**M231 - Pattern Recognition and Machine Learning**

**DUCK-HA HWANG**

**UID#: 404589112**

**Project II: Human face detection by Boosting techniques**

**1. Objectives.**

Boosting is a general method for improving the accuracy of any given learning algorithm. Specially, one can use it to combine simple “rules” (or weak learner), 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, but cascade is not required.

**2. The project includes the following steps.**

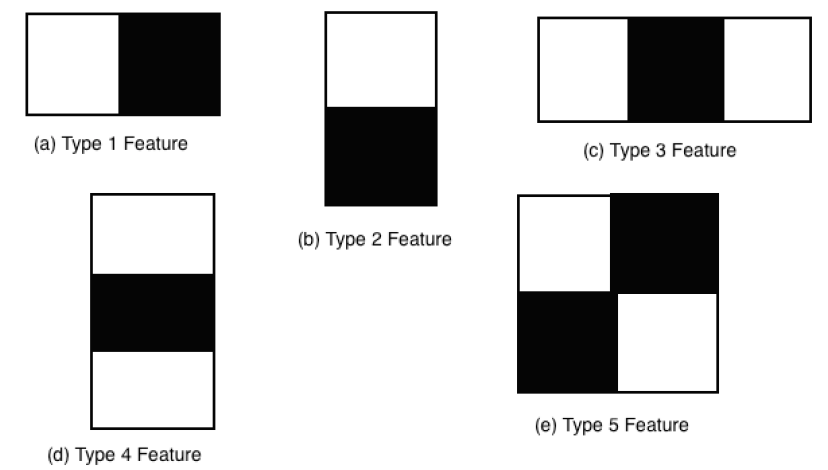
**2.1. Construction of weak classifiers**

Design a few types of features as we discussed in class, and vary their size and location to generate about 10^5 features. For each feature calculate the histograms for the positive and negative populations. Determine the threshold. Each feature corresponds to a weak classifier, and is also called a tree-stump. [Note that as the samples change their weights over time, the histogram and threshold will change.]

Answer:

As a part of the Adaboosting or Adaptive Boosting Procedure, I am to design some number of features. These features have two portions to them, the shaded and the un-shaded areas, and a difference in the intensity between these portions is calculated for every image in the dataset. This is what taken as the feature value for a particular image, for a specific feature type. Here, I have designed 5 different types of Haar features as shown below. Thus for every image the feature value is calculated for every feature type and for all possible valid scales and positions of these feature windows.

Feature: [1 2; 2 1; 1 3; 3 1; 2 2]



For the above designs of the features in a 16x16 image I was able to obtain 32384 valid features, with split being 8704 of the Type 1 and Type 2 features each, 5440 of the Type 3 and Type 4 features each and the rest 4096 of the Type 5 features. I found the feature values i.e the intensity difference values of every feature on application on an image, for 10000 face and 10000 non-face 16x16 images. I have used only in 16x16 images throughout my computations due to the limitation of time.

**2.2. AdaBoosting**

Implement the Adaboost algorithm to boost the weak classifiers you got in (2.1).

i) Display the best ten features as images (after boosting);

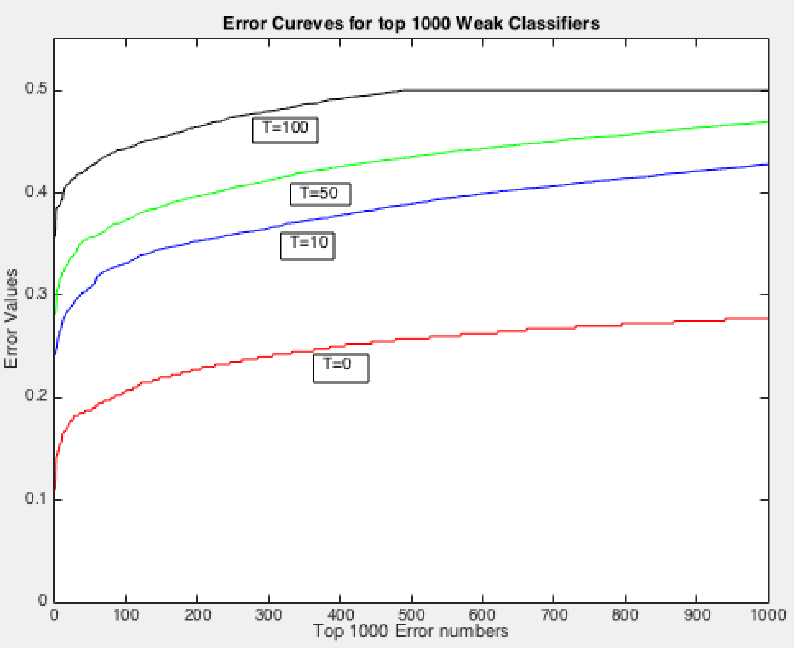
Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

