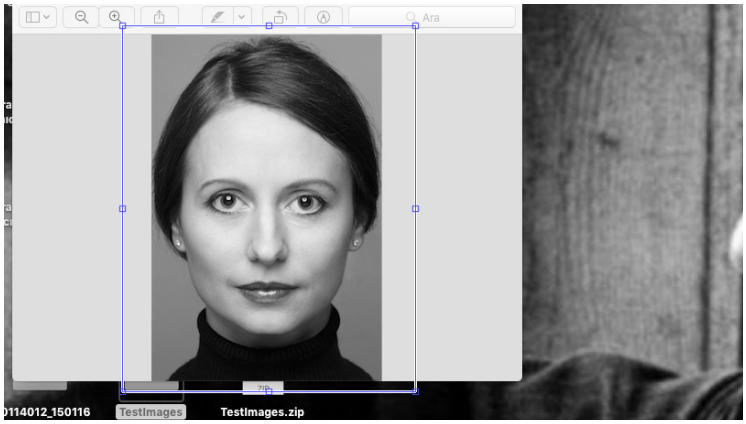
Face Detection by Cross-Correlation on MatLab



Mustafa Onur İzmitlioğlu

Face Detection Using Cross-Correlation in MatLab

In this project, we aimed to implement a face-detection application in MatLab using the cross-correlation between an average face image and 5 test images.

Cross correlation:

In signal processing, **cross-correlation (in other words, sliding dot product)** is a measure of similarity of two series as a function of the displacement of one relative to the other. These two series are "images" in our case, and the outcome of the cross-correlation and the relationship between the similarity of two images can be described as follows:

" The range of the data is -1 to 1 such that the closer the cross-correlation value is to 1, the more closely the information sets are."

The cross correlation can be mathematically expressed as given:

$$M_I[x,y] = \frac{1}{pq}\sum_{r=0}^{p-1}\sum_{s=0}^{q-1} I[x+r,y+s] \quad (1)$$

$$M_T = \frac{1}{pq}\sum_{r=0}^{p-1}\sum_{s=0}^{q-1} T[r,s] \quad (2)$$

$$A[x,y] = \frac{\sum_{s=0}^{q-1}\sum_{r=0}^{p-1}(T[r,s]-M_T)\times(I[x+r,y+s]-M_I[x,y])}{\sqrt{\sum_{s=0}^{q-1}\sum_{r=0}^{p-1}(T[r,s]-M_T)^2}\times\sqrt{\sum_{s=0}^{q-1}\sum_{r=0}^{p-1}(I[x+r,y+s]-M_I[x,y])^2}} \quad (3)$$

M(I): Image matrix
M(T): Test Matrix
A[x,y]: Correlation of two images at point [x,y]

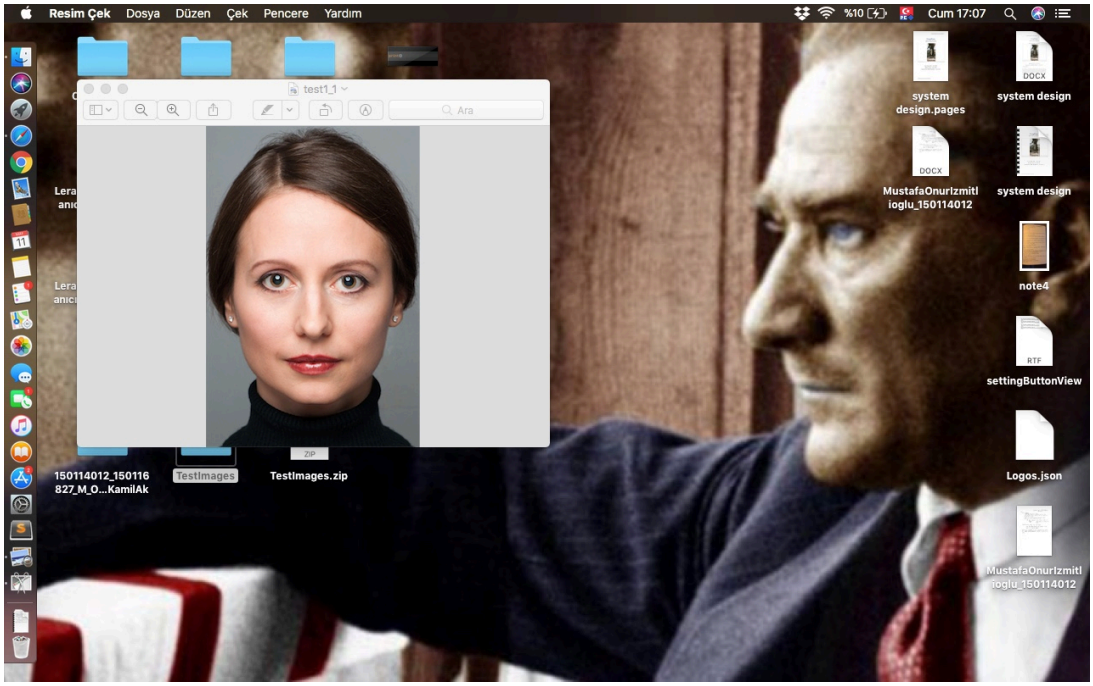
Implementation of Cross-Correlation:

First of all, we needed to take the average of face images (in this experiment, 3 faces). The average of three face is outputted as following:

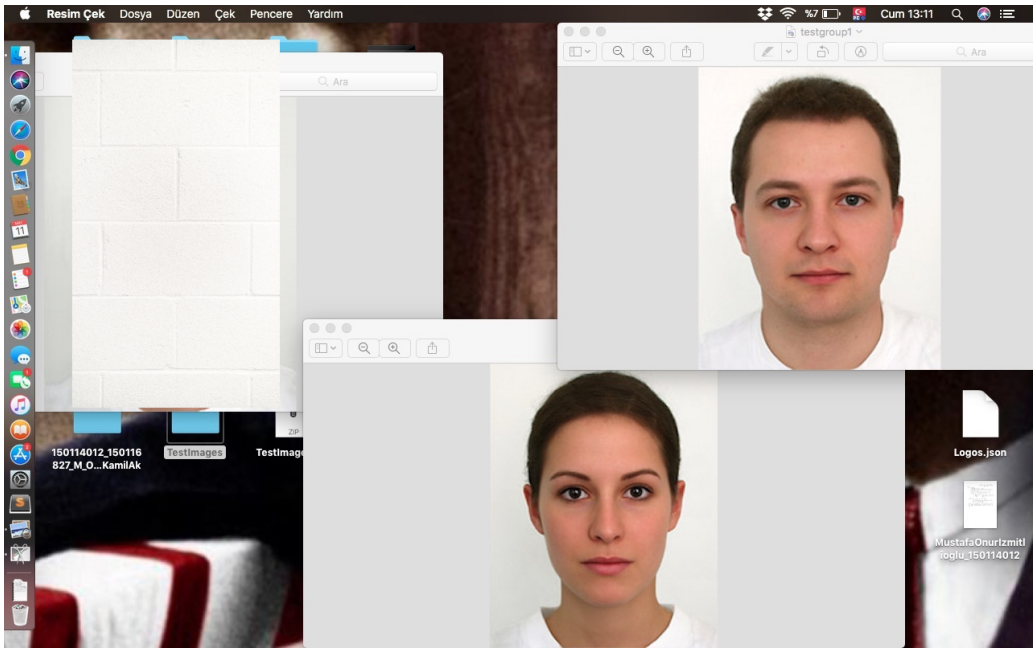
(a) (b) (c) (d)Average

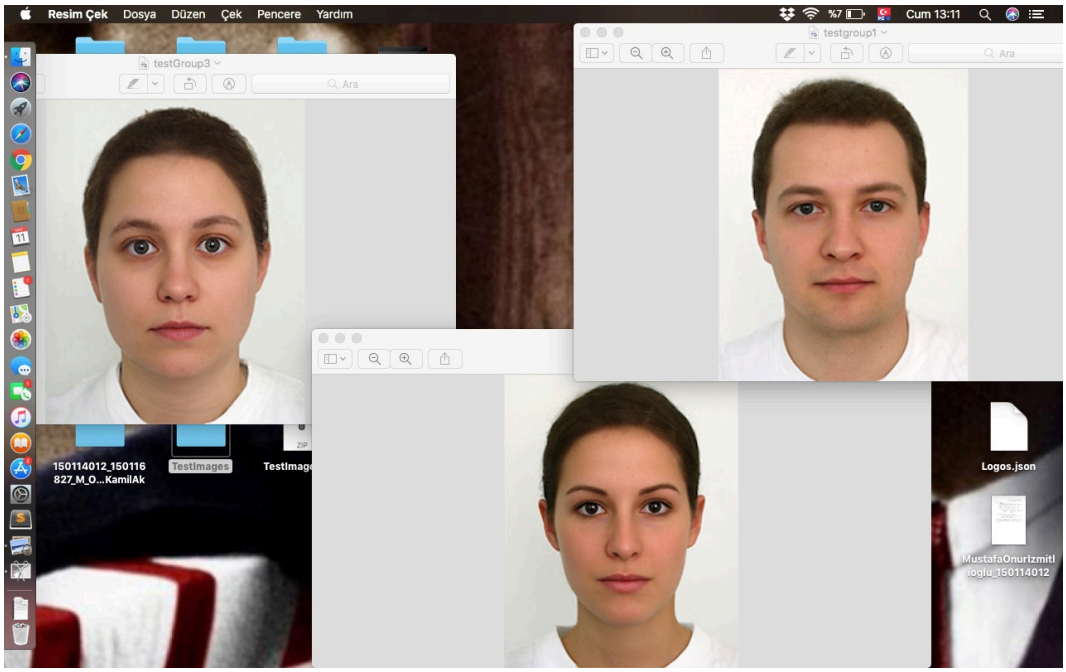
