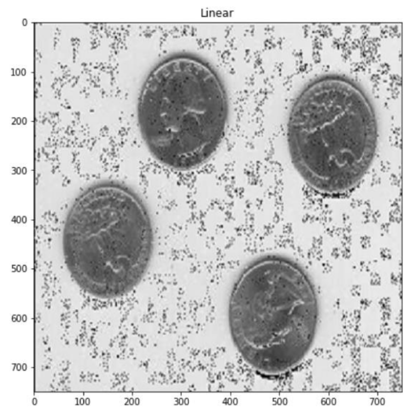
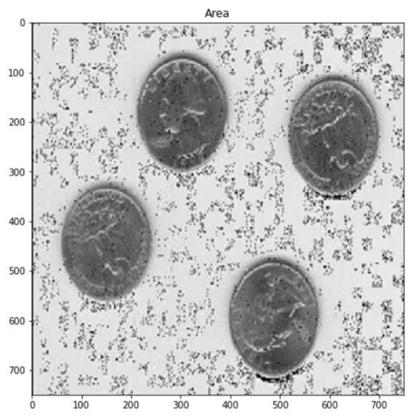
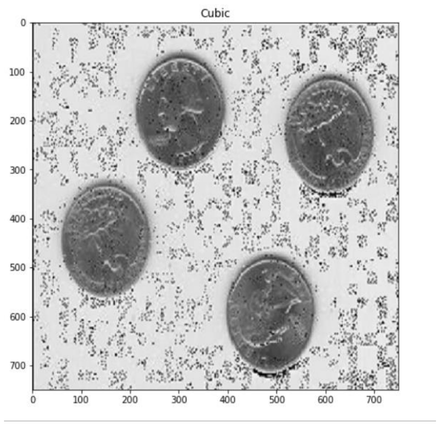
Geometrical spatial operations

Scaling

In our report, we utilized the cv2 library extensively to accomplish our desired tasks. We explored various interpolation methods for scaling purposes. Specifically, in our exercise, we employed linear, area, and cubic interpolation techniques. Linear interpolation utilizes bilinear interpolation, while area resampling is based on pixel area relation. Cubic interpolation, on the other hand, employs bicubic interpolation. The obtained results showed minimal differences discernible to the naked eye.



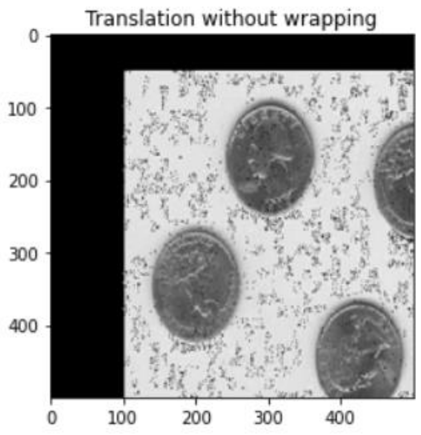


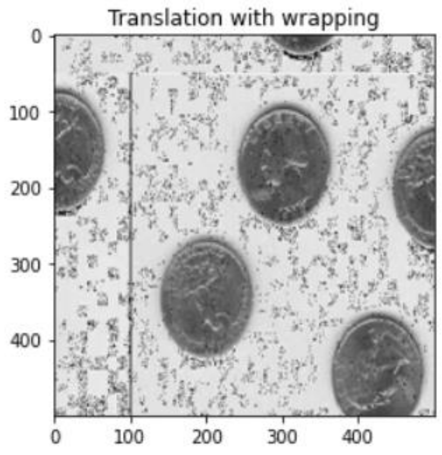


Overall, the area interpolation performed slightly better, as it is also the preferred method for scaling with factors greater than one.

Translation

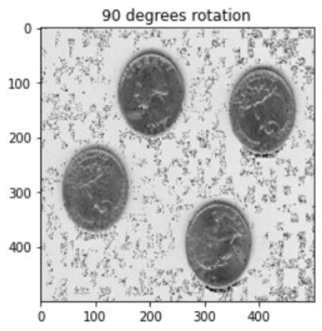
Translation is essentially shifting, implemented through matrix multiplication. We experimented with two approaches: wrapped and unwrapped. In the unwrapped version, we simply applied the shifting, while in the wrapped version, the matrix's elements that extend beyond the borders wrap around to the opposite side. Below, you can observe the results.





Rotation

Rotation was also conducted using the cv2 library. We rotated the picture 90 degrees clockwise, and the resulting image is as follows.

