OUTPUT FROM running main() function in Viola_Jones program

The following output is from an actual run of the attached ***Viola_Jones_FINAL.py*** program. I experimented with various numbers and sizes (2 x 1, 4 x 2, 6 x 4, etc.) of features and it appeared that the best results were from 60 features, mostly 6 x 2. These resulted in a very low False Negative Rate (FNR) (about 3.2%) and a reasonably low False Positive Rate (FPR) (about 10%). After the THETA adjustments, the FNR was down to near-zero but the FPR jumped to about 28%, still below the 30% suggested threshold.

The code in the attached ***Viola_Jones_FINAL.py*** file is extensively commented and is separated into three parts: preliminary and training functions, testing functions (on the test image ***test_img.jpg*** being used in this case), and the main program that calls these functions. Accordingly, the following output is divided into TRAINING OUTPUT and TEST-CASE output. The TEST-CASE output shows the coordinates of the top-left corners of the 64 x 64 frames within which faces were detected, and the picture below also shows these faces outlined with an orange-colored box. The positioning of the squares is not ideal, sometimes a little below or above the face. This is easy to fix but I did not have time to do that before the deadline.

To run ***Viola_Jones_FINAL.py***, please load the file and type main(). You will then see the output which I have captured below. Please note that the training part of the program takes almost two hours to run, most of that time being on the AdaBoost (which is evident from the output below). The actual testing of the ***test_img.jpg*** file takes about 30 seconds to run (after the training has completed).

OUTPUT FROM LOADING AND TRAINING PHASE

---

main()

START loading faces and backgrounds and form label array (+1 and -1) ....

FINISHED loading faces/backgrounds and labels into a tuple of arrays

START computing the integral array .....

FINISHED computing the integral array

START forming the feature list .....

FINISHED forming the feature list

START performing AdaBoost. Iterations TAKE LONG!! Alert is provided on entering every FIFTH feature.

Starting iteration# 1 in AdaBoost - calling opt_weaklearner .....

    opt_weaklearner: starting feature # 1 of 60

    opt_weaklearner: starting feature # 6 of 60

opt_weaklearner: starting feature # 11 of 60

opt_weaklearner: starting feature # 16 of 60

opt_weaklearner: starting feature # 21 of 60

opt_weaklearner: starting feature # 26 of 60

opt_weaklearner: starting feature # 31 of 60

opt_weaklearner: starting feature # 36 of 60

opt_weaklearner: starting feature # 41 of 60

opt_weaklearner: starting feature # 46 of 60

opt_weaklearner: starting feature # 51 of 60

opt_weaklearner: starting feature # 56 of 60

Optimal learner index: 46 Adding it to weaklearner list - p and theta are (respectively) -1.0 -2243.0

FALSE NEGATIVE/FALSE POSITIVE rates after iteration # 1 are: 0.06 / 0.1225 Weights have been updated

Starting iteration# 2 in AdaBoost - calling opt_weaklearner .....

opt_weaklearner: starting feature # 1 of 60

opt_weaklearner: starting feature # 6 of 60

opt_weaklearner: starting feature # 11 of 60

opt_weaklearner: starting feature # 16 of 60

opt_weaklearner: starting feature # 21 of 60

opt_weaklearner: starting feature # 26 of 60

opt_weaklearner: starting feature # 31 of 60

opt_weaklearner: starting feature # 36 of 60

opt_weaklearner: starting feature # 41 of 60

opt_weaklearner: starting feature # 46 of 60

opt_weaklearner: starting feature # 51 of 60

opt_weaklearner: starting feature # 56 of 60

Optimal learner index: 14 Adding it to weaklearner list - p and theta are (respectively) -1.0 -796.0

FALSE NEGATIVE/FALSE POSITIVE rates after iteration # 2 are: 0.042 / 0.109 Weights have been updated

Starting iteration# 3 in AdaBoost - calling opt_weaklearner .....

opt_weaklearner: starting feature # 1 of 60

opt_weaklearner: starting feature # 6 of 60

opt_weaklearner: starting feature # 11 of 60

opt_weaklearner: starting feature # 16 of 60

opt_weaklearner: starting feature # 21 of 60

opt_weaklearner: starting feature # 26 of 60

opt_weaklearner: starting feature # 31 of 60

opt_weaklearner: starting feature # 36 of 60

opt_weaklearner: starting feature # 41 of 60

opt_weaklearner: starting feature # 46 of 60

opt_weaklearner: starting feature # 51 of 60

opt_weaklearner: starting feature # 56 of 60

Optimal learner index: 2 Adding it to weaklearner list - p and theta are (respectively) 1.0 515.0

FALSE NEGATIVE/FALSE POSITIVE rates after iteration # 3 are: 0.037 / 0.1025 Weights have been updated

Starting iteration# 4 in AdaBoost - calling opt_weaklearner .....

