Project 2: Human Face Detection using Boosting

(Venkata Sasank Pagolu, 804945548)

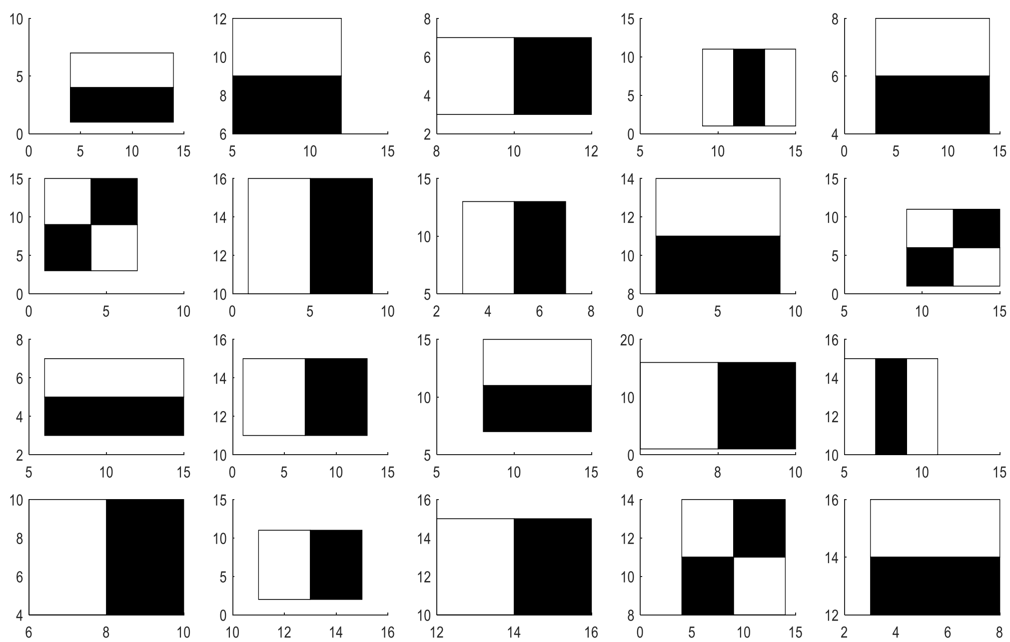
Part-A: Adaboost

1) Haar filters: Display the top 20 Haar fillters after boosting. Report the corresponding voting weights.

Voting Weights of top 20 filters:

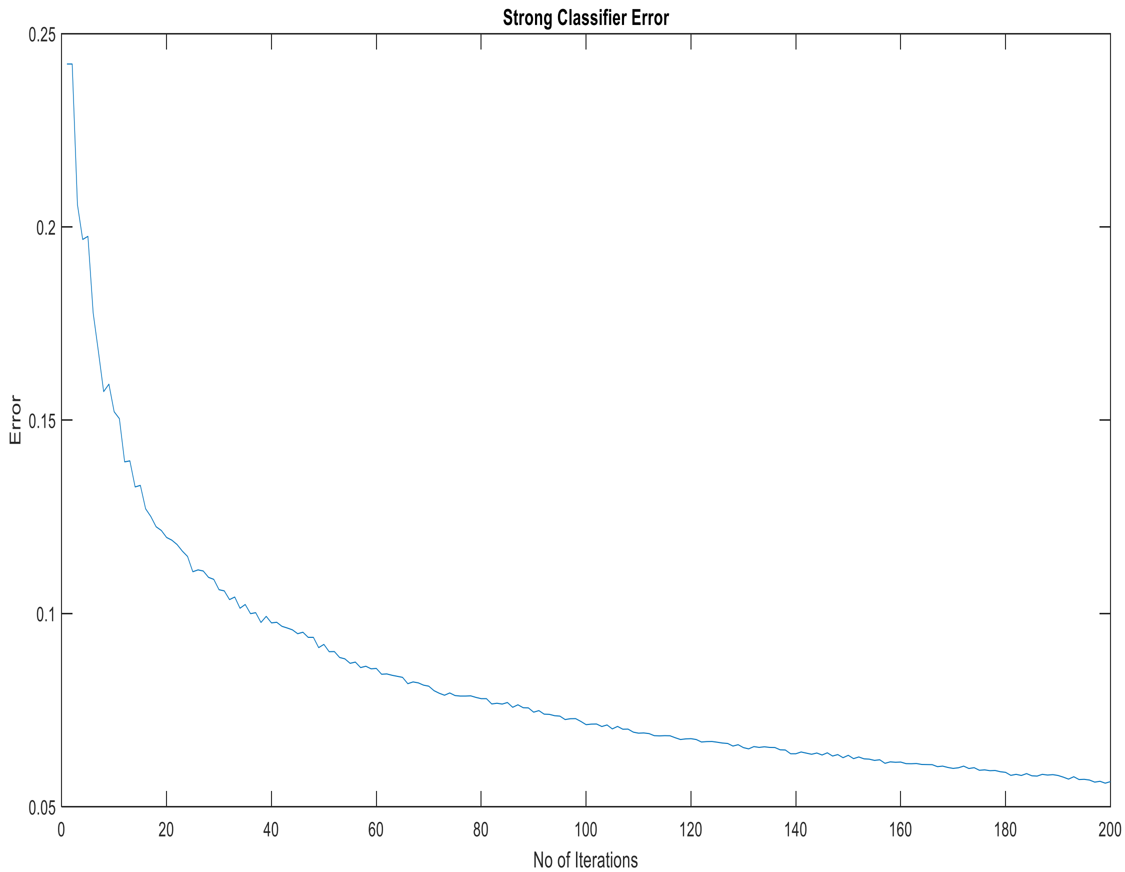
{0.570502666792819, 0.350873684103734, 0.379510809211362, 0.360934500508181, 0.258603888935515, 0.271977044404877, 0.230119079774740, 0.259792603010869, 0.186786348461131, 0.198628205662515, 0.181702548018053, 0.197587740854787, 0.167880641109761, 0.197840946120262, 0.195560949213443, 0.179677589213352, 0.138073438570107, 0.149970689258770, 0.127496344143354, 0.161567779430654}

Top 20 Haar Filters:



When the filter is applied on the image, the sum of the image pixels on the white colored side and sum of the image pixels on the black colored side are subtracted and the result obtained is stored as the feature for the image corresponding to that filter. The main purpose of haar filters is to generate features for the raw images which will be later used in the Adaboost classification for detecting the faces in the images. The top 20 filters achieved from the Adaboost algorithm are displayed above.

