Project 3: Eigen Faces

CSE559A - Computer Vision

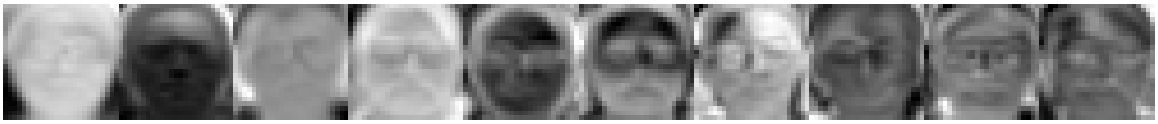
Daniel de Cordoba Gil

10/23/2016

1. Testing recognition with cropped class images

We obtained the following average face from the neutral image set:

Using it, we computed the 10 eigenvectors with the lowest eigenvalue:



When trying to recognize all 24 interesting faces and determine whether they were faces or not, we got different errors, and depending on the threshold chosen, more or less were considered faces or not. The following is a list of the mse obtained after projecting and reconstructing the faces:

- interesting/01.tga, MSE: 658.199
- **interesting/02.tga, MSE: 1641.91**
- **interesting/03.tga, MSE: 790.613**
- **interesting/04.tga, MSE: 1522.98**
- **interesting/05.tga, MSE: 1758.03**
- interesting/06.tga, MSE: 557.482
- interesting/07.tga, MSE: 646.532
- interesting/08.tga, MSE: 434.591
- interesting/09.tga, MSE: 484.556
- interesting/10.tga, MSE: 571.732
- interesting/11.tga, MSE: 276.432
- interesting/12.tga, MSE: 527.628
- interesting/13.tga, MSE: 441.073
- interesting/14.tga, MSE: 451.231
- **interesting/15.tga, MSE: 1610.11**
- interesting/16.tga, MSE: 565.273
- interesting/17.tga, MSE: 322.44
- interesting/18.tga, MSE: 519.553
- interesting/19.tga, MSE: 454.637
- interesting/20.tga, MSE: 382.199
- **interesting/21.tga, MSE: 913.998**
- interesting/22.tga, MSE: 683.683
- interesting/23.tga, MSE: 289.163
- interesting/24.tga, MSE: 385.814

This gives us a mean of 703.7, a median of 542.6 and a standard deviation of 451.3. We can see the distribution of the points in Figure 1 and 2.

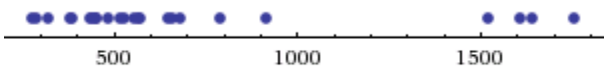


Figure 1. Mse distribution of the interesting image set computing isface

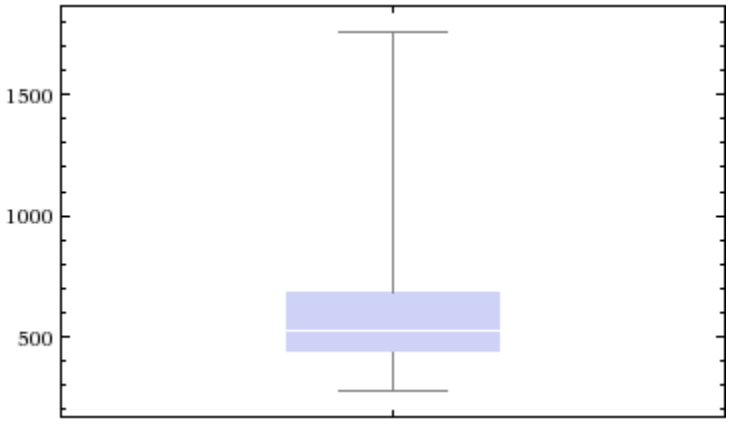


Figure 2. Box and whisker chart of the mse in the interesting image set computing isface

If we choose a threshold of 700, we get a correct classification in 18 of the 24 interesting faces, which is a 75% accuracy. In the above list, the results that overcome this threshold are in bold.

When verifying the interesting faces in comparison with their corresponding neutral faces, using just 10 eigenfaces we get a decent accuracy, with just 3 out of 24 faces showing a mse greater than 60000, which shows a 88% accuracy in our sample. The following is a list of the MSE obtained for all images. In bold, the values with a mse > 60000 are shown:

- interesting/01.tga - neutral/01, MSE: 20335.1
- interesting/02.tga - neutral/02, MSE: 154036
- interesting/03.tga - neutral/03, MSE: 33053.6
- **interesting/04.tga - neutral/04, MSE: 197479**
- interesting/05.tga - neutral/05, MSE: 59766.8
- interesting/06.tga - neutral/06, MSE: 21208.9
- **interesting/07.tga - neutral/07, MSE: 109273**
- interesting/08.tga - neutral/08, MSE: 12656.9
- interesting/09.tga - neutral/09, MSE: 11581.1
- interesting/10.tga - neutral/10, MSE: 33461.6
- interesting/11.tga - neutral/11, MSE: 18382.3
- interesting/12.tga - neutral/12, MSE: 45451.3
- interesting/13.tga - neutral/13, MSE: 21164

