Project 2: Human Face Detection using Boosting

Yufei Hu (404944367)

November 10, 2017

1 Objectives

Boosting is a general method for improving the accuracy of any given learning algorithm. Specifically, one can use it to combine weak learners, each performing only slightly better than random guess, to form an arbitrarily good hypothesis. In this project, you are required to implement an AdaBoost and RealBoost algorithms for frontal human face detection.

2 Tasks

Be aware: the number of positive training samples is 11838; and the number of negative training samples is 35356 in my codes for both AdaBoost (without hard negative mining) and RealBoost. Besides, I did not reduce the number of filters in both cases.

2.1 Construct weak classifiers $\{h_j\}$

Load the predefined set of Haar filters. Compute the features $\{f_j\}$ by applying each Haar filter to the integral images of the positive and negative populations. Determine the polarity s_j and threshold θ_j for each weak classifier h_j:

$$h_j(x) = \begin{cases} -s_j & \text{if } f_j < \theta_j \\ s_j & \text{otherwise} \end{cases} \tag{1}$$

Write a function which returns the weak classifier with lowest weighted error. Note, as the samples change their weights over time, the histograms and threshold θ will change.

Q (a): Haar filters

Display the top 20 Haar filters after boosting. Report the corresponding voting weights $\{\alpha_t: t = 1, \ldots, 20\}$.

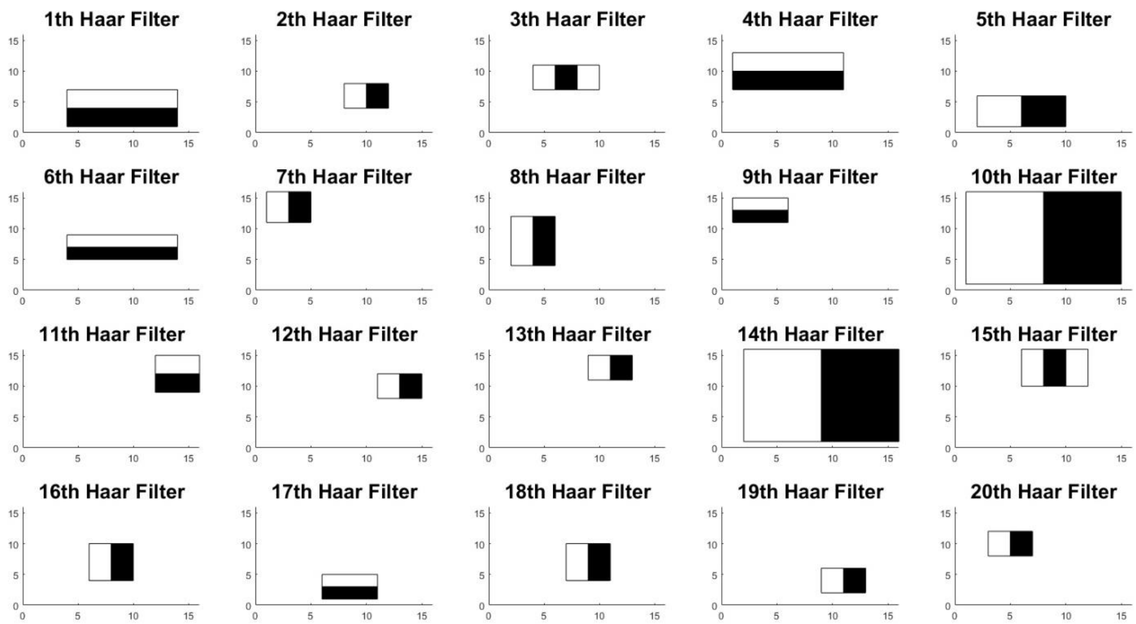

Figure 1: Top 20 Haar filters selected after boosting

Table 1: Voting weights of top 20 Haar filters selected after boosting

$\alpha_1 = 0.71093$	$\alpha_6 = 0.25207$	$\alpha_{11} = 0.23764$	$\alpha_{16} = 0.19238$
$\alpha_2 = 0.44505$	$\alpha_7 = 0.26459$	$\alpha_{12} = 0.20164$	$\alpha_{17} = 0.18339$
$\alpha_3 = 0.40212$	$\alpha_8 = 0.24448$	$\alpha_{13} = 0.19236$	$\alpha_{18} = 0.16933$
$\alpha_4 = 0.33781$	$\alpha_9 = 0.23296$	$\alpha_{14} = 0.20143$	$\alpha_{19} = 0.17858$
$\alpha_5 = 0.29271$	$\alpha_{10} = 0.23009$	$\alpha_{15} = 0.22118$	$\alpha_{20} = 0.15356$