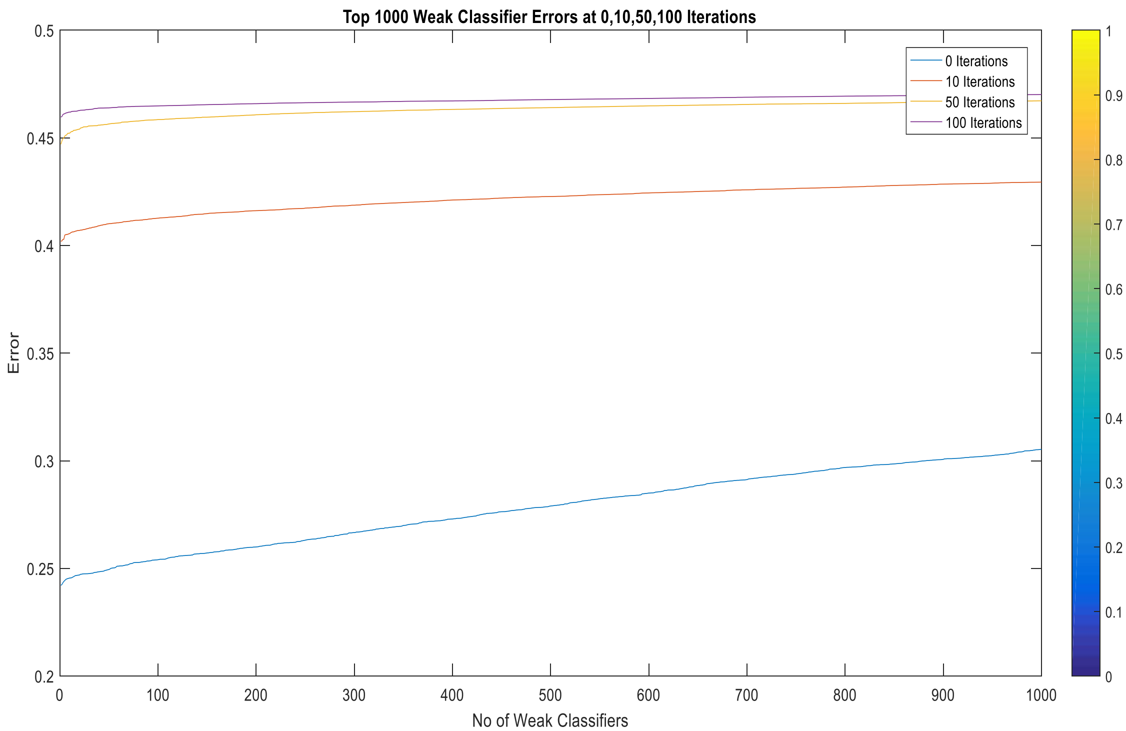
2) **Training error of strong classifier: Plot the training error of the strong classifier over the number of steps T.**



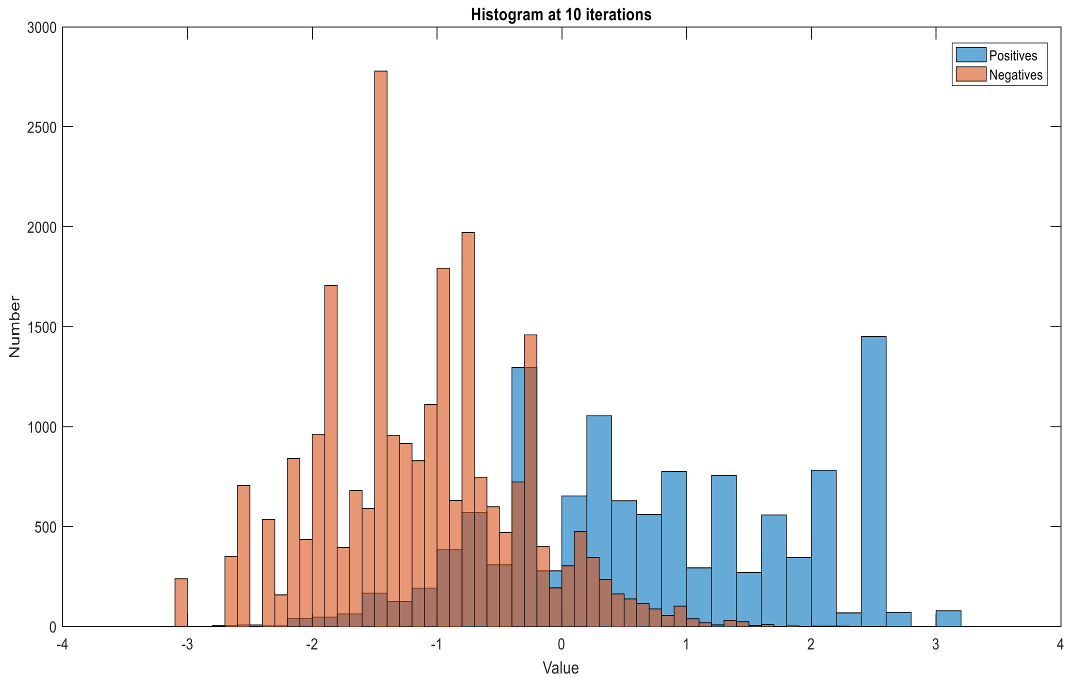
We observe that the strong classifier error decreases as we increase the no of weak classifiers. The strong classifier which is a weighted sum of 200 weak classifiers in this case was able to reduce the final misclassification error to 0.05. At 100 iterations, the error of the strong classifier is approximately 0.07. By the time we choose 200 weak classifiers, the error is minimal which is 0.056.

