Facial Recognition for Visually Impaired

A Sai Charan	- 15404
Sriram Yasaswi	- 15441
Sangeeth Raaj S R	- 15442
Vasanth J S	-15454

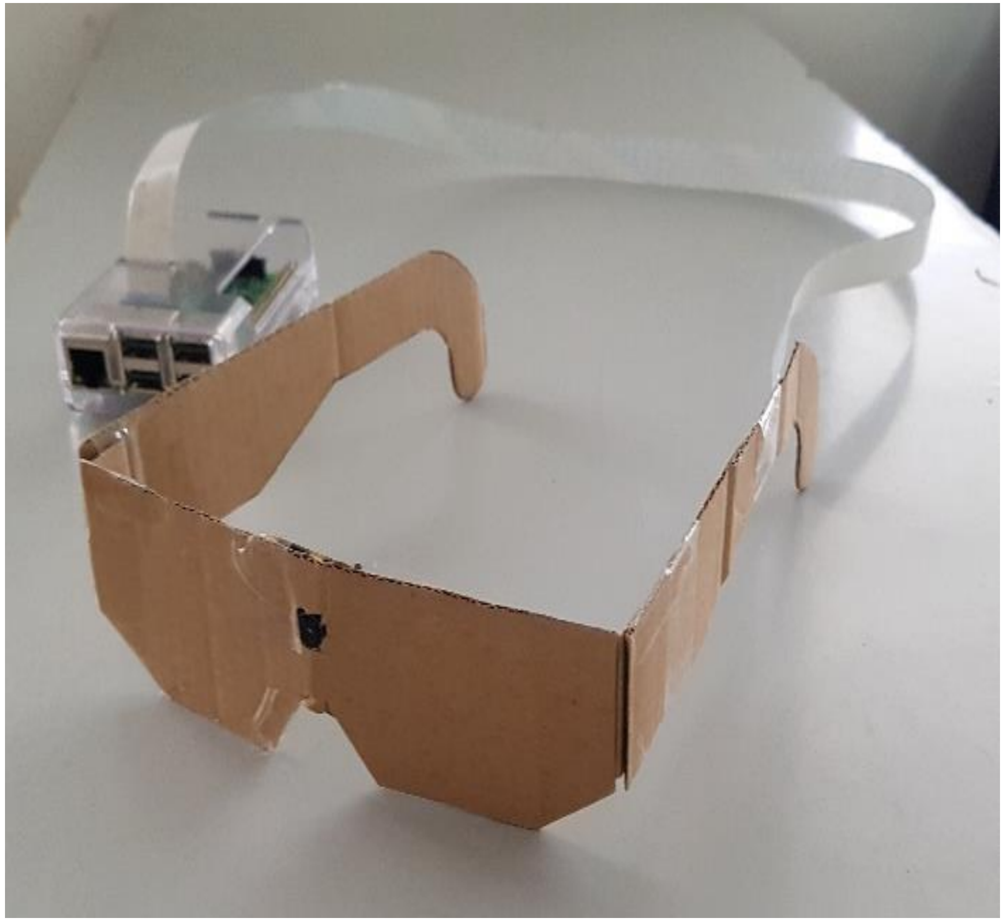
Introduction

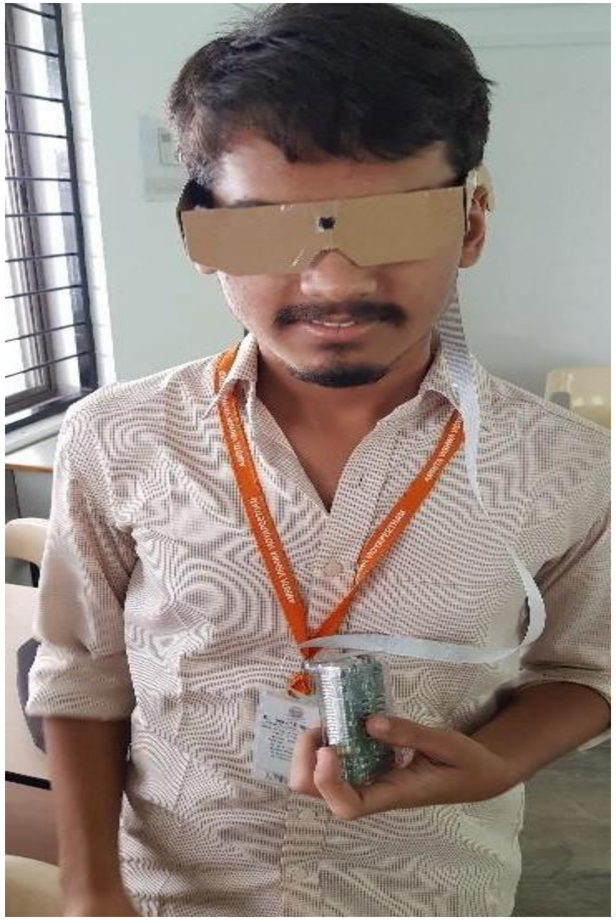
- Visual Impairment refers to a loss of vision. It creates difficulties to people in their day to day activities such as socializing, reading and walking. A major concern for such people is recognizing known individuals who are in their presence.
- Facial Recognition tools and technology can help them overcome that disability.
- The objective is designing an efficient system that will accurately detect and also recognize faces irrespective of changes in environmental and lighting conditions.

