**Assignment #2 Report – EigenFaces, Adaboost, Detection**

**Written By: mhtay**

**P1)**

Eigen-Face was computed from the corpus of images, and the mean and divided by subsequent zero-mean variance to obtain the normalized Eigen-face:



Figure : Eigen-Face

The dot product of the normalized eigen-face was taken against the image, cornered at every pixel of the image except the corners using a sliding window.

The maximum values of the dot product can be seen in an image of the scores obtained.

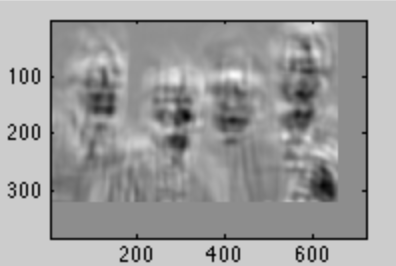


Figure :DotProductImage

However, due to the high intensity of features like the hand, false positives is possible. Thus, a normalized dot product of the faces with the eigenface might produce better results, by not only subtracting the mean of the patches, but also dividing by the sum of each zero mean patch. The resulting score is more informative as the brighter regions correspond to patches cornered at that area.

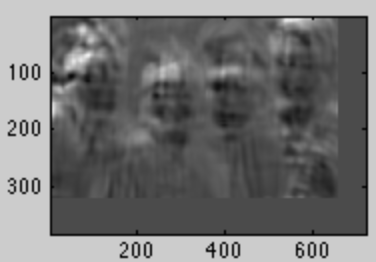


Figure :Zero-Mean, 1 variance patches dot product

Dividing each zero mean patch with the sum of the patch before the mean normalization, however, gives better distinction of the faces:

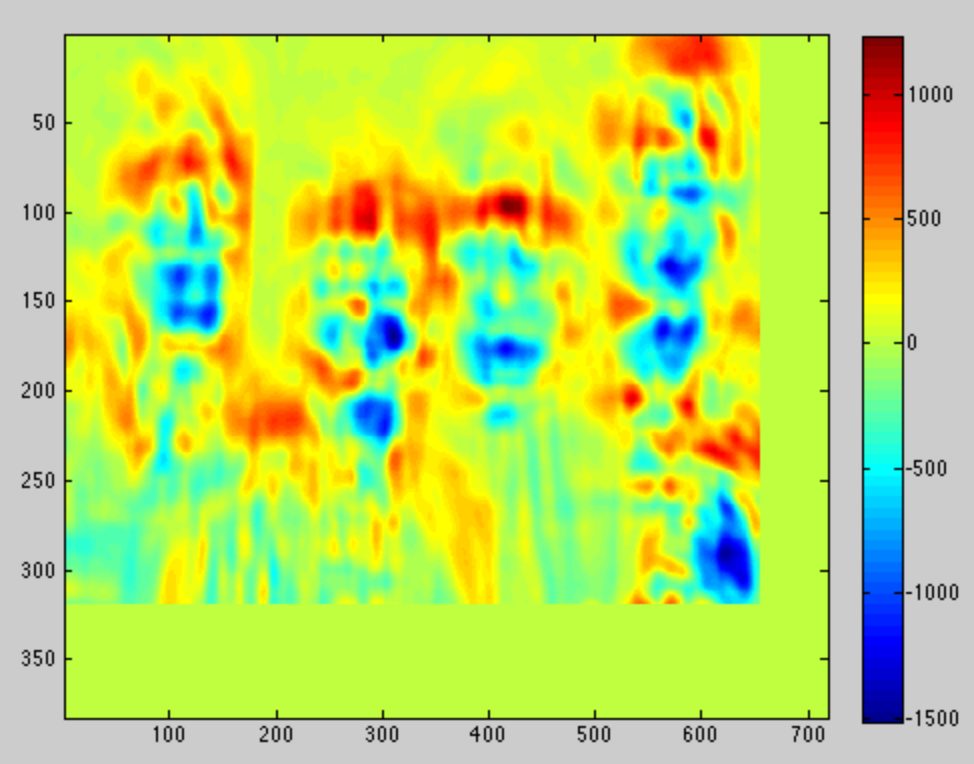
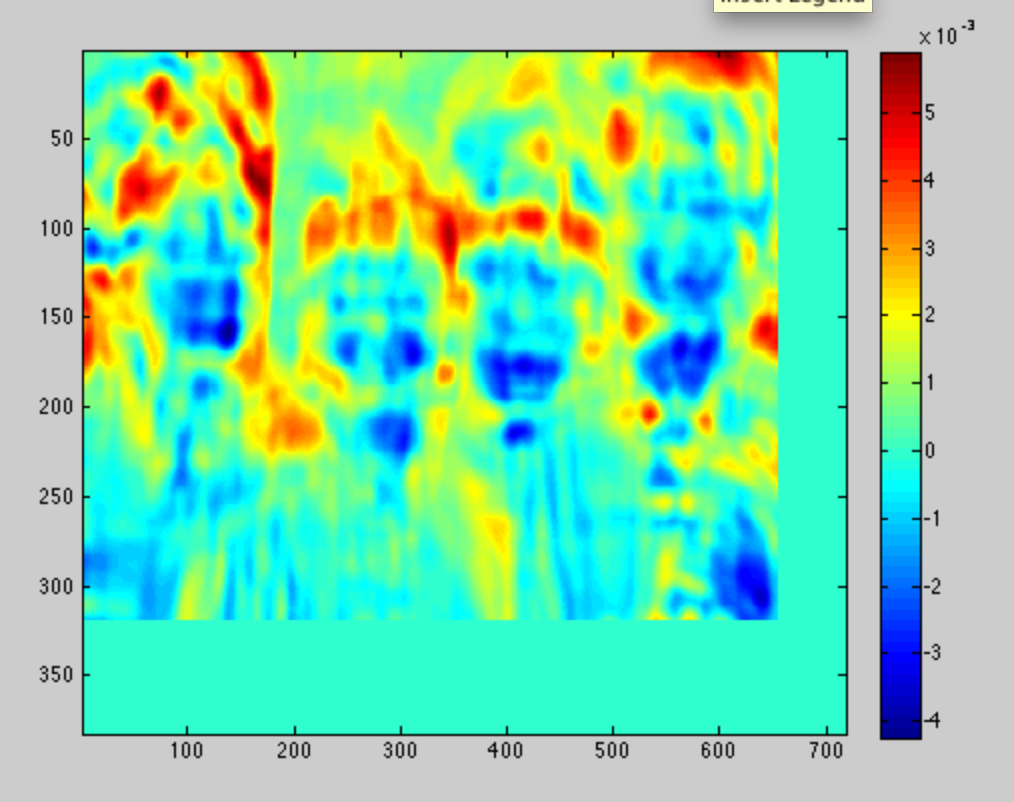


Figure 5: After dividing by sum of patch (before mean norm)

