# **Computer vision**

# **Assignment 3 report**

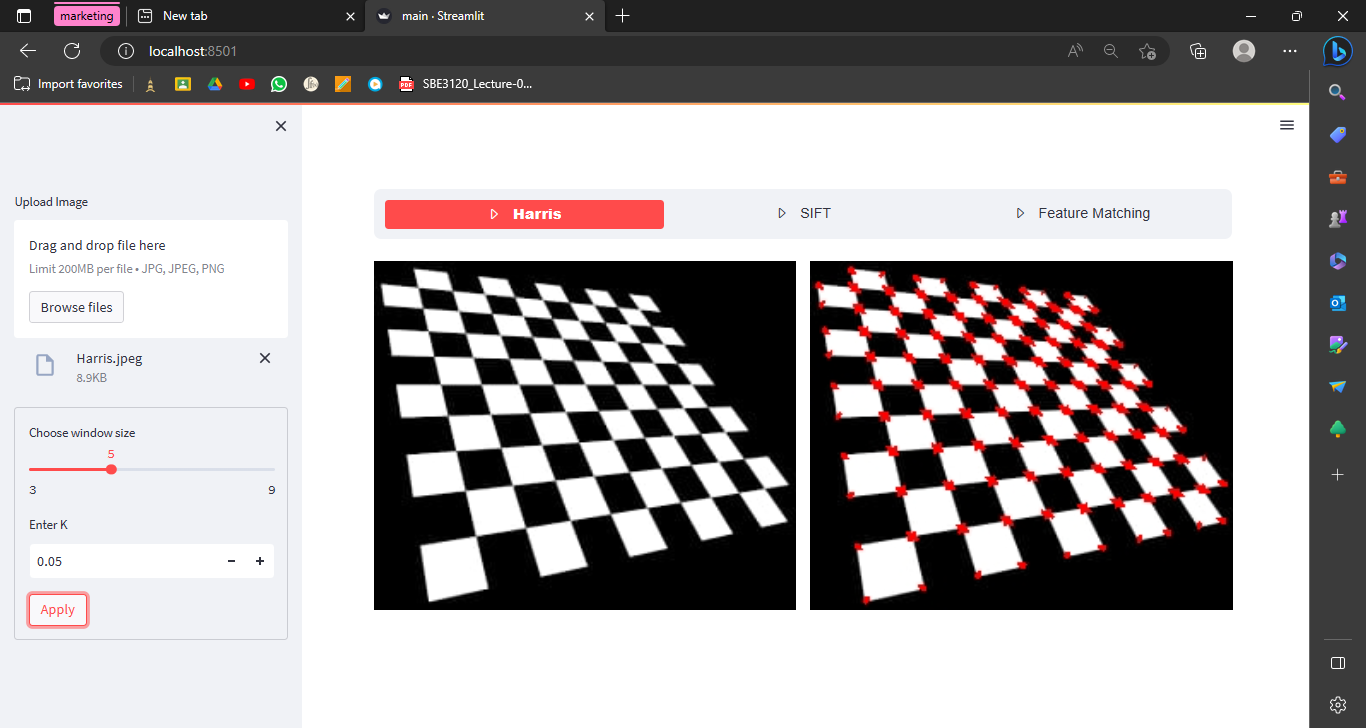
# HARRIS and LAMBDA operators

1. **Harris operator**

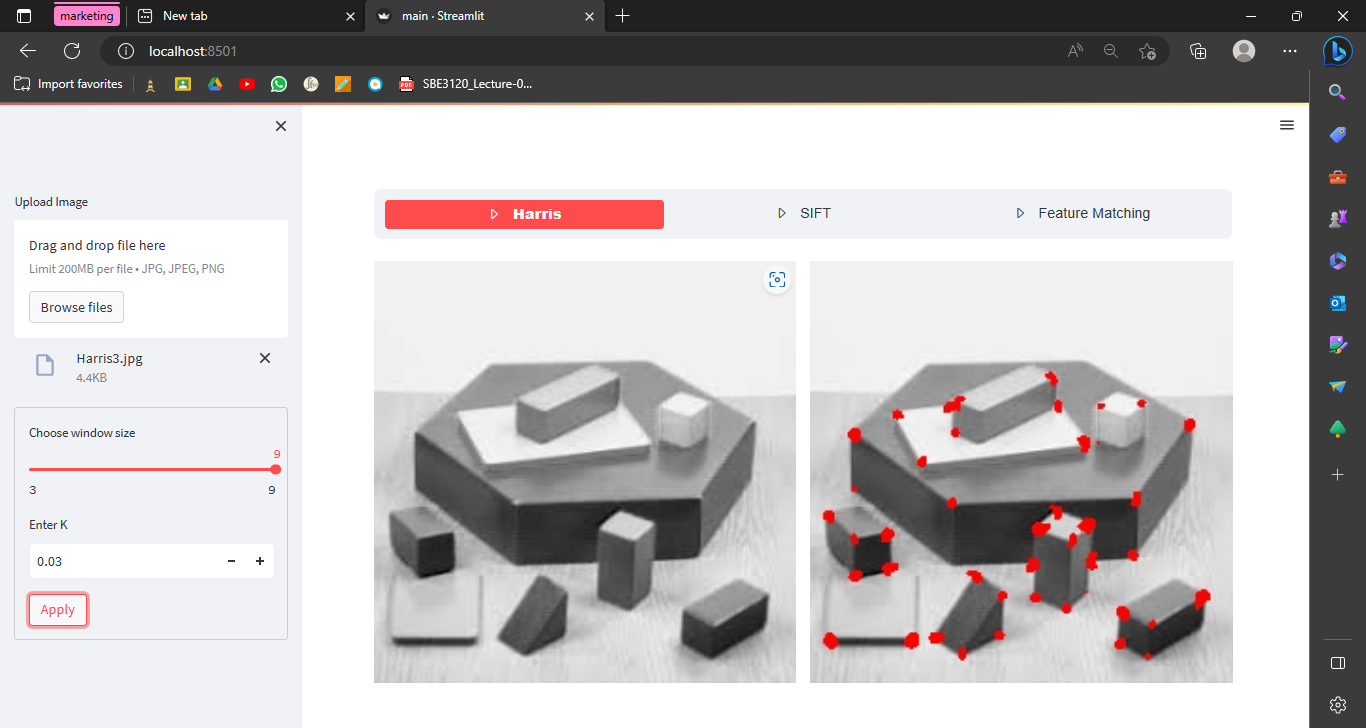
* Brief explanation:

- The Harris corner detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image. It was first introduced by Chris Harris and Mike Stephens in 1988 upon the improvement of Moravec’s