Kimberly Insigne

Project 2: Human face detection by Boosting techniques

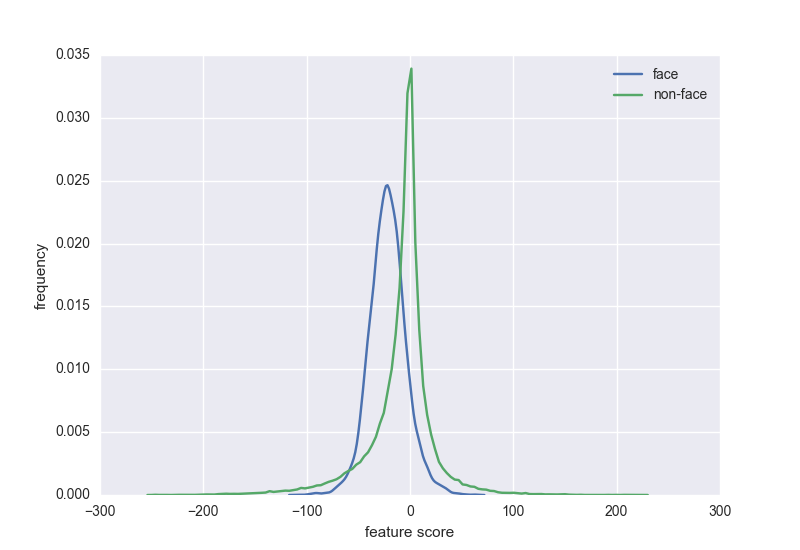
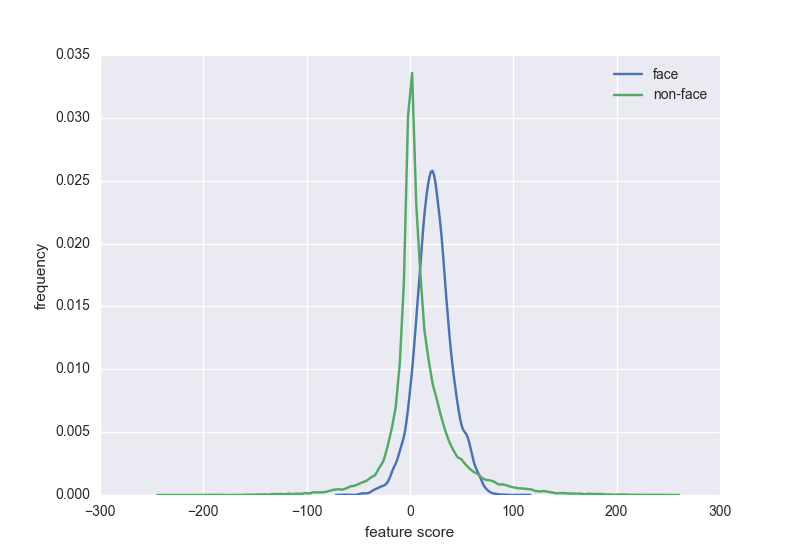
In this project I implemented face detection using thousands of weak Haar classifiers, boosted them with Adaboost and Realboost to train a strong classifier. I used this strong classifier to detect faces (with hard negative mining and non-maximum suppression) on two test images of our class. We are provided with a training set of ~11,000 positive and ~45,000 negative 16x16 training images and three background images of our test images containing no faces.

