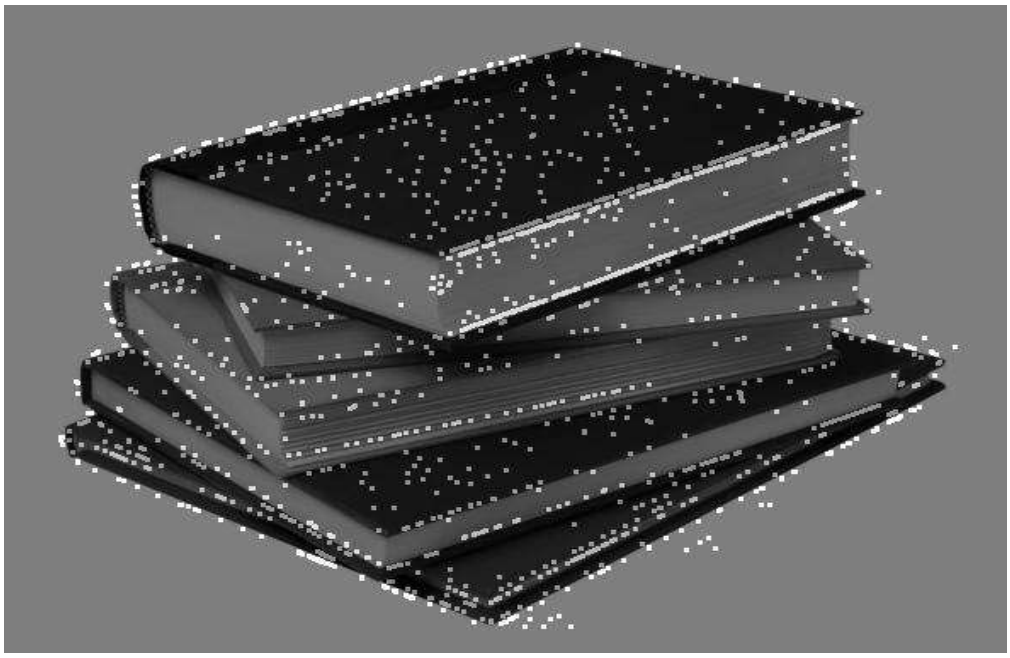
IMPLEMENTATION

- The 2 images on which features have to be found are uploaded to colab from drive.
- These images RGBA type and hence converted to grey scale to process them for features. My image pixel values are in range 0 to 1, hence my highest pixel value is 1.
- Two functions are created
    - octave
    - extrema_points
- **octave()** – This function takes in image, its row and column as input and returns the features of each scale as a list of arrays. It first applies Gaussian filter on images and then find Difference of Gaussian between the scales and then it calls the extrema_points function that returns "*feature matrix*" for each scale. Then it appends these matrices in a list and returns it to user.
- I have used number of octaves as 1 and number of levels as 10(because of 1 octave I am increasing number of levels in that octave) and sigma of Gaussian as 1.6. Hence the number of feature matrix becomes 7(Number of DOG output = 9 and then top most and bottom most scale space images are not considered for extrema detection hence 7).
- Feature matrix –The features detected in each scale are represented by 1(one can use 255) in this matrix and rest points are zero, to increase the visibility to human eye a 3x3 neighbourhood around detected feature is marked as 1(one can use 255).
- **extrema_points()** – It takes lower level, mid level and upper level scale space images, row and column as input and calculates the minimum and maximum points across the mid level scale space image with respect to a 3 x 3 size filter across all the 3 images around the point and returns matrix of all such extrema points. But these points(1 pixel wide) are not visible to naked eye hence a 3 x 3 region around the extrema points are given high value(1 or 255) to make it visible to naked eye.
- Then for any image the octave function is called(inside of which extrema_points function is called) and it returns back a matrix containing extrema points(high pixel value in extrema point and its 3 x 3 surrounding).

- All the features of the different scale spaces are combined into one matrix and shown out in the original image. To make the features visible over the image the overall feature matrix is added with image matrix.

