Face recognition and real –time face detection

Saloni Chudgar	– 1641013
Abhi Patel	– 1641021
Manav Chotalia	– 1641036
Aanal Shah	– 1844009

Problem Statement

- **Emotion Recognition and comparison of different methods for face detection**

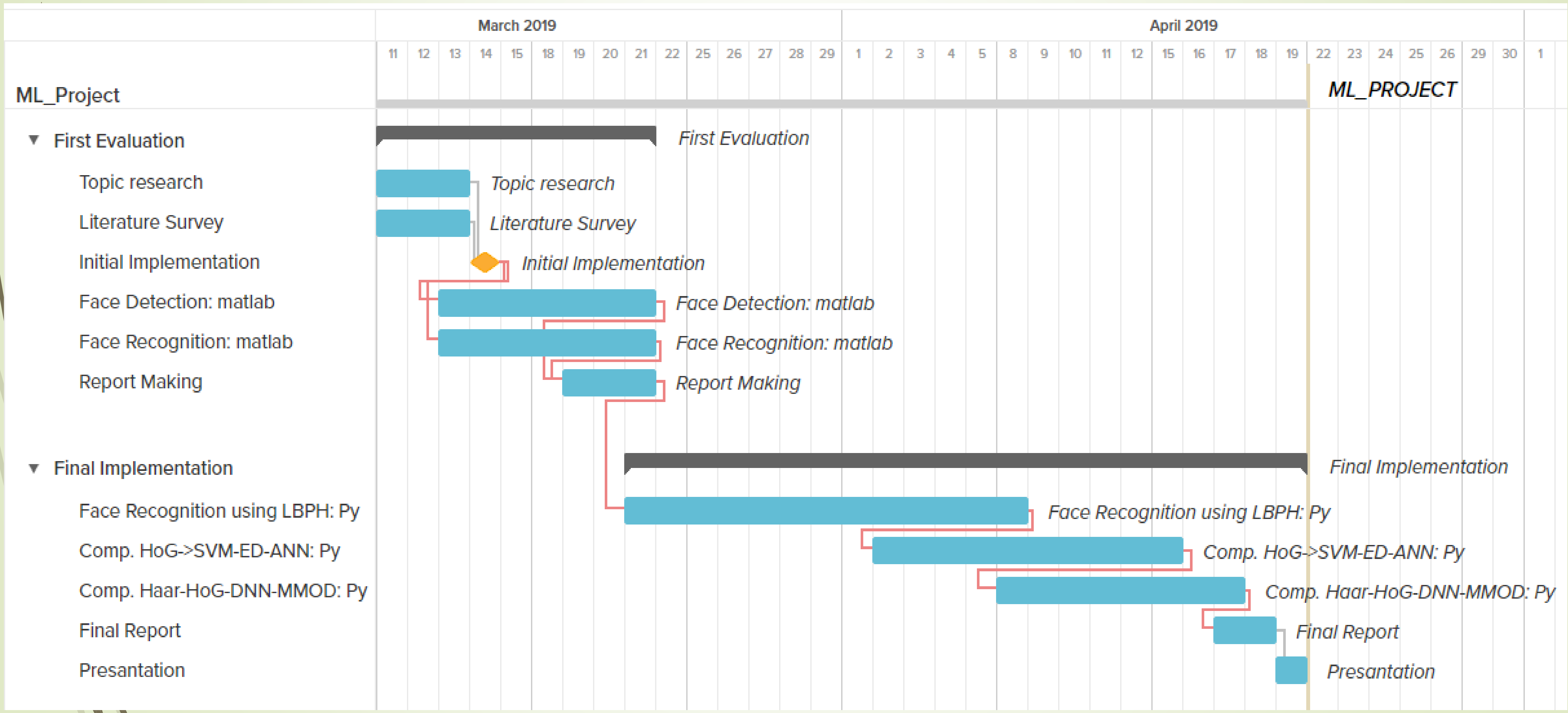
Introduction

- The various problems that arise during the process of face detection include the diversity among faces and lighting conditions.
- Thus, ideally the face detection algorithm used should be able detect the faces irrespective of the age, gender, skin conditions or lighting effects.
- In recent times, the accuracy of face detection algorithms has steadily increased. The approach is divided into two major types namely Feature Based (Geometric) and View Based (Photo-metric).
- The geometric approach works by identifying the various facial features like eyes, nose, mouth, etc first and then on the basis of distances and angles, the faces are classified. Thus, it works on the spatial configuration of features.
- While the photo metric approach works on the basis of a gradient mapping where the normal of different surfaces obtained from different images with a similar face are used.