3) Training errors of weak classifiers: At steps T = 0; 10; 50; 100, plot the curve for the training errors of the top 1000 weak classifiers among the pool of weak classifiers in increasing order. Compare these four curves.

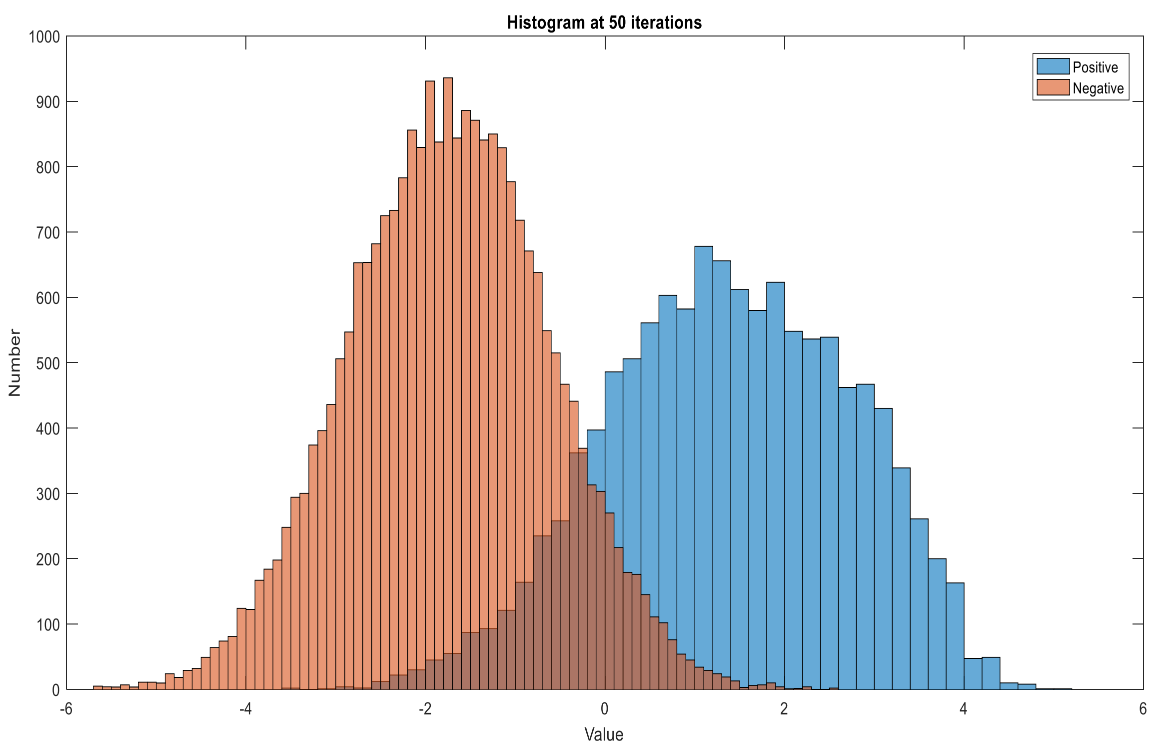
Below the top 1000 weak classifier errors at 0, 10, 50, 100 iterations are plotted using different colors. We observe that the weak classifiers error range increases as we increase the number of iterations. This is because of the fact that during the start, we pick the classifier with the smallest error. In the later iterations, we are picking the classifier with next smallest errors. So, it is reasonable to see a shift in the weak classifier errors as the iterations increase.



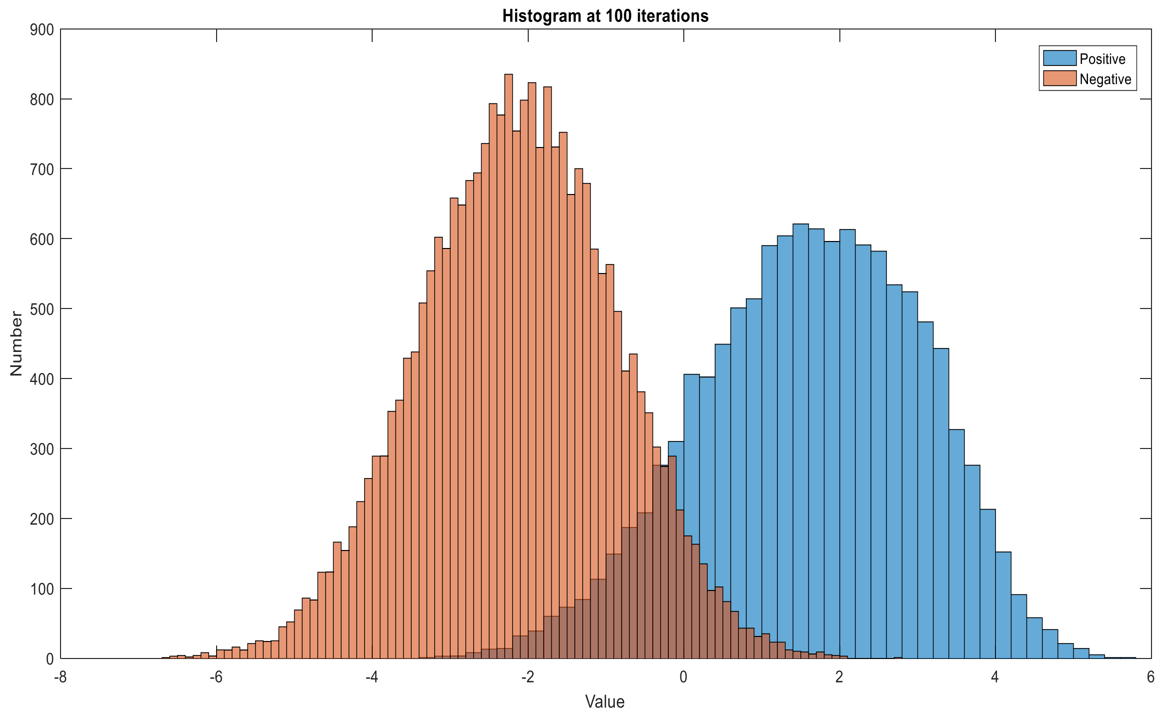
4) Histograms: Plot the histograms of the positive and negative populations over F(x), for T = 10; 50; 100, respectively.



