

Assignment – 2

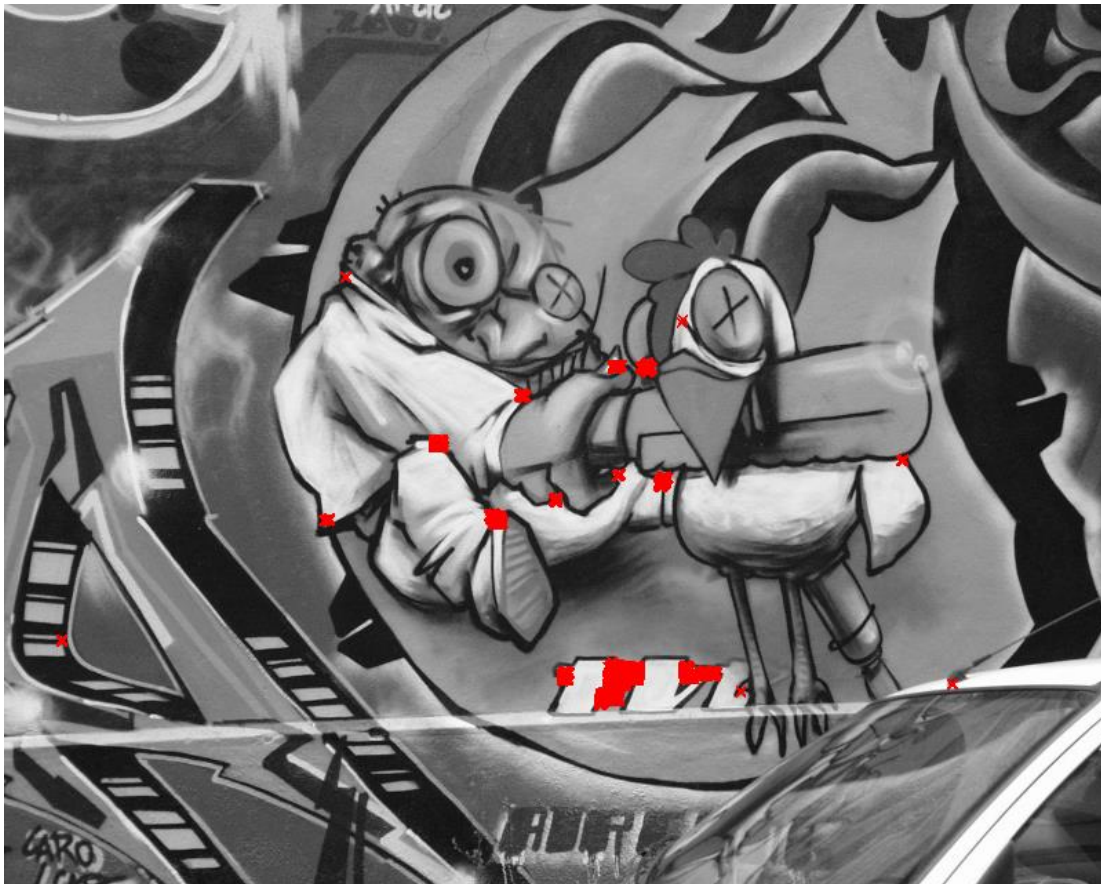
Elkhan Ismayilzada

[GITHUB LINK](#)

Task 1 – For the task 1, here are the configurations I made

- **Sobel kernel** – This is used for getting gradients of the image
- **11x11 patch** – This is used for constructing H (harris) tensor
- $H = \begin{bmatrix} \sum I_{xx} & \sum I_{xy} \\ \sum I_{xy} & \sum I_{yy} \end{bmatrix}$ - This is the construction of H tensor.
- $R = \det(H) - k * \text{trace}(H)$ – This is used for the response and **0.05** is used as **k** value

In the end I got following picture, the red dots are the features.