Our Approach

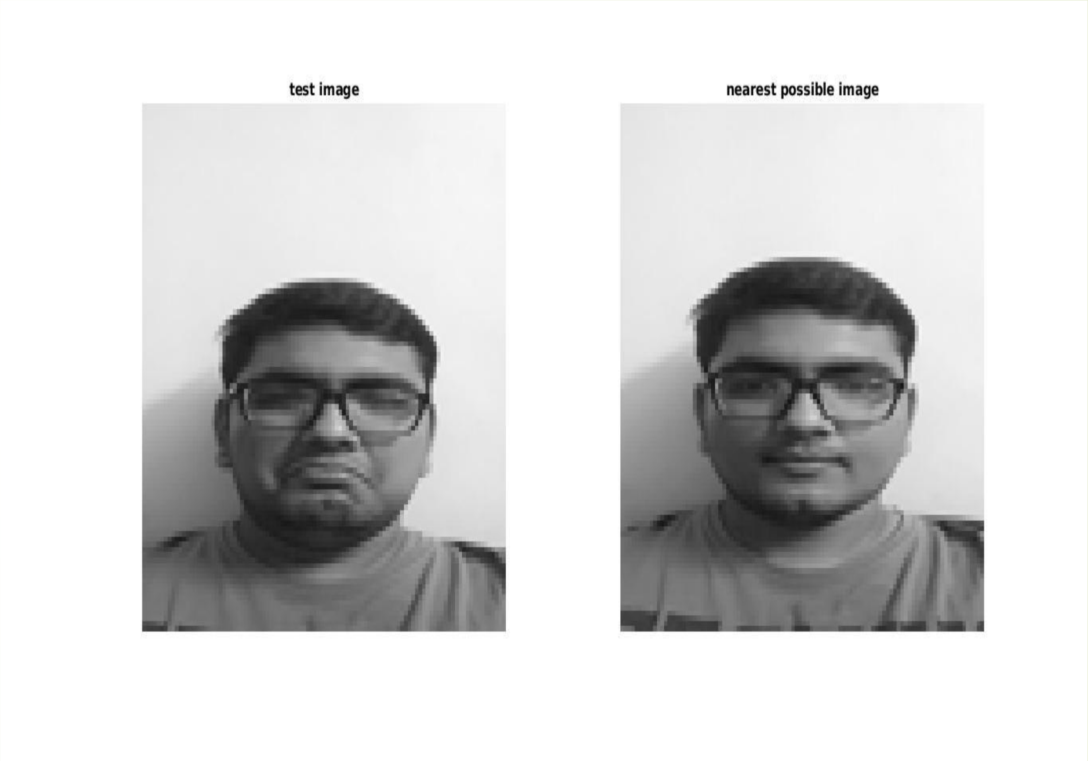
- We divided our work into modules to make it simpler:
- Face Detection in MATLAB using SVD and Euclidian Distance
- Face detection in C++.
- Face Detection in Python.
- Face Recognition in Python using ANN.
- Comparison of SVM, ED and ANN with HoG as feature extractor.
- Realtime comparison of DNN, HoG, Haar and MMOD(Max-Margin Object Detection)

Project Progress



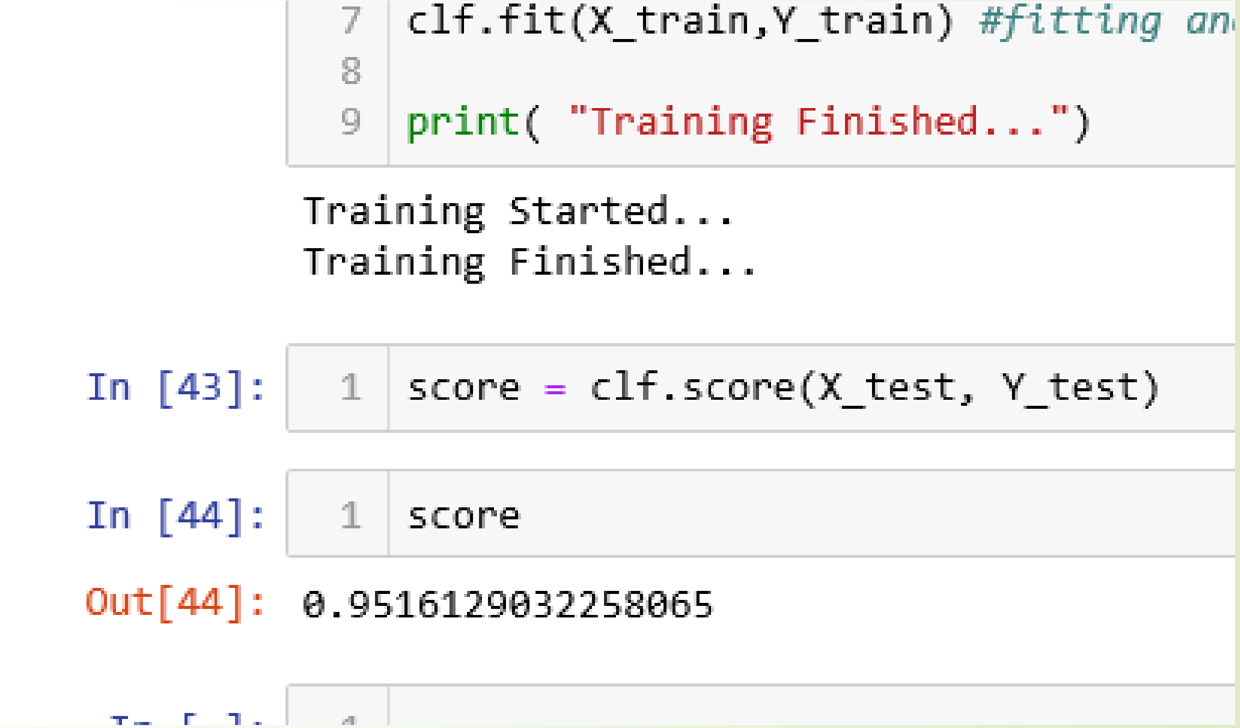
Approach - In MATLAB and C++

- Method used : SVD and Euclidian Distance.
- The image is stored in the form of column vectors.
- Dataset Used : 62 students of Ahmedabad University.