Histogram showing the separation between face image and non-face images is plotted at 10 iterations. The values are not clearly separable. This is because of the fact that our strong classifier which is a weighted combination of 10 weak classifiers is not robust enough. Separability would be good if we take more weak classifiers.

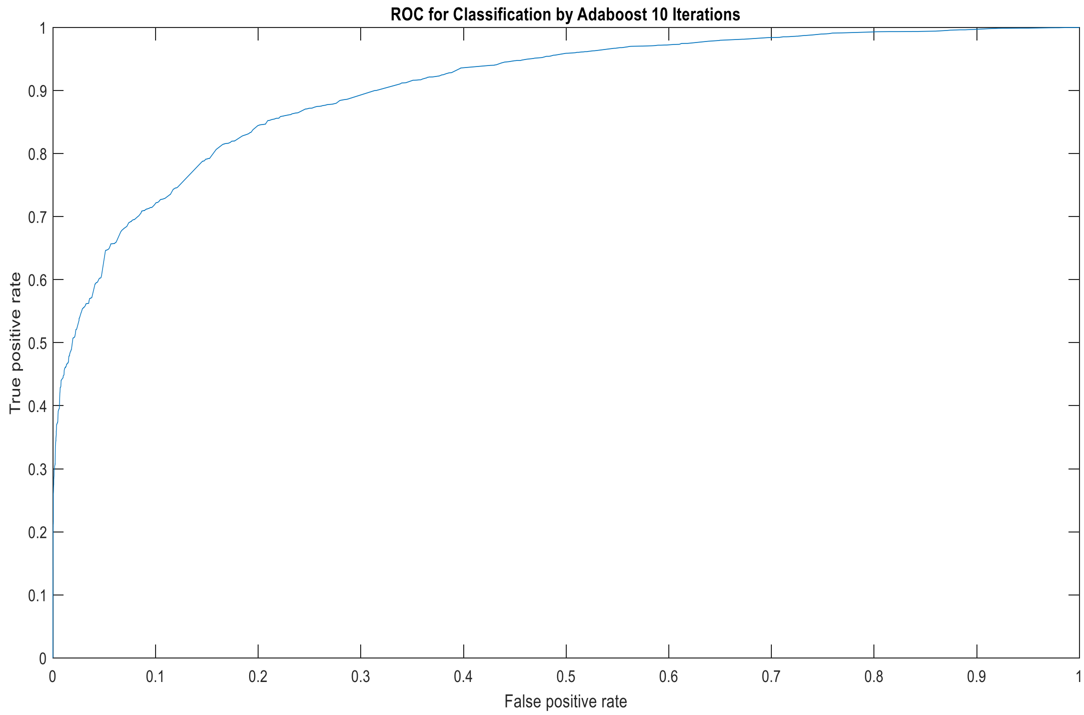


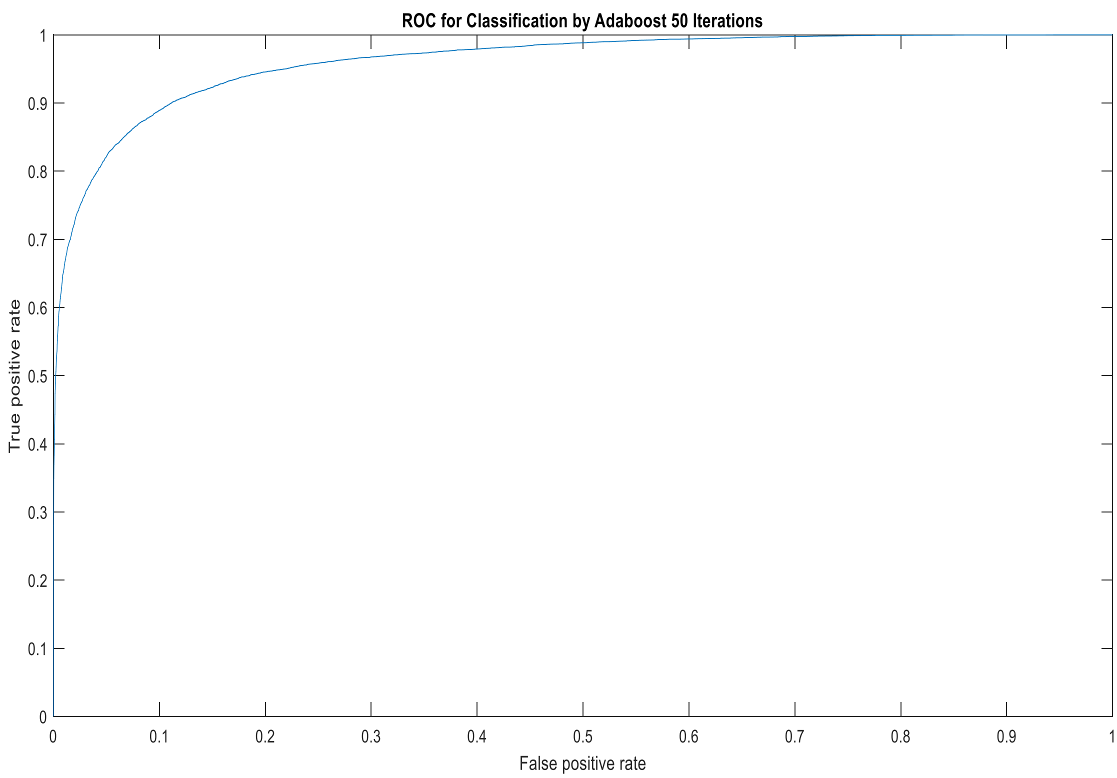
Histogram showing the separation between face image and non-face images is plotted at 50 iterations. The values are not entirely separable. This is because of the fact that our strong classifier which is a weighted combination of 50 weak classifiers is not robust enough. But the better separation is given compared to previous histogram at 10 iterations. Separability would be good if we take more weak classifiers.