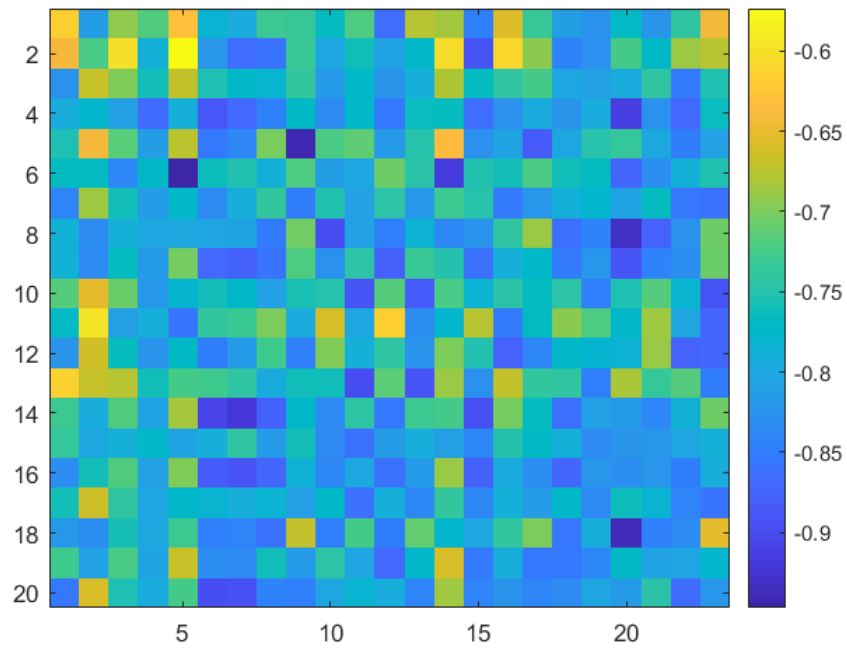
Task 2 – As you noticed from the picture, features are clustered together. To get rid of these clusters, I compared feature value with the features around it in a radius of 10 and if it is not the maximum value I discarded. After maxima suppression I got following picture



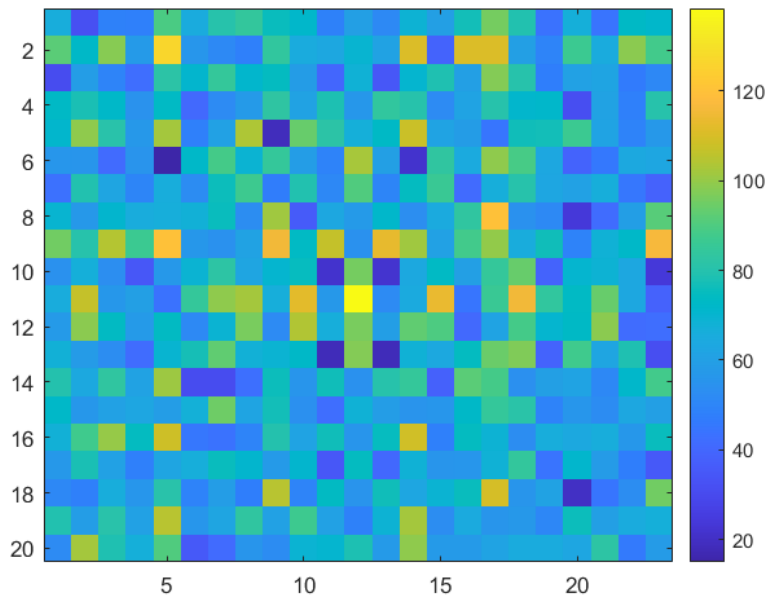
As expected, clusters are gone, and we have only one value at each feature.