- A prototype is to be developed that can be worn by the user. This prototype has embedded devices which perform these tasks in real time. It performs face recognition using a variety of methods and sends a audio message to the user regarding the identity of any individual in front of them.

- It comprises of a raspberry pi and cardboard goggles. A PiCam was attached to the center of the goggle. This captures images and sends them to the Raspberry Pi that will be carried in a pouch along with its battery source.

Prototype

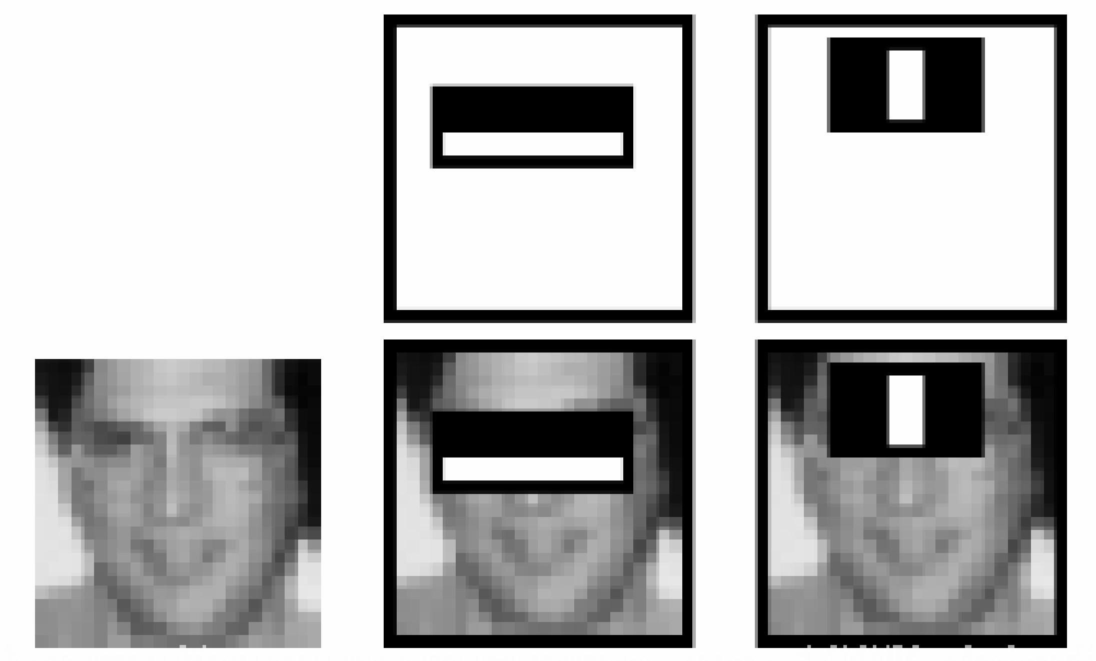




Working

- Initially, we made use of Haar cascade classifier for detecting faces and eigen faces algorithm for recognizing faces. This classifier is trained with several positive (face) and negative (non-face) images.
- It uses Haar-like features to identify regions in the image that contain faces. These features are applied at all windows of any image. Since there are many features, they are cascaded to form a system that has different levels through which each image has to go through.
- The window of an image that goes through all stages is a face region.

- The image below shows a feature which compares the darkness of the eye and the nose bridge to detect faces. In any image most regions are non face regions and so the windows that are classified as negative are removed.



- Eigen faces method was used for face recognition. It is performed using feature values that are projected by one eigenface set obtained from principal component analysis (PCA). The idea of PCA is to reduce the dimensionality of a data set while retaining the variations in the data set as much as possible.
- The geometric distance between the feature vectors of an observed image x and an image in the dataset xt is calculated by using the formula