ii) At steps T=0, 10, 50, 100 respectively, plot the curve for the training errors of top 1000 weak classifiers among the pool of weak classifiers in increasing order.

Compare these four curves and see how many weak classifiers have errors close to 1/2;

Answer:



After Adaboosting, iterations T=0,T=10,T=50,T=100 each the top 1000 errors. The errors are sorted order and the top 1000 are taken and plotted as shown in the above image. As it can be seeing, with the increase of T, the error of weak classifiers increases.

At T=100 between 500 and 1000 classifiers’ errors close to 1/2.

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

From the three histograms, you plot their corresponding ROC curves.

[Note: You can continue to T= 200, or 500 as long as its performance keeps improving. ]

Answer:

For T=10

|  |  |
| --- | --- |
|  |  |

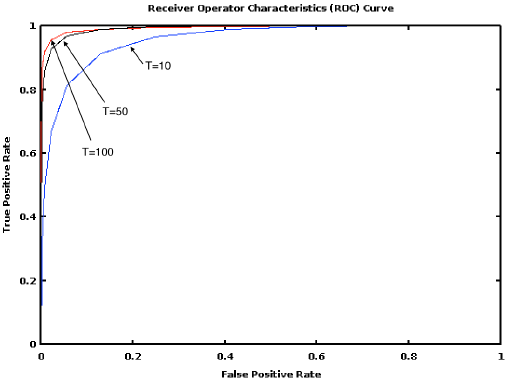
For T=50

|  |  |
| --- | --- |
|  |  |

For T=100

|  |  |
| --- | --- |
|  |  |

Total ROC curves



**2.3. RealBoosting**

Implement RealBoosting algorithm using the top T=10, 50, 100 features you have chosen at Adaboosting step 2.2.

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

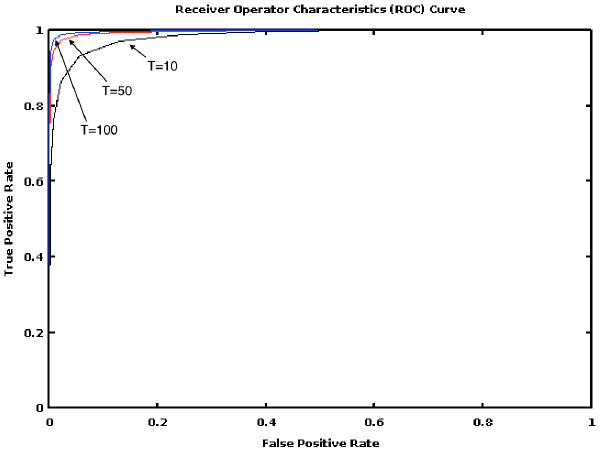
Answer:

For T=10

For T=50

For T=100

(v) Plot the three ROC curves and comparing them against the ROC's in 2.2.



**3. Datasets**

**Training data**: Face dataset and Non-face dataset in the size of 16x16 pixels are given to you.

[Warning: The training of Boosting code takes a long time, so use a small number of examples when you test your codes, and then run the full dataset after you verify your code.]

**Test data**: Two class photos are given. [Note that you need to scale the image into a few scales so that the faces at the front and back are 16x16 pixels in one of the scaled image]. Run you code and do some non-maximum suppression, i.e. when two positive detections overlap significantly, choose the one has higher score.

The classroom image was rescaled to include as many faces as possible. In the comparison below it can be clearly seen that the Real Boost results are much better than the adaboost ones, as there are comparatively more non faces also being detected as faces (false positives) in adaboost over realboost.

Ada Boosted Classifiers testing on Classroom image:





Real Boosted Classifiers testing on Classroom image:





**Hard negative mining**. 3 background images are taken without faces, you can run your detector on these images. Any “faces” detected by your code are called “hard negatives”. Add them to the training set and re-train (don’t start over, just add them at the end of where you stopped).