The selected top 20 Haar filters are shown in Figure 1. It is interesting to see that the shapes of some selected filters are close to certain facial features. For example, it seems that the first selected Haar filter is targeted for detecting mouths. Thus, the selection is quite convincing.

2.2 Implement AdaBoost

Implement the AdaBoost algorithm to boost the weak classifiers. Construct the strong classifier H(x) as an weighted ensemble of T weak classifiers:

$$H(x) = sign(F(x)) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \quad (2)$$

Two class photos are given. Note, you need to scale the image into a few scales so that the faces at the front and back are 16×16 pixels in one of the scaled image. Run your classifier on these images. Perform non-maximum suppression, i.e. when two positive detections overlap significantly, choose the one that has higher score. Perform hard negatives mining. You are given background images without faces. Run your strong classifier on these images. Any "faces" detected by your classifier are called "hard negatives". Add them to the negative population in the training set and re-train your model. Include the following in your report:

Q (b): Training error of strong classifier

Plot the training error of the strong classifier over the number of steps T.

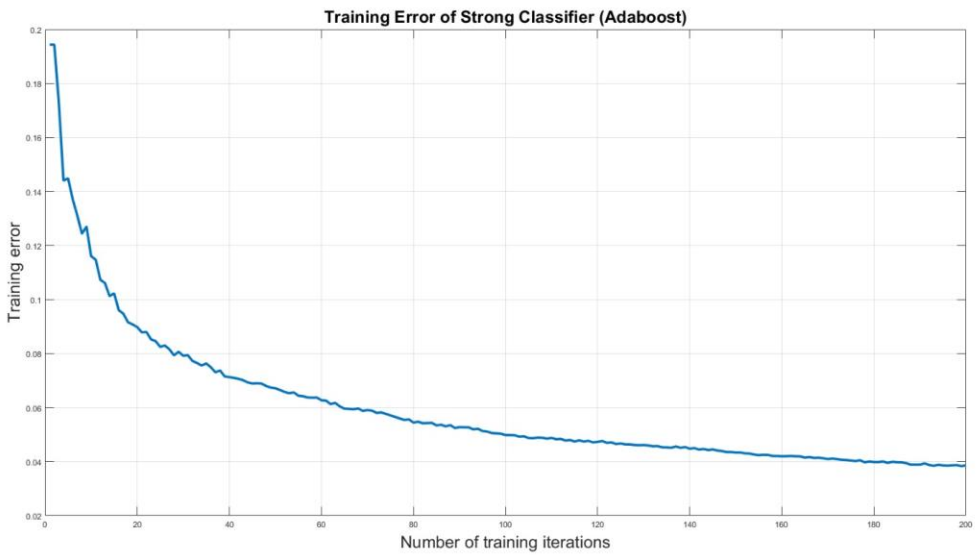

Figure 2: Training errors of the strong classifier (AdaBoost)

It is reasonable to see that the training error decreases way down to somewhere around 4%. According to the slope of the curve at latter iterations in Figure 2, training with more than 200 iterations seems to be unnecessary considering the minor improvements it can bring.

Q (c): Training error of weak classifiers

At steps T = 0, 10, 50, 100, 200, plot the curve for the training errors of the top 1,000 weak classifiers among the pool of weak classifiers in increasing order. Compare these four curves.