corner detector. The Harris operator works by computing a score R for each pixel in an image. The score R is based on the eigenvalues of a matrix M that is computed from the image gradient. The Harris operator uses these eigenvalues to determine whether a pixel is located at a corner or not.

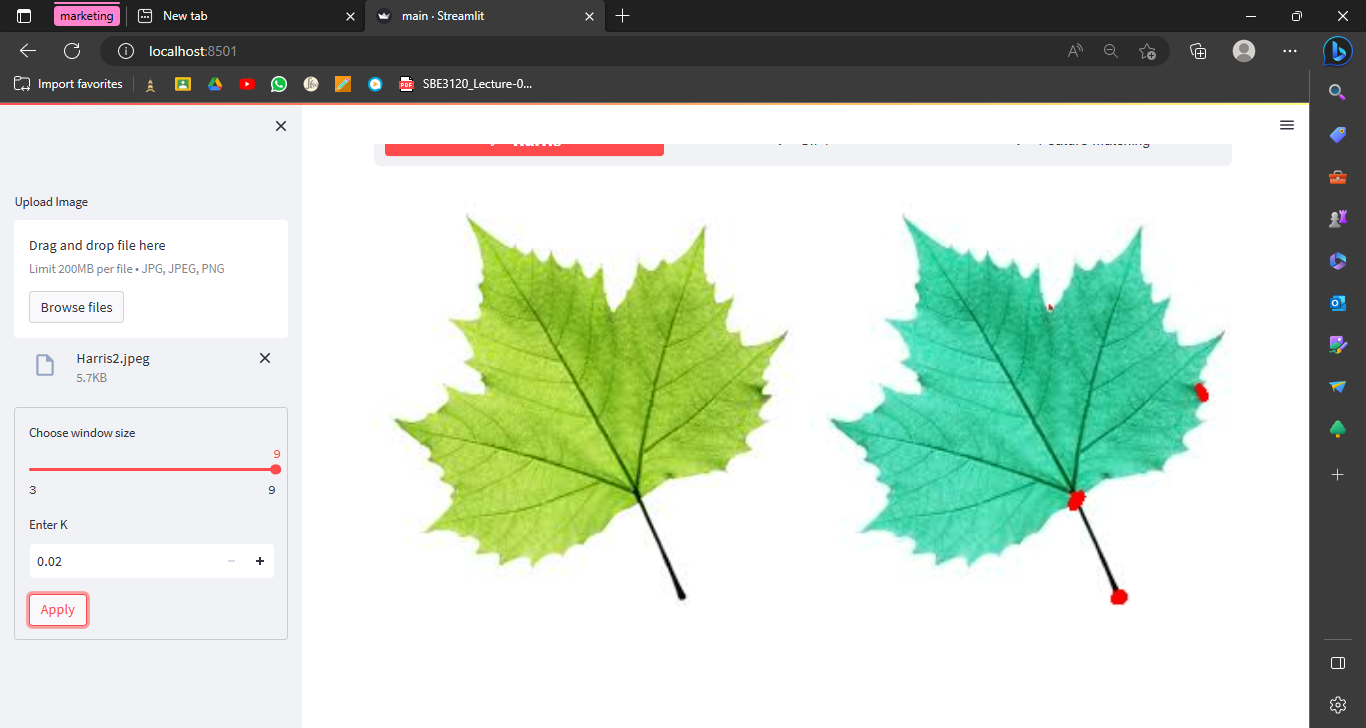
* Before and after applying Harris operator:



* Sample 2:



* Sample 3:



1. **Lambda operator:**

- It can be thought of as a variant of the Harris operator but it is much more computationally expensive because of the square root calculations.

1. Color to grayscale.

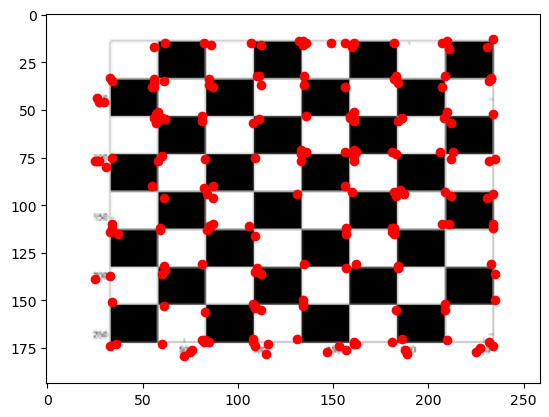
2. Spatial derivative calculation.

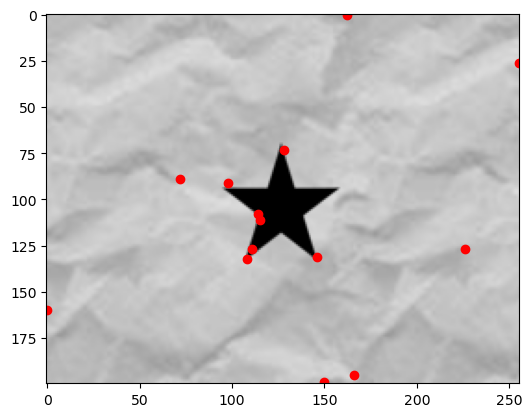
3. Create the H matrix from gradient entries.

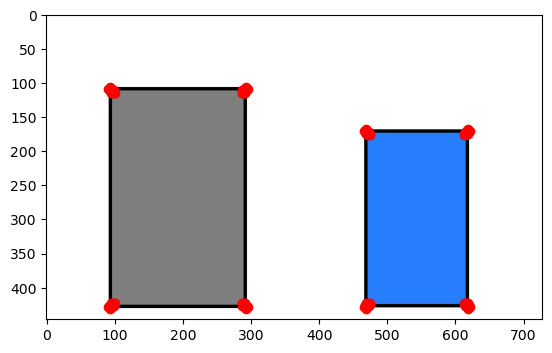
4. Compute the eigenvalues for each pixel.

5. Non-maximum suppression; we can also use local thresholding to choose our points.

* Some samples after applying Lambda operator:







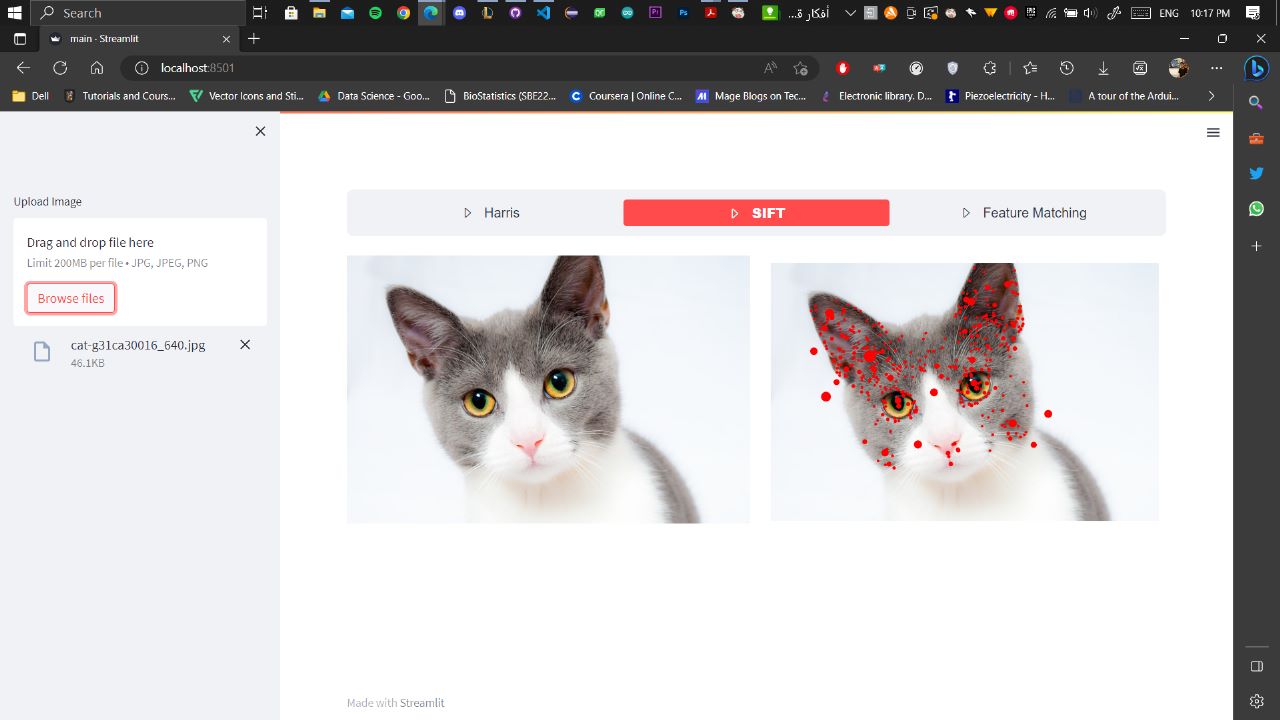
## Feature descriptors using scale invariant features (SIFT):

* Brief explanation:

The Scale-Invariant Feature Transform (SIFT) is a feature descriptor used in computer vision and image processing for object recognition, image matching, or classification problems. It was invented by David Lowe in 1999. SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. SIFT features are extracted from an image by first identifying key points or interest points using a difference-of-Gaussian function applied to a scale space representation of the image. Then, a descriptor is computed for each key point based on the gradient magnitudes and orientations of the image pixels surrounding it.

* There are mainly four steps involved in SIFT algorithm:

1. Scale-space Extrema Detection: This step involves detecting potential locations for finding features by identifying local extrema over scale and space1.
2. Keypoint Localization: This step involves accurately localizing the key points by fitting a model to the potential locations detected in step 12.
3. Orientation Assignment: This step involves assigning an orientation to each key point based on local image gradient directions2.
4. Descriptor Generation: This step involves computing a descriptor for each key point based on the gradient magnitudes and orientations of the image pixels surrounding it1.