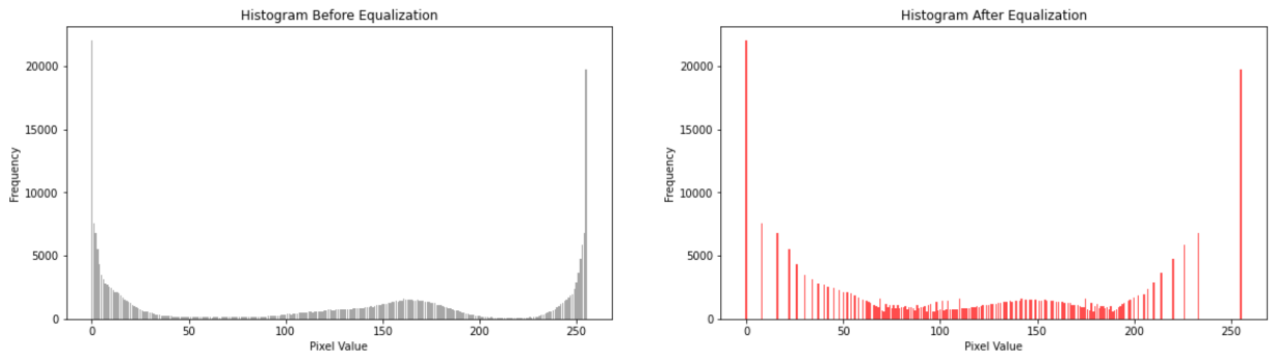


Histogram equalization

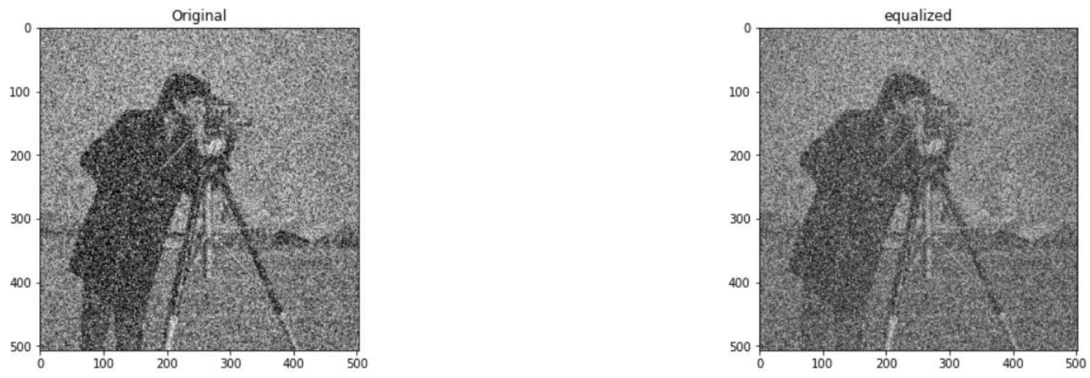
Once again, we employed the cv2 library to initially compute the histogram of the original image. Subsequently, we applied histogram equalization to normalize the frequency distribution of pixels. Since our images are in grayscale, our histograms have only one dimension. However, histogram equalization did not yield significant improvements in our data. This was likely due to the high contrast present in our images, as evidenced by the histogram chart where very low and very high pixel values were the most frequent.

Before and after equalization

Histogram



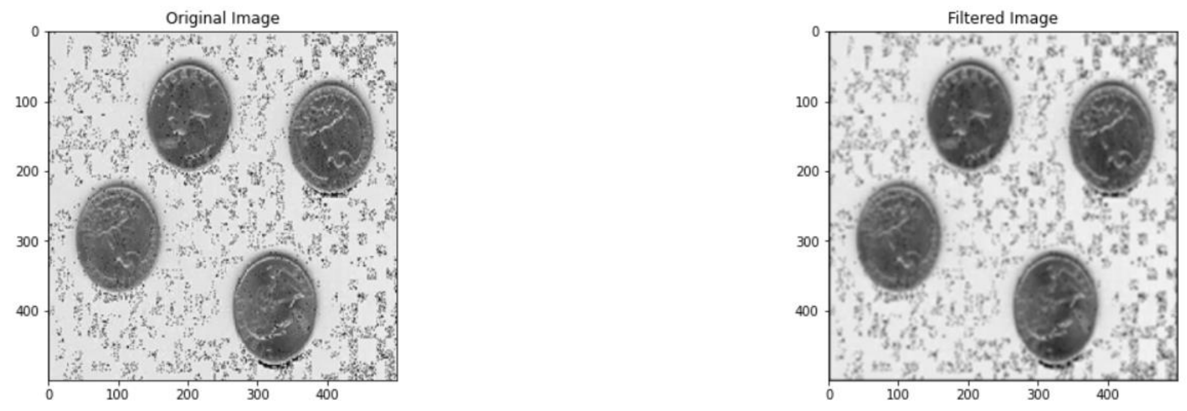
Image



Filtering in Spatial Domain:

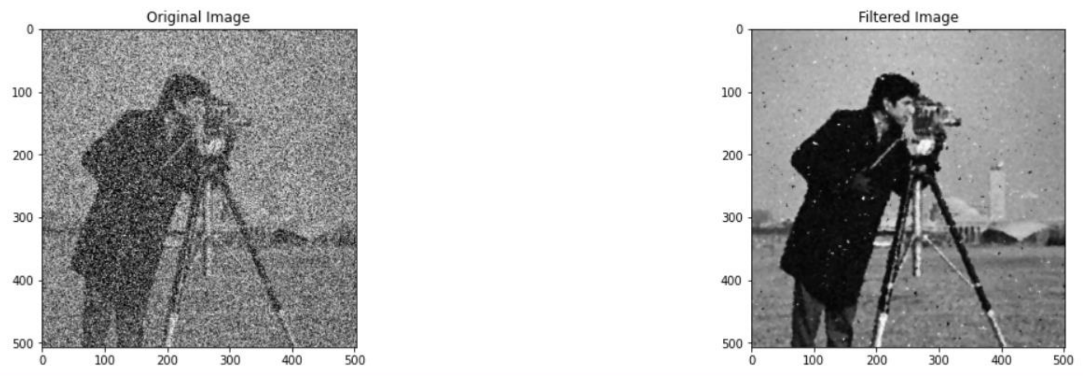
Average (Mean) filter

As we couldn't find suitable libraries for average filtering, we resorted to convolution with a matrix of the desired dimension (5 in our case), normalizing it so that all entries are 1/25. Average filtering, as expected, functions similarly to a blurring operation, which can be confirmed by the following result.



Median filter

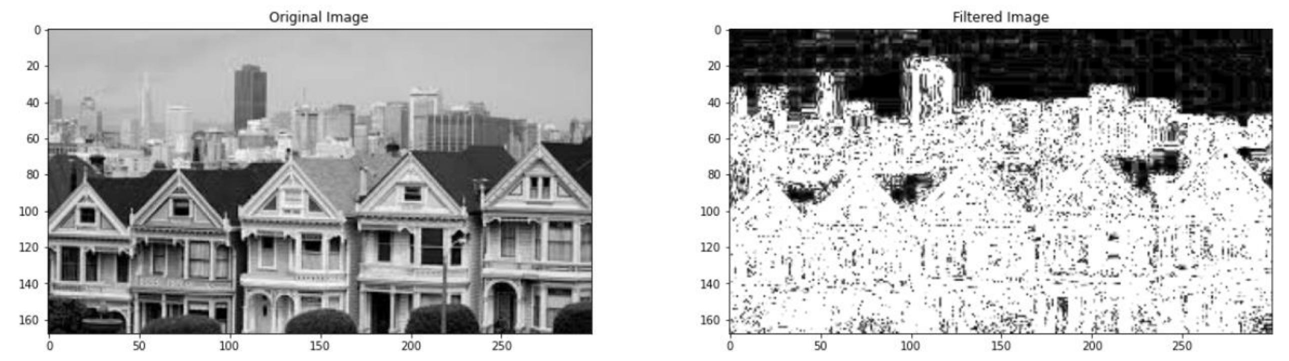
We utilized the ndimage module from the scipy library for this filtering process, and the results were remarkable. The rationale behind this is that when we introduce noise to an image, we typically add noise from the same distribution. If our data has sufficient contrast, this wouldn't significantly alter the ordering. Therefore, by replacing each pixel with the median value in median filtering, we can obtain an image with significantly reduced noise.