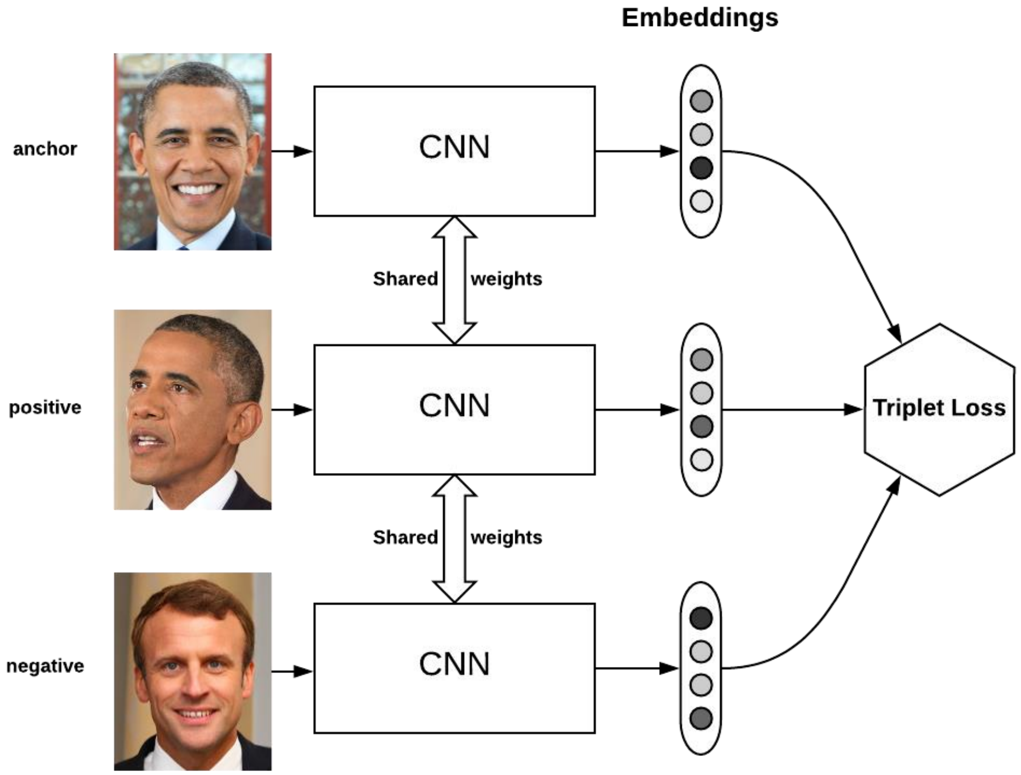
$$d(x, xt) = || y - yt ||$$

- Here y and yt represent the feature vectors of the respective images. It will be recognised based on the image in the dataset to which it has the shortest distance.

FaceNet

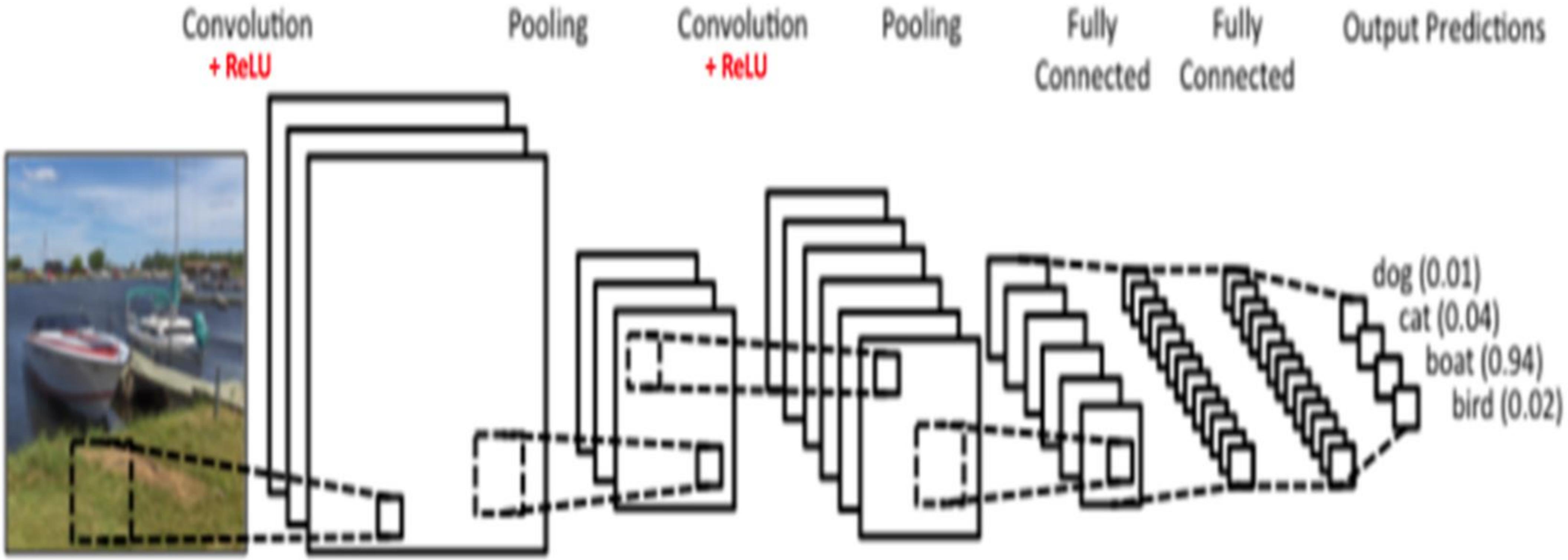
- Google’s FaceNet neural network claimed to give an enhanced accuracy and so we replaced the eigenfaces method with the FaceNet model. It is a deep learning architecture consisting of convolutional layers and returns a 128 dimensional vector embedding for each face.
- It learns a mapping from face images to a compact Euclidean Space where distances directly correspond to a measure of face similarity.
- Once this is done, tasks such as face recognition, verification, and clustering are easy to do using standard techniques

