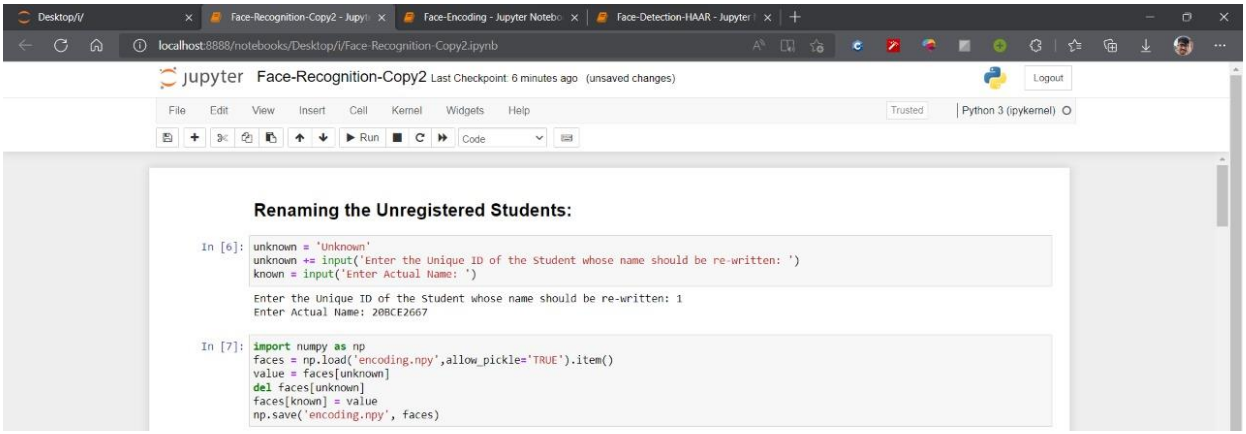
```
       -0.07639059,  0.02469255,  0.05804119]), 'Vanshit Kandoi': array([-1.59672290e-01,  1.96112692e-02,  1.11261196e-01, -
3.50892879e-02,
       -1.81720201e-02, -1.50887966e-01,  6.20571524e-03, -5.24500981e-02,
        9.20103788e-02, -4.12407443e-02,  2.03296244e-01, -8.85563940e-02,
       -1.93627387e-01, -7.52765164e-02, -9.59302038e-02,  6.98724464e-02,
       -1.25679061e-01, -8.63505825e-02, -4.77491245e-02, -1.26266167e-01,
        8.70791972e-02,  5.91885000e-02, -1.05181960e-02,  3.33385635e-03,
       -1.02025390e-01, -3.73058081e-01, -1.06779605e-01, -1.29653871e-01,
        2.29245815e-02, -3.42432000e-02,  3.06795370e-02,  3.60245034e-02,
       -1.47740379e-01, -3.09694782e-02, -1.83308311e-02,  6.50172085e-02,
        7.26916268e-02, -1.81740783e-02,  1.81280181e-01, -4.80571762e-05,
       -6.58320934e-02, -1.73891187e-02,  7.62748867e-02,  2.74499387e-01,
        1.32743359e-01,  1.40445501e-01,  1.23700872e-03,  8.99623707e-03,
        8.61251280e-02, -2.23701075e-01,  5.86553589e-02,  1.25154734e-01,
        6.45780861e-02,  4.93005663e-02,  6.86604977e-02, -1.03383288e-01,
        5.65614849e-02,  3.46926674e-02, -2.30291411e-01,  7.48862997e-02,
        1.62570812e-02, -7.19684288e-02, -1.96266435e-02,  1.44510493e-02,
        2.30308324e-01,  6.96720630e-02, -7.66332224e-02, -5.26155606e-02,
        1.30531341e-01, -2.41366014e-01, -2.21611392e-02, -1.50451735e-02,
```


Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Detection-HAAR.ipynb
jupyter Face-Detection-HAAR Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
    videoWriter.write(frame)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE
    )

    cv2.imwrite('load-wb/'+ name +'.jpg', frame)

    s=0
    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
        s = 1

    # Display the resulting frame
    cv2.imshow('Video', frame)

#     cv2.imwrite('crop/'+ name +'.jpg', frame[x:w, y:h])
    status = cv2.imwrite('load/'+ name +'.jpg', frame)

    if s :
        print("[INFO] Image "+ name +".jpg written to filesystem: ", status)
        break

