

Project Exhibition

(Review II Content)

**Staff and Visitor Logging**

**& Management System**

Semester III

By Abhijay Wadhwani(18BCE10005)

Hritik Ranjan(18BCE100117)

Utkarsha(18BCE10284)

16th October, 2019

Initial Version (v1.0)

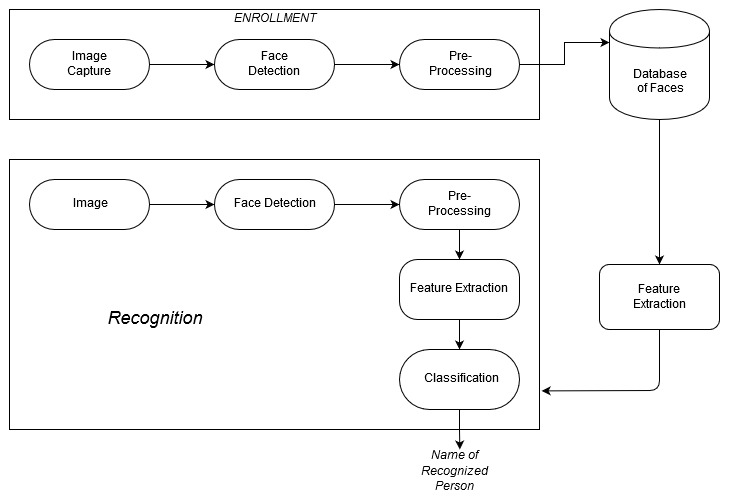
Guided by:

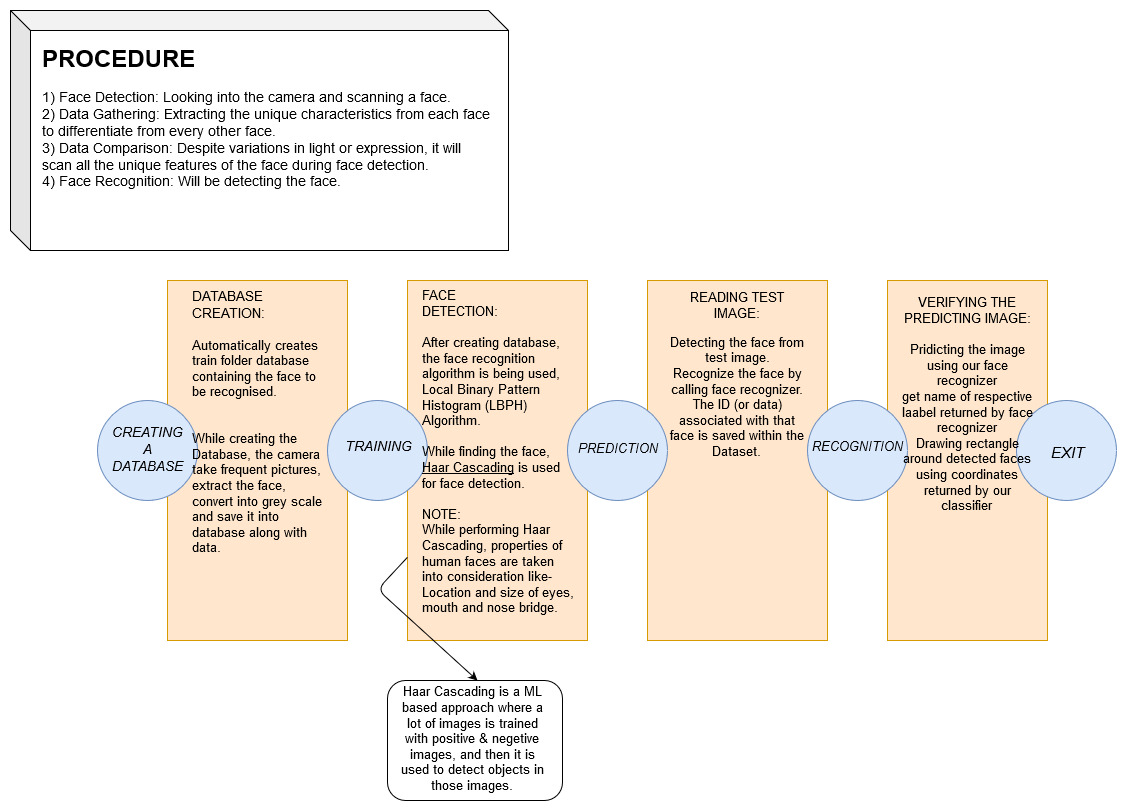
Dr. Manas Kumar Mishra

* Objectives:

To demonstrate abilities to:

* Handle camera setup and calibrate video faces
* Pre-process images and extract features
* Perform face recognition
* This system will provide a format for propagating operation of real time usage of computer vision. The actual usage of our project is that when a person enters, then its entry time is recorded and stored in the csv file. Now, if any visitor enters (Not in our database) in the block then its entity is registered and stored in the database and the stored info is used for future use. Our system can be used for verification as well as identification. If any person whose record is already present in the system, then it will show the record otherwise it will capture the image of person and save it as unidentified which can be used further as per requirement.
* System Architecture:



* Flow Diagram:
* Modules and Explanation

1. **Capturing and Storing Images:**

Asking the user to enter its name and then creating the folder having the entered name which will be used for labelling the Data. With the help of inbuilt Video capture method in openCV starting the process of capturing images of an individual, prompting the user to press the shutter in order to capture a minimum of 12 images for better feature extraction. All the captured images are stored into the labeled folder with numerical name of the image(1.jpg,2.jpg,3.jpg etc).

Hence, Creating the database for further training.

1. **Training the recognizer:**

We are using the LBPHFaceRecognizer with is inbuilt in the Cv2 library, we are using this Recognizer because it is not significantly affected by light and, in real life, we can't guarantee perfect light conditions, also we are using the Haar cascade Classifier for detecting the Region of Interest(Face).With the help of OS library walking into the Directory of our Database and reading images label by label, by using pillow opening image in Grayscale and converting into numpy Array and then passing this array into our Cascade Classifier to detect the face part of the image and then creating a list of obtained Region of Interest. Using the dump method of Pickle library serialzing the label list object. In the end training our recognizer with the ROIs list and corresponding labels and finally saving the obtained YML file(trainer.yml)

1. **Prediction:**

This is where we get to see if our algorithm is recognizing our individual faces or not. We are implementing real time face detection and Recognition, using the same cascade classifier and the same recognizer that we used in our Training module. First loading the dumped object using pickle and de-serializing the list object for use in our script. Detecting the face(Region of Interest) of the image frame obtained from the device camera and passing this ROI as a parameter in our recognizers predict function, here we are using a confidence value that shows the lower bound of match percentage. Printing the ID corresponding to the given face.

* References:

1.Research Papers

* + Ali, Ammad, Shah Hussain, Farah Haroon, Sajid Hussain and Mohammad Farhan Khan. “Face Recognition with Local Binary Patterns.” (2012).
  + S. Chintalapati and M. V. Raghunadh, "Automated attendance management system based on face recognition algorithms," *2013 IEEE International Conference on Computational Intelligence and Computing Research*, Enathi, 2013, pp. 1-5.
  + P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, pp. I-I.

2.Websites

* https://software.intel.com/en-us/node/754941
* <https://www.superdatascience.com/blogs/opencv-face-recognition>
* Cascade Classification — OpenCV 2.4.13.7 documentation
* <https://docs.python.org/3/>
* <https://github.com/codingforentrepreneurs/OpenCV-Python-Series>
* <https://github.com/informramiz/opencv-face-recognition-python>
* <https://stackoverflow.com/questions/22249579/opencv-detectmultiscale-minneighbors-parameter>
* **Future Improvements and Long term Goals**

In the current version of our project we have used the inbuilt LBPHFaceRecognizer, but we can use Deep Learning model for better accuracy and more efficiency.

To avoid spoofing we can further add eye blinking detection.