For normal image (without any modification) the number of features for book image is and for building image is 1483 and for building image it is 6538.

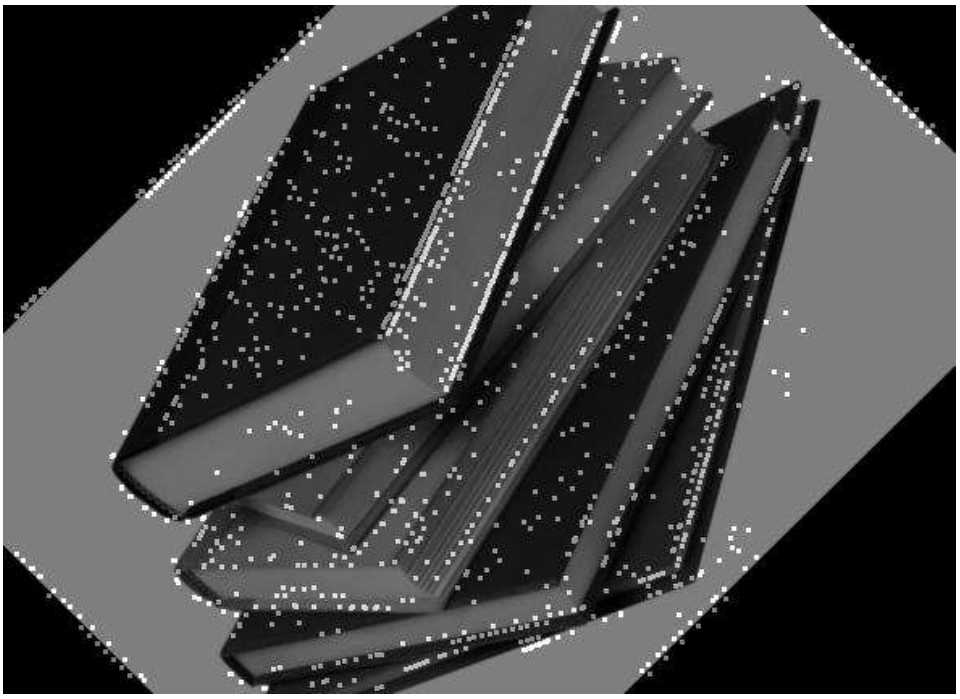
Feature on book image:



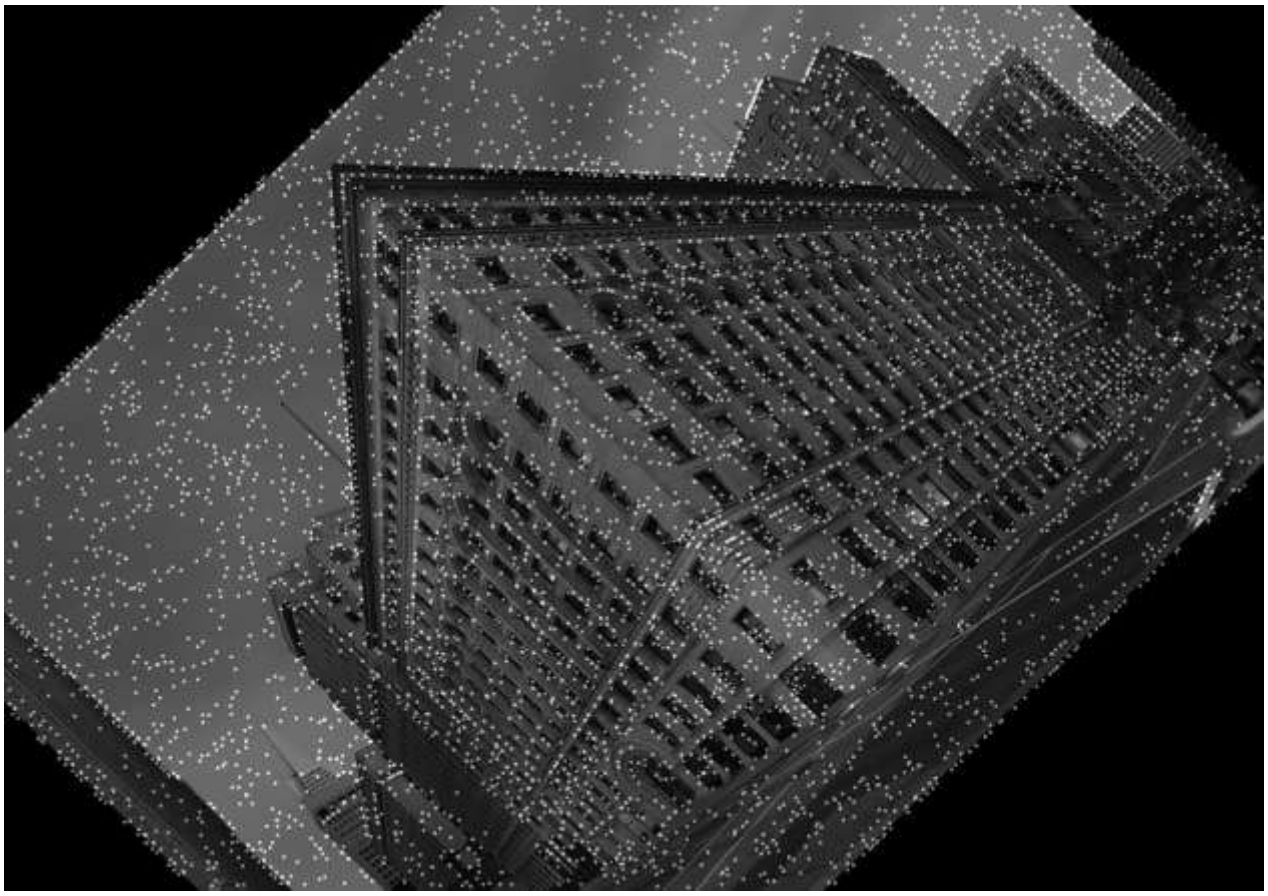
Feature on building image:

For rotated image(45 degree)) the number of features for book image is and for building image is 1353 and for building image it is 5857.

Feature on rotated book image:

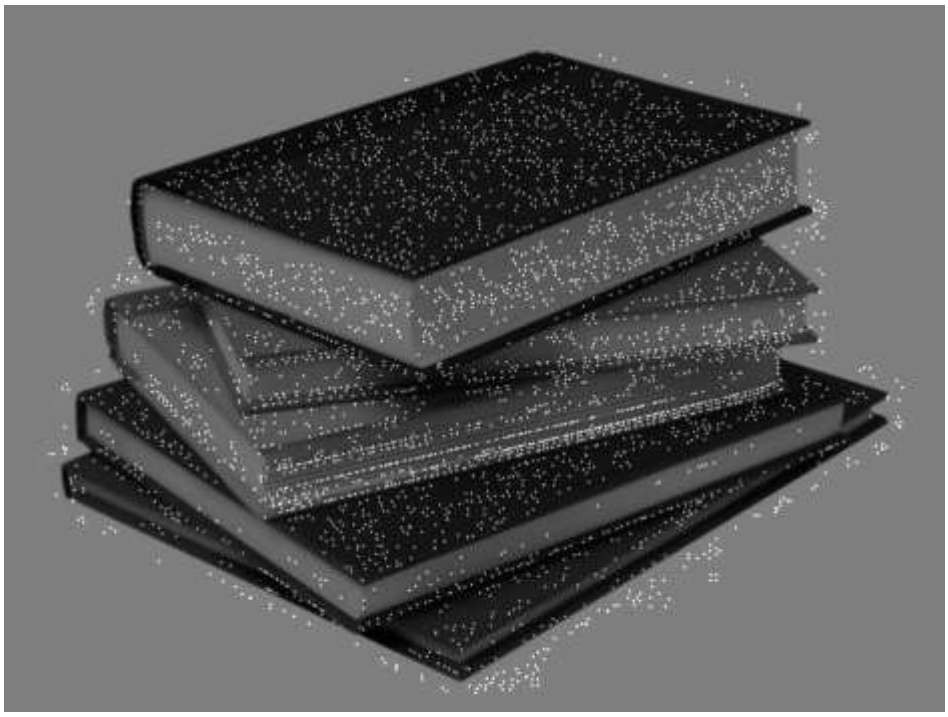


Feature on rotated building image:

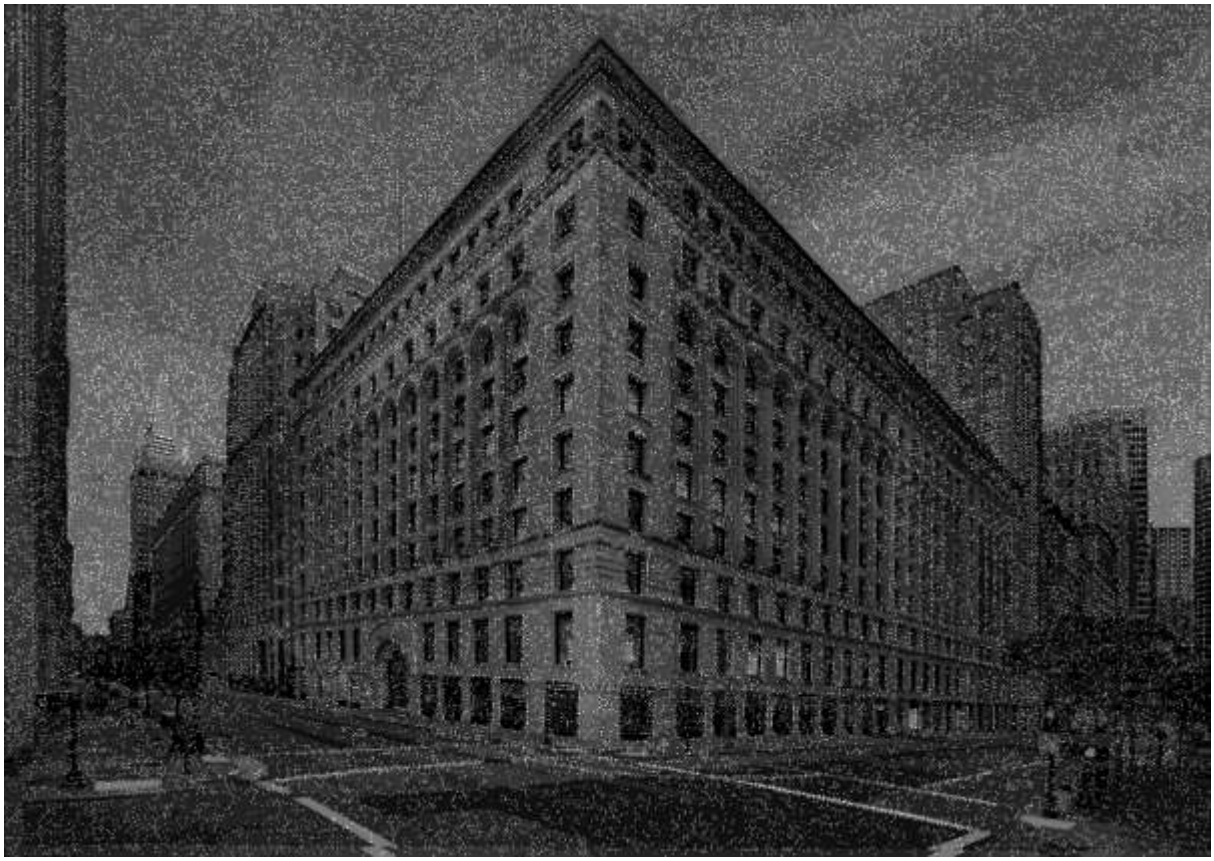


For upsampled image(factor of 2) the number of features for book image is and for building image is 3555 and for building image it is 39884.