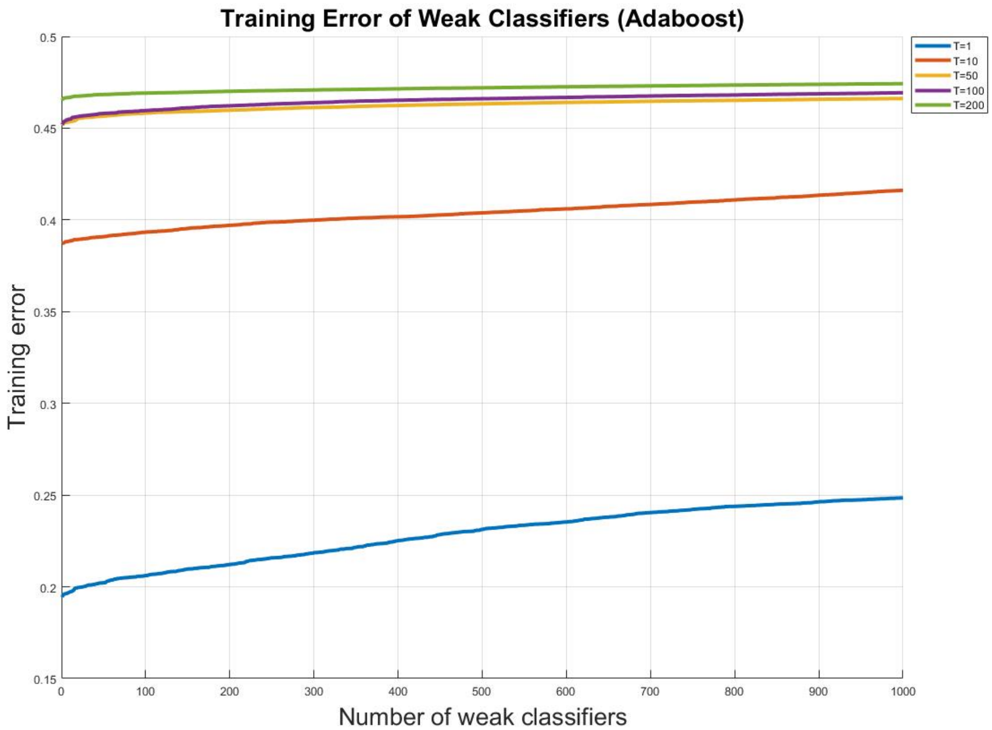

Figure 3: Training errors of weak classifiers (AdaBoost)

According to Figure 3, the more iterations the training goes, the higher the training errors are for the top 1000 weak classifiers. The reason is because the weights for these wrongly classified data keeps increasing with more iterations go on. At latter stages of training, the errors for all the top 1000 weak classifiers almost approach 50%. This phenomenon explains why the improvement of performance of the strong classifier slows down after hundreds of iterations as all weak classifiers can no longer shrink the errors for those magnified wrongly classified data.