- Training is done using triplets: one image of a face is anchor , another image of that same face is the positive and an image of a different face is the negative. It makes use of triplet loss. It achieves very high accuracy. FaceNet makes use of Convolutional layers.



Convolutional Neural Network

- A Convolutional Neural Network is a network with some convolutional layers . These layers have a number of filters that does various operations.
- The input images are represented in the form of tensors which are multidimensional arrays and manipulation of these matrices is performed.
- After passing through several layers , they are converted into a feature vector that is of 128 dimensions.


Convolution
+ ReLU
Pooling
Convolution
+ ReLU
Pooling
Fully Connected
Fully Connected
Output Predictions
dog (0.01)
cat (0.04)
boat (0.94)
bird (0.02)

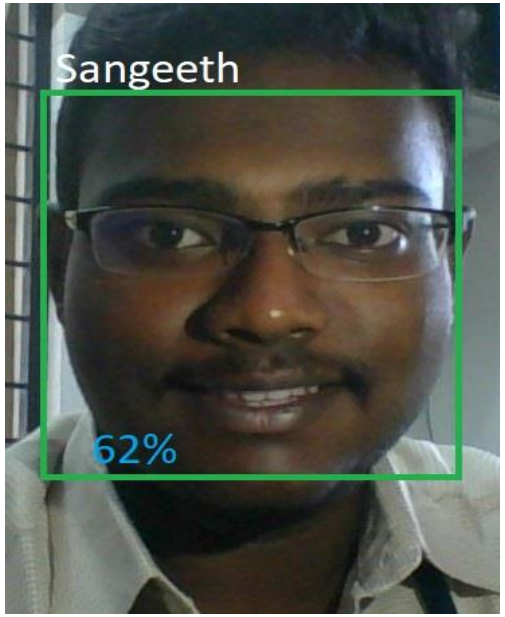
Enhancing the performance

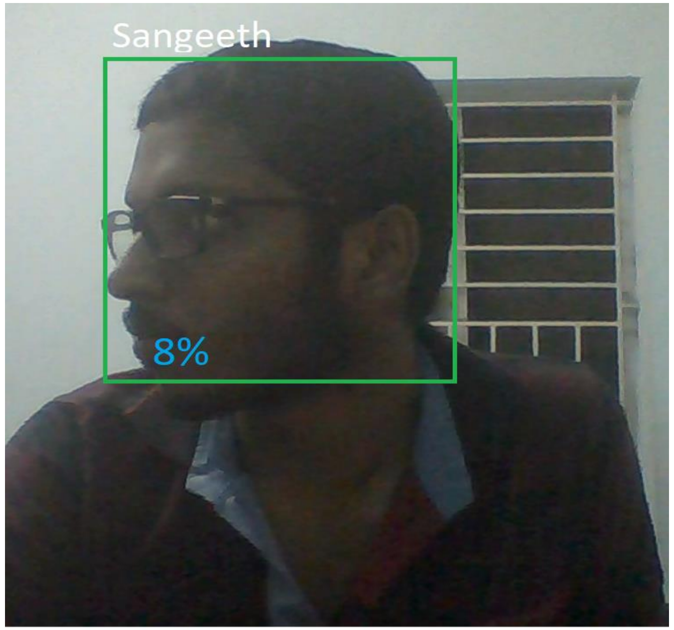
- In order to improve the accuracy offered by FaceNet, we add a additional layer over the neural network layer. The 128-d value obtained is fed into a classifying or clustering algorithm like SVM, Naive Bayes, KNN and K-means.

- The dataset is a set of faces that are trained in different angles and lighting conditions. When applying these algorithm during the testing phase it was found that SVM and KNN offered the highest accuracy and fastest recognition rates.