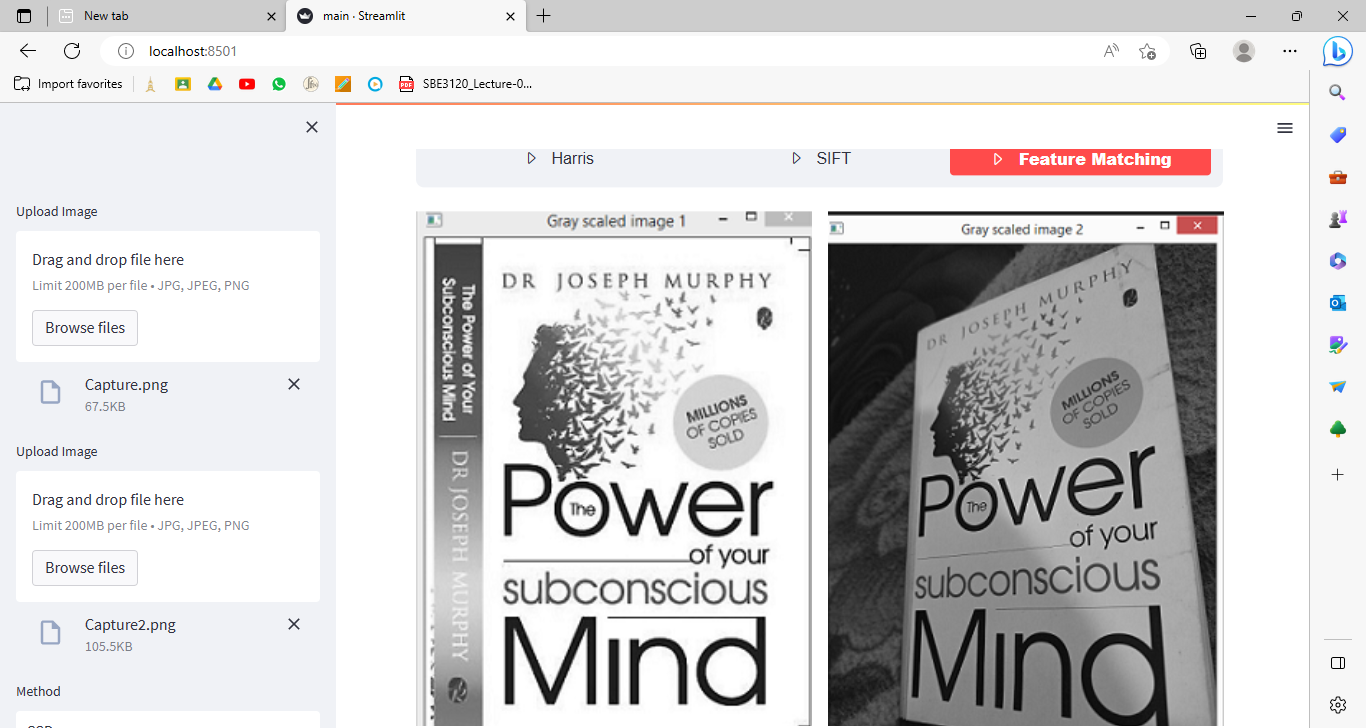


## Feature matching:

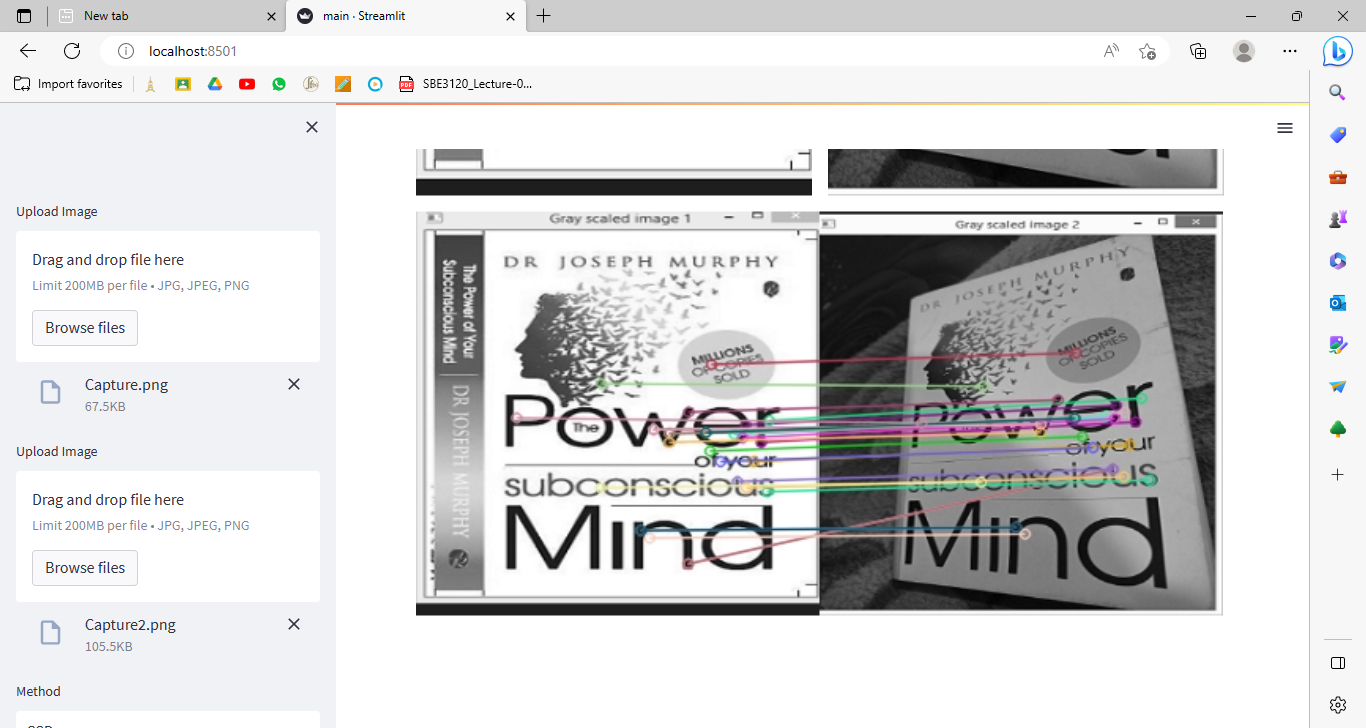
* 1st , using SSD method:

The basic idea of feature matching is to calculate the sum square difference between two different feature descriptors (SSD). So feature will be matched with another with minimum SSD value. SSD = ∑(v1 −v2)^2.