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video_capture.release()
cv2.destroyAllWindows()
```

```
[INFO] Image 20BCE2667.jpg written to filesystem:  True
```

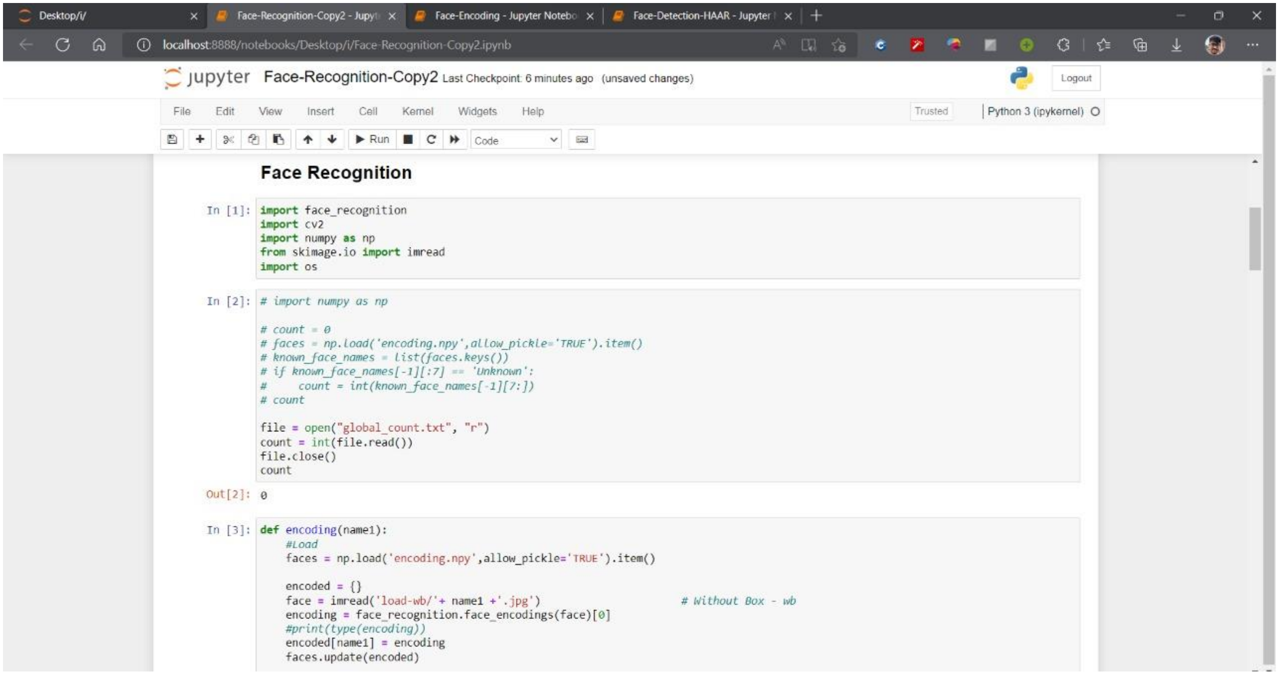
FACE RECOGNITION



Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Recognition-Copy2.ipynb
jupyter Face-Recognition-Copy2 Last Checkpoint: 6 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

Renaming the Unregistered Students:

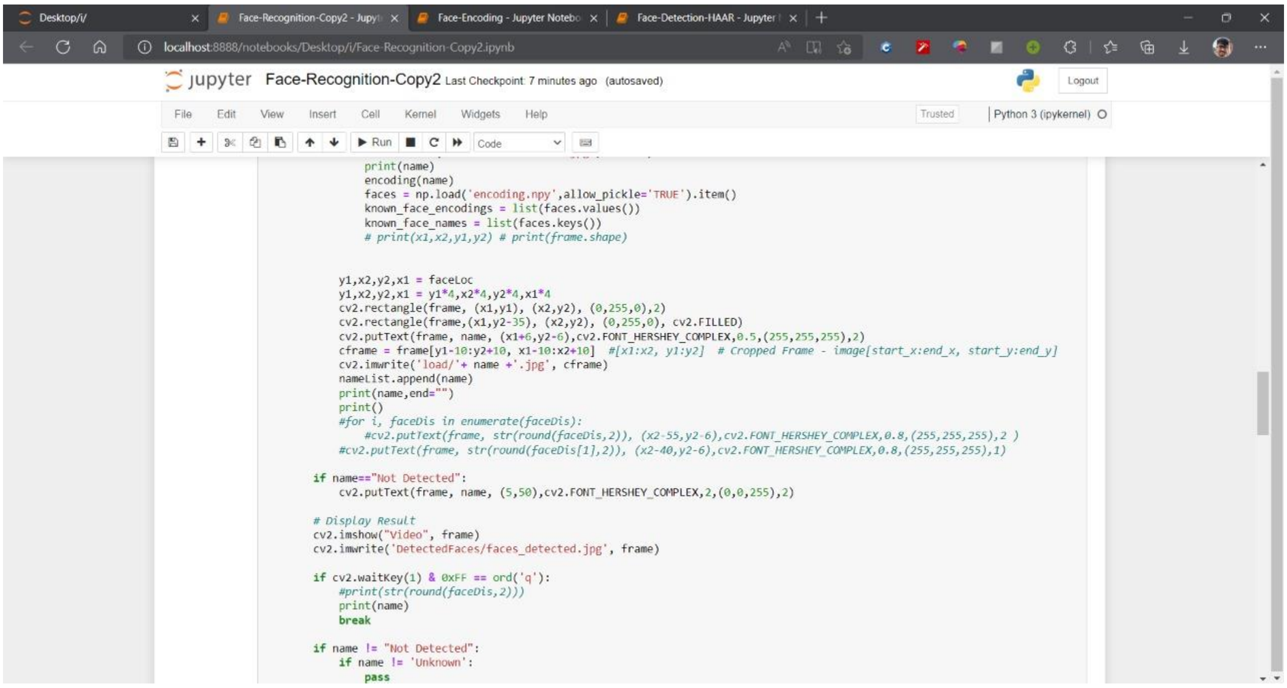
```
In [6]: unknown = 'Unknown'
        unknown += input('Enter the Unique ID of the Student whose name should be re-written: ')
        known = input('Enter Actual Name: ')