- interesting/14.tga - neutral/14, MSE: 9223.79
- **interesting/15.tga - neutral/15, MSE: 184151**
- interesting/16.tga - neutral/16, MSE: 52580.2
- interesting/17.tga - neutral/17, MSE: 6018.54
- interesting/18.tga - neutral/18, MSE: 12470.4
- interesting/19.tga - neutral/19, MSE: 35719.8
- interesting/20.tga - neutral/20, MSE: 12178.7
- interesting/21.tga - neutral/21, MSE: 19435.6
- interesting/22.tga - neutral/22, MSE: 13188.1
- interesting/23.tga - neutral/23, MSE: 3351.31
- interesting/24.tga - neutral/24, MSE: 5884.7

This gives us a mean of 45502, a median of 20750 and a standard deviation of 56624. We can see the distribution of the points in Figure 3 and 4.

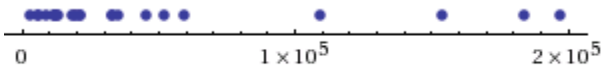


Figure 3. Mse distribution of the interesting image set computing verifyface

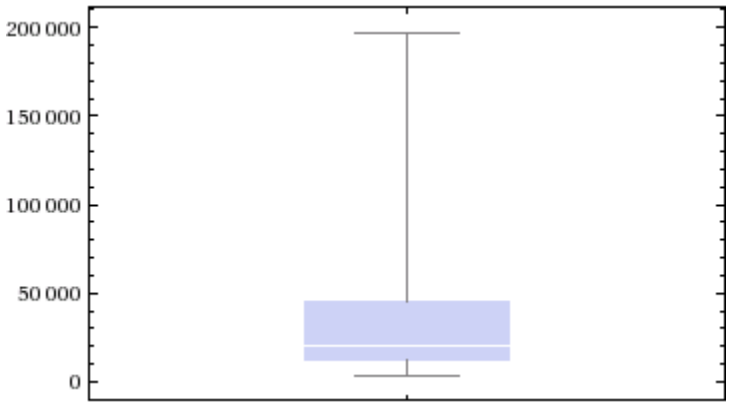


Figure 4. Box and whisker chart of the mse in the interesting image set computing verifyface

To end with the analysis of the interesting images set using 10 eigenfaces, let's look at how many of the images are correctly matched with their corresponding neutral face. In the following list, the recognize face program is run for every image, which returns the closest neutral image for every image received. The wrongly matched images are marked in bold:

- interesting/01.tga closest to neutral/01; MSE: 20335.1
- **interesting/02.tga closest to: neutral/13; MSE: 96835**
- **interesting/03.tga closest to: neutral/14; MSE: 22589.4**
- **interesting/04.tga closest to: neutral/14; MSE: 73442.4**

- interesting/05.tga closest to: neutral/05; MSE: 59766.8
- interesting/06.tga closest to: neutral/06; MSE: 21208.9
- **interesting/07.tga closest to: neutral/24; MSE: 58974.3**
- interesting/08.tga closest to: neutral/08; MSE: 12656.9
- interesting/09.tga closest to: neutral/09; MSE: 11581.1
- interesting/10.tga closest to: neutral/10; MSE: 33461.6
- interesting/11.tga closest to: neutral/11; MSE: 18382.3
- interesting/12.tga closest to: neutral/12; MSE: 45451.3
- interesting/13.tga closest to: neutral/13; MSE: 21164
- interesting/14.tga closest to: neutral/14; MSE: 9223.79
- **interesting/15.tga closest to: neutral/24; MSE: 146803**
- **interesting/16.tga closest to: neutral/12; MSE: 27121.1**
- interesting/17.tga closest to: neutral/17; MSE: 6018.54
- interesting/18.tga closest to: neutral/18; MSE: 12470.4
- interesting/19.tga closest to: neutral/19; MSE: 35719.8
- interesting/20.tga closest to: neutral/20; MSE: 12178.7
- **interesting/21.tga closest to: neutral/23; MSE: 13893.2**
- interesting/22.tga closest to: neutral/22; MSE: 13188.1
- interesting/23.tga closest to: neutral/23; MSE: 3351.31
- interesting/24.tga closest to: neutral/24; MSE: 5884.7

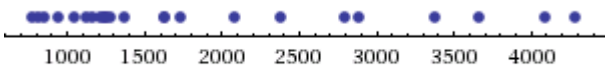
We get 17 imaged matched correctly from the 24 images dataset. Therefore, the accuracy measured is 71%, as expected. The unmatched pictures are seen below. The three faces shown are the original interesting face, the reconstructed interesting face and the neutral face matched to such face:

We see that the errors make sense and are reasonable, as all mismatched faces are similar to the original ones. Also, the correct answer at least appear highly in the sorted results, normally second or third, although in some cases fourth or fifth.