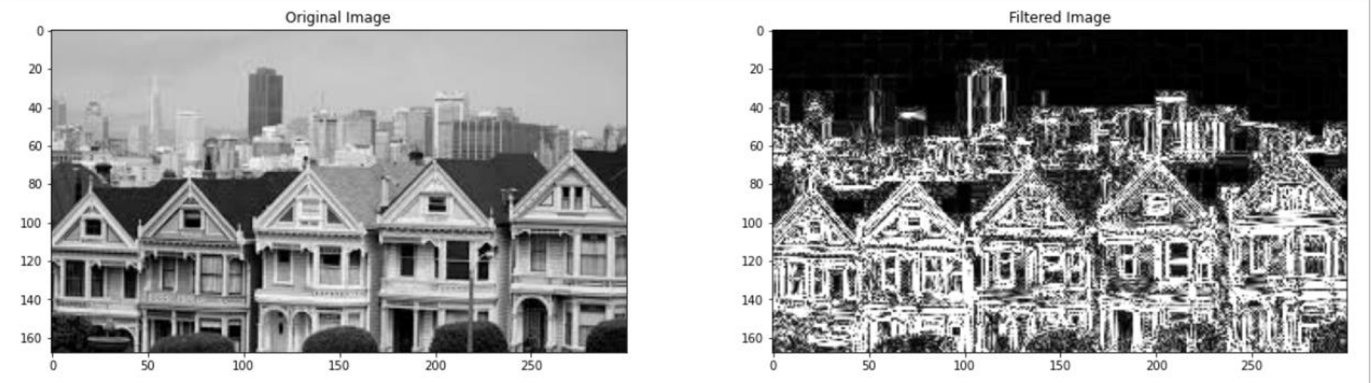
Laplacian filter

The Laplacian filter is designed to detect edges by identifying sudden changes in pixel values. It achieves this by computing the second derivative. In image processing, discrete approximations of the Laplacian operator are used due to the discrete nature of digital images. The Laplacian filter is applied to each pixel in the image by convolving the image with a Laplacian kernel. The Laplacian kernel is a small matrix that approximates the second derivative operation. However, the results of this filtering were not satisfactory, primarily due to our choice of a large window size. Retrying the process with a smaller window size yielded better results in edge detection.

K = 5:



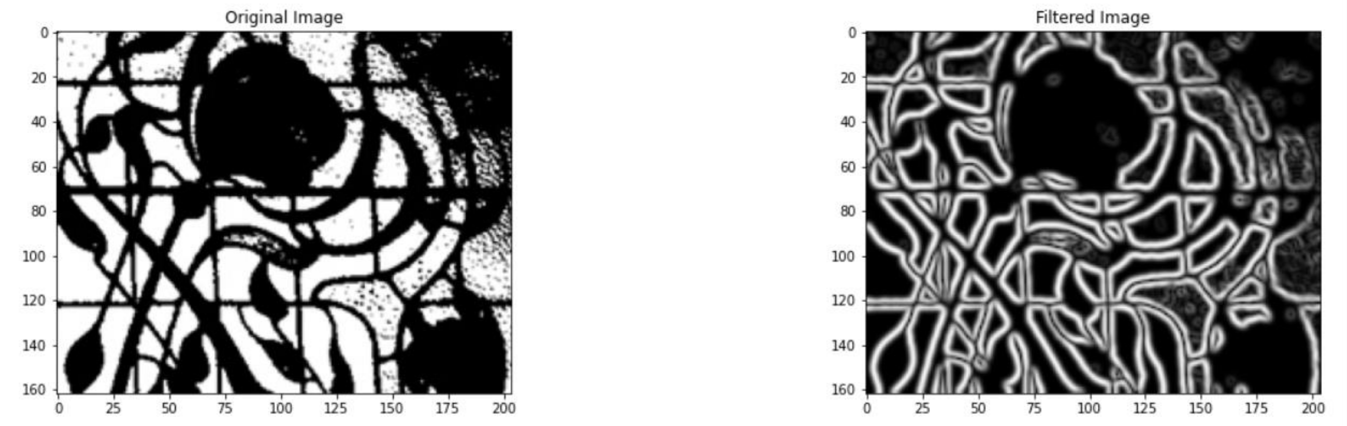
K = 3:



Sobel filtering

Sobel filtering is a widely used technique in image processing for edge detection. It aims to highlight edges in an image by computing the gradient magnitude, which indicates the rate of change of intensity, at each pixel location.

The Sobel operator consists of two separate kernels (one for computing the gradient in the horizontal direction, and one for the vertical direction). These kernels approximate the derivative of the image intensity with respect to the x and y spatial dimensions. The result is as follows, which was very successful in detecting the edges.


Original Image
Filtered Image