        Enter the Unique ID of the Student whose name should be re-written: 1
        Enter Actual Name: 20BCE2667

In [7]: import numpy as np
        faces = np.load('encoding.npy',allow_pickle='TRUE').item()
        value = faces[unknown]
        del faces[unknown]
        faces[known] = value
        np.save('encoding.npy', faces)
```



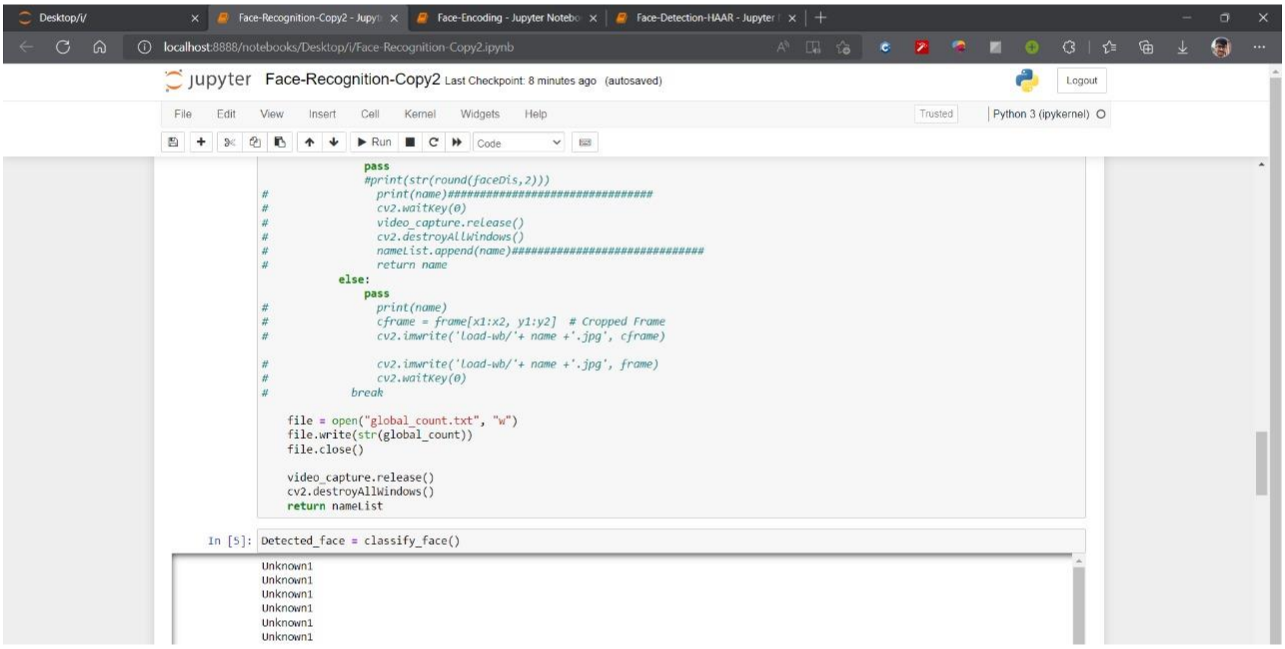
Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Recognition-Copy2.ipynb
jupyter Face-Recognition-Copy2 Last Checkpoint: 6 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

Face Recognition