Feature on upsampled book image:

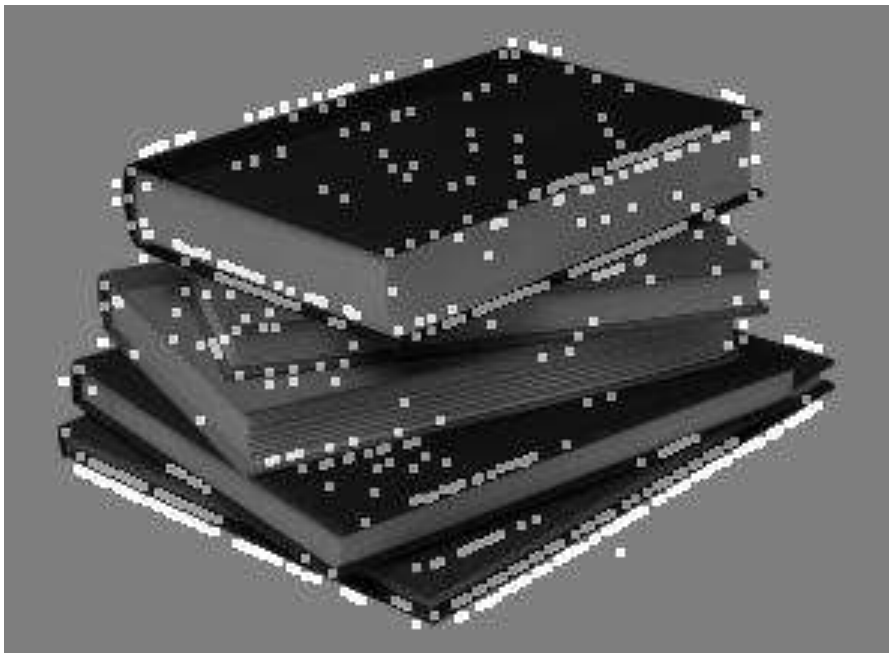


Feature on upsampled building image:



For downscaled image (factor of 2) the number of features for book image is and for building image is 584 and for building image it is 1854.

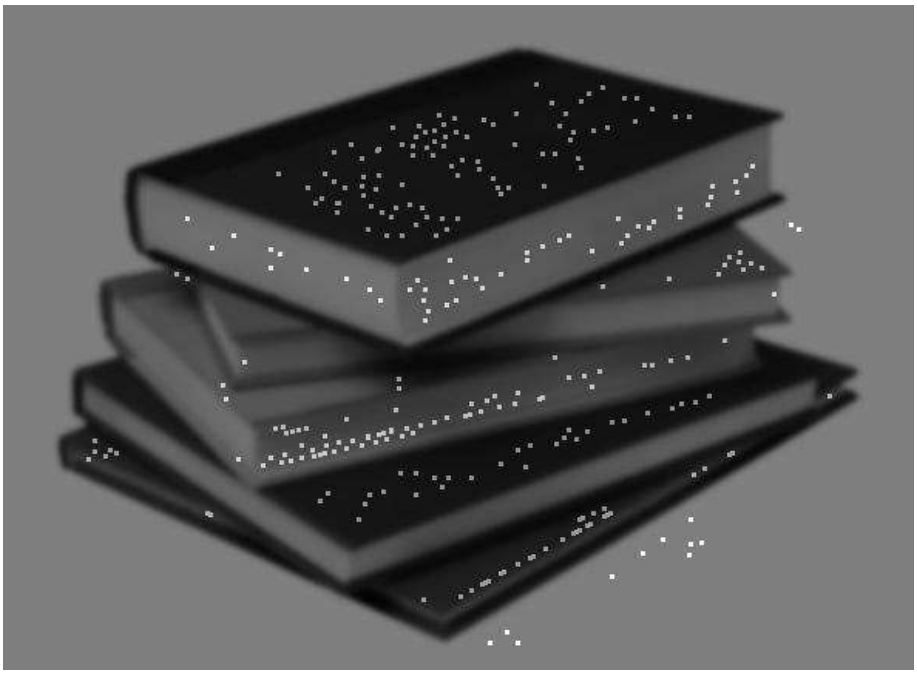
Feature on downscaled book image:



Feature on downscaled building image:

For Gaussian blurred(of sigma = 2) image the number of features for book image is and for building image is 293 and for building image it is 5108.