Then, this average face is used to apply the cross-correlation into 5 different test images. These test images include similar faces in-between, and also some of them check the effect of the background color of the test images.

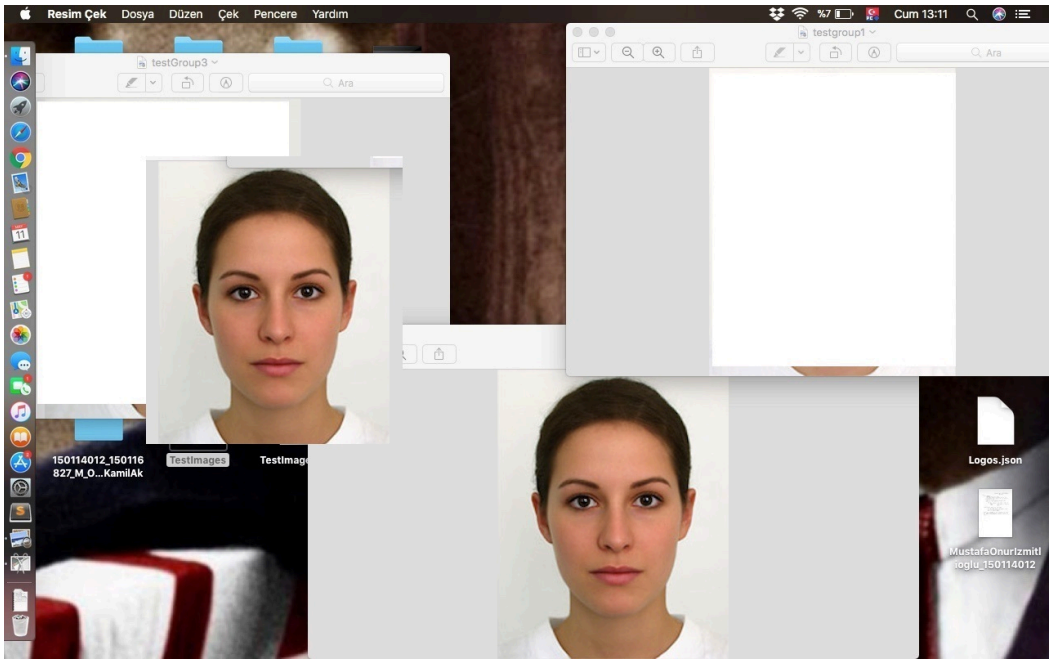
The 5 test images are:




Resim Çek Dosya Düzen Çek Pencere Yardım
Cum 17:11
test1_2
system design.pages
system design
MustafaOnurIzmitlioglu_150114012
system design
note4
settingButtonView
Logos.json
MustafaOnurIzmitlioglu_150114012
150114012_150116827_M_O...KamilAk
TestImages
TestImages.zip


Resim Çek Dosya Düzen Çek Pencere Yardım
Cum 13:11
testgroup1
Logos.json
MustafaOnurIzmitlioglu_150114012
150114012_150116827_M_O...KamilAk
TestImages


Resim Çek
Dosya
Düzen
Çek
Pencere
Yardım
Cum 13:11
testGroup3
testgroup1
Ara
150114012_150116
827_M_O...KamilAk
TestImages
Logos.json
MustafaOnurIzmitl
oglu_150114012

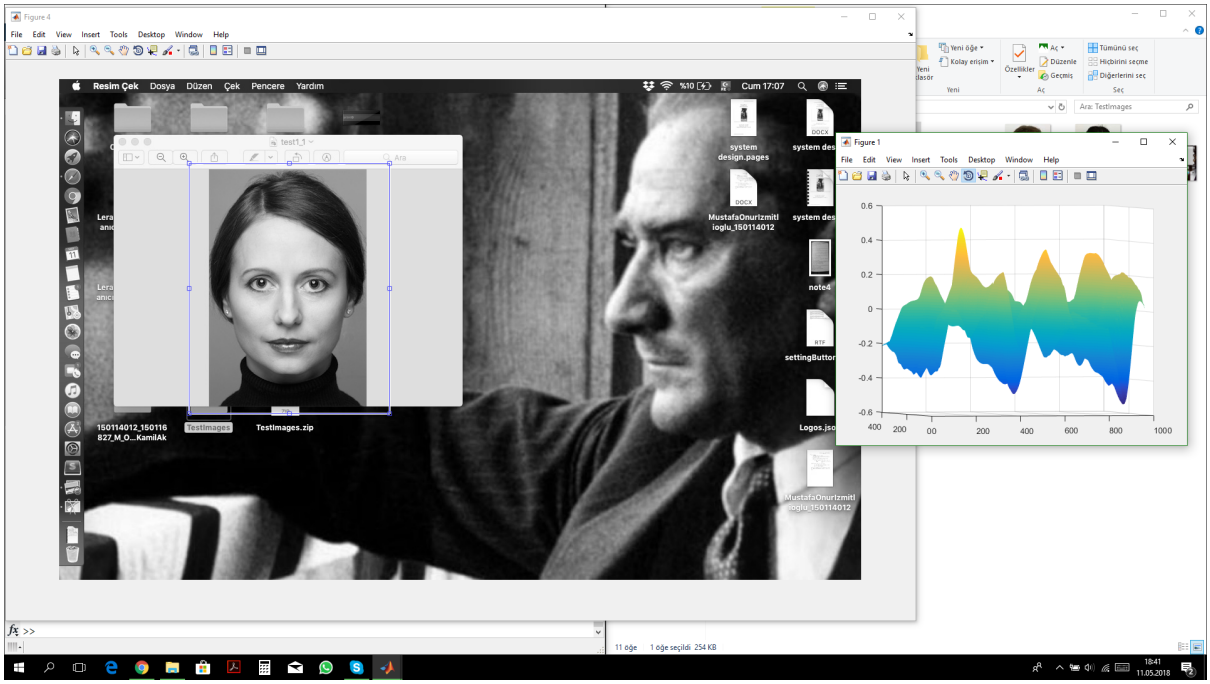

Resim Çek
Dosya
Düzen
Çek
Pencere
Yardım
Cum 13:11
testGroup3
testgroup1
Ara
150114012_150116
827_M_O...KamilAk
TestImages
Logos.json
MustafaOnurIzmitl
oglu_150114012