```
In [1]: import face_recognition
        import cv2
        import numpy as np
        from skimage.io import imread
        import os

In [2]: # import numpy as np

        # count = 0
        # faces = np.load('encoding.npy',allow_pickle='TRUE').item()
        # known_face_names = list(faces.keys())
        # if known_face_names[-1][:7] == 'Unknown':
        #     count = int(known_face_names[-1][7:])
        # count

        file = open("global_count.txt", "r")
        count = int(file.read())
        file.close()
        count

Out[2]: 0

In [3]: def encoding(name1):
            #Load
            faces = np.load('encoding.npy',allow_pickle='TRUE').item()

            encoded = {}
            face = imread('load-wb/'+ name1 +'.jpg')                # Without Box - wb
            encoding = face_recognition.face_encodings(face)[0]
            #print(type(encoding))
            encoded[name1] = encoding
            faces.update(encoded)
```


Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Recognition-Copy2.ipynb
jupyter
Face-Recognition-Copy2 Last Checkpoint: 7 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

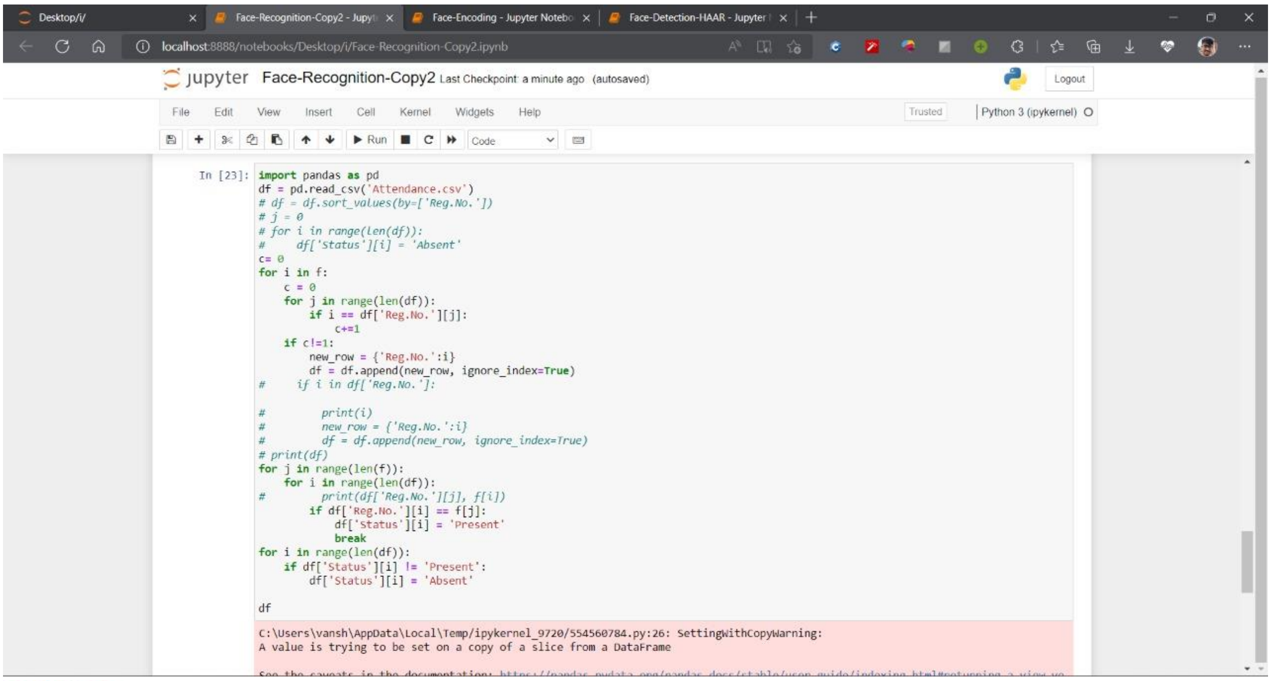
```
                print(name)
                encoding(name)
                faces = np.load('encoding.npy',allow_pickle='TRUE').item()
                known_face_encodings = list(faces.values())
                known_face_names = list(faces.keys())
                # print(x1,x2,y1,y2) # print(frame.shape)