Approach - Face Recognition Python

- Method Used : SVM
- Supervised Method
- Accuracy :
- Initial – 42 %
- Final – 95%



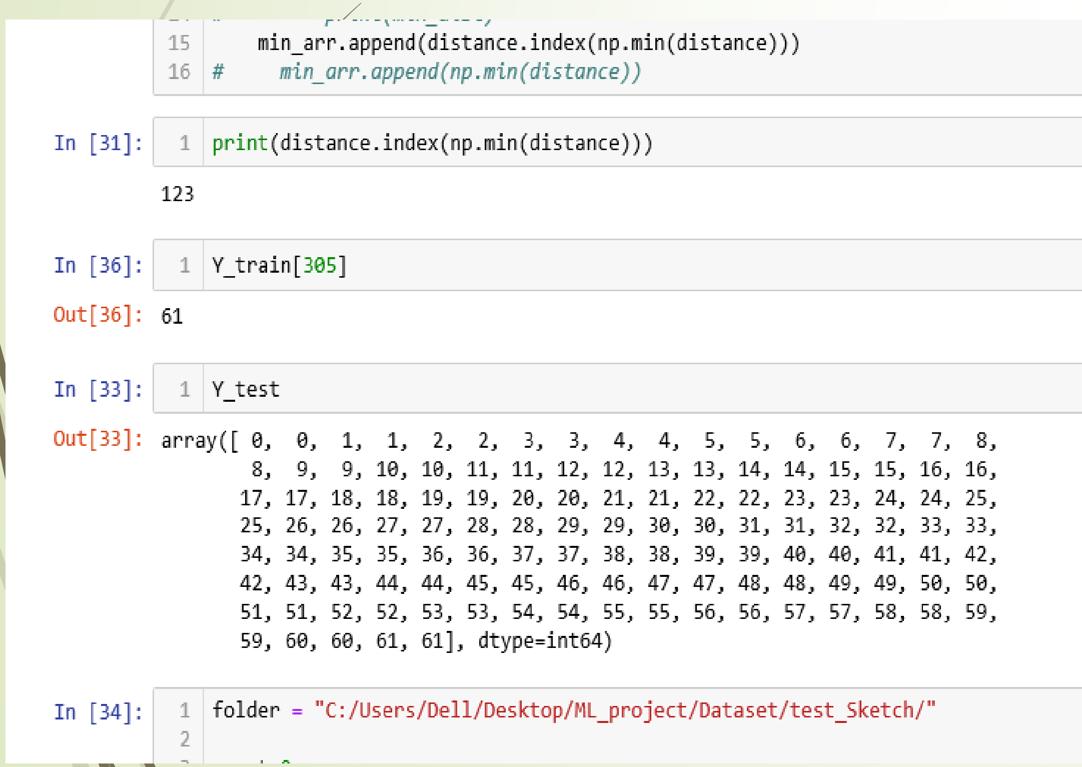
```
clf.fit(X_train,Y_train) #fitting an

print( "Training Finished...")
```

```
Training Started...
Training Finished...
```

```
In [43]: score = clf.score(X_test, Y_test)

In [44]: score

Out[44]: 0.9516129032258065
```