Figure :Without dividing by sum

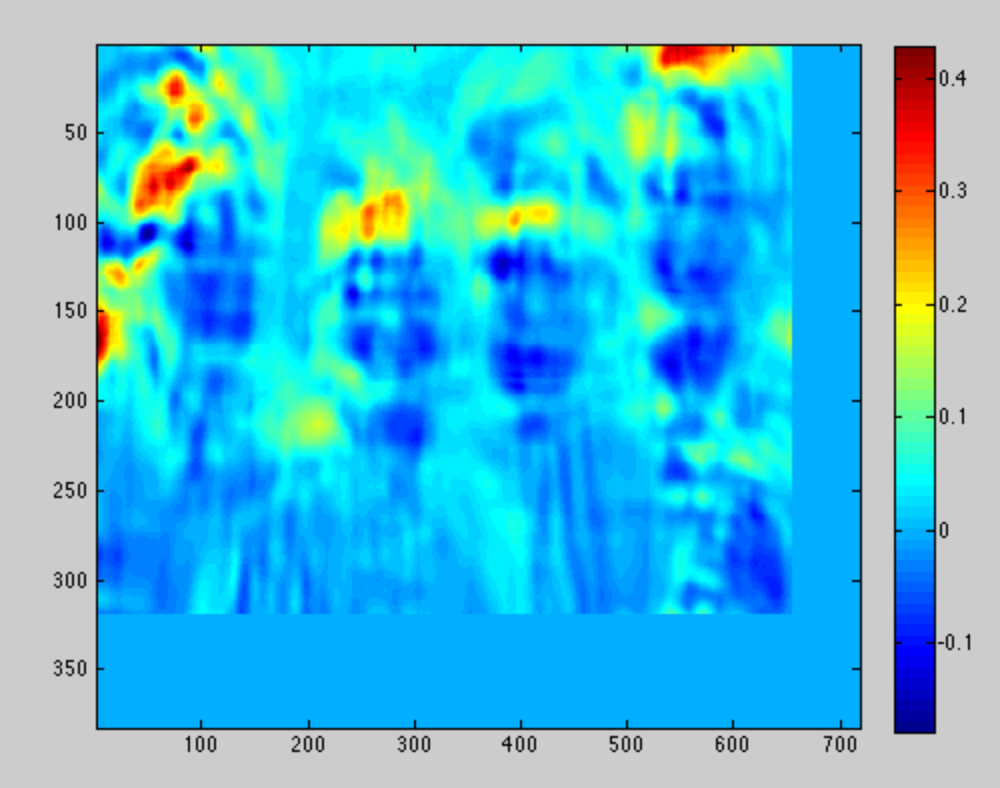


Figure 6: after dividing by sum of mean norm patch

However, the approach taken in figure 6 does not work well on the other image, so the approach was taken to carry out the normalized dot product on each patch after subtracting the patch mean and without dividing by norm, using the eigenface learned after mean and standard deviation normalization.

A query sliding face window, which indicated a 1 in the presence of a face, and 0 otherwise was carried out. The decision on whether a face was present was based on a threshold that was empirically found to be 70% of the maximum patch score.

This was repeated at each of the 5 scales, from a value of 0.5, 0.75, 1.0, 1.5 and 2.0. The resulting faces were detected at:

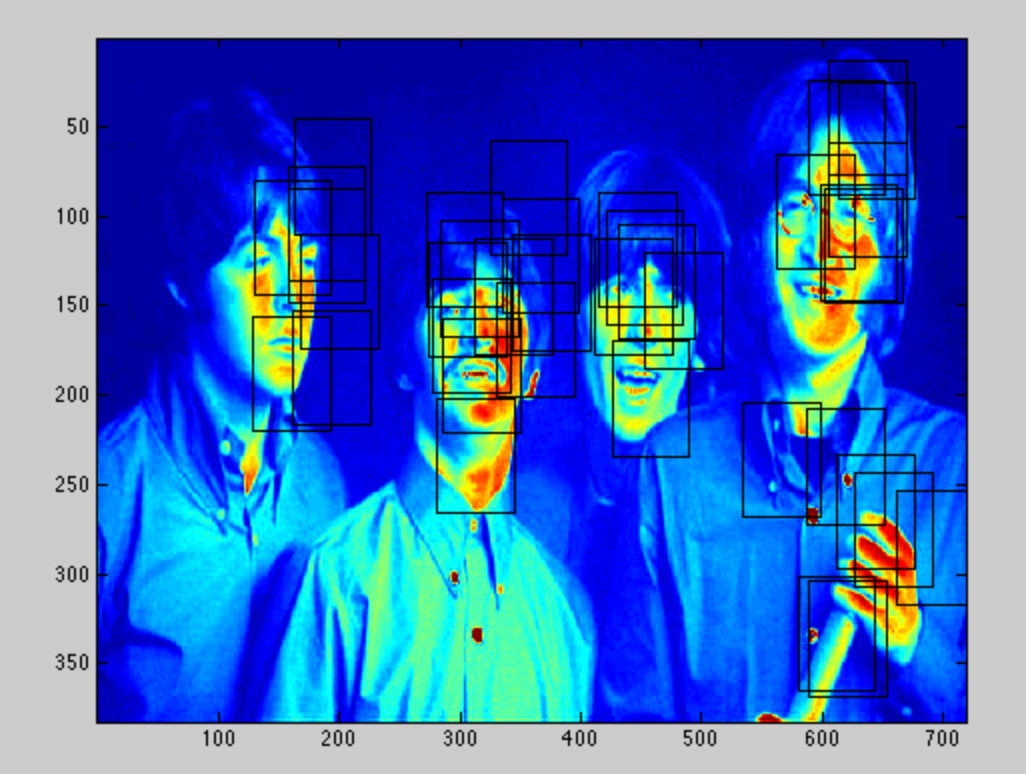


Figure 7: Multi-scale sliding window face query

Thereafter, bounding boxes in proximity were coalesced into a single bounding box. This was done by searching for the nearest bounding box to the existing box, then taking the average of the 2 boxes as the new box. However, the new box produced might be close to an existing box in the shortlisted list, thus the process of searching for nearest boxes needs to be repeated until no closest boxes are found. The limit for the closest box is set to 64, the approximate size of the Eigen-face rectangle.

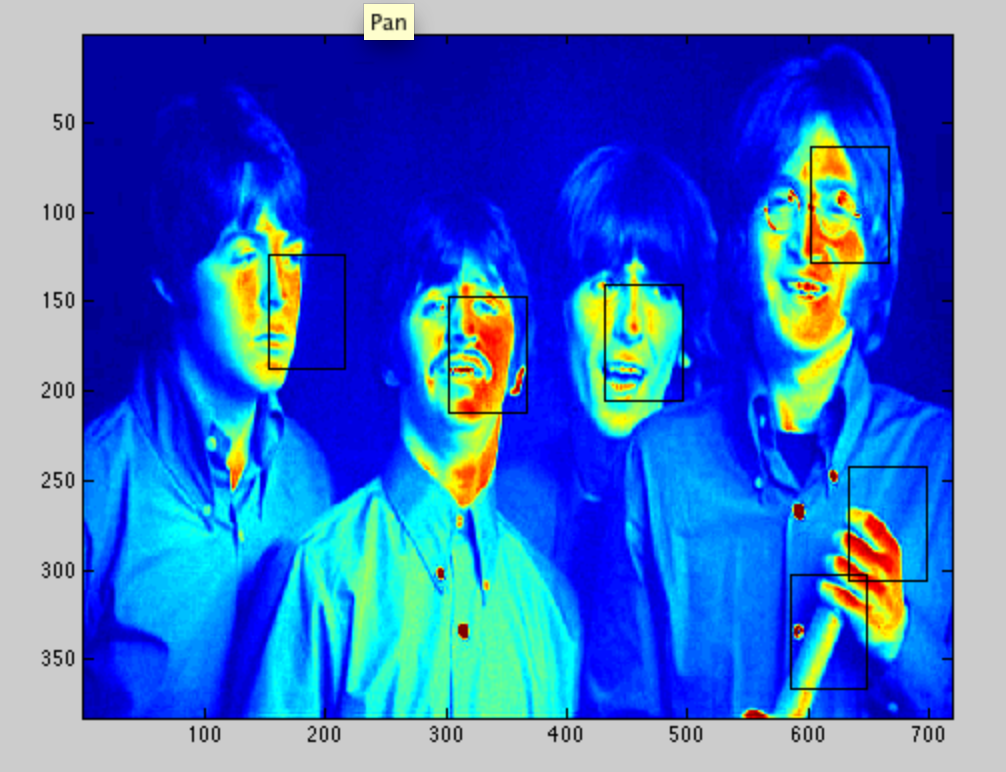


Figure 8: Short-listed faces