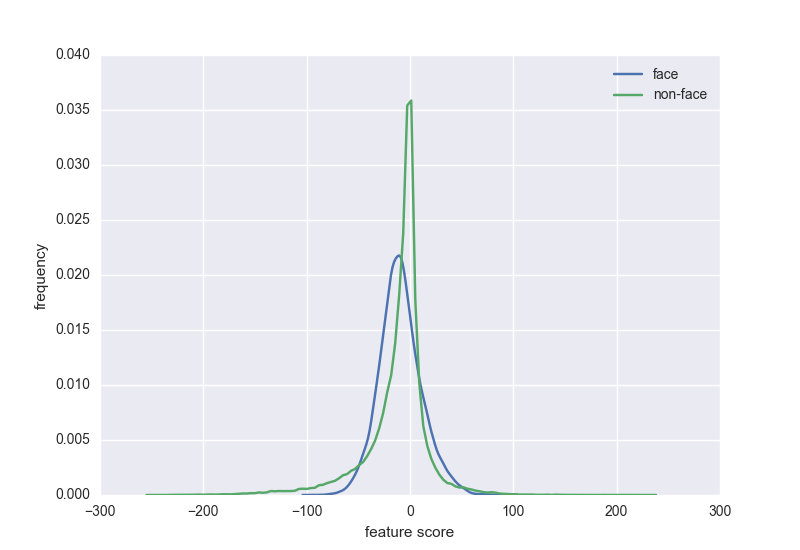
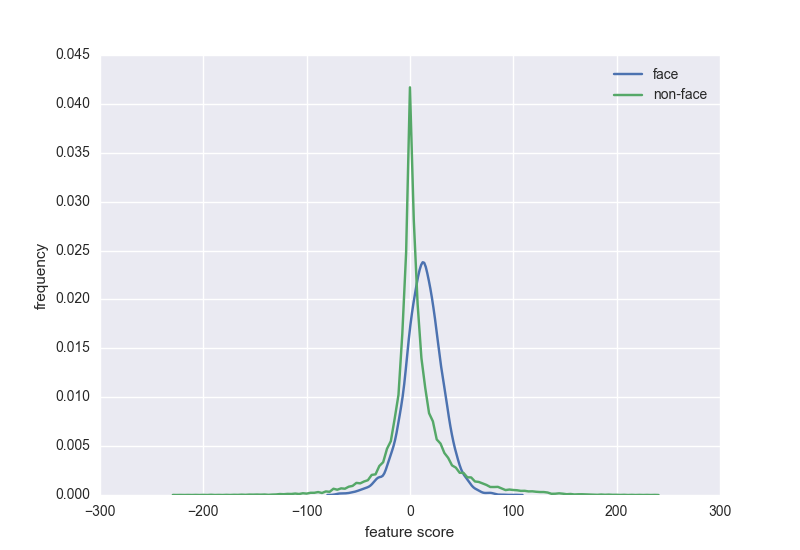
1. **Construction of weak classifiers**

Throughout the entire project I used Haar features (described in the Viola and Jones paper) as weak classifiers. Four classes of features were used (shown below), and for each class its respective shape was moved to every possible position in the 16x16 training images. The value for each feature is the sum of pixel intensities in the dark section minus the sum of pixel intensities in the white section. For each feature class, the inverse arrangement was calculated as well, for example for Class I there are features with the dark section on the left and those with the dark section on the right.

|  |  |  |  |
| --- | --- | --- | --- |
| Class I: 2 vertical rectangles side by side | Class II: 2 horizontal rectangles on top of each other | Class III: 3 vertical rectangles side by side | Class IV: a 2x2 grid of rectangles |
|  |  |  |  |

Shown below are the performance of a few weak classifiers that are more discriminative between positive and negative training images.





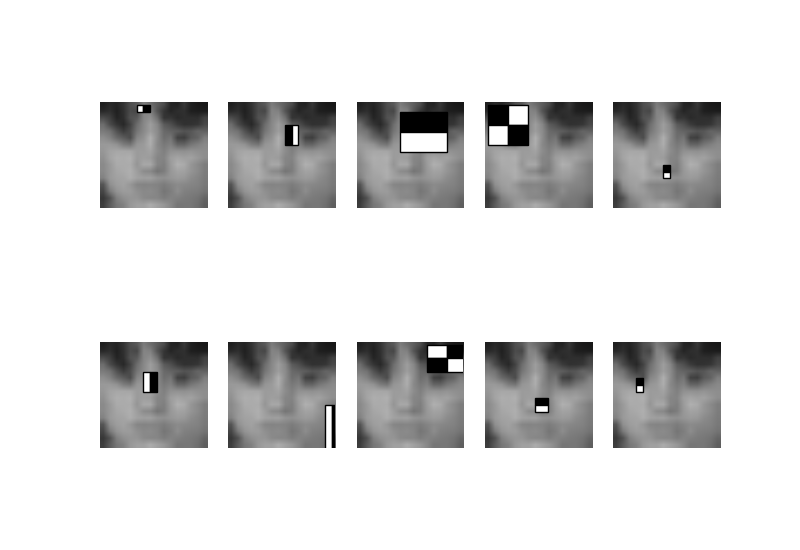
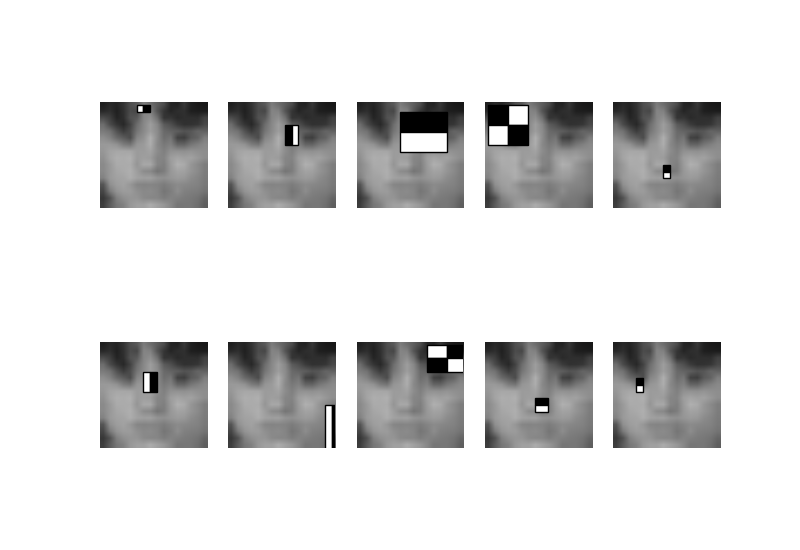
**Figure 1.** Examples of weak classifier performance