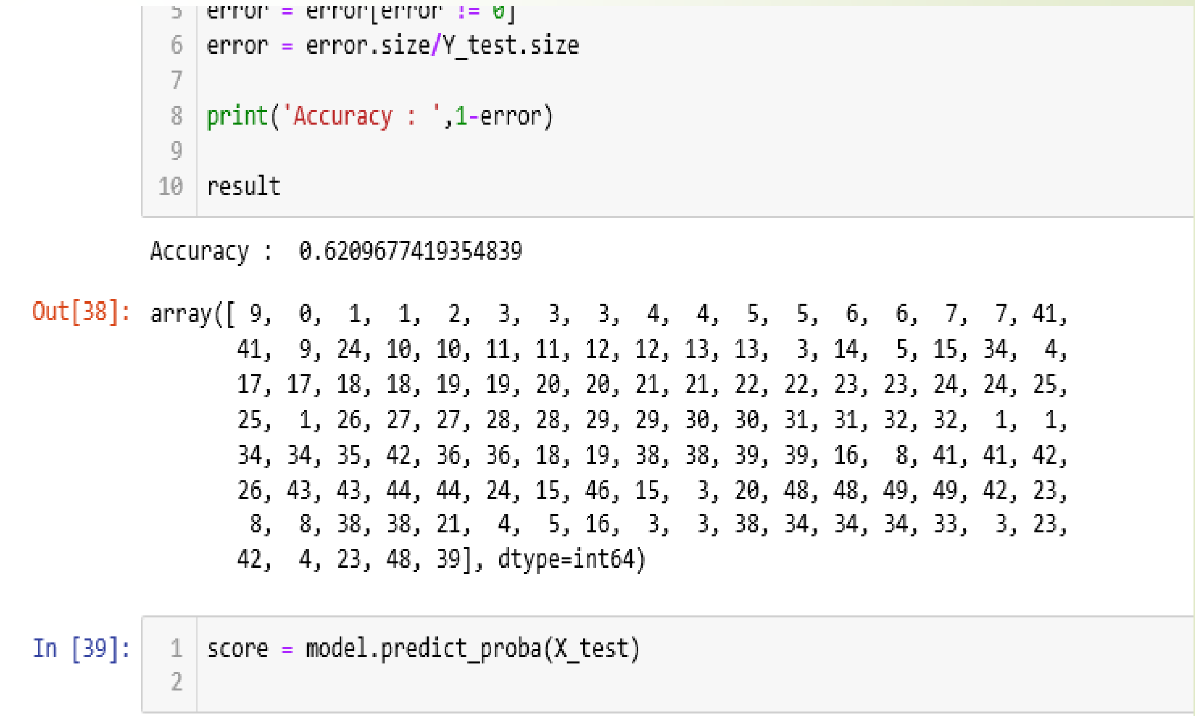
Approach - Face Recognition Python

- Method Used :Euclidian Distance
- Accuracy : 62 %



```
    min_arr.append(distance.index(np.min(distance)))
#     min_arr.append(np.min(distance))
```

```
In [31]: print(distance.index(np.min(distance)))

         123

In [36]: Y_train[305]

Out[36]: 61

In [33]: Y_test

Out[33]: array([ 0,  0,  1,  1,  2,  2,  3,  3,  4,  4,  5,  5,  6,  6,  7,  7,  8,
                 8,  9,  9, 10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 16,
                17, 17, 18, 18, 19, 19, 20, 20, 21, 21, 22, 22, 23, 23, 24, 24, 25,
                25, 26, 26, 27, 27, 28, 28, 29, 29, 30, 30, 31, 31, 32, 32, 33, 33,
                34, 34, 35, 35, 36, 36, 37, 37, 38, 38, 39, 39, 40, 40, 41, 41, 42,
                42, 43, 43, 44, 44, 45, 45, 46, 46, 47, 47, 48, 48, 49, 49, 50, 50,
                51, 51, 52, 52, 53, 53, 54, 54, 55, 55, 56, 56, 57, 57, 58, 58, 59,
                59, 60, 60, 61, 61], dtype=int64)

In [34]: folder = "C:/Users/Dell/Desktop/ML_project/Dataset/test_Sketch/"
```

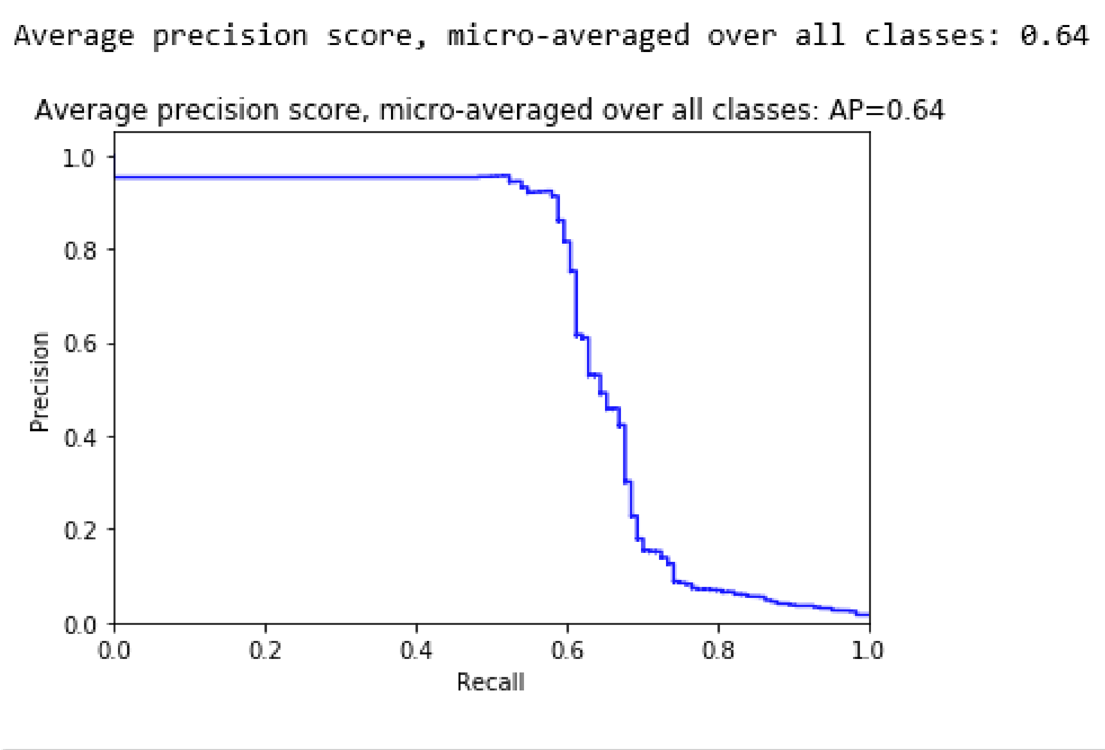


```
error = error[error != 0]
error = error.size/Y_test.size

print('Accuracy : ',1-error)

result
```

```
         Accuracy :  0.6209677419354839

Out[38]: array([ 9,  0,  1,  1,  2,  3,  3,  3,  4,  4,  5,  5,  6,  6,  7,  7, 41,
                41,  9, 24, 10, 10, 11, 11, 12, 12, 13, 13,  3, 14,  5, 15, 34,  4,
                17, 17, 18, 18, 19, 19, 20, 20, 21, 21, 22, 22, 23, 23, 24, 24, 25,
                25,  1, 26, 27, 27, 28, 28, 29, 29, 30, 30, 31, 31, 32, 32,  1,  1,
                34, 34, 35, 42, 36, 36, 18, 19, 38, 38, 39, 39, 16,  8, 41, 41, 42,
                26, 43, 43, 44, 44, 24, 15, 46, 15,  3, 20, 48, 48, 49, 49, 42, 23,
                 8,  8, 38, 38, 21,  4,  5, 16,  3,  3, 38, 34, 34, 34, 33,  3, 23,
                42,  4, 23, 48, 39], dtype=int64)

In [39]: score = model.predict_proba(X_test)
```