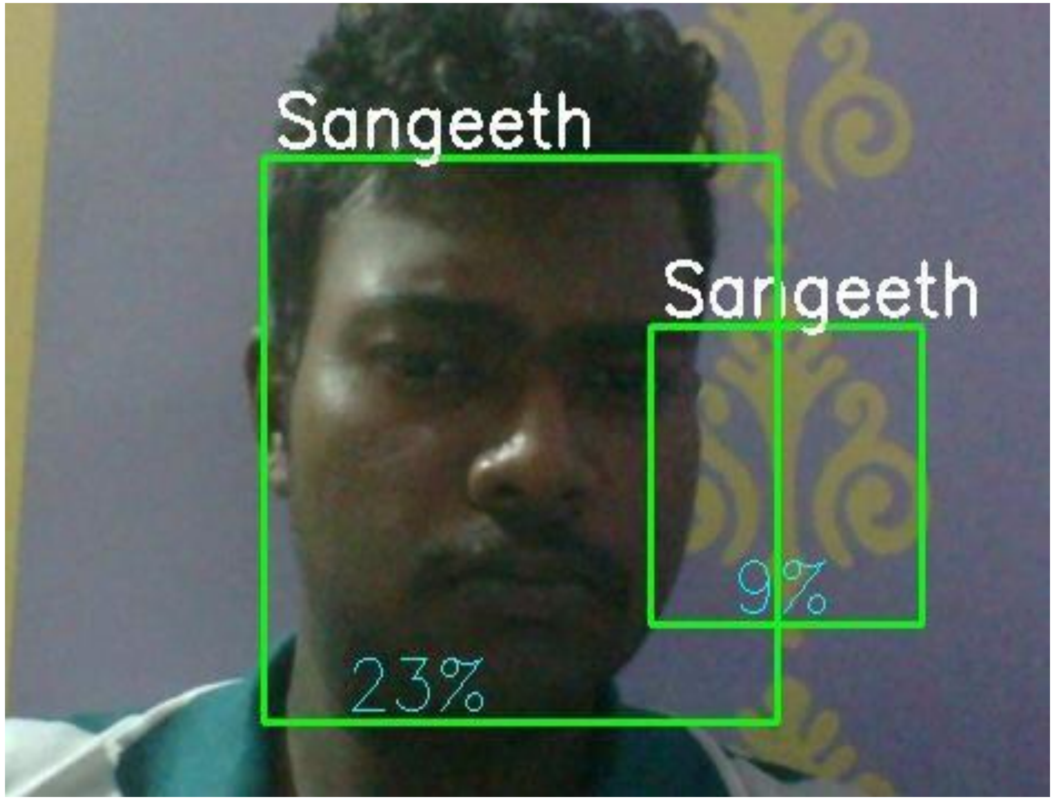
Results

- Frontal Recognition with SVM offered very high accuracy and correctly recognised images on most occasions. However , there were wrong recognitions when faces at different angles were used.