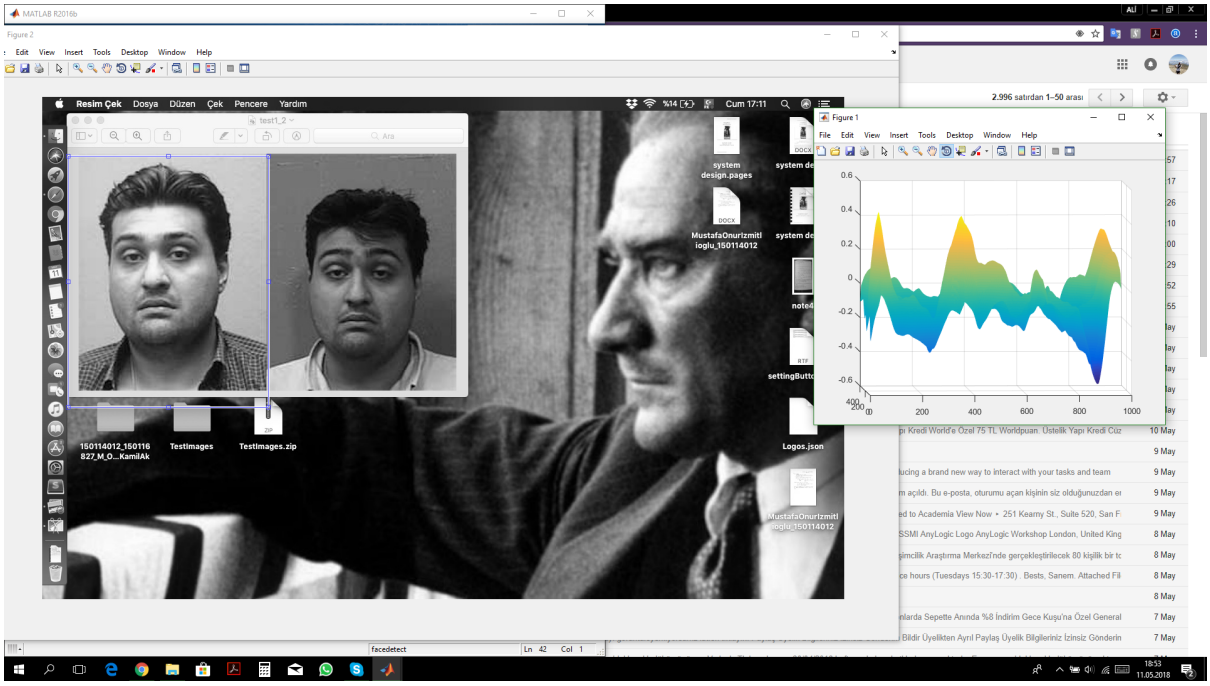
Output Results:

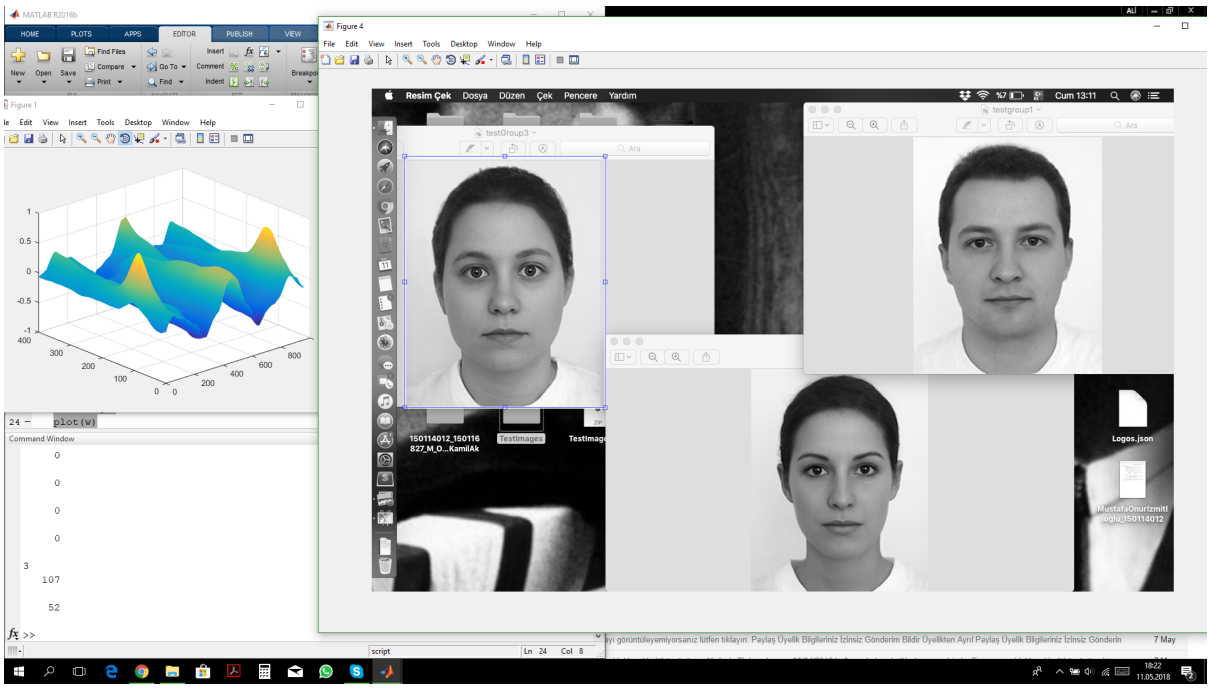
After running the MatLab code, we obtained the following results, where the most similar face image is highlighted by a built-in draggable rectangle object outputted by:

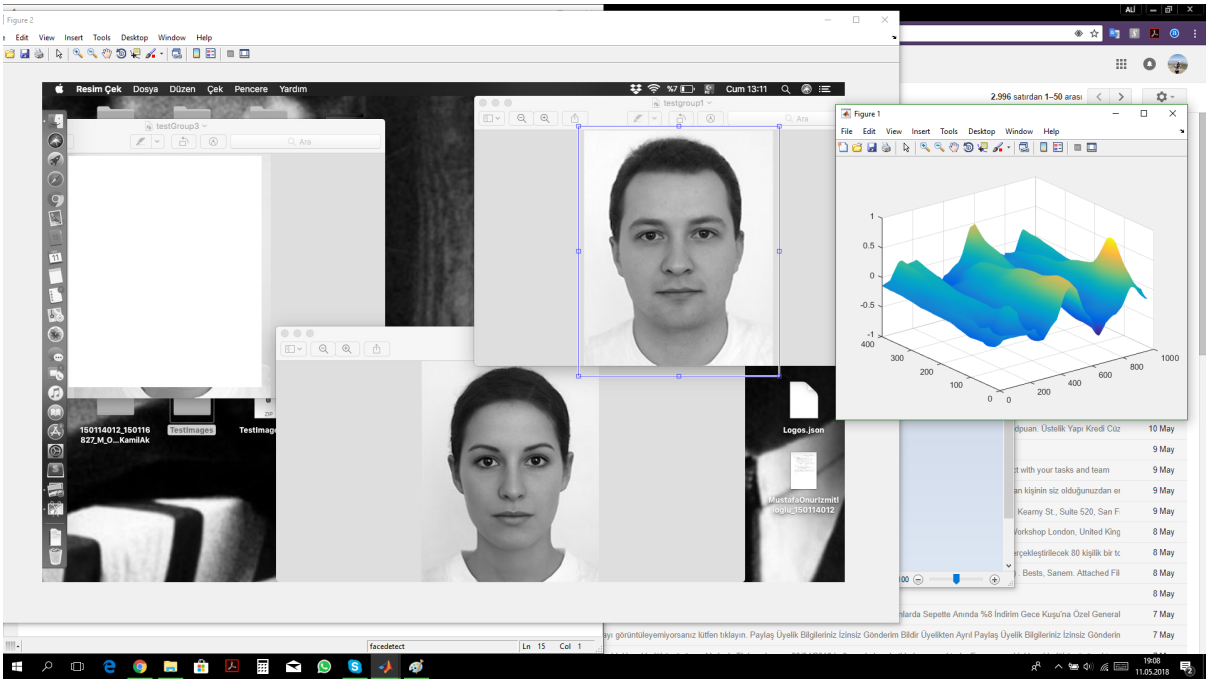
```
imshow(GroupImage1);
imrect(gca, [xpeak+1, ypeak+1, (size(averageImage,2)), (size(averageImage,1))]);
```