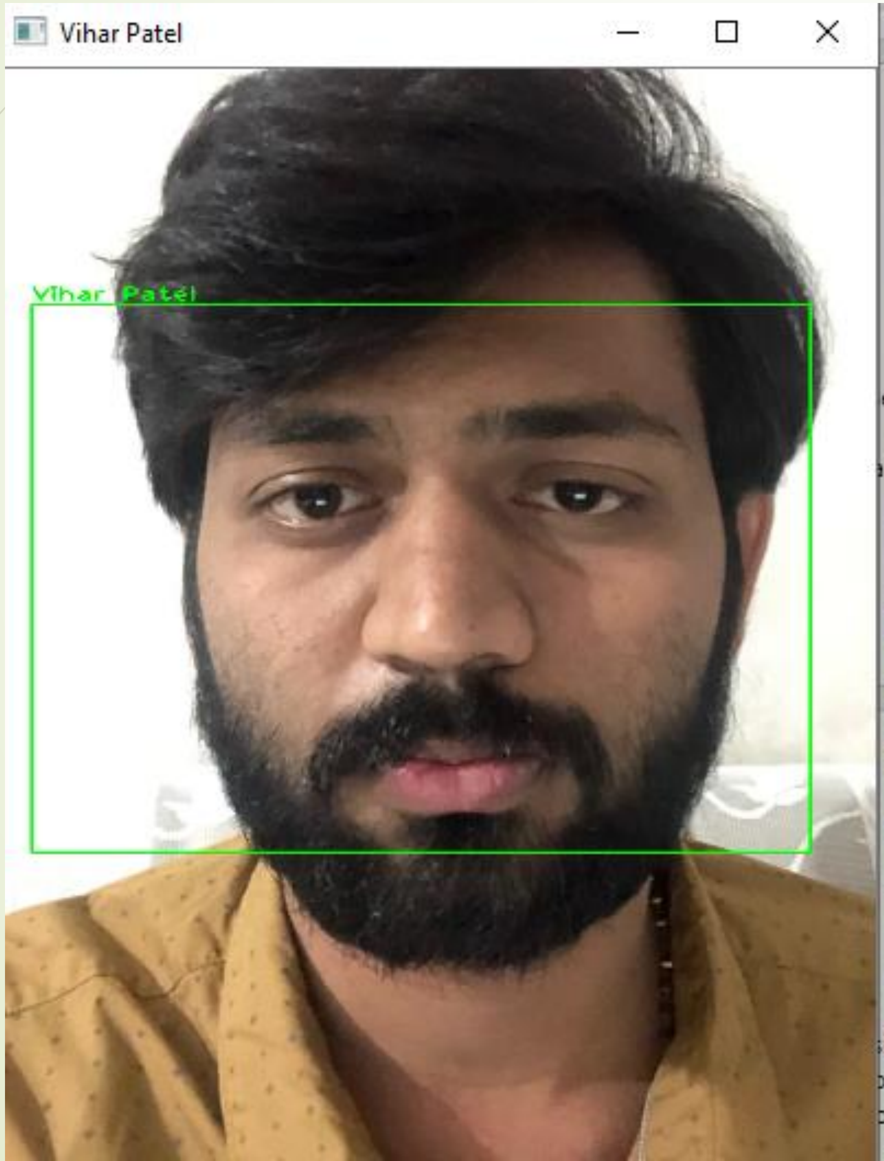
Approach - Face Recognition Python

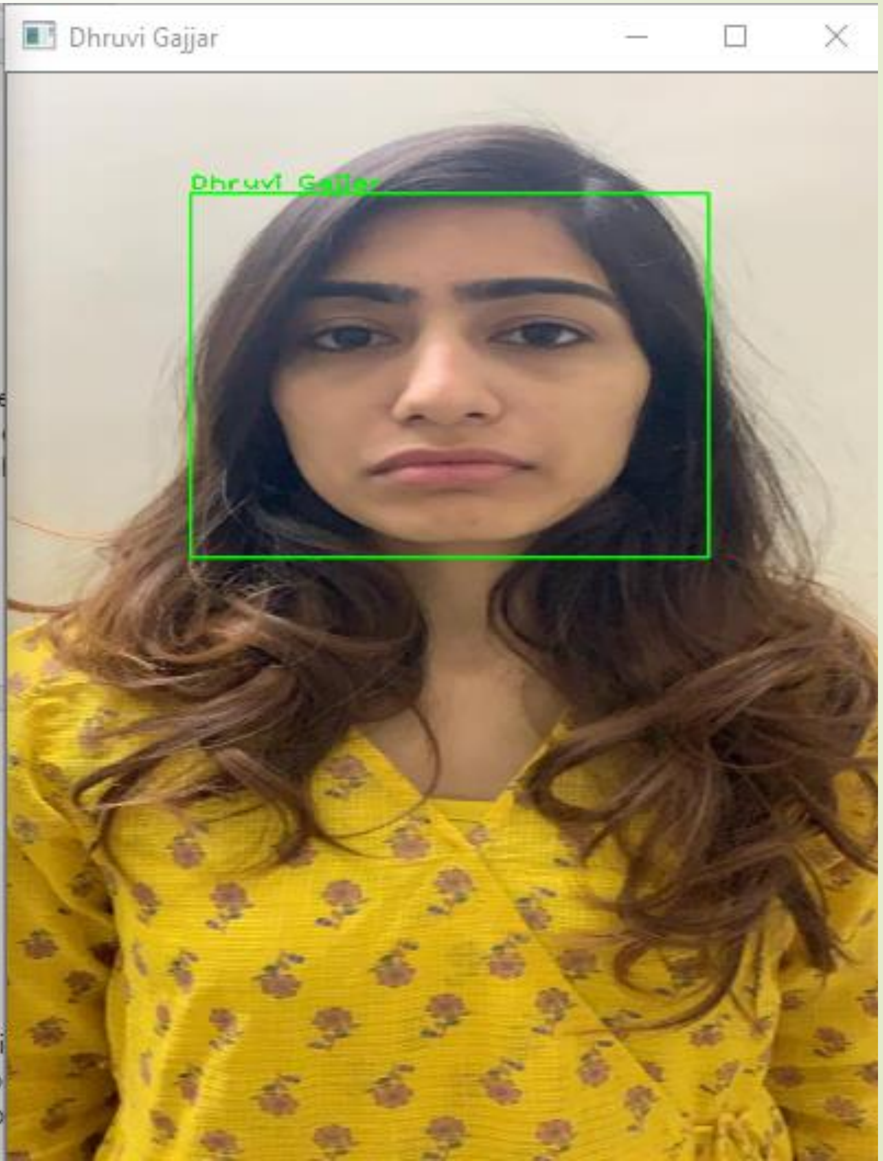
- Method Used : ANN
- Precision vs Recall
- Precision - 64%



```
Average precision score, micro-averaged over all classes: 0.64
```