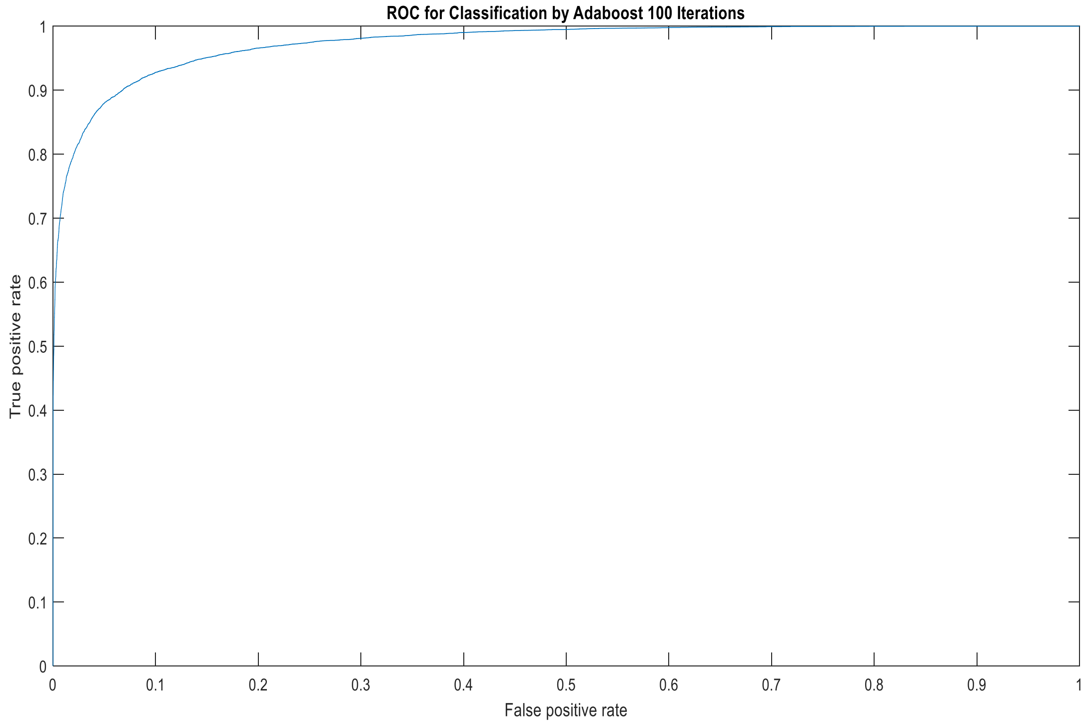


Histogram showing the separation between face image and non-face images is plotted at 100 iterations below. The values are separated reasonably well. This is because of the fact that our strong classifier which is a weighted combination of 100 weak classifiers is robust. The better separation is visible compared to previous histogram at 10 and 50 iterations.

5) ROC: Based on the histograms, plot their corresponding ROC curves for T = 10; 50; 100, respectively.







Roc curves for the strong classifier at 10, 50 and 100 iterations are plotted. We observe that the area under the curve is larger at 100 iterations. This indicates that the classifier performs more accurately and reduces false positive detection as we increase the number of iterations and take more weak classifiers weighted combinations into the strong classifier.

6) Detections: Display the detected faces in both of the provided images without hard negative mining.

Considered 10 scales for the image throughout the project: 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0.

The below image shows the detected faces by the strong classifier. I have used all the scales for detection first and removed the overlaps between different scales by saving only the rectangle with maximum value outputted by the strong classifier. This procedure is followed for all the 3 test images and the results are shown below.

Test Image 1 at all Scales

Test Image 1 at Scale 1.5

Test Image 1 at Scale 3.5

Test Image 1 at Scale 5.5

Test Image 2 at all Scales

Test Image 3 at all Scales

We can see that there are false positives in the images. Ceiling, table and non face images are detected as faces sometimes. To eliminate this problem, negative mining is performed on the images.

7) Hard negative mining: Display the detected faces in both of the provided images with hard negative mining.

The obtained strong classifier from Adaboost is applied on the test image which has turned heads of students. All the detected images on all scales are treated as false positives. Those images were extracted and the Adaboost is retrained on top of those additional negative images. The weak classifiers and their corresponding weights change due to the addition of new features from negative faces. After obtaining the new strong classifier, the faces in the test images were detected.

After Hard Negative Mining, Test Image 1

We can see that lot of false positives are removed after performing the hard negative mining. Compared to the faces detected before hard negative mining, the false positives are significantly reduced. The false positives count in the image dropped to a single digit. This shows the effectiveness of the hard negative mining task.

After Hard Negative Mining, Test Image 2 at all Scales

We can see that lot of false positives are removed after performing the hard negative mining. Compared to the faces detected before hard negative mining, the false positives are significantly reduced. The false positives count in the image dropped to a single digit. This shows the effectiveness of the hard negative mining task.