            y1,x2,y2,x1 = faceLoc
            y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
            cv2.rectangle(frame, (x1,y1), (x2,y2), (0,255,0),2)
            cv2.rectangle(frame,(x1,y2-35), (x2,y2), (0,255,0), cv2.FILLED)
            cv2.putText(frame, name, (x1+6,y2-6),cv2.FONT_HERSHEY_COMPLEX,0.5,(255,255,255),2)
            cframe = frame[y1-10:y2+10, x1-10:x2+10]  #[x1:x2, y1:y2]  # Cropped Frame - image[start_x:end_x, start_y:end_y]
            cv2.imwrite('load/'+ name +'.jpg', cframe)
            nameList.append(name)
            print(name,end="")
            print()
            #for i, faceDis in enumerate(faceDis):
                #cv2.putText(frame, str(round(faceDis,2)), (x2-55,y2-6),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),2 )
            #cv2.putText(frame, str(round(faceDis[1],2)), (x2-40,y2-6),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),1)

        if name=="Not Detected":
            cv2.putText(frame, name, (5,50),cv2.FONT_HERSHEY_COMPLEX,2,(0,0,255),2)

        # Display Result
        cv2.imshow("Video", frame)
        cv2.imwrite('DetectedFaces/faces_detected.jpg', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            #print(str(round(faceDis,2)))
            print(name)
            break

        if name != "Not Detected":
            if name != 'Unknown':
                pass
```


Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Recognition-Copy2.ipynb
jupyter
Face-Recognition-Copy2 Last Checkpoint: 8 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
                pass
                #print(str(round(faceDis,2)))
#                 print(name)#################################
#                 cv2.waitKey(0)
#                 video_capture.release()
#                 cv2.destroyAllWindows()
#                 nameList.append(name)##############################
#                 return name
            else:
                pass
#                 print(name)
#                 cframe = frame[x1:x2, y1:y2]  # Cropped Frame
#                 cv2.imwrite('load-wb/'+ name +'.jpg', cframe)

#                 cv2.imwrite('load-wb/'+ name +'.jpg', frame)
#                 cv2.waitKey(0)
#             break

    file = open("global_count.txt", "w")
    file.write(str(global_count))
    file.close()

    video_capture.release()
    cv2.destroyAllWindows()
    return nameList
```

```
In [5]: Detected_face = classify_face()
```

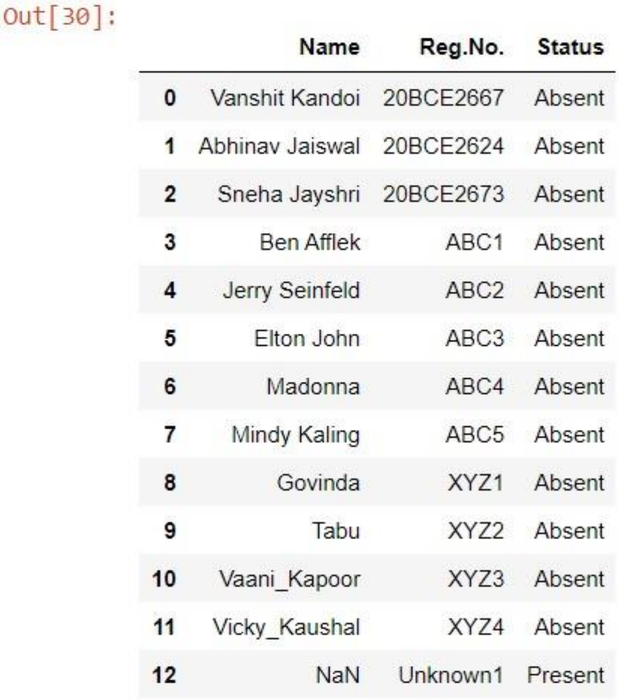
```
Unknown1
Unknown1
Unknown1
Unknown1
Unknown1
Unknown1
```