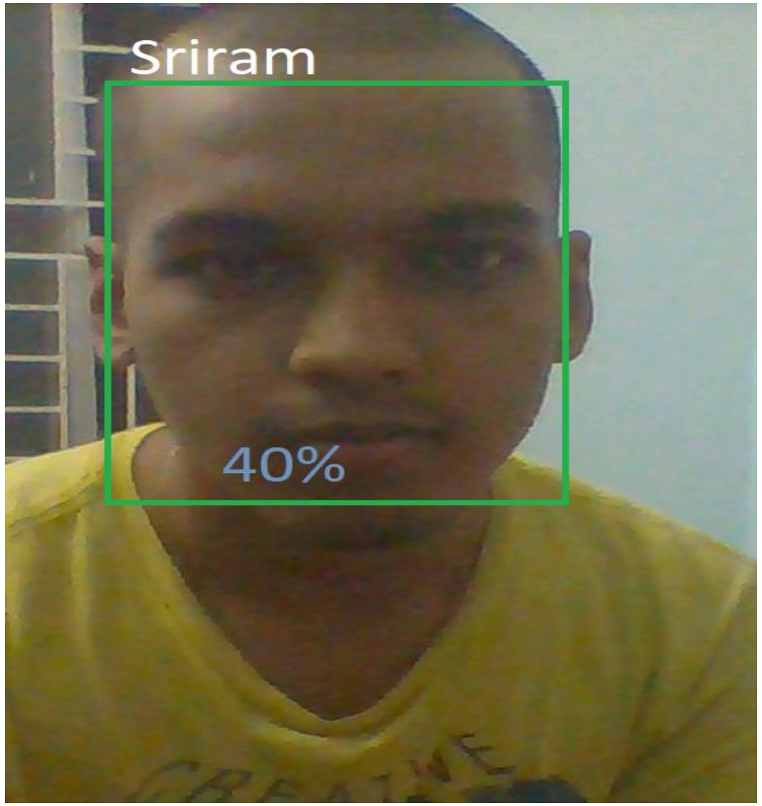




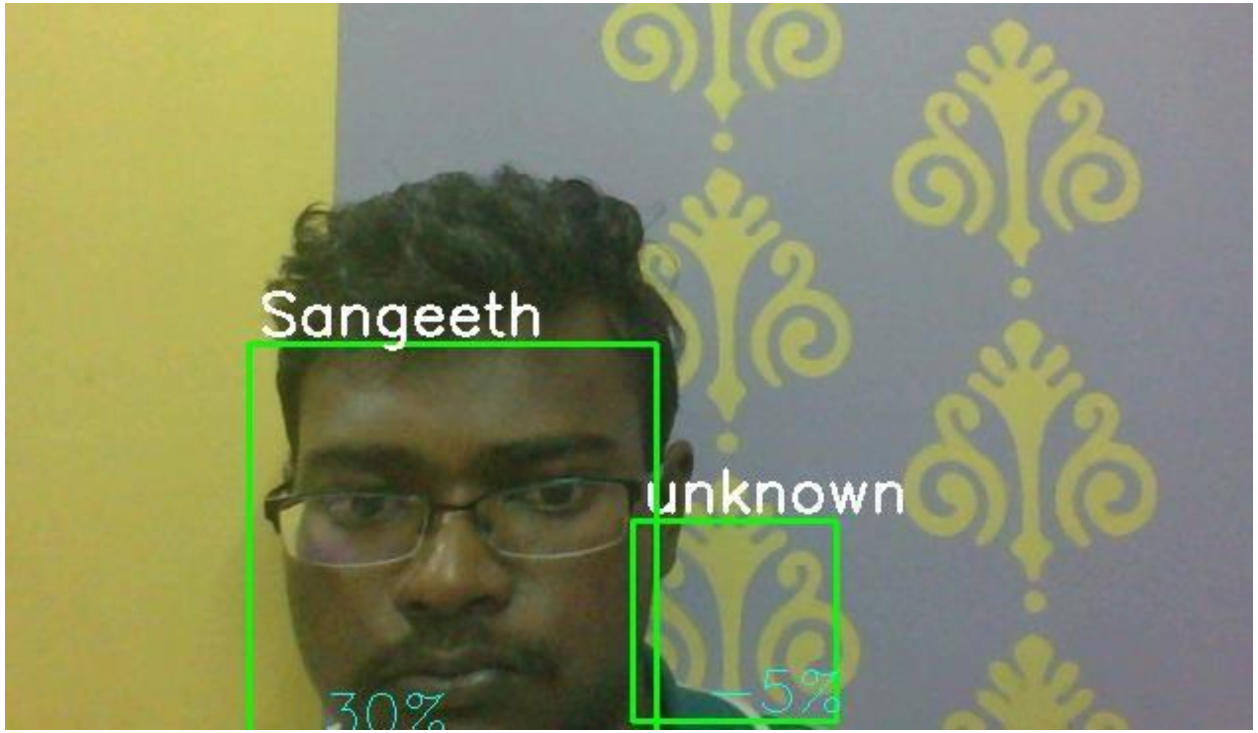
- There were some false positives based on lighting conditions. When a non-face region is detected as a face , it is a false positive.



- Naive Bayes classifier works best when there are independent variables. Since the dataset has features that are independent of each other and the degree to which classes overlap each other are small, it is suitable to use this classifier.

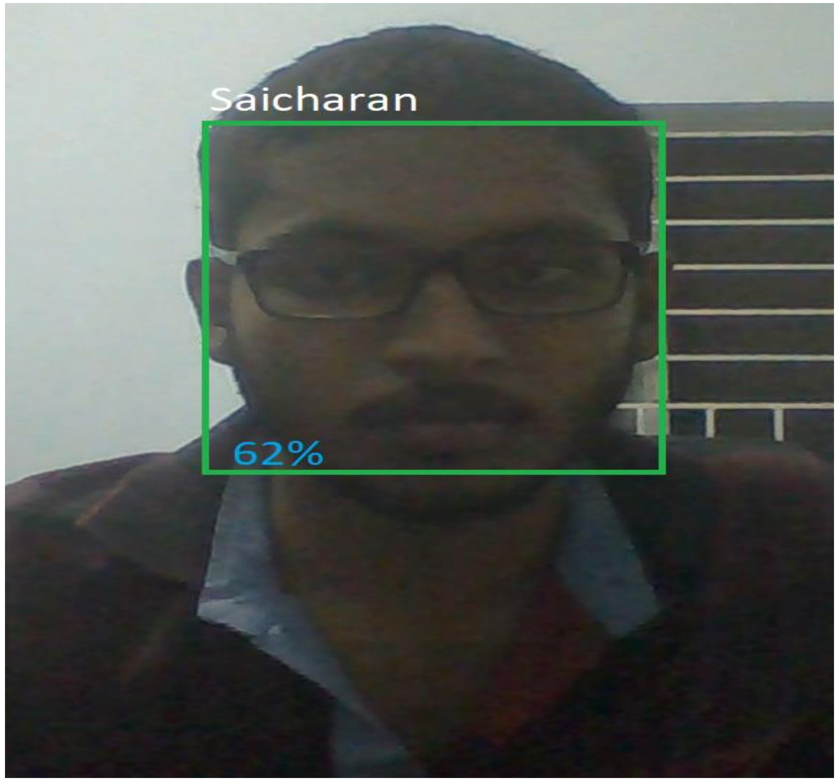


- A few cases of wrong recognition were found. There were some cases where non-face regions were wrongly detected to be faces and during the recognition phase, they were given an unknown tag with very less accuracy. This was seen on rare occasions.