Q (d): Histograms

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

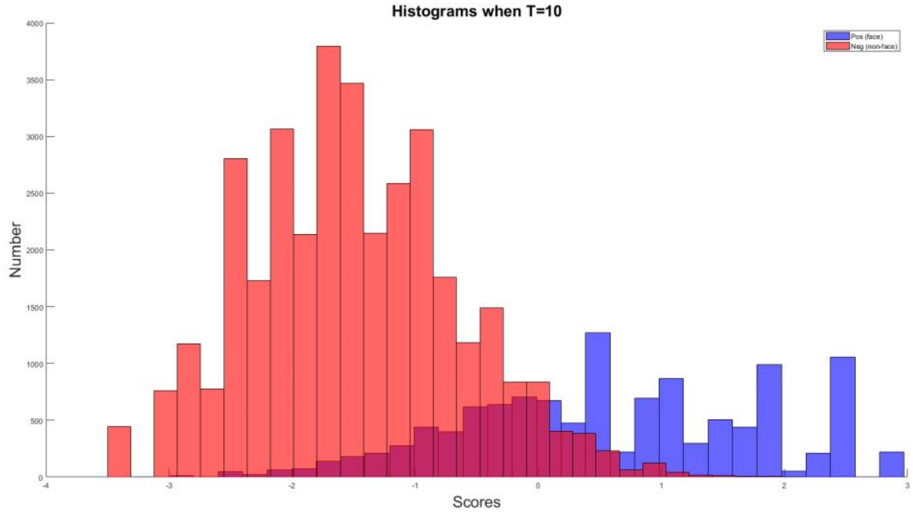

Figure 4: Histograms of the positive and negative populations at T=10

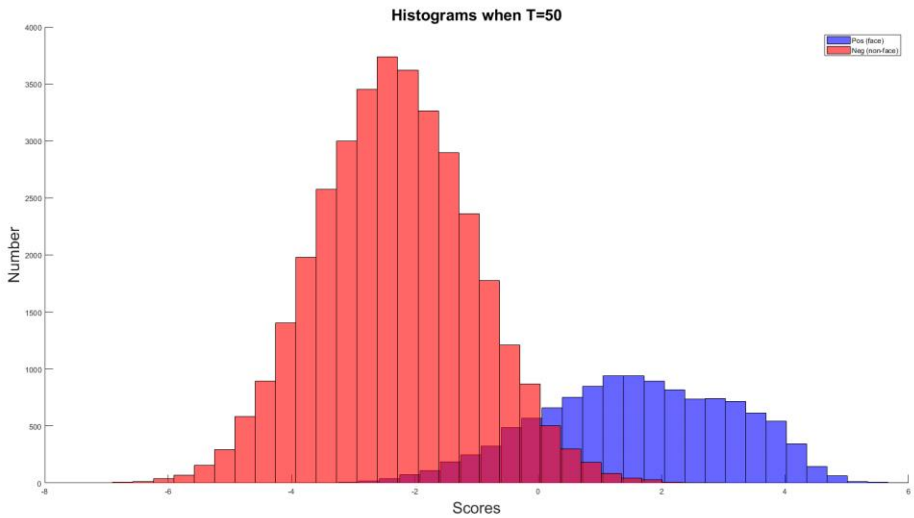

Figure 5: Histograms of the positive and negative populations at T=50

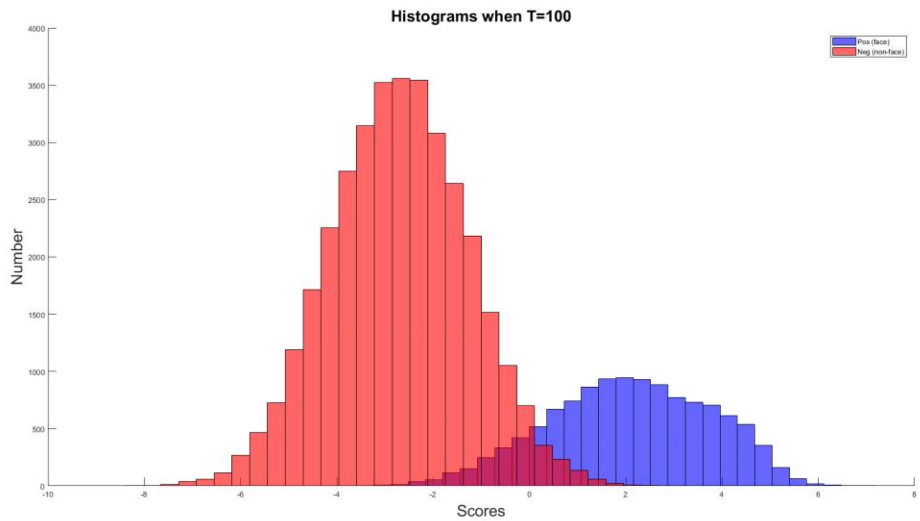

Figure 6: Histograms of the positive and negative populations at T=100

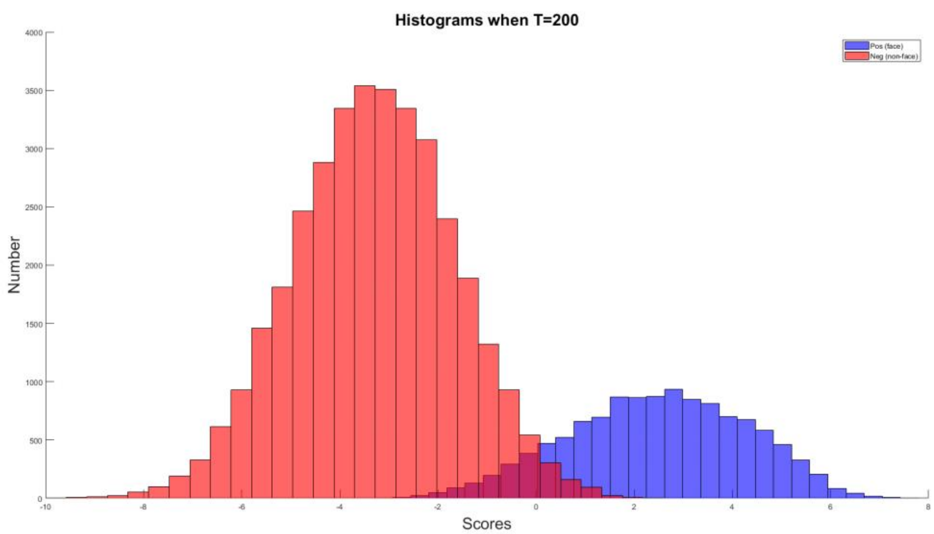

Figure 7: Histograms of the positive and negative populations at T=200

It can be seen that the positive and negative populations are better distinguished at the 200th iteration by comparing to the 10th iteration. This phenomenon well fits the fact that the training error of the strong classifier keeps increasing with more iterations go on.