This figure shows weak classifiers with poor separation between faces and non-faces, but boosting will improve the performance of these weak classifiers.

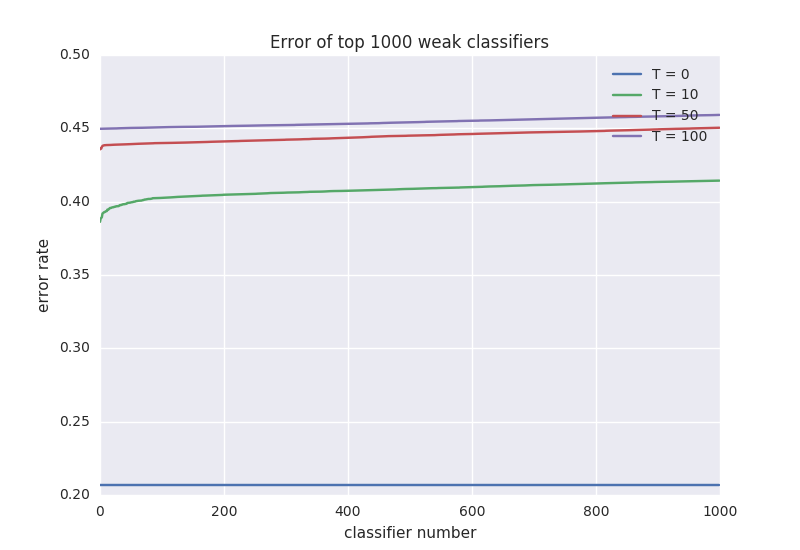
1. **Adaboost**

I trained the set of weak classifiers constructed in Part 1 for 100 iterations using Adaboost, resulting in a strong classifier with 100 features.