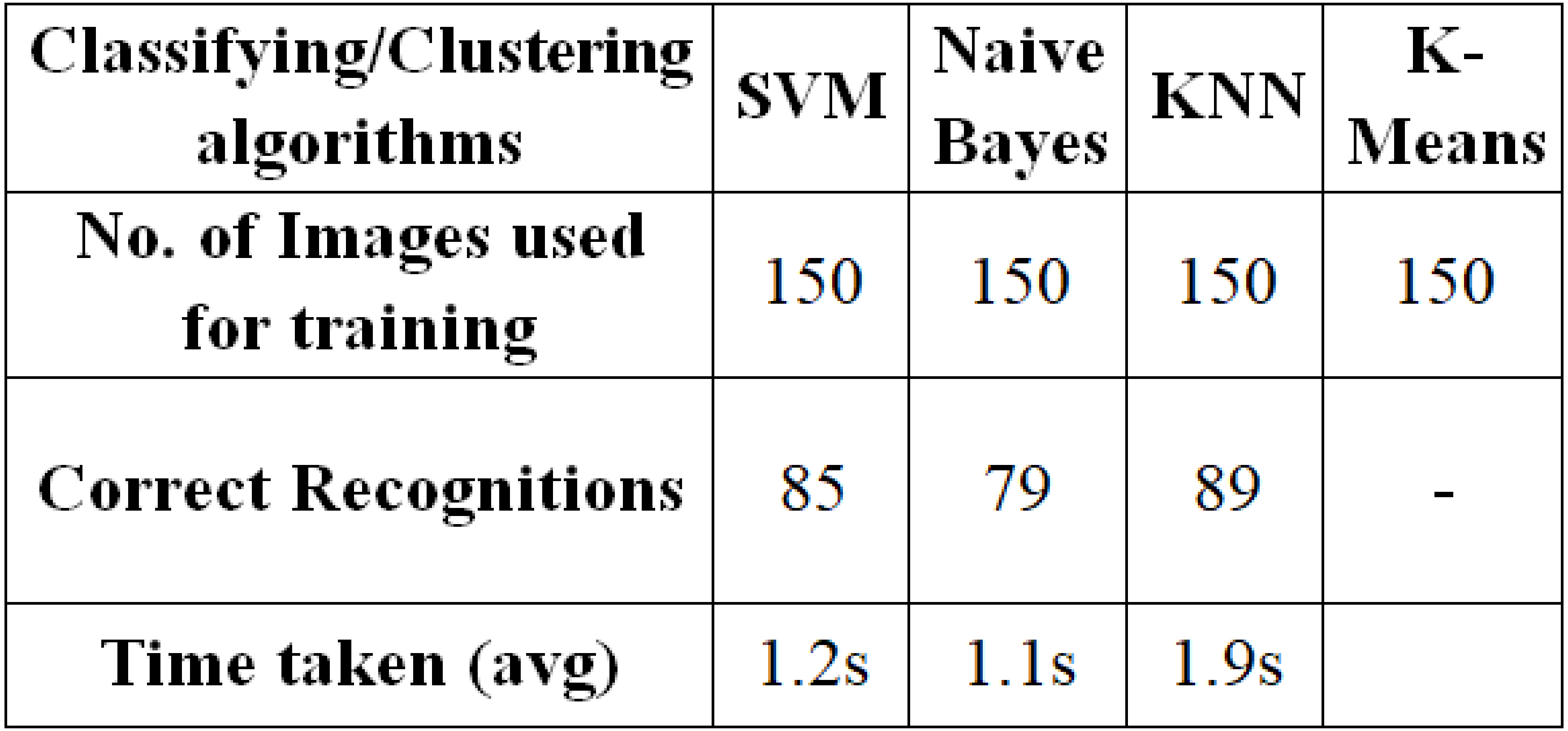
- A common drawback we found was that when using images captured in the dark , the algorithms failed to even detect the faces on majority occasions.

- It was found that KNN algorithm offered the highest accuracy. The faces were recognized correctly but KNN is slow in comparison to other classifiers. Since it needs to calculate the distance and the training data must be sorted at every prediction, it generally takes more time.