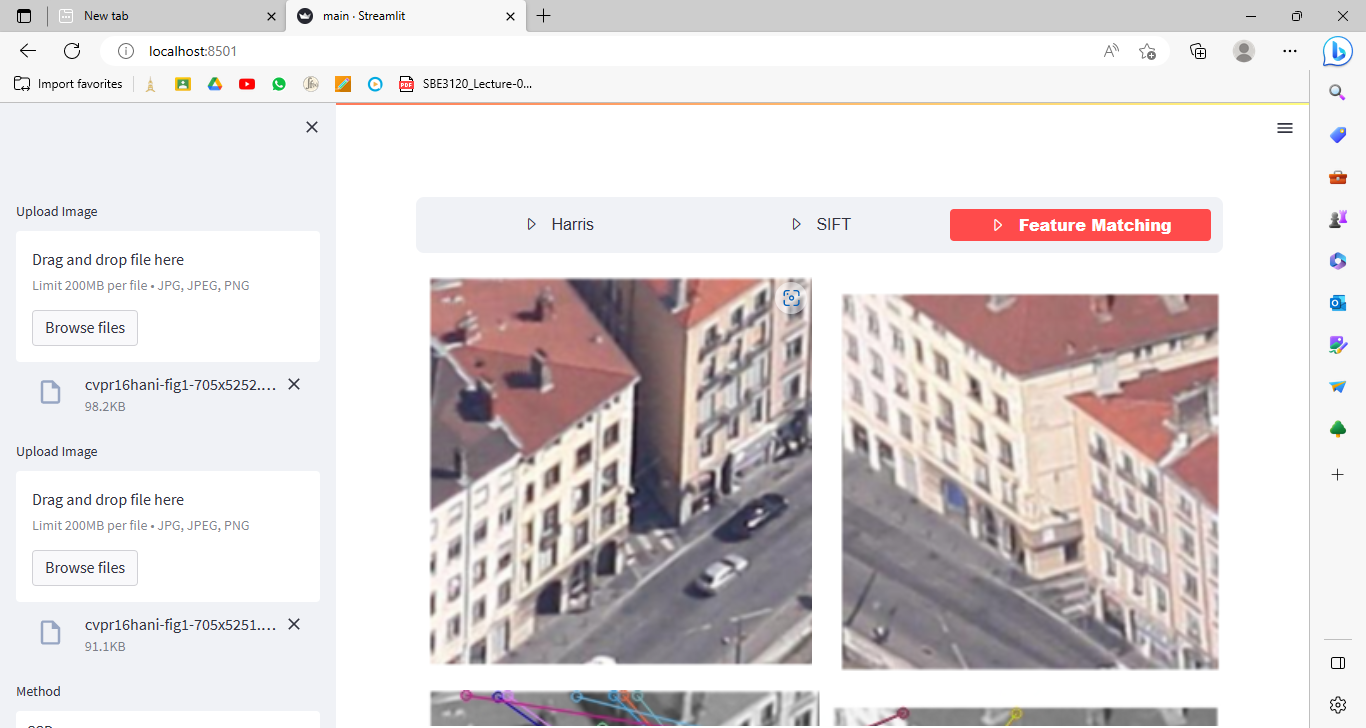
* Sample 1



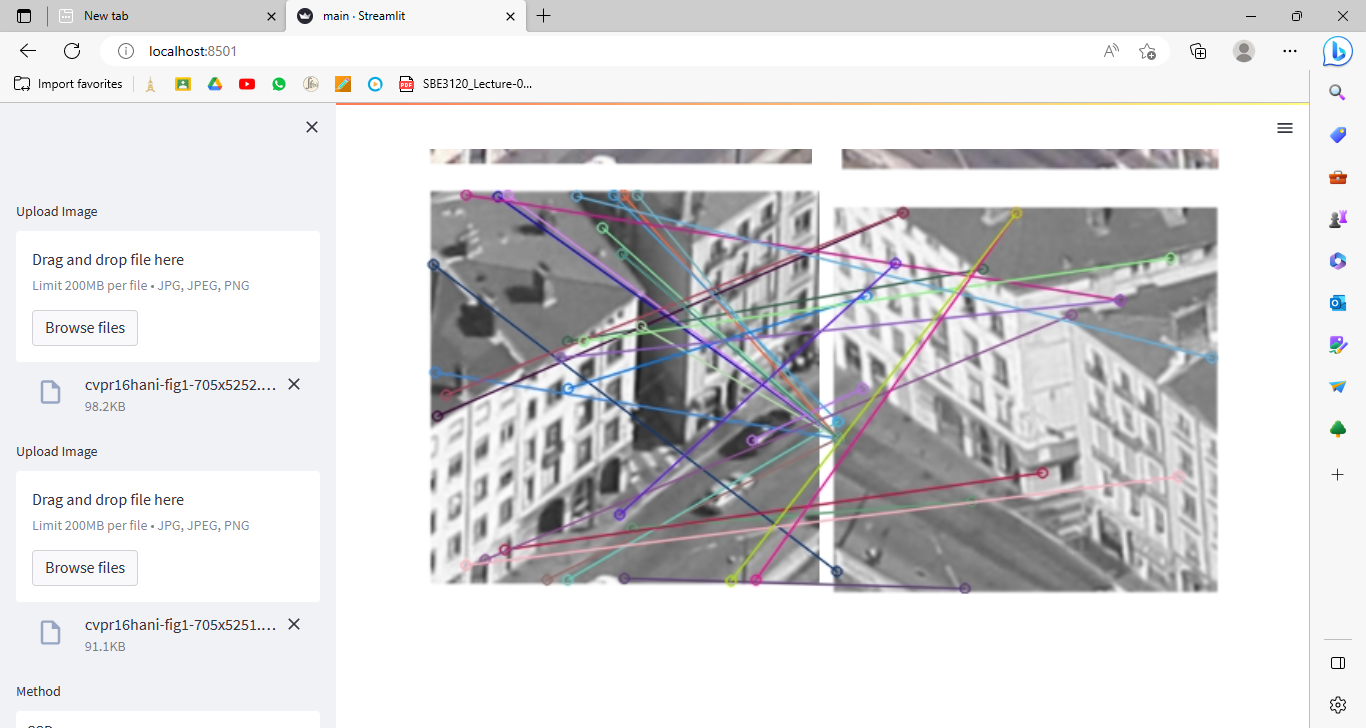
The images we will apply feature matching to