Q (e): ROC

Based on the histograms, plot their corresponding ROC curves for T = 10; 50; 100; 200, respectively.

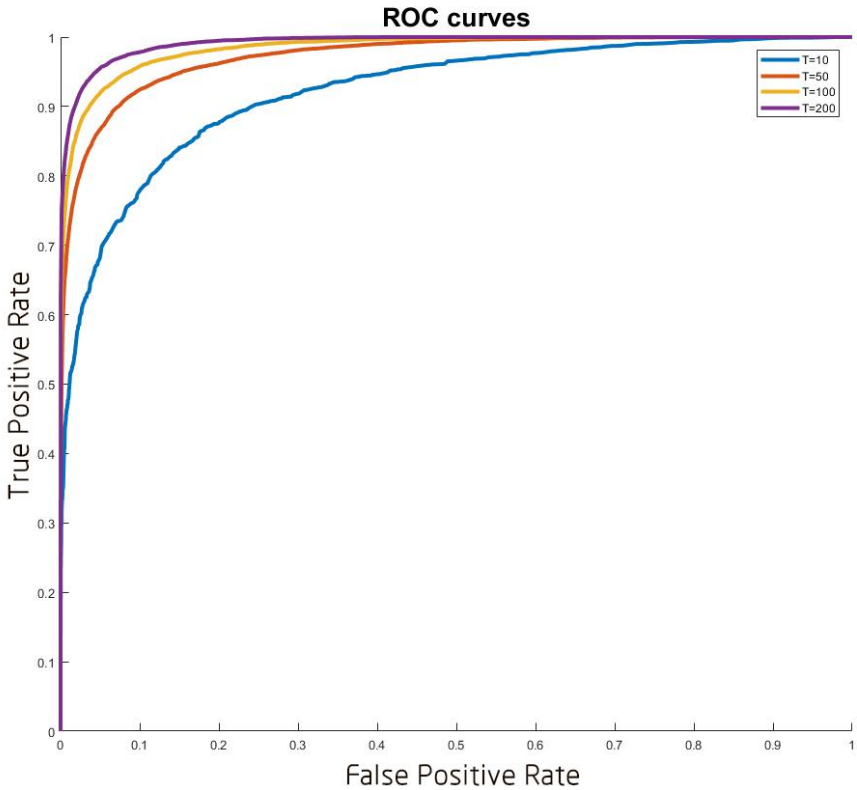

Figure 8: ROC curves of the strong classifier

According to Figure 8, the ROC curve goes to up-left corner with more iterations go on. It means that the strong classifier at latter stage achieves a better classification performance by comparing to previous stages as the true positive rate increases while the false positive rate decreases. This fits the phenomenon depicted in Figure 3 - 7 quite well. However, the ROC curve at the 200^{th} iteration seems to have approached a limit, indicating a performance upper bound in this case.

Q (f): Detections

Display the detected faces in all of the provided images without hard negative mining.