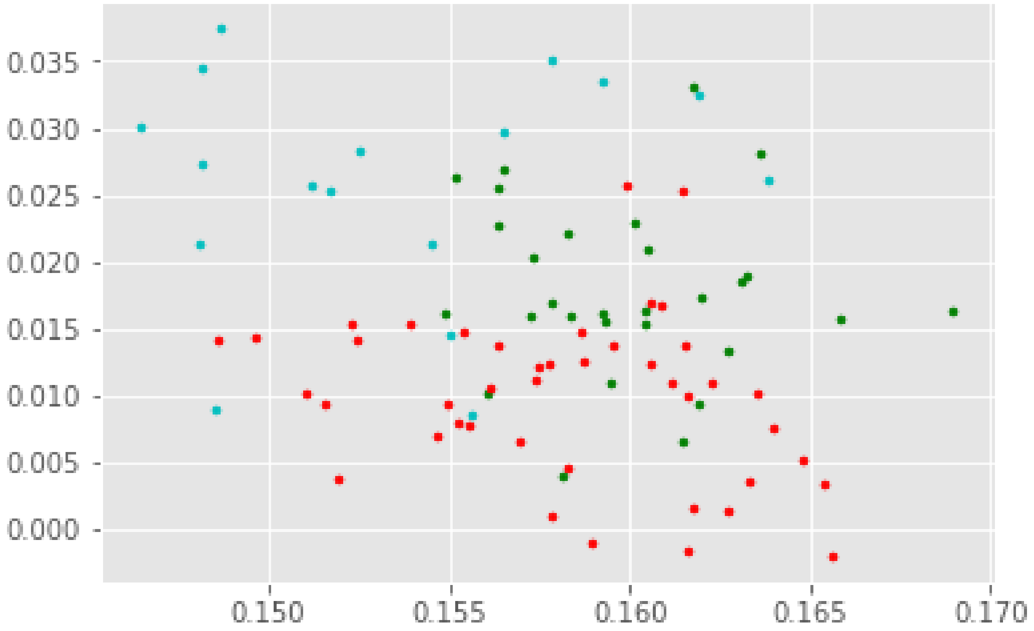
Working Prototype

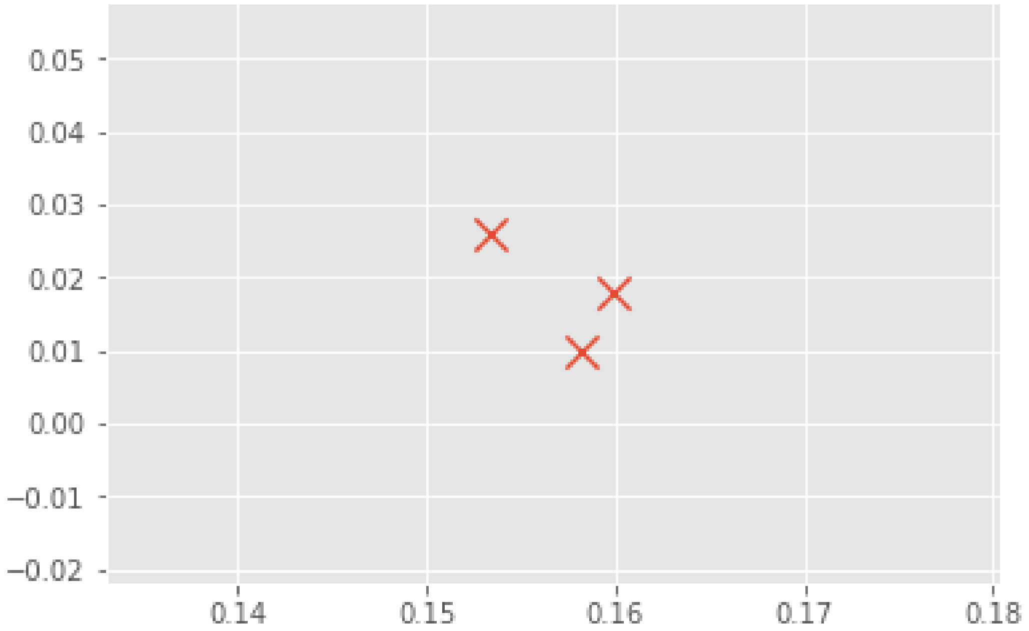
Comparison of classifiers and clusters



Classifying/Clustering algorithms	SVM	Naive Bayes	KNN	K-Means
No. of Images used for training	150	150	150	150
Correct Recognitions	85	79	89	-
Time taken (avg)	1.2s	1.1s	1.9s	

- The first graph below shows a representation of the embeddings of three different classes when K-means clustering is used. It is seen that they don’t form distinct clusters and they are spread throughout the graph. The other graph shows that when the center points of the clusters of different classes were plotted, only two entries appear closer to each other


0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0.150
0.155
0.160
0.165
0.170


0.05
0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
0.14
0.15
0.16
0.17
0.18