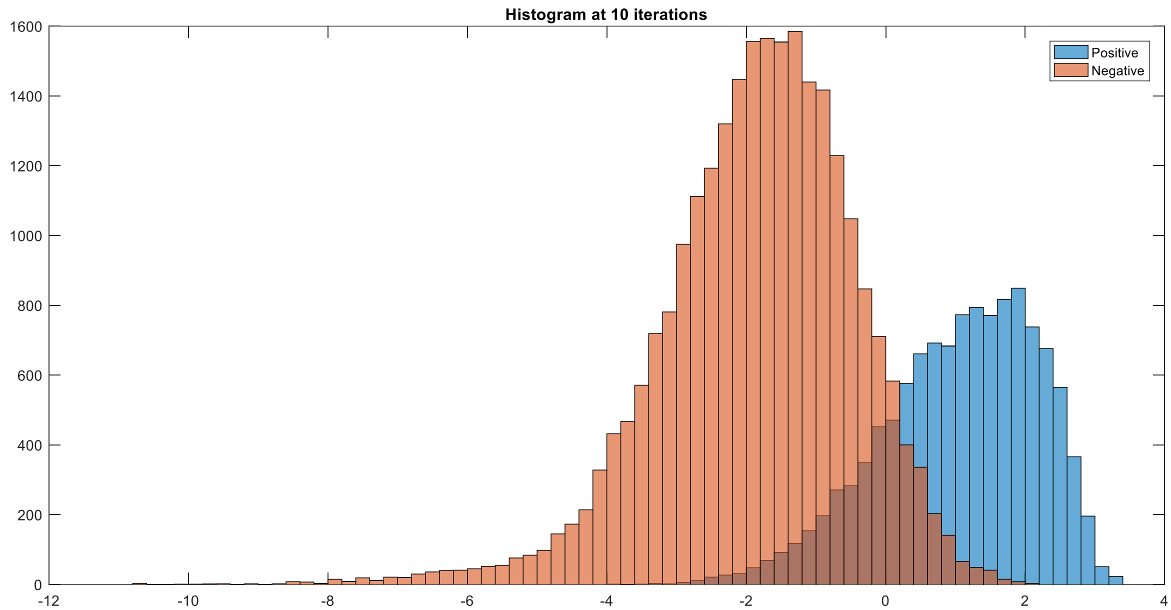
After Hard Negative Mining, Test Image 3 at all Scales

We can see that lot of false positives are removed after performing the hard negative mining. Compared to the faces detected before hard negative mining, the false positives are significantly reduced. The false positives count in the image dropped to a single digit. This shows the effectiveness of the hard negative mining task.

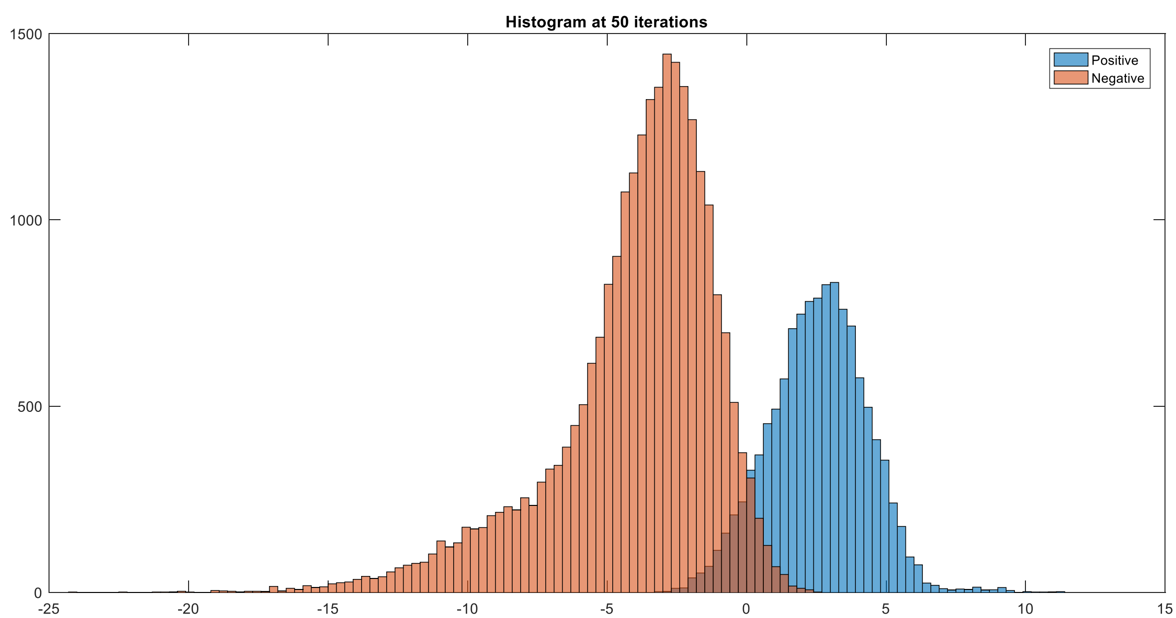
Part B – RealBoost

1) Histograms: Plot the histograms of the positive and negative populations over F(x), for T = 10; 50; 100, respectively.

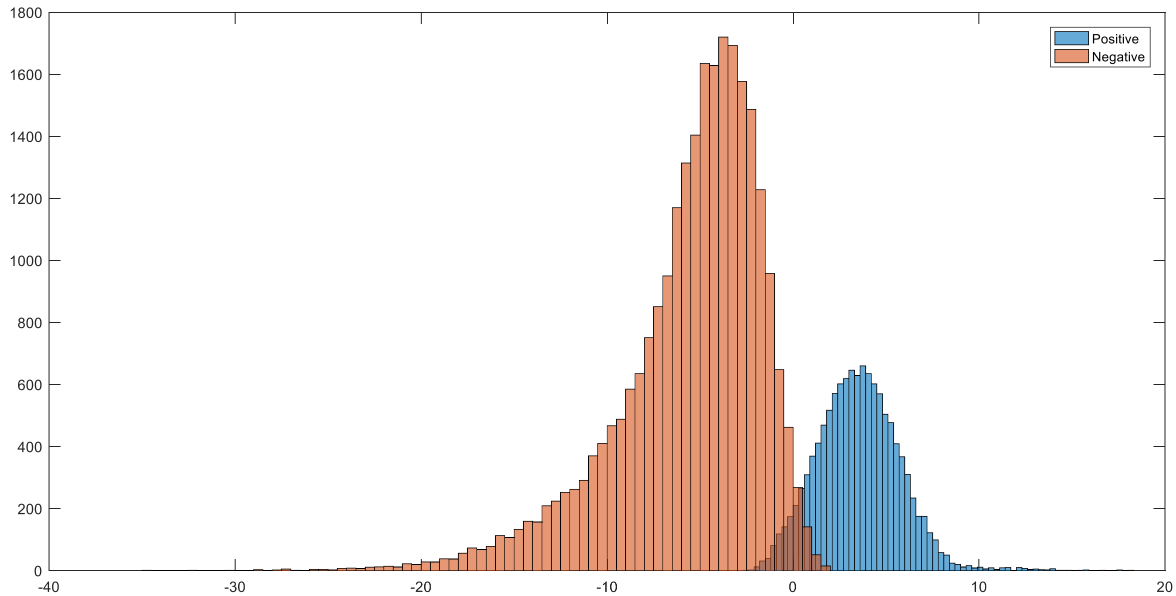
No of bins taken is 20. Ran for 200 iterations. The final error is 0.0099.