Figure 9: Detection results of the first testing image

Figure 10: Detection results of the second testing image

Figure 11: Detection results of the third testing image

In Figure 9 - 11, the red bounding boxes refer to those who have a classification scores bigger than 0.8 but less than 1.7. The yellow bounding boxes refer to those who have a classification scores bigger than 1.7 but less than 1.8. The green bounding boxes are those who have a classification scores bigger than 1.8, indicating the strong classifier is quite confident that the input patch is a face.

One phenomenon is that some non-face areas are mistakenly detected as face with a high classification scores. Some mistakes are quite reasonable as they indeed look like faces, others appear to be a little unreasonable. But it is shown in the rest of this report that most of those mistakes can be corrected after conducting hard-negative mining.

Q (g): Hard negative mining

Display the detected faces in all of the provided images with hard negative mining.

Figure 12: Detection results of the first testing image after hard negative mining

Figure 13: Detection results of the second testing image after hard negative mining

Figure 14: Detection results of the third testing image after hard negative mining

As you can tell from Figure 12 - 14, the number of false faces indeed decreases after conducting hard negative mining. However, the strong classifier becomes more uncertain to some faces as the bounding boxes go red. In Figure 14, these is still an seemingly unreasonable mistake at the bottom-right corner, but it is quite negligible comparing to the overall performance improvement.

2.3 Implement RealBoost

Implement the RealBoost. Compute the histograms of negative and positive populations and the corresponding ROC curves.

Q (h): Histograms

Plot the histograms of the positive and negative populations over F(x), for T = 10; 50; 100; 200, respectively.

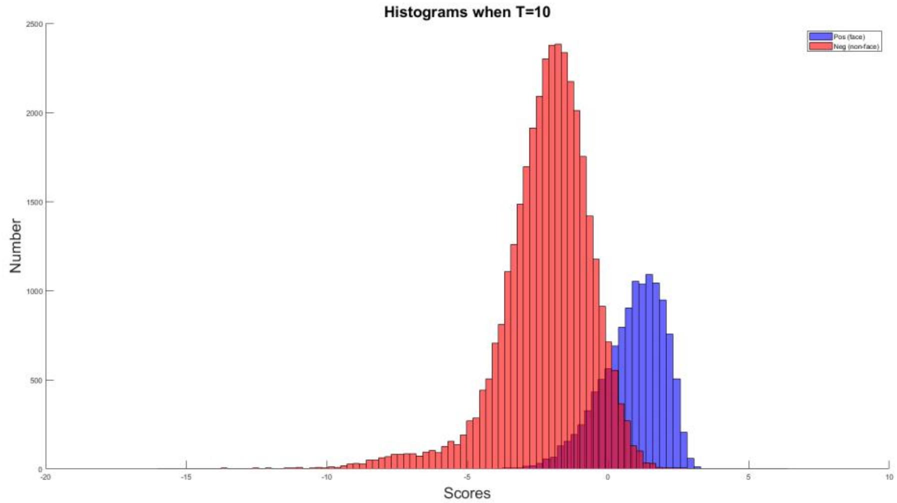

Figure 15: Histograms of the positive and negative populations at T=10 (RealBoost)

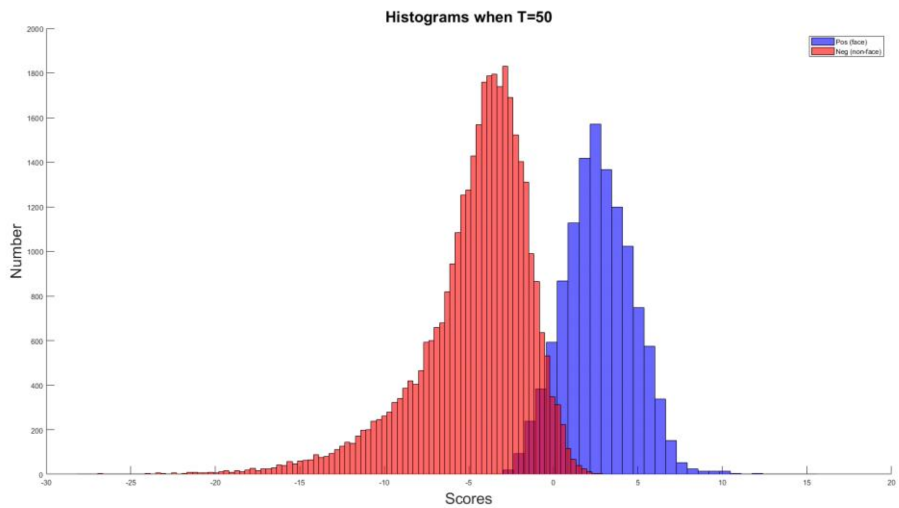

Figure 16: Histograms of the positive and negative populations at T=50 (RealBoost)