Let’s now experiment changing the number of eigenfaces. In the following images, we can see the distribution of the mse for different number of eigenfaces used when computing, as well as the mean, median and standard deviation:

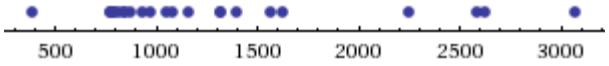
Eigenfaces: 1

mean	1873
median	1332
sample standard deviation	1079



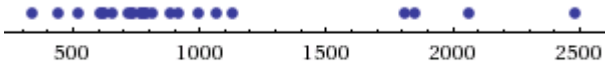
Eigenfaces: 3

mean	1278
median	1012
sample standard deviation	692.2



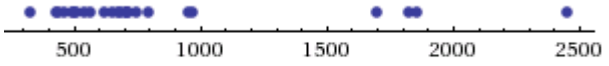
Eigenfaces: 5

mean	966
median	779.5
sample standard deviation	540.2



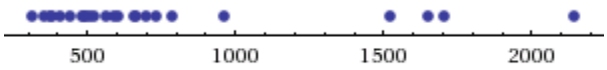
Eigenfaces: 7

mean	848.7
median	681.3
sample standard deviation	543.4



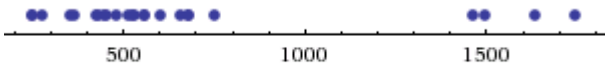
Eigenfaces: 9

mean	754.9
median	579.6
sample standard deviation	491.7



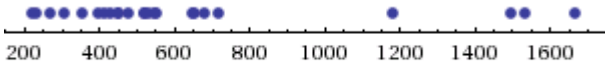
Eigenfaces: 11

mean	673.3
median	529.2
sample standard deviation	440.2



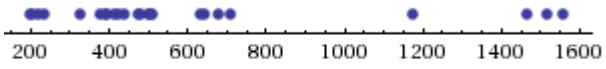
Eigenface: 13

mean	636.8
median	522.4
sample standard deviation	410.2



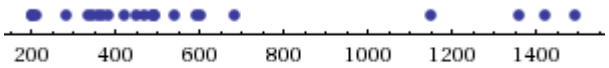
Eigenfaces: 15

mean	604.8
median	478.7
sample standard deviation	407.3



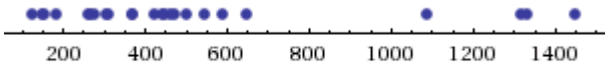
Eigenfaces: 17

mean	565.3
median	461.5
sample standard deviation	387.7



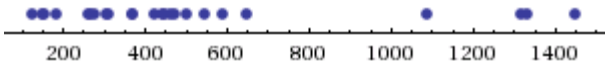
Eigenfaces: 19

mean	542
median	444.3
sample standard deviation	382.7



Eigenfaces: 21

mean	521.7
median	435.8
sample standard deviation	384.1



We can see that as the number of eigenfaces increases, the mean mse decreases. We see that most faces are better detected, and for this reason, the mse distribution goes to the left, but the classification of the outliers that don't fit well in the face space still have a big mse which does not get much better with an increase in the number of eigenfaces.

Let's look at the number of faces correctly and incorrectly classified using the --recognizeface option, for different number of eigenfaces used (Figure 5 and 6):

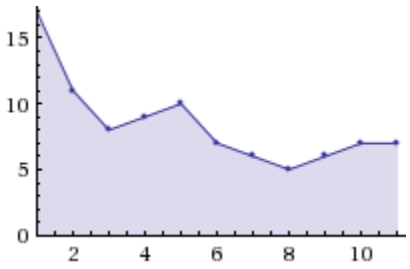


Figure 5. Faces incorrectly recognized (y's) versus number of eigenfaces used (x's)

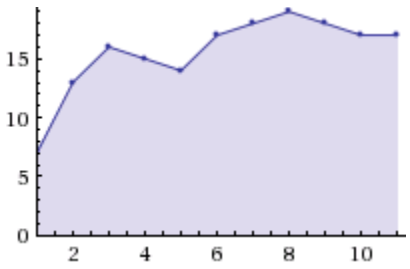


Figure 6. Faces correctly recognized (y's) versus number of eigenfaces used (x's)