Task 3 – In task 3, first I found the orientation of **61x61** patch of each feature of the image using the eigen vector of **H** tensor constructed as in Task 1. After finding the eigen vector, I found the angle with respect to X axis by simply computing inverse of arctan of the value $V(2)/V(1)$ where **V** is the eigen vector. After finding the angle I rotated the **61x61** patch and took **21x21** patch in center of **61x61** patch. For each feature of the first image, I calculated SSD and NCC scores with the features of second image and constructed a distance matrix and here are the results.

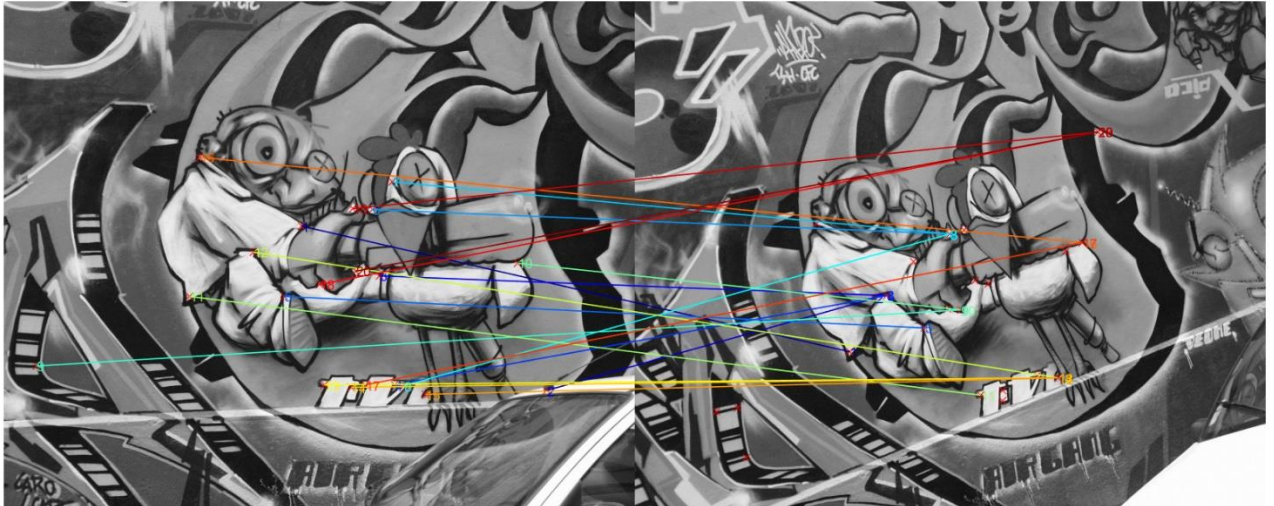
NCC distance matrix



SSD distance matrix



To check the similarity between two distance matrices, I negated NCC scores (so that the less score means less difference). As you noticed from the pictures, the NCC and SSD matrices are look alike with minor differences. However, I chose NCC score for the putative matches, and I got following picture.