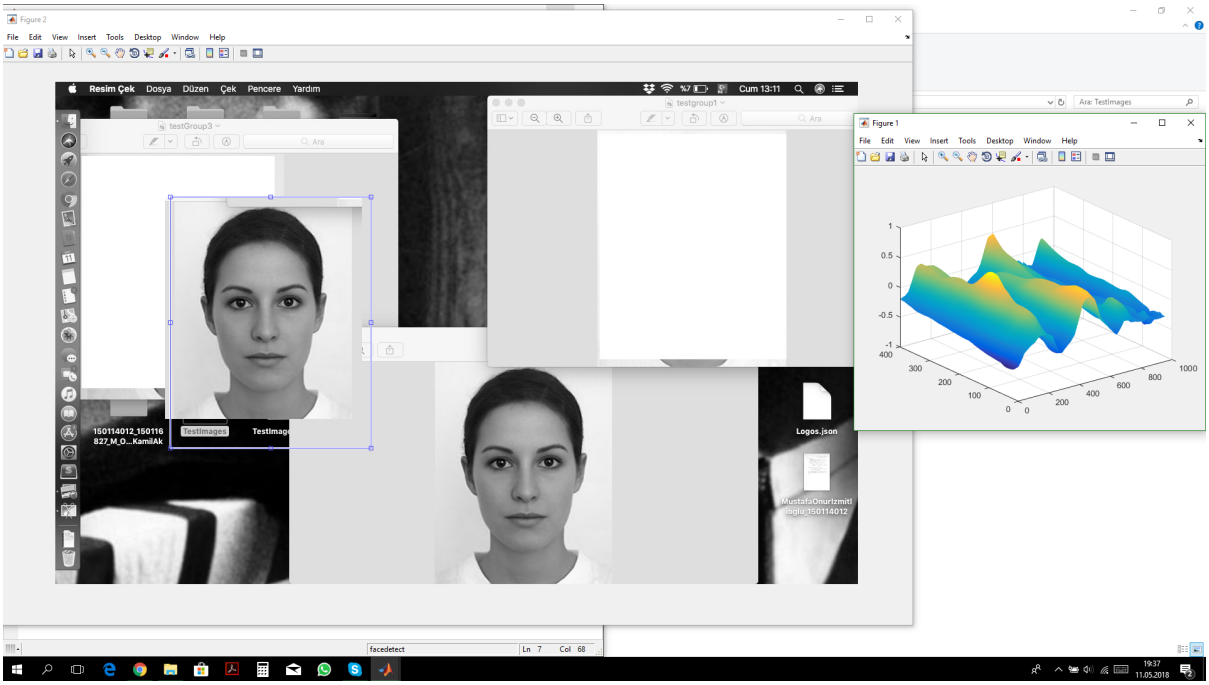
where “gca” is a MatLab draggable object. The 3D figures show the cross-correlation coefficient on each point (x,y) of the bigger images:











Performance & Conclusion:

In conclusion, after taking everything into consideration, we observed that the mathematical calculation of cross correlation between an average face image (320px, 400px) to a set of test images (each are approximately 1920px, 1080px) takes nearly 7 minutes in the PC which the experiment is run on. This run-time may change in different computers with higher performance, nevertheless this algorithm we applied would take a time complexity of O(n^2), since we perform a nested for loop for large number of iteration. To check and understand the performance of the code we implemented, we also implemented the built-in cross-correlation method in MatLab. Surprisingly, it took almost the same time to run the code, since the (m x n) calculation must definitely be performed.

Notes Regarding To The Implementation:

- The imrect() function draws rectangles with a few pixels of difference, this is caused by the fact that the test faces within the large images are not exactly 320px x 400px, and it is assumed that the dimensions are not scaled, this would take much more time to compute.
- The peaks in 3D figures correspond to the maximum correlation observed after running the MatLab code. Check the figures' (x, y) axes to understand better how this works.
- This algorithm extracts the maximum similarity. To detect multiple faces in an image, the algorithm can be re-constructed where the peak points (e.g. which are above 0.4) would output each face.