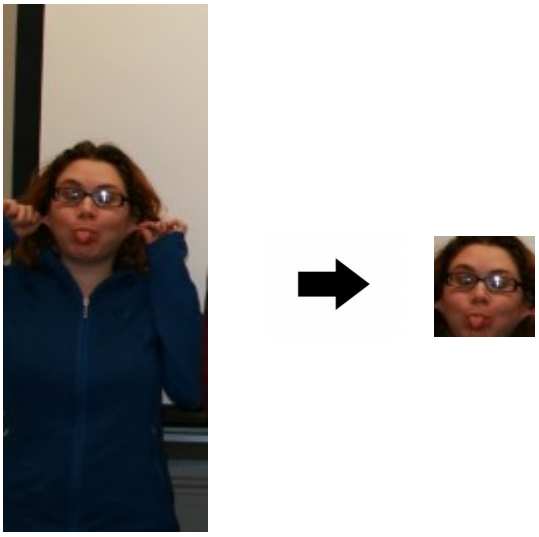
The sweet spot where the maximum number of faces are correctly matched with their original face is 15 eigenfaces, where 19 out of 24 faces are correctly recognized (79% accuracy). The minimum number of faces recognized is acquired when using only 1 eigenface, in which only 7 faces are recognized (29% accuracy). The trend is that for a small number of eigenfaces the accuracy is low, but as we use more eigenfaces the accuracy gets better, until we get to the maximum, in which the accuracy can descend again, but the accuracy does not change much after this point. Therefore, we would need to find this sweet point in which the accuracy is maximized, but it is not an easy task, as we don't have any tool to find it besides testing many images and finding out which number of eigenfaces maximizes the accuracy. Therefore it is not clear how to choose the best number of eigenfaces, but we know we have to use enough so the accuracy is high enough, but not all of them because the speed of the algorithm will be lower for every extra eigenface.

2. Cropping and finding faces

When we execute:

```
./main --findface faces/group/test_single.tga neutral.face 0.35 0.55 0.01 crop 1 cropped_face.tga
```

we get a cropped image of the face found in test_single-tga:

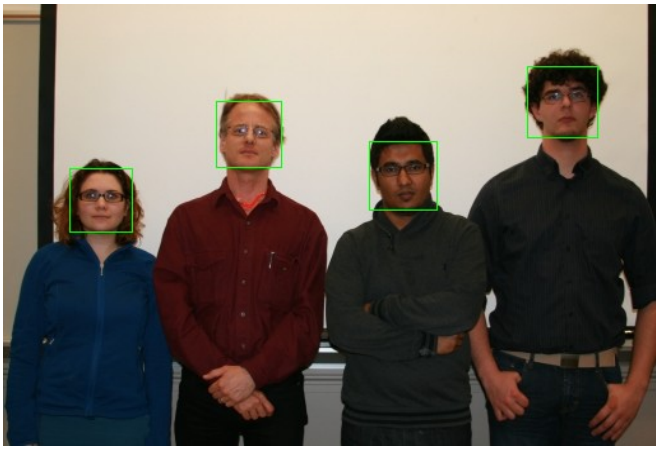


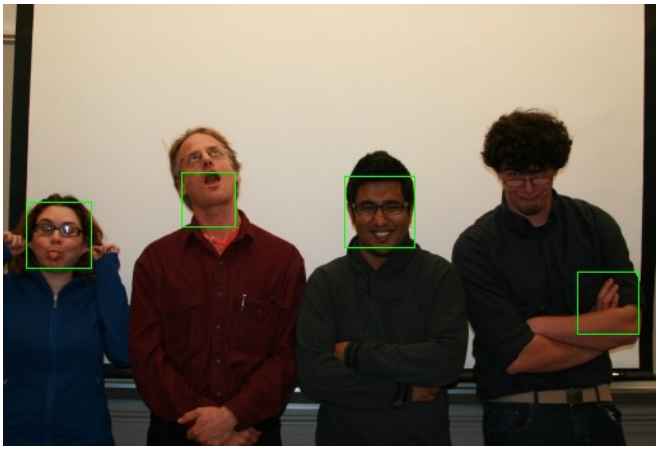
Let's try with a picture of myself (min_scale=0.1, max_scale=0.4, step=0.01):

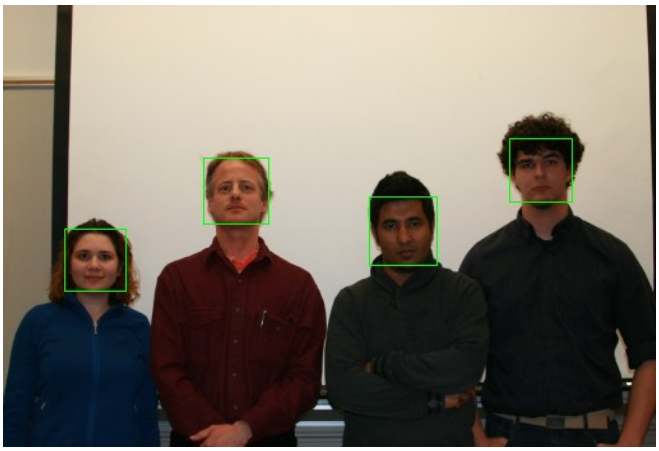
The cropped image obtained is not perfect, as it cuts my chin (probably because I have a thin face, and most faces in the sample are more round), but is successfully detects my face. The best scale detected was 0.29. Let's try another picture:

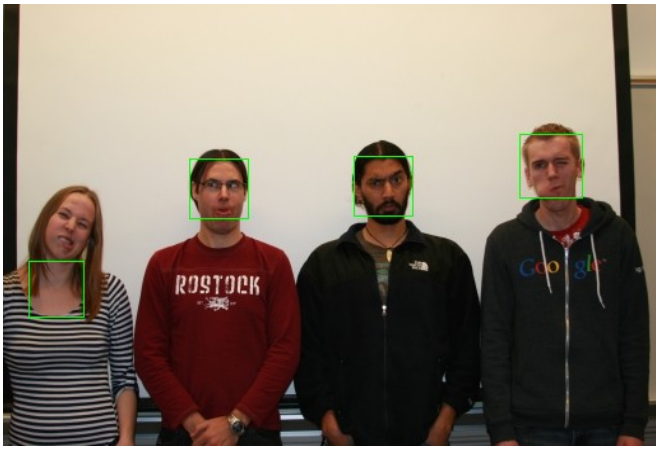
This one is quite better! We used a min_scale=0.3, max_scale=0.5, step=0.01, and the best scale found was 0.44, which is the applied to the above image.

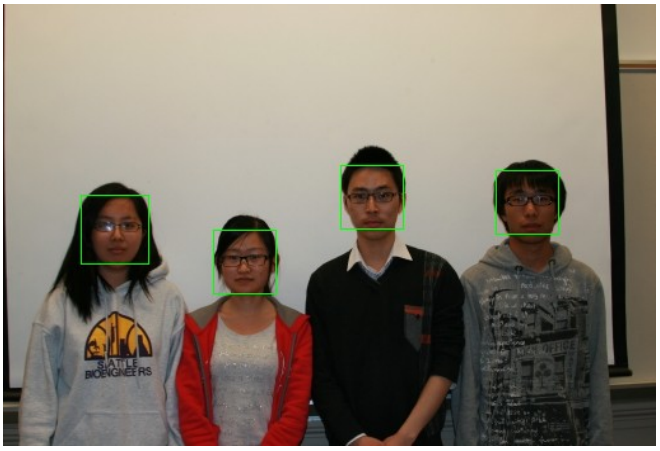
Now, let's see some of the images obtained in group photos:

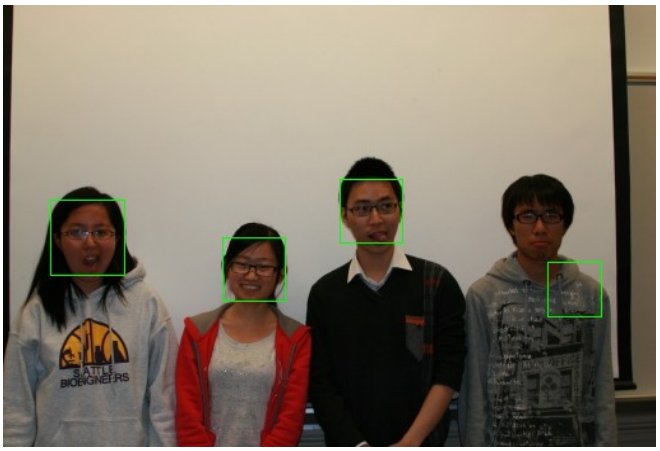


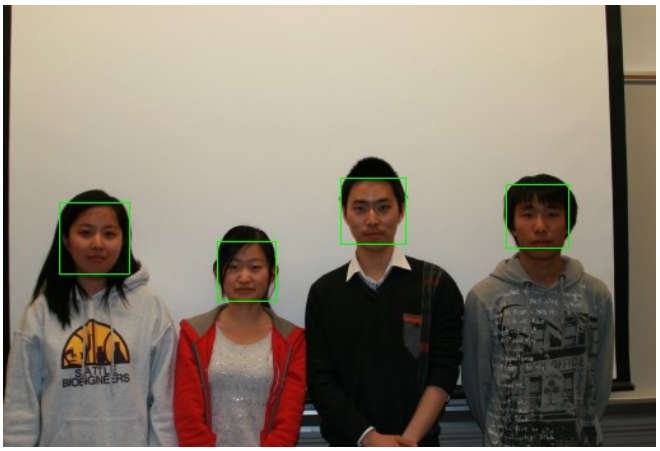


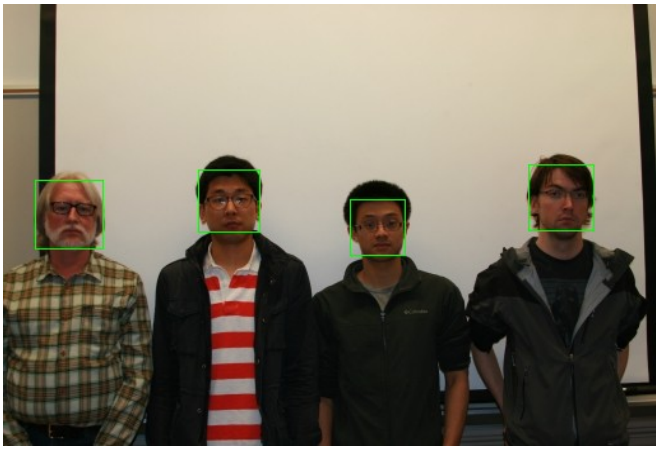


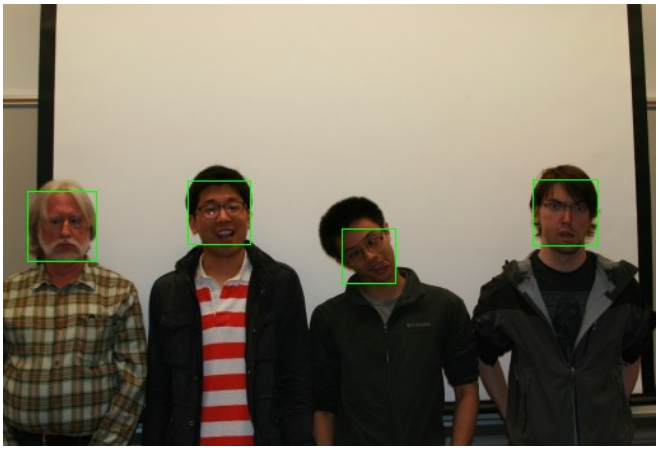

ROSTOCK

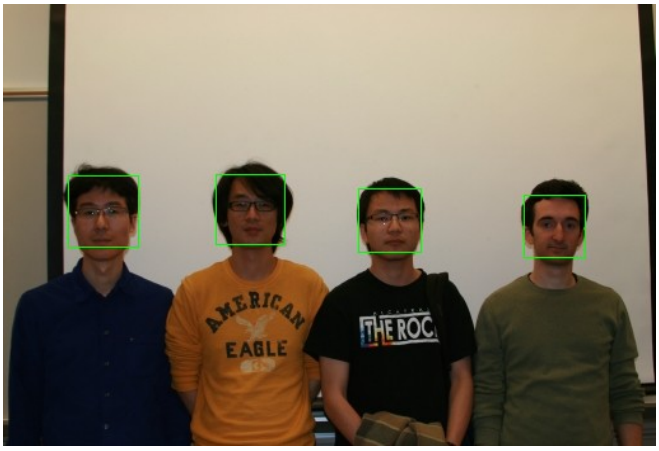
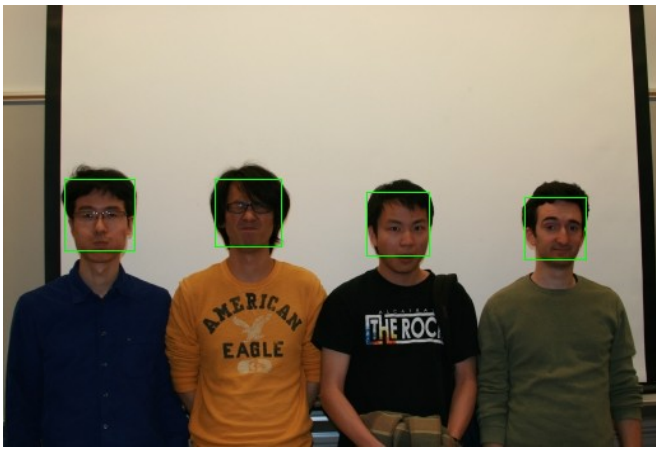
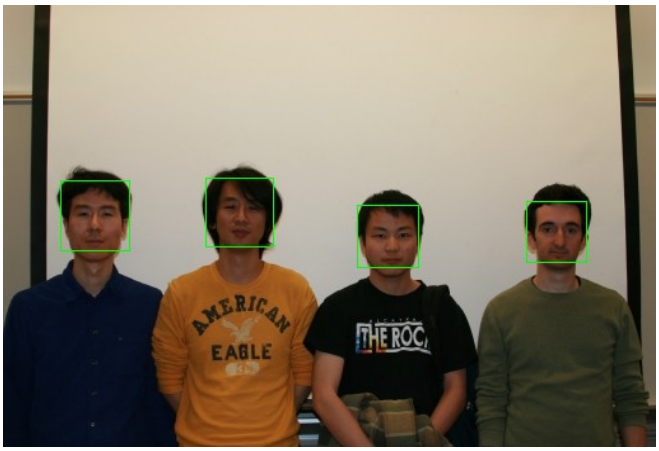
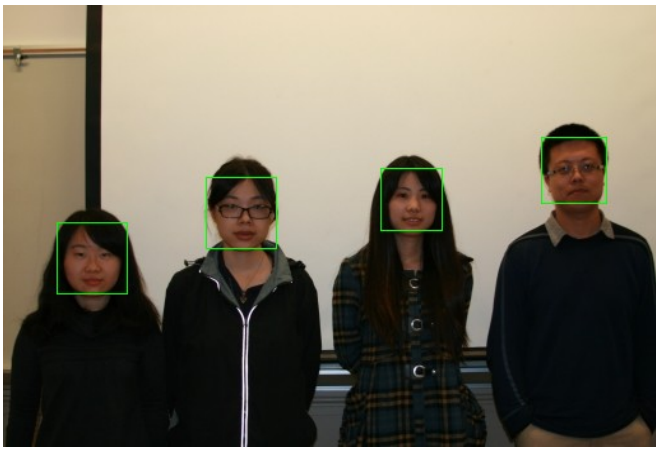
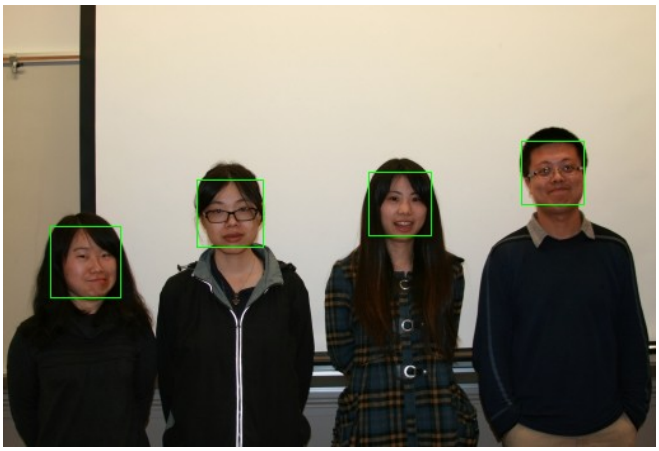