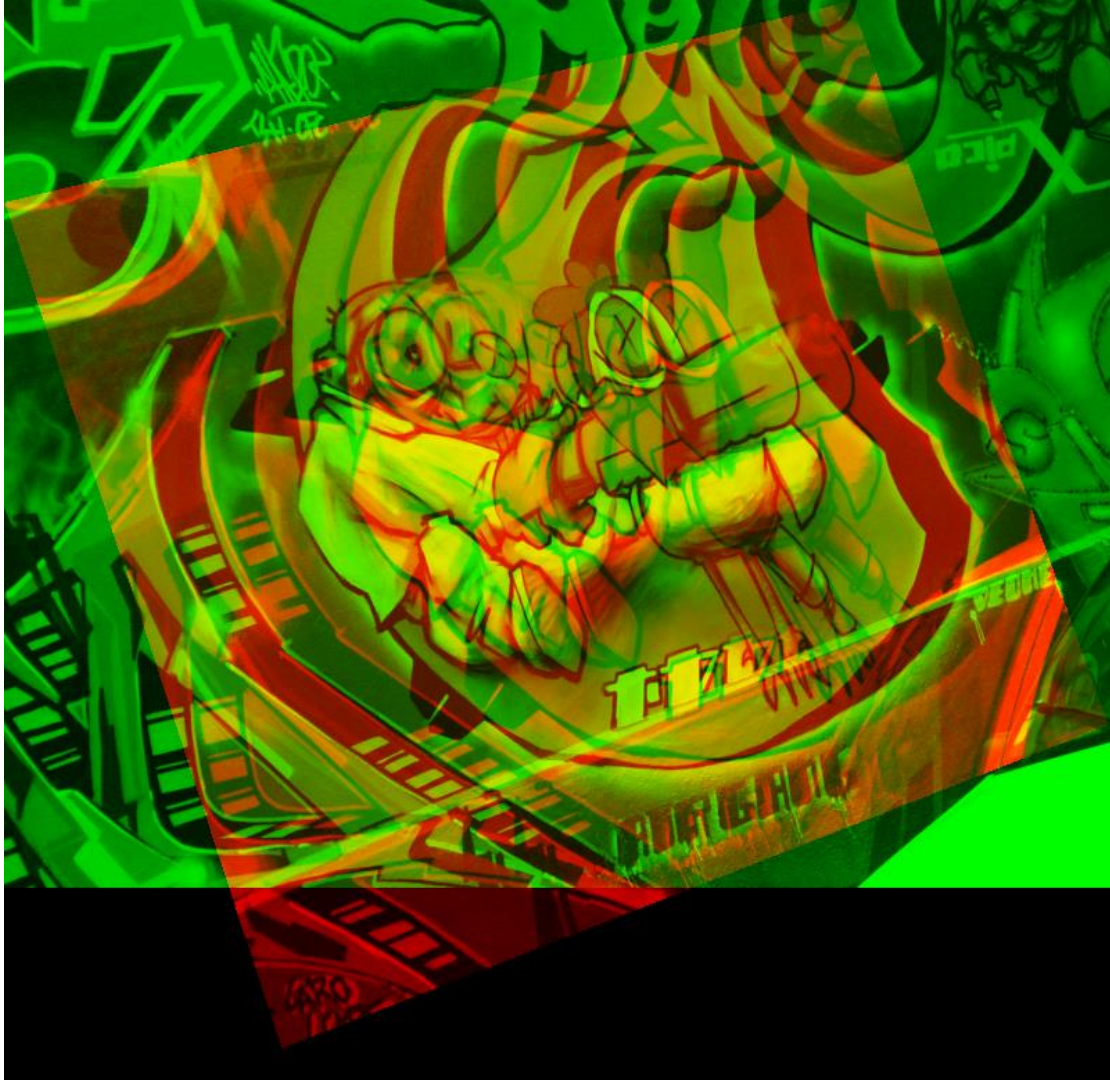
Task 4 – I used the same parameters for this task as in **Task 1** except that I did not select strongest 1000 points and I put threshold value (900) for response R. (in total more than 12000 points before maxima suppression). I used NCC score for selecting putative matches. Then I sampled 4 points randomly, computed homography matrix, transformed the feature coordinates of the first image by applying this matrix and computed Euclidean distance between the first image feature coordinates and second image ones. I considered the points inlier if the distance is less than 2 pixels. I ran this procedure for 1000 times and found approximately 30 inliers out of 122 total points. The equation to estimate iteration count is given by

$$N = \frac{\log(1 - p)}{\log(1 - \omega^k)}$$

Where p is the chance of point to be inlier, k is the number of sampled points, and ω is number of inliers divided by total points. Therefore,

$$N = \frac{\log(1 - 0.99)}{\log(1 - \left(\frac{30}{122}\right)^4)} = 1257$$

Therefore, in the next experiment, I ran it for 1257 times and found following picture



My estimations aren't perfect but at least it knows where to rotate so it does what it is supposed to do.

Task 5 - I have no idea...