After applying feature matching (using SSD)



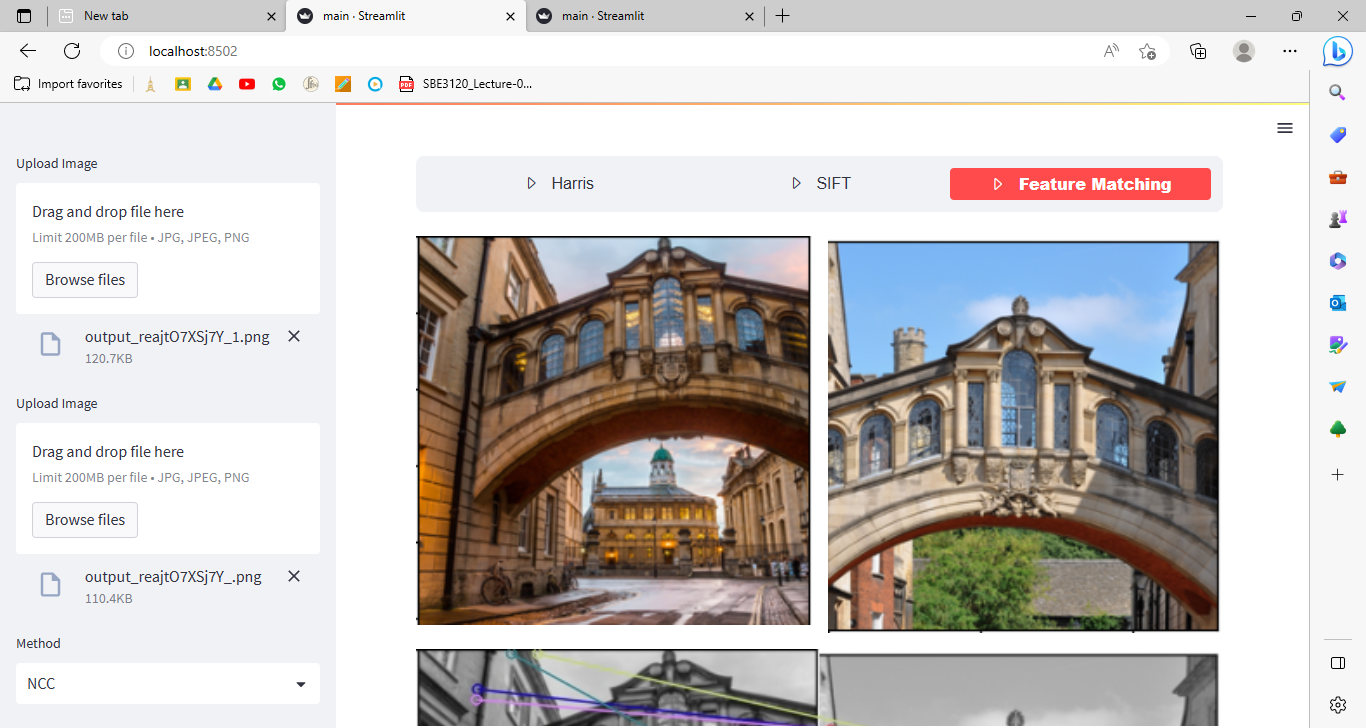
The images we will apply feature matching to