Histogram showing the separation between face image and non-face images is plotted at 10 iterations. The values are not clearly separable. This is because of the fact that our strong classifier which is a weighted combination of 10 weak classifiers is not robust enough. Separability would be good if we take more weak classifiers. The results and separation enhanced compared to Adaboost.



Histogram showing the separation between face image and non-face images is plotted at 50 iterations. The values are not entirely separable. This is because of the fact that our strong classifier which is a weighted combination of 50 weak classifiers is not robust enough. But the better separation is given compared to previous histogram at 10 iterations. Separability would be good if we take more weak classifiers. The results and separation enhanced compared to Adaboost.

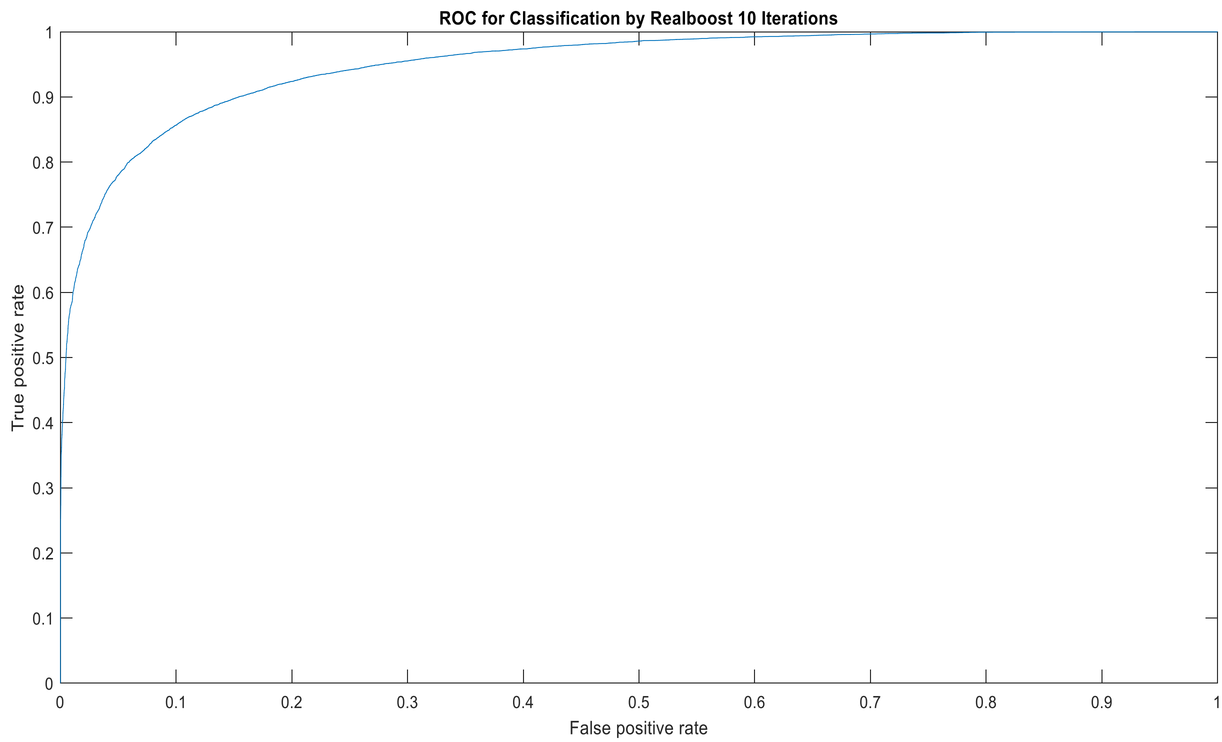


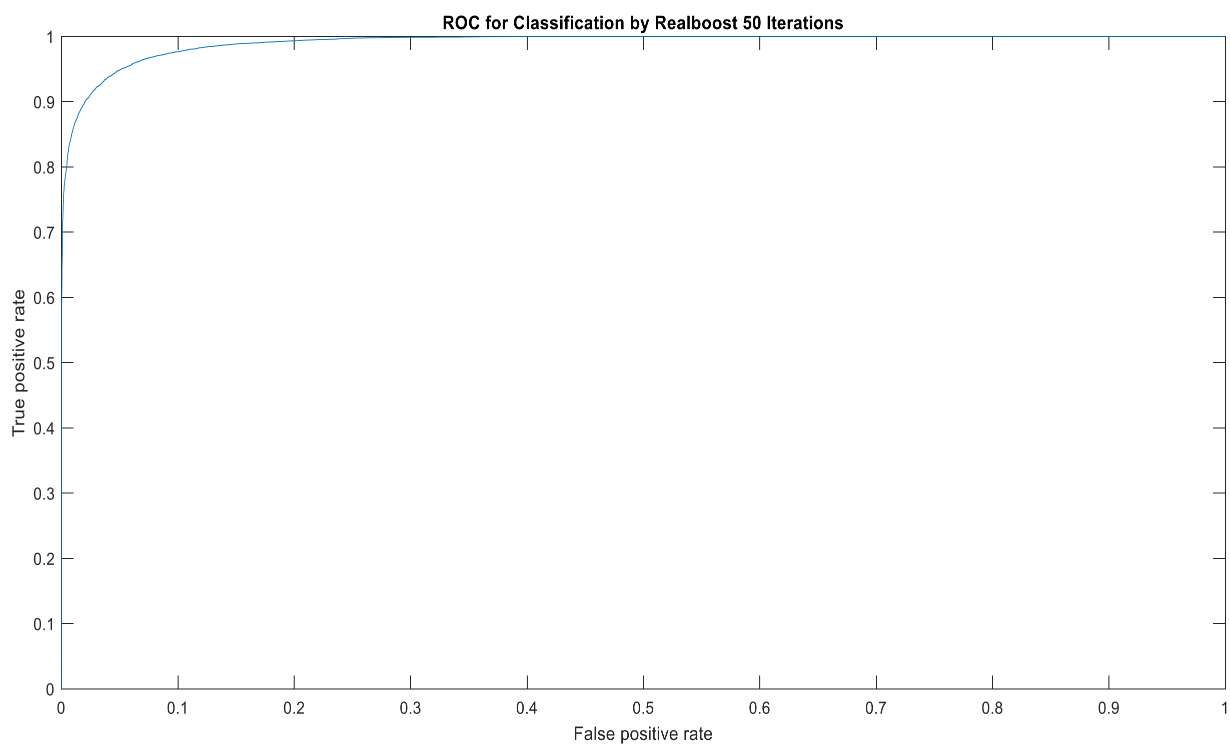
Histogram showing the separation between face image and non-face images is plotted at 100 iterations below. The values are separated reasonably well. This is because of the fact that our strong classifier which is a weighted combination of 100 weak classifiers is robust. The better separation is visible compared to previous histogram at 10 and 50 iterations. The results and separation enhanced compared to Adaboost.