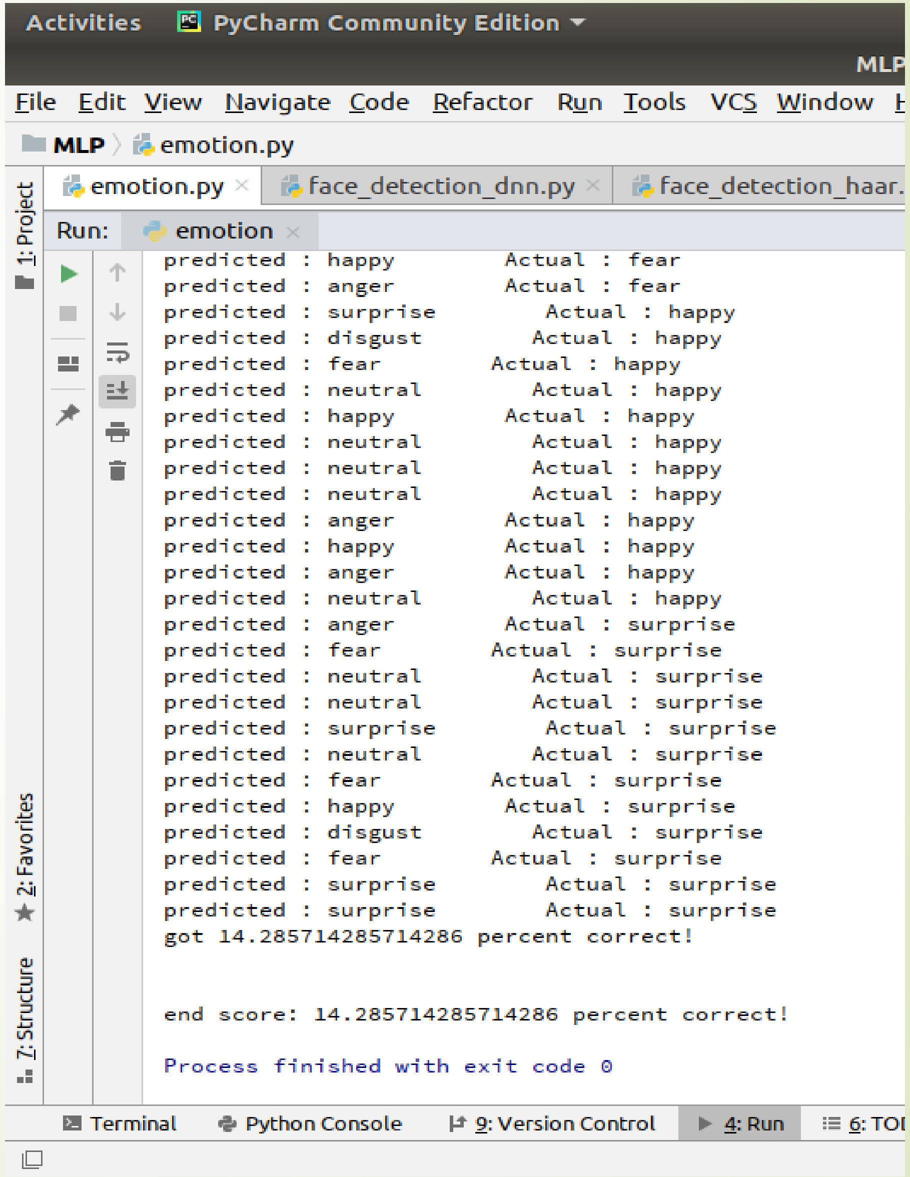
Approach - Face Recognition Python





Approach - Emotion Recognition Python

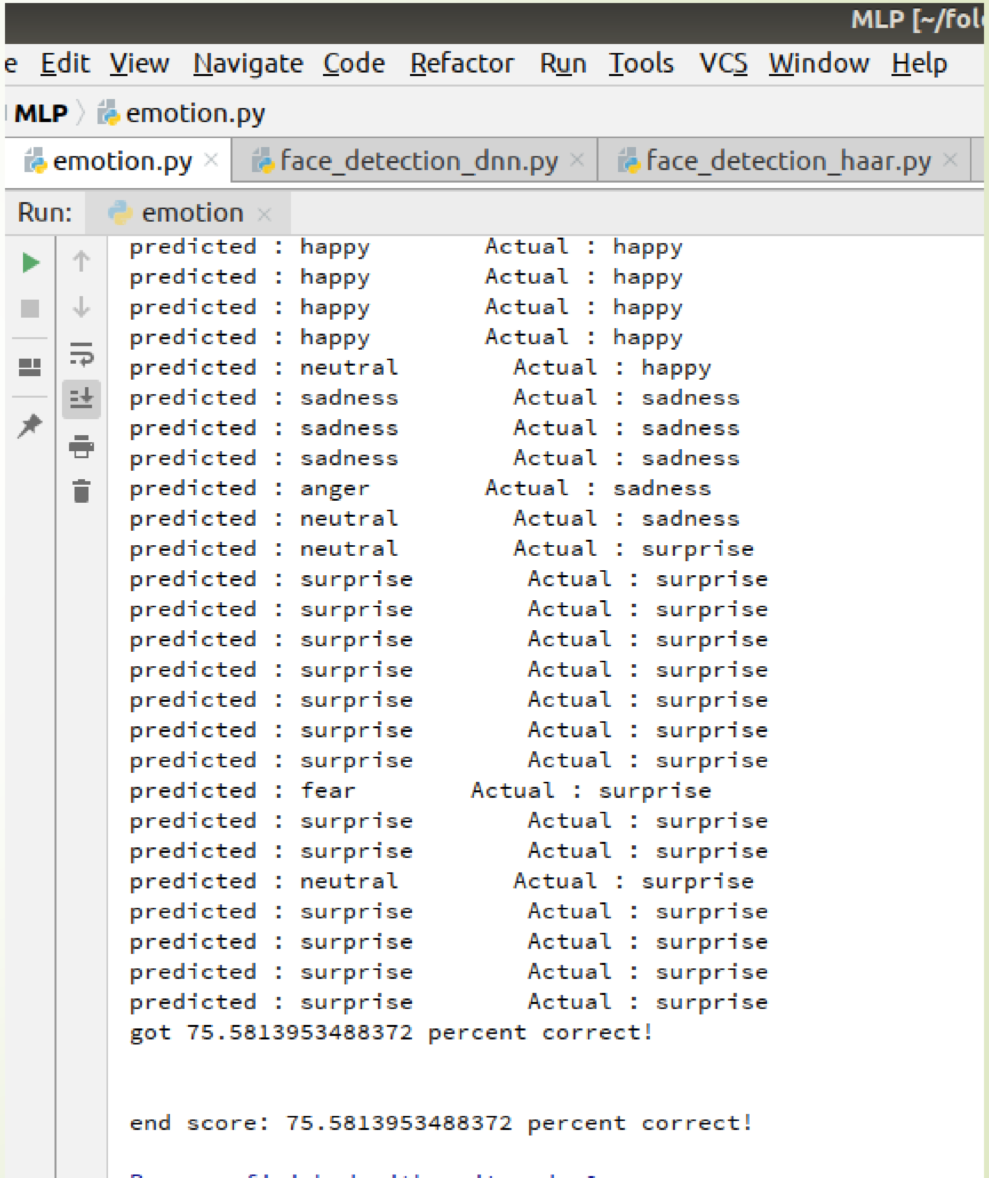
- Method Used : Fischer Faces
- Data Set : Ahmedabad University Students
- Accuracy : 14 -20 %



```
predicted : happy        Actual : fear
predicted : anger        Actual : fear
predicted : surprise        Actual : happy
predicted : disgust        Actual : happy
predicted : fear        Actual : happy
predicted : neutral        Actual : happy
predicted : happy        Actual : happy
predicted : neutral        Actual : happy
predicted : neutral        Actual : happy
predicted : neutral        Actual : happy
predicted : anger        Actual : happy
predicted : happy        Actual : happy
predicted : anger        Actual : happy
predicted : neutral        Actual : happy
predicted : anger        Actual : surprise
predicted : fear        Actual : surprise
predicted : neutral        Actual : surprise
predicted : neutral        Actual : surprise
predicted : surprise        Actual : surprise
predicted : neutral        Actual : surprise
predicted : fear        Actual : surprise
predicted : happy        Actual : surprise
predicted : disgust        Actual : surprise
predicted : fear        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
got 14.285714285714286 percent correct!


end score: 14.285714285714286 percent correct!

Process finished with exit code 0
```