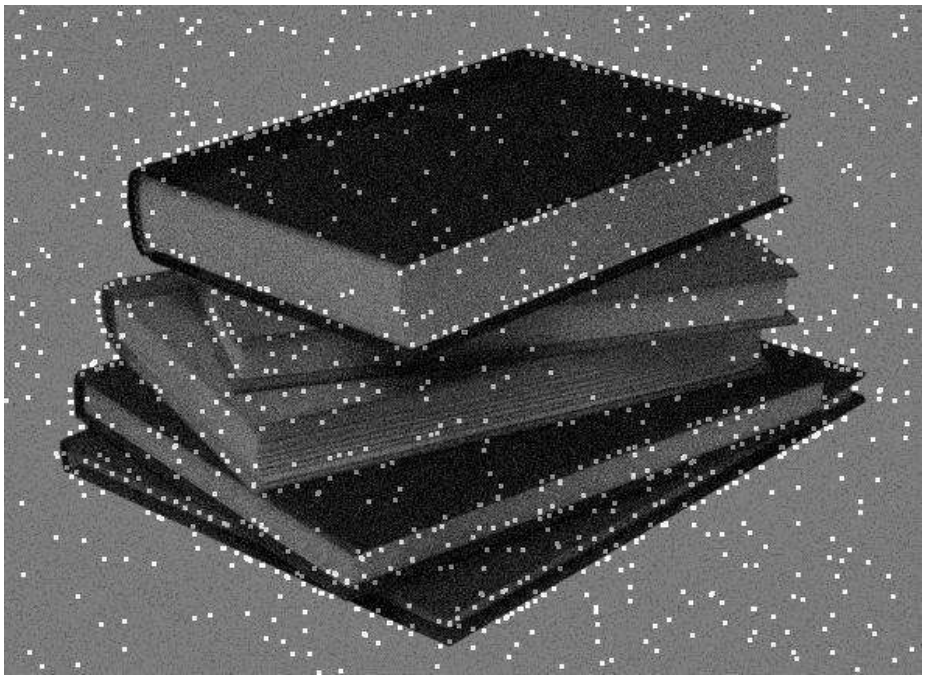
Feature on Gaussian blurred book image:



Feature on Gaussian blurred building image:

For Gaussian noise added image the number of features for book image is and for building image is 1106 and for building image it is 4603.

Feature on Gaussian noise added book image:



Feature on Gaussian noise added building image:

RESULTS AND ANALYSIS

From the above images we can draw the following inferences:

- In the book image (without changes) most of the features are observed in the edges and corners. In building image (without changes) we observe a lot of features this is because there are a lot of corners and edges in the buildings like windows and doors. Still there are many unimportant features(like features in sky) that can be removed in the second step of SIFT.
- In the rotated image (by 45 degree) also the number of features are almost same as original image (for both book and building) and more or less at same points except at the edge of rotation.
- In the upsampled image the number of feature is more compared to original images. Building image has a lot of features detected some of which I believe will be removed in the 2^{nd} step of SIFT.
- In the down sampled image the number of features detected is less as number of pixel is also less for comparisons (for both building and book). Features are located more or less at the same parts.
- In Gaussian blurred image the number of feature detected is less for the book image but for building image the number of features detected is nearly equal(slightly less) with respect to original image.
- For Gaussian noise image the number of features detected is more for book image. Earlier features in original book image were detected mainly around corners and edges and not in empty space but now after addition of Gaussian noise they are detected everywhere. But for building image the number of features has decreased with comparison to original image.

PROBLEMS FACED

The first problem I faced was while converting ‘rgb’ image to ‘grey scale’ as this image was ‘rgba’ image so it took me a while to debug the error.

The second problem I faced was the features were not visible after finding extremas and adding with images, then I rresolved it by taking a 3x3 matrix across each extrema and giving them a high value of 1.