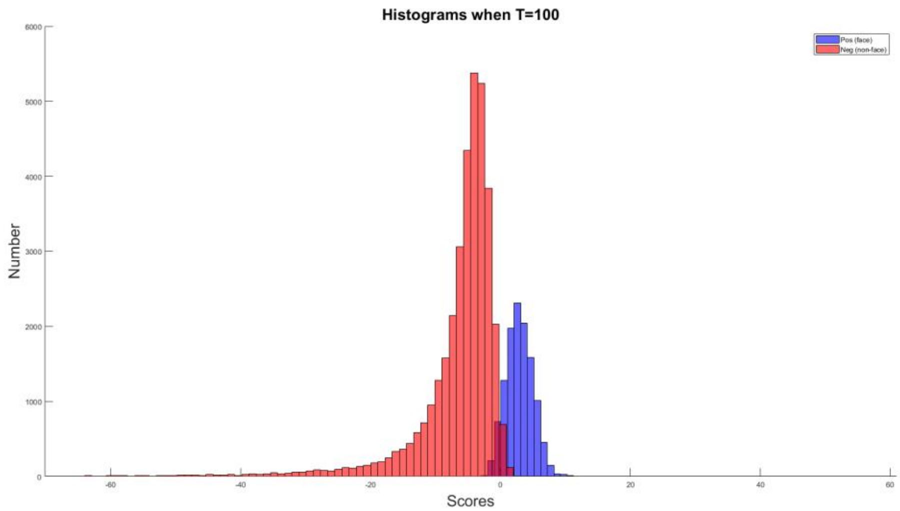

Figure 17: Histograms of the positive and negative populations at T=100 (RealBoost)

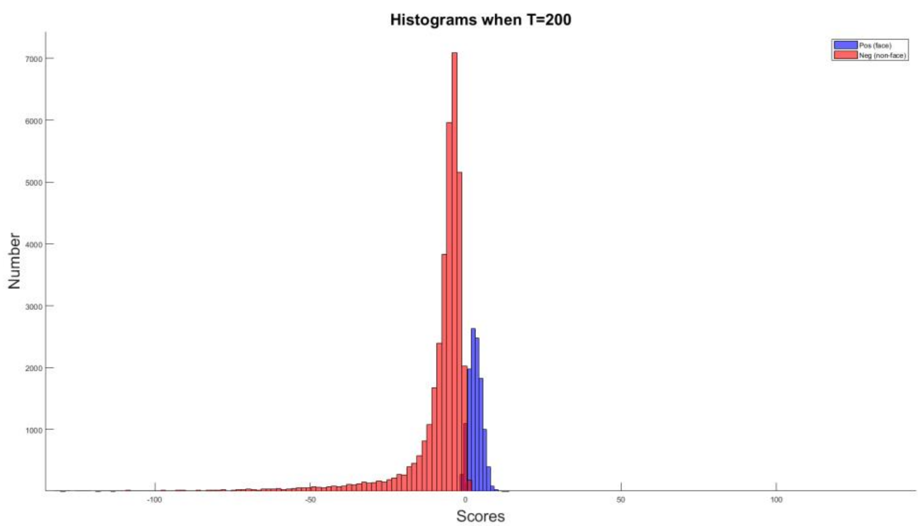

Figure 18: Histograms of the positive and negative populations at T=200 (RealBoost)