Desktop/i/
Face-Recognition-Copy2 - Jupyt
Face-Encoding - Jupyter Notebo
Face-Detection-HAAR - Jupyter
localhost:8888/notebooks/Desktop/i/Face-Recognition-Copy2.ipynb
jupyter
Face-Recognition-Copy2 Last Checkpoint: a minute ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel) ○
Run
Code

```
In [23]: import pandas as pd
         df = pd.read_csv('Attendance.csv')
         # df = df.sort_values(by=['Reg.No.'])
         # j = 0
         # for i in range(len(df)):
         #     df['Status'][i] = 'Absent'
         c= 0
         for i in f:
             c = 0
             for j in range(len(df)):
                 if i == df['Reg.No.'][j]:
                     c+=1
             if c!=1:
                 new_row = {'Reg.No.':i}
                 df = df.append(new_row, ignore_index=True)
         #     if i in df['Reg.No.']:

         #         print(i)
         #         new_row = {'Reg.No.':i}
         #         df = df.append(new_row, ignore_index=True)
         # print(df)
         for j in range(len(f)):
             for i in range(len(df)):
         #         print(df['Reg.No.'][j], f[i])
                 if df['Reg.No.'][i] == f[j]:
                     df['Status'][i] = 'Present'
                     break
         for i in range(len(df)):
             if df['Status'][i] != 'Present':
                 df['Status'][i] = 'Absent'

         df
```

```
C:\Users\vansh\AppData\Local\Temp/ipykernel_9720/554560784.py:26: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
```



Out[30]:

	Name	Reg.No.	Status
0	Vanshit Kandoi	20BCE2667	Absent
1	Abhinav Jaiswal	20BCE2624	Absent
2	Sneha Jayshri	20BCE2673	Absent
3	Ben Afflek	ABC1	Absent
4	Jerry Seinfeld	ABC2	Absent
5	Elton John	ABC3	Absent
6	Madonna	ABC4	Absent
7	Mindy Kaling	ABC5	Absent
8	Govinda	XYZ1	Absent
9	Tabu	XYZ2	Absent
10	Vaani_Kapoor	XYZ3	Absent
11	Vicky_Kaushal	XYZ4	Absent
12	NaN	Unknown1	Present

7. Conclusions and Future work

This project demonstrated a method for detecting and recognising faces which will make attendance easy. We were able to properly discover and recognise the individual using HAAR Cascading and DLIB library. Different algorithms were used to increase the accuracy of our face recognition model.

For Face Detection, HAAR Cascading Model was used and for Face Recognition, DLIB library was used. This algorithm may be employed in any secure structure; it can also be used as a biometric for a person, assisting security forces in detecting criminals and keeping society safe. Under today's constraints, face recognition systems perform admirably.

We can enhance this Smart Attendance AI system by applying nested AI model and higher flexibility applications which are beneficial for the users.

Whenever a new face which is not registered which when scanned will get an unknown id and is temporarily stored in the database. Later on, his details can be updated. This will give the authority in-charge (in this case faculty) the flexibility to later update details without missing the attendance of the newcomer (in this case student).

Program (copy the complete code to a folder and submit the same during review, include all the dependencies and also a read me file having the instructions to execute your code)

Google Account

Username: aiattendanceface@gmail.com

Password: AIattendance

Link:

https://drive.google.com/drive/folders/1uctQPJ-VpMjz9Bg-YseWbHOS6dPHKxt7?usp=sharing

8. References

http://www.ijracse.com/olvolume7issue2/Roshini-2.pdf

https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/

https://pypi.org/project/dlib/

https://www.analyticsvidhya.com/blog/2022/04/face-detection-using-the-dlib-face-detector-model/

https://www.crunchbase.com/organization/keyless-technologies-ltd/technology

https://en.wikipedia.org/wiki/Facial_recognition_system

https://www.electronicid.eu/en/blog/post/face-recognition/en

https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2

http://winteknologi.com/img/product/pdf/ede8225c99f6e1883d4ae14c66fb20191117.pdf

https://doi.org/10.1109/ICEEICT53905.2021.9667836

https://www.kaggle.com/havingfun/100-bollywood-celebrity-faces

https://www.sciencedirect.com/science/article/pii/S2666285X21000728

https://www.intechopen.com/chapters/51031