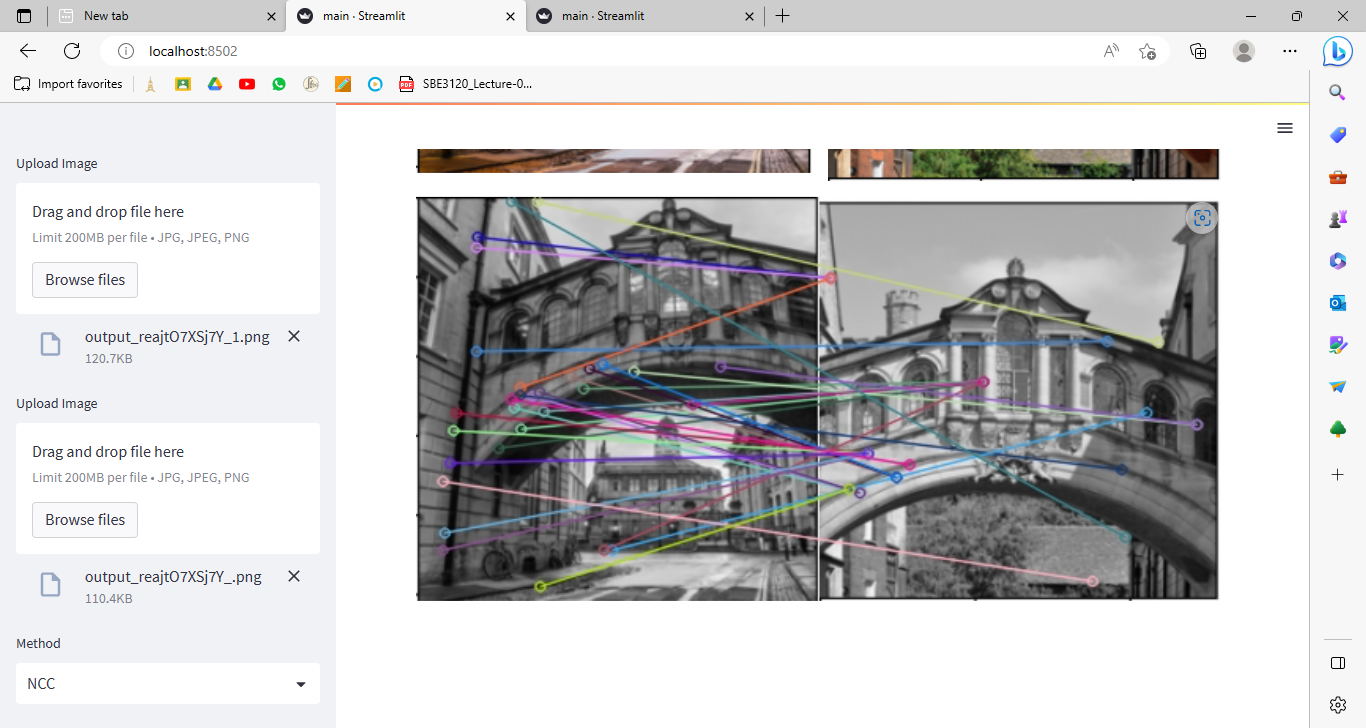
After applying feature matching (using SSD)

* 2nd, using NCC:

Feature matching using NCC (Normalized Cross-Correlation) is a method for matching feature points. In the traditional NCC method, it is necessary to calculate the correlation coefficient of every pair of feature points in the point set for many times. The points with correlation coefficients less than the threshold values in the point set are filtered successively to determine the final matching pairs. The matching result and speed of the calculation method are affected by the threshold value and the number of feature points.



Before applying feature matching



After applying feature matching

* **Note please**: the computational time is calculated in the code for each image/case, as it varies from one image/mode to another.