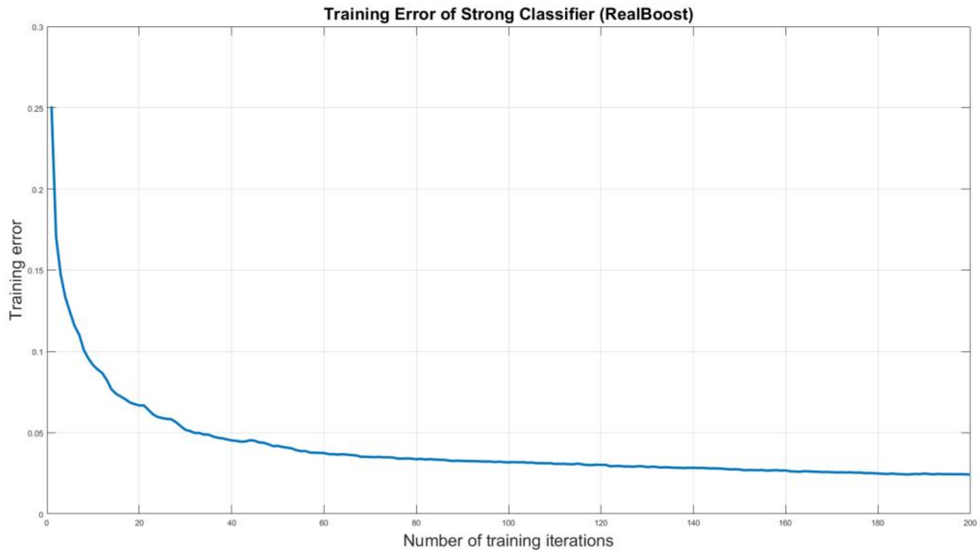

Figure 19: Training error of the strong classifier (RealBoost)

It is surprised to see that RealBoost achieves a much better classification performance as the training error approaches 2.5% comparing to the 4% achieved by AdaBoost previously. This improvement becomes more straightforward by looking at the well-distinguished positive and negative populations in Figure 15 - 19. The shapes of both populations are more pointed comparing to AdaBoost, indicating a smaller variance and a bigger classification confidence.

Q (i): ROC

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

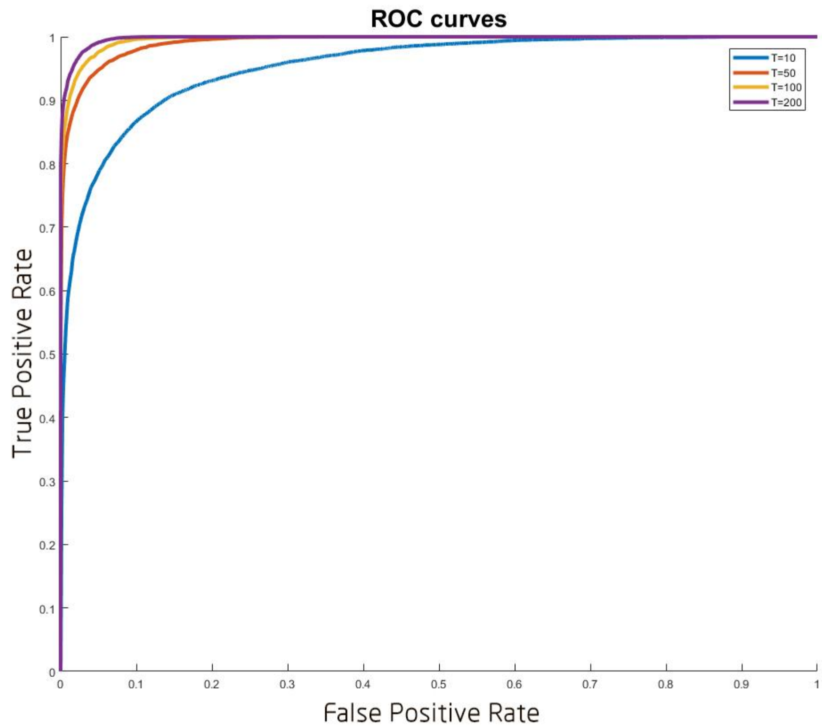

Figure 20: ROC curves of the strong classifier (RealBoost)

By comparing Figure 8 which is the ROC curves of the strong classifier of AdaBoost, the curves in Figure 20 are much closer to the top-left corner at all training stages. This indicates that RealBoost indeed has a better classification performance comparing to AdaBoost.