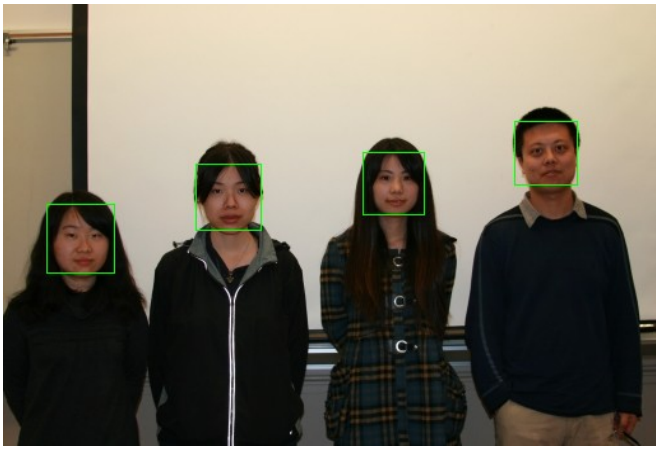






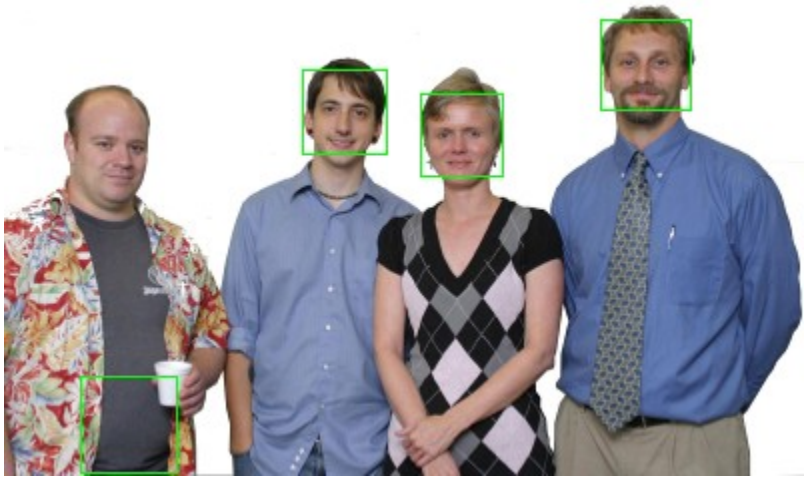




As it can be seen, in most pictures, the faces are found perfectly. The program fails in some cases for some of the weird faces, and recognizes similar structures like the neck or the arm. From the pictures above, we get only four faces not perfectly classified out of 68 faces (accuracy 95%), and two of them point to the neck, which is quite close to the face anyway. To get such good results, we added a filter that takes into account the variance of the face, and discards the images with variances that are too low. This is done to prevent the algorithm from getting fooled by low-texture areas or areas close to the face space but far from the facial mean. This is not a mandatory deliverable, but it makes the performance quite better (before, the accuracy was closer to 80%, as many times the shirt of the people were detected instead of their face), and therefore, it should count as extra credit, from my point of view. To get all these pictures, we used a min_scale=0.5, max_scale=0.6, step=0.01.