Approach - Emotion Recognition Python

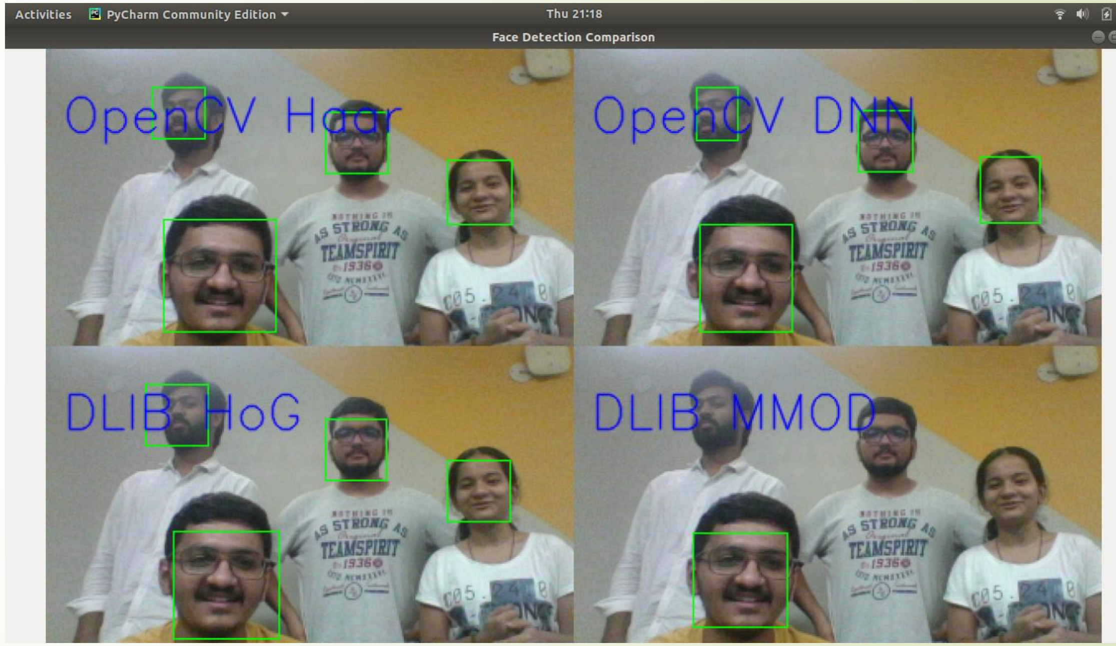
- Method Used : Fischer Faces
- Data Set : Pol-Van
- Accuracy : 75 - 82 %



```
predicted : happy        Actual : happy
predicted : happy        Actual : happy
predicted : happy        Actual : happy
predicted : happy        Actual : happy
predicted : neutral        Actual : happy
predicted : sadness        Actual : sadness
predicted : sadness        Actual : sadness
predicted : sadness        Actual : sadness
predicted : anger        Actual : sadness
predicted : neutral        Actual : sadness
predicted : neutral        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : fear        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : neutral        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
predicted : surprise        Actual : surprise
got 75.5813953488372 percent correct!


end score: 75.5813953488372 percent correct!
```

Approach - Real Time comparison of 4 different methods.

- Methods Used :
- HAAR Cascade
- HoG
- DNN – Single Shot Multi-Box Detector
- CNN - MMOD(Max-Margin Object Detector)



Final Results

	DNN	HAAR Cascade	HoG	CNN
Real time Detection	always works	works mostly	works sometimes	works sometimes
Non-frontal faces	always works	doesn’t work	works sometimes	works mostly
Scaled Face	always works	works mostly	works mostly	works sometimes
Occlusions	works mostly	doesn’t work	works sometimes	always works

Contribution

	Saloni Chudgar (1641013)	**Abhi Patel (1641021)**	**Manav Chotalia (1641036)**	**Aanal Shah (1844009)**
C and Matlab code	Worked on C-Code	Worked on MATLAB - Code	Worked on MATLAB - Code	Worked on C-Code
Face Detection Python	Worked on SVM	Worked on ANN	Worked on HoG	Worked on MMOD
Other	Worked on Face Recognition Python	Worked on Emotion Detection	Worked on Face Recognition Python	Worked on Emotion Detection

References

- Turk, Matthew, and Alex Pentland. ”Eigenfaces for recognition.” Journal of cognitive neuroscience 3.1 (1991): 71-86.
- Belhumeur, Peter N., Joo P. Hespanha, and David J. Kriegman. ”Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.” IEEE Transactions on Pattern Analysis and Machine Intelligence 7 (1997): 711-720.
- Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. ”Deep face recognition.” bmvc. Vol. 1. No. 3. 2015.
- Dniz, Oscar, et al. ”Face recognition using histograms of oriented gradients.” Pattern Recognition Letters 32.12 (2011): 1598-1603.
- King, Davis E.. Max-Margin Object Detection. CoRR abs/1502.00046 (2015): n. pag.