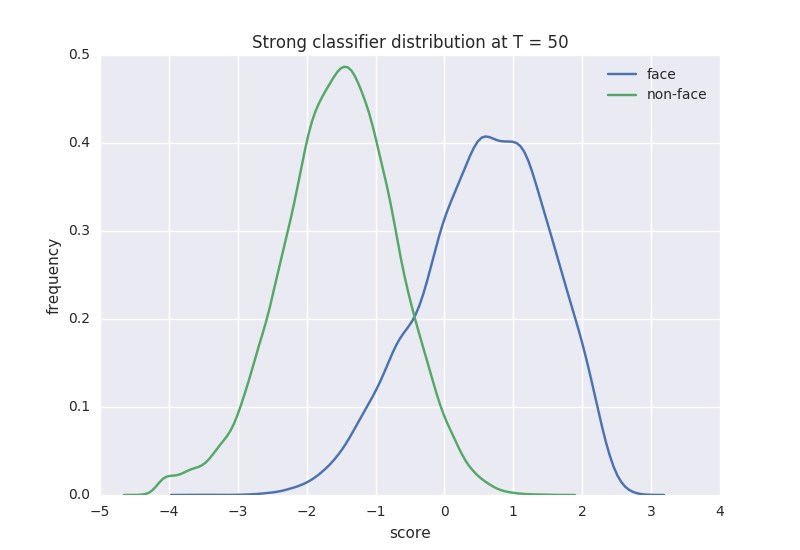
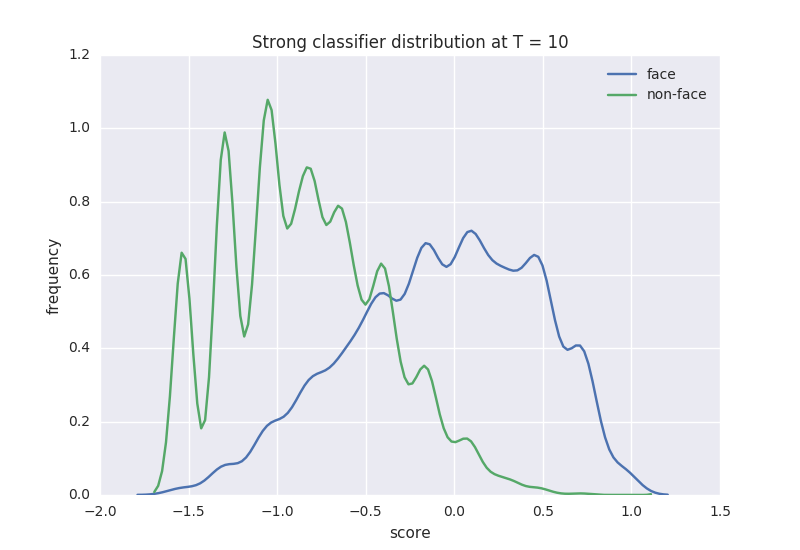
1. Shown below are the top 10 features with the highest weights in the final strong classifier, in order from left to right and top to bottom. The features are displayed at their corresponding positions on top of a sample training face.



1. Shown below are the training errors of the remaining top 1000 weak classifiers at different iterations of Adaboost training.



1. Shown below are the histograms for the positive and negative training example as scored by the strong classifier trained by Adaboost.



1. **Realboost**

Next, I ran Realboost for 100 iterations on the top 100 features selected by Adaboost. Realboost uses multiple bins to approximate an arbitrary 1D function for each classifier, introducing a few issues that need to be considered. For a certain classifier, there may be no samples that fall within a certain bin, or it may contain all positive or all negative training samples. The bin weight is calculated by where p(b) and q(b) are the sum of all positive and negative weights that fall in bin *b*, respectively, therefore you cannot have zero positive or negative weights in each bin. To account for this, I used a weight of 0 if no samples fell into a bin, because this bin offers no information. If there were only one type of samples in the bin, I used a very low “pseudo-count” for the bin that was not present. If there are only positive samples in a bin, the bin weight will be very positive and if there are only negative samples in a bin, the bin weight will be very negative, reflecting a high level of confidence for that bin. I hesitated to use +Inf for bins with all positive samples and –Inf for bins with all negative samples because I would obtain +Inf for some sample weights during the weight update stage. It is important to note that these modifications may lead to over-fitting to the training data (usually requires further cross-validation to protect against).

1. **Testing on class image**

I ran the strong classifier trained with Adaboost and Realboost on the two test images using a sliding window approach at multiple scales. Each 16x16 window in the test images were classified at multiple image scales from 0.06 to 0.22 the size of the original image, in 0.01 increments (16 scales per image), so hopefully each face will be captured by the 16x16 window at some scale. Shown below are the initial results, with no hard negative mining and no non-maximum suppression.

To improve performance, I ran the strong classifier over the three background images and used any faces detected as “hard negatives”. These hard negatives were incorporated into the training set and the Realboost classifier was re-trained. The classifier was re-trained beginning at the last iteration of Realboost, using the sample weights from the final iteration. Hard negative samples were all given the maximum negative sample weight, and the negative sample weights were rescaled. The Realboost classifier was re-trained for 100 more iterations so that each feature is refined using the hard negative examples. Shown below are the face detection results after hard negative mining.

Finally, to reduce overlapping boxes resulting from detection at multiple scales, I implemented non-maximum suppression. If any boxes overlapped by more than 50%, I selected the box with the highest score. Shown below are the face detection results after non-maximum suppression.