opt_weaklearner: starting feature # 1 of 60

opt_weaklearner: starting feature # 6 of 60

opt_weaklearner: starting feature # 11 of 60

opt_weaklearner: starting feature # 16 of 60

opt_weaklearner: starting feature # 21 of 60

opt_weaklearner: starting feature # 26 of 60

opt_weaklearner: starting feature # 31 of 60

opt_weaklearner: starting feature # 36 of 60

opt_weaklearner: starting feature # 41 of 60

opt_weaklearner: starting feature # 46 of 60

opt_weaklearner: starting feature # 51 of 60

opt_weaklearner: starting feature # 56 of 60

Optimal learner index: 24 Adding it to weaklearner list - p and theta are (respectively) 1.0 374.0

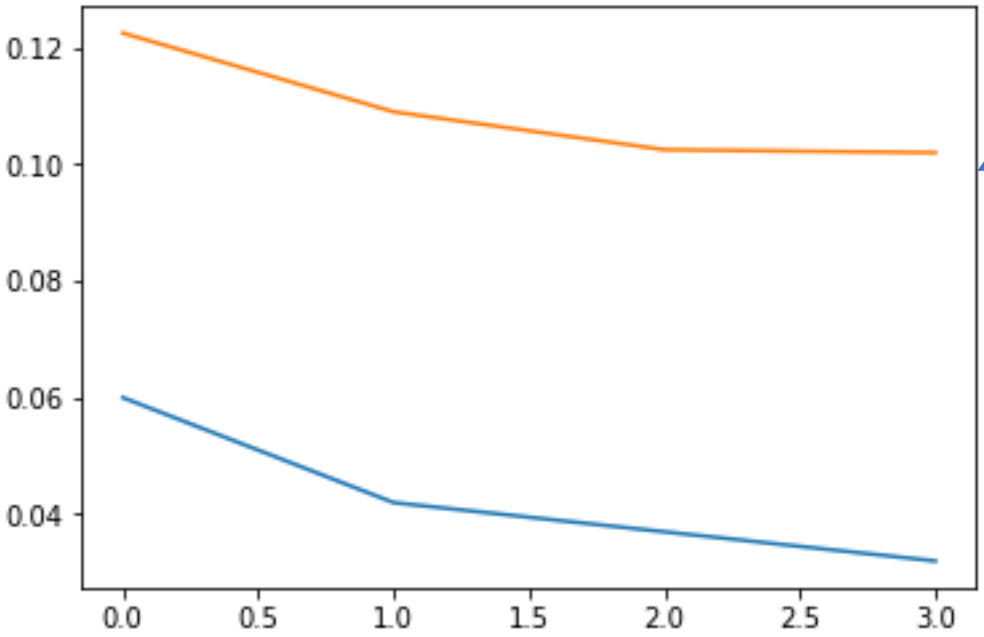
FALSE NEGATIVE/FALSE POSITIVE rates after iteration # 4 are: 0.032 / 0.102 Weights have been updated

Plots for FALSE NEGATIVES (BLUE) and FALSE POSITIVES (ORANGE) are shown below (x-axis = iteration #):

AdaBoost HAS COMPLETED!

THETA-adjusted FALSE NEGATIVE rate is: 0.0055 ← **Almost perfect**

THETA-adjusted FALSE POSITIVE rate is: 0.2805 ← **Less than 30%**



FP (Orange) and FN (blue) rates during AdaBoost iterations (on x-axis). This is BEFORE the THETA adjustments

---

TEST-CASE OUTPUT

---

load_image("test_img.jpg")

runSlider(64,64)

Faces were detected in 64 x 64 pixel squares whose TOP LEFT CORNERS have the following coordinates:

[192 , 320]

[192 , 704]

[192 , 768]

[192 , 896]

[192 , 1216]

[256 , 192]

[256 , 320]

[256 , 384]

[256 , 512]

[256 , 768]

[256 , 1216]

[256 , 1280]

[256 , 1344]

[320 , 192]

[320 , 256]

[320 , 384]

[320 , 448]

[320 , 512]

[320 , 576]

[320 , 640]

[320 , 704]

[320 , 768]

[320 , 832]

[320 , 896]

[320 , 960]

[320 , 1216]

[320 , 1408]

[384 , 320]

[384 , 384]

[384 , 576]

[384 , 704]

[384 , 960]

[384 , 1024]

[384 , 1152]

[384 , 1344]

[448 , 256]

[448 , 320]

[448 , 704]

[448 , 768]

[448 , 832]

[448 , 1344]

[512 , 256]

[512 , 576]

[512 , 768]

[512 , 832]

[512 , 896]

[512 , 1088]

[512 , 1152]

[512 , 1216]

[512 , 1280]

[576 , 320]

[576 , 384]

[576 , 448]

[576 , 576]

[576 , 640]

[576 , 1024]

[576 , 1216]

[576 , 1280]

Total number of faces detected by Viola-Jones: 58 Actual total: 50

Here is a picture of the faces bounded by orange squares

Out[183]: (140, 140, 58)


0
200
400
600
800
1000
1200
0
200
400
600
800
1000
1200
1400
1600