Let’s use this in a picture not provided by the professor:



Unfortunately, the face of the man on the left is not detected. This is probably because he is not looking straight at the camera. To achieve such results we had to modify slightly the filter for high variance to 1.3e6 instead of 5e6 to filter the low variance parts of the image that are close to the face space.

3. Verify Face

We have already talked about this in section 1, but let's see it again. The following list shows the results for 10 eigenfaces when verifying the images in the interesting image set against the right neutral face. In bold, the images that go over a threshold of 60000 are shown.

- interesting/01.tga - neutral/01, MSE: 20335.1
- interesting/02.tga - neutral/02, MSE: 154036
- interesting/03.tga - neutral/03, MSE: 33053.6
- **interesting/04.tga - neutral/04, MSE: 197479**
- interesting/05.tga - neutral/05, MSE: 59766.8
- interesting/06.tga - neutral/06, MSE: 21208.9
- **interesting/07.tga - neutral/07, MSE: 109273**
- interesting/08.tga - neutral/08, MSE: 12656.9
- interesting/09.tga - neutral/09, MSE: 11581.1
- interesting/10.tga - neutral/10, MSE: 33461.6
- interesting/11.tga - neutral/11, MSE: 18382.3
- interesting/12.tga - neutral/12, MSE: 45451.3
- interesting/13.tga - neutral/13, MSE: 21164
- interesting/14.tga - neutral/14, MSE: 9223.79
- **interesting/15.tga - neutral/15, MSE: 184151**
- interesting/16.tga - neutral/16, MSE: 52580.2
- interesting/17.tga - neutral/17, MSE: 6018.54
- interesting/18.tga - neutral/18, MSE: 12470.4
- interesting/19.tga - neutral/19, MSE: 35719.8
- interesting/20.tga - neutral/20, MSE: 12178.7
- interesting/21.tga - neutral/21, MSE: 19435.6
- interesting/22.tga - neutral/22, MSE: 13188.1
- interesting/23.tga - neutral/23, MSE: 3351.31
- interesting/24.tga - neutral/24, MSE: 5884.7

The following list shows the results for 10 eigenfaces when verifying the interesting images against the wrong neutral face. In bold, the images that go over a threshold of 60000 are shown (all of them).

- **interesting/01.tga - neutral/02, MSE: 321369**
- **interesting/02.tga - neutral/03, MSE: 316715**
- **interesting/03.tga - neutral/04, MSE: 205345**
- **interesting/04.tga - neutral/05, MSE: 469018**
- **interesting/05.tga - neutral/06, MSE: 239879**
- **interesting/06.tga - neutral/07, MSE: 121234**
- **interesting/07.tga - neutral/08, MSE: 264747**
- **interesting/08.tga - neutral/09, MSE: 433733**

- **interesting/09.tga - neutral/10, MSE: 332114**
- **interesting/10.tga - neutral/11, MSE: 146206**
- **interesting/11.tga - neutral/12, MSE: 156746**
- **interesting/12.tga - neutral/13, MSE: 522392**
- **interesting/13.tga - neutral/14, MSE: 339816**
- **interesting/14.tga - neutral/15, MSE: 108744**
- **interesting/15.tga - neutral/16, MSE: 225578**
- **interesting/16.tga - neutral/17, MSE: 69228.3**
- **interesting/17.tga - neutral/18, MSE: 276630**
- **interesting/18.tga - neutral/19, MSE: 224183**
- **interesting/19.tga - neutral/20, MSE: 61391**
- **interesting/20.tga - neutral/21, MSE: 379566**
- **interesting/21.tga - neutral/22, MSE: 212703**
- **interesting/22.tga - neutral/23, MSE: 192558**
- **interesting/23.tga - neutral/24, MSE: 332604**
- **interesting/24.tga - neutral/01, MSE: 277781**

We can see that 60000 is the best threshold as it is the one that offers the lowest number of false positives and false negatives: 0 false positives and only three false negatives. If we choose a higher threshold, like 70000, the number of false positives goes to two, and the number of false negatives stays the same. If we try a lower threshold, like 50000, the number of false positives stays the same, but the number of false negatives goes to 5. Therefore, the optimal threshold is 60000. We discovered this with trial and error, and for a different number of eigenfaces used, it may vary.

To end, for extra credit I have also added a boolean called speed_up to apply the speedup when computing the eigenfaces. If this bool is set to true in the code, the eigenfaces will be computed much faster: the time goes from about 20 seconds to almost immediate.