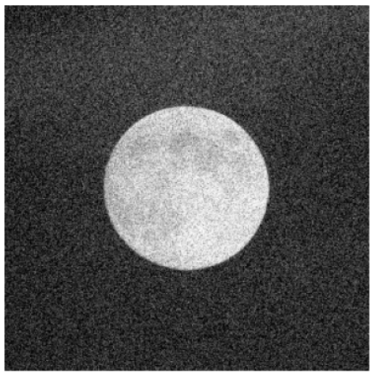
Digital Image Processing
Assignment 2

Tejash More
M.Tech SP

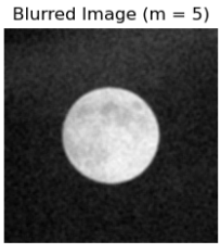
Packages used: NumPy, SciPy, Pillow, Matplotlib

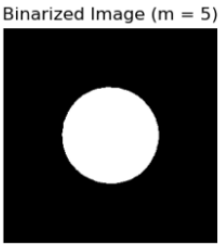
1. Spatial Filtering and Binarization

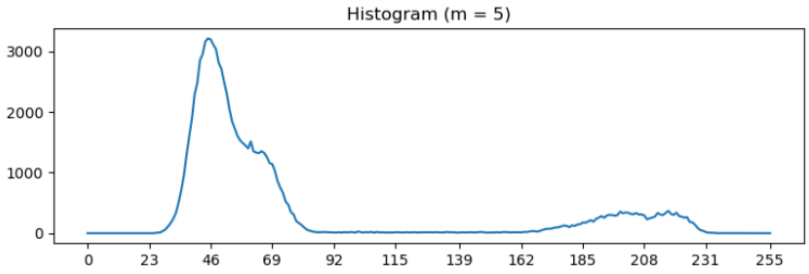
This is the grayscale converted version of the input image.



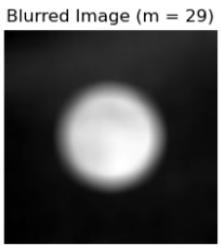
- For m = 5:
  - Optimal Threshold: 126
  - Least Intra-Variance: 169.775

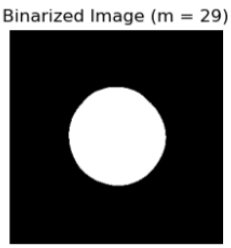


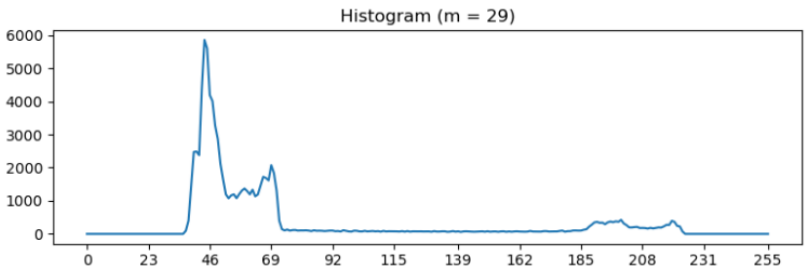




- For m = 29
  - Optimal Threshold: 120
  - Least Intra-Variance: 287.466



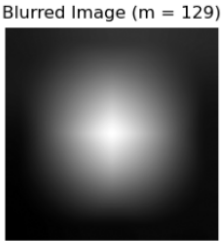


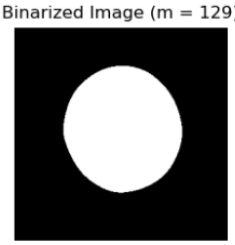


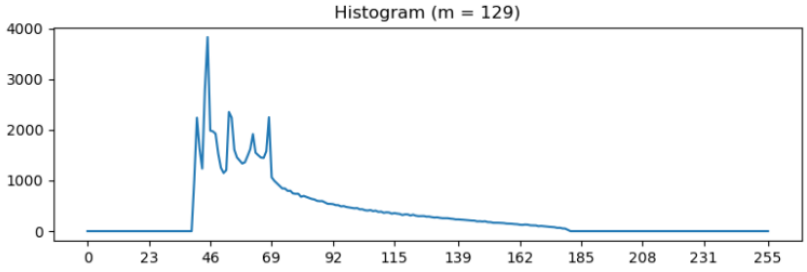
- For m = 129
  - Optimal Threshold: 91
  - Least Intra-Variance: 270.689

Digital Image Processing
Assignment 2

Tejash More
M.Tech SP







Box filter used is:

$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$
$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$
$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$
$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$
$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$	$1/m^2$

Observation:

For m = 5, It gave the least intra-class variance out of the 3 filters used.

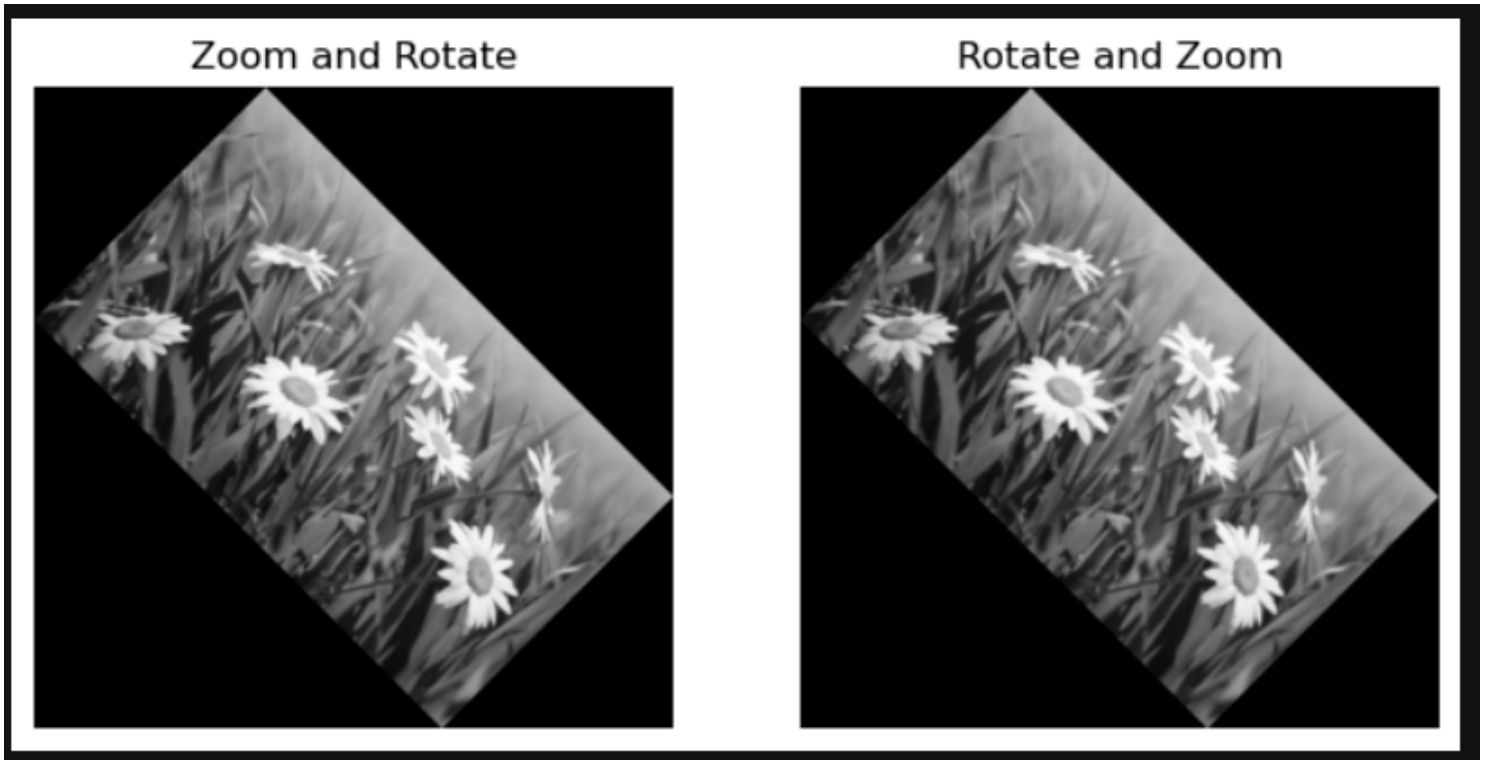
Now, difference between the intra-class variance between m=5, and m=29 is 117.69. Now, Difference between their Optimal Threshold is only 6, that is, in terms of Binarization, they are very close still there is huge difference in intra-class variance for them. One possible reason can be, because of Blur, on the edges of moon, there is a vast range of pixel values in the blurred region, which increases the overall $\sum (k - mean)^2$.

This shows that Intra-Class Variance alone is not a good measure to determine the optimal Box filter size to reduce the noise.

2. Scaling and Rotation with Interpolation

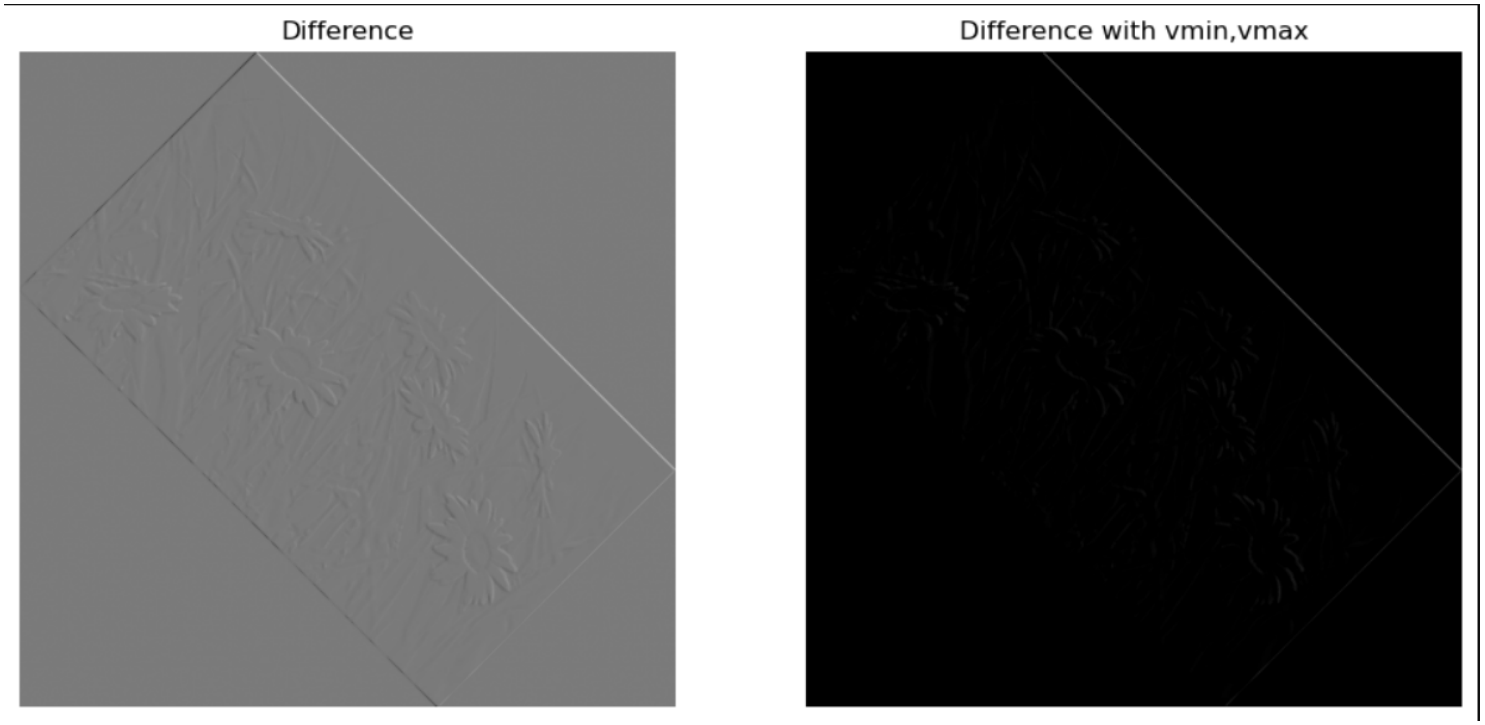
Digital Image Processing
Assignment 2

Tejash More
M.Tech SP



Task A **Task B**

Analysis of the Result:



In Task A, Image is Upsampled by 2x and then it was Rotated. Interpolation during Rotating was done on an Upsampled image where there are more detailed pixels present, whereas In Task B, Image was rotated first and then it was Upsampled, so Interpolation was done on a small image with comparatively lesser details present.

Now, the difference betweent them:

Digital Image Processing
Assignment 2

Tejash More
M.Tech SP

1. For smooth regions will be very close to 0 as not much detail present for the interpolation to give different values
2. For detailed regions like edges, pixel values vary greatly and hence interpolation is more likely to give different values for them.

Difference is grayish because of matplotlib.pyplot.imshow.

MinPixel is -178 and MaxPixel is 250. Imshow maps the least pixel value to 0 and greated pixel value to 255. Therefore, the smooth regions where the pixel value is close to 0 is grayish in colour.

2nd image where vmin and vmax is used was to clip the pixel value at 0 and 255. Since the algebraic operation on images can lead to values outside the [0, 255] range. Thus, all the pixel value <= 0 are clipped to 0 and then it was plotted.

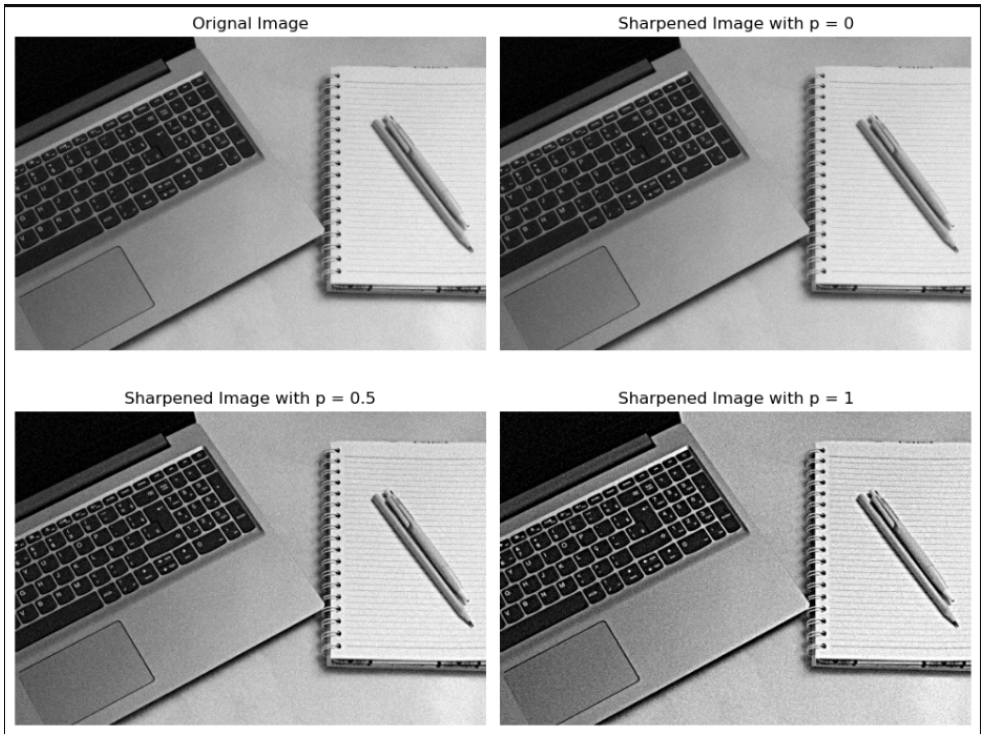
Some Problem I faced was:

a. Image was getting cropped during rotation. To solve this, canvas of the rotated image was determined using the 4 corners of the image.
b. Another decision making was, since the canvas dimension was coming in float, ceil value should be taken, but using ceil, dimension of the both tasks were not matching. Task A was giving (887, 887) and task B was giving (888, 888) due to the fact that (ceil(x) *2) == ceil(x * 2) is not true for all x. So, instead of Ceil, Floor value was taken to make the dimension match and to avoid padding of 1 pixel on one side of the image.

Conclusion: Scaling and Rotation operation are NOT commutative

3. Image Sharpening Concept