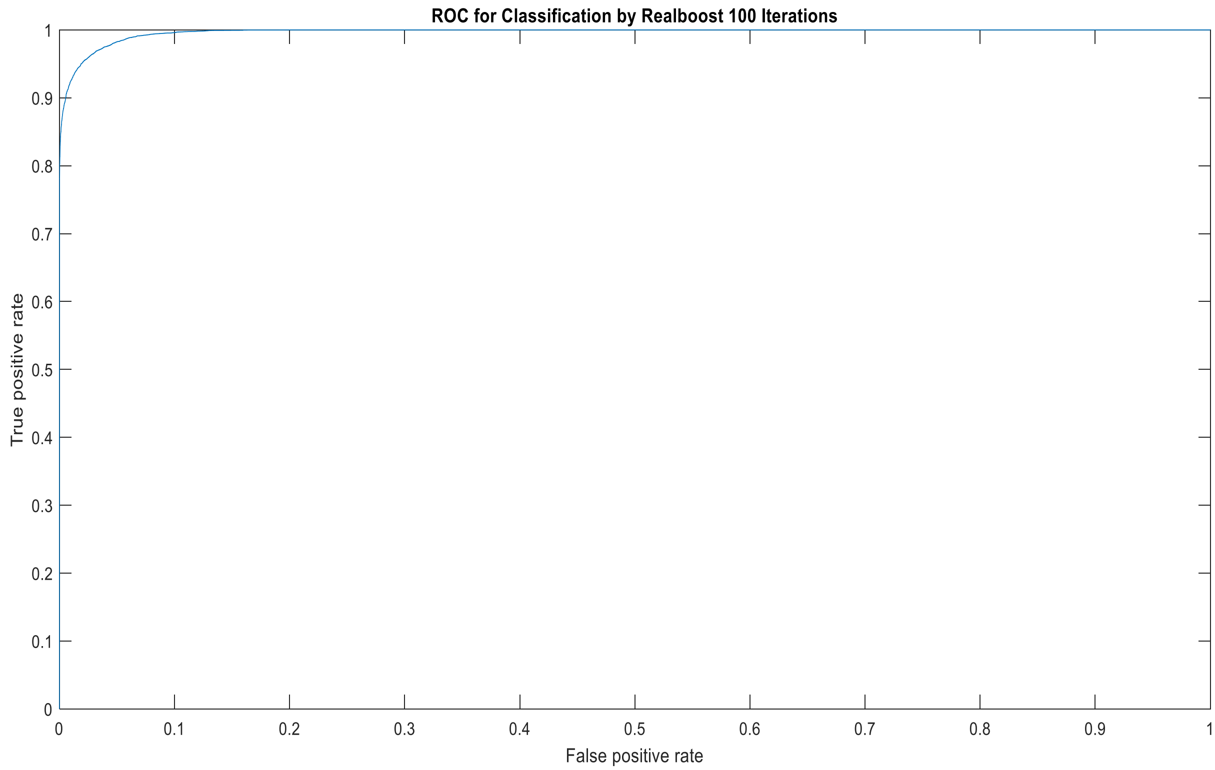
2) ROC: Based on the histograms, plot their corresponding ROC curves. Compare them with the ROC curves in (e).

Roc curves for the strong classifier at 10, 50 and 100 iterations are plotted. We observe that the area under the curve is larger at 100 iterations. This indicates that the classifier performs more accurately and reduces false positive detection as we increase the number of iterations and take more weak classifiers weighted combinations into the strong classifier.

Roc curves obtained from Realboost are significantly outperforming the curves obtained from Adaboost. AUC (Area under the curve) is higher for plots obtained from Realboost. This shows that the Realboost is more capable of detecting true positives and eliminating false positives.


ROC for Classification by Realboost 10 Iterations
True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC for Classification by Realboost 50 Iterations
True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



RealBoost decreasing Error over iterations from 1 to 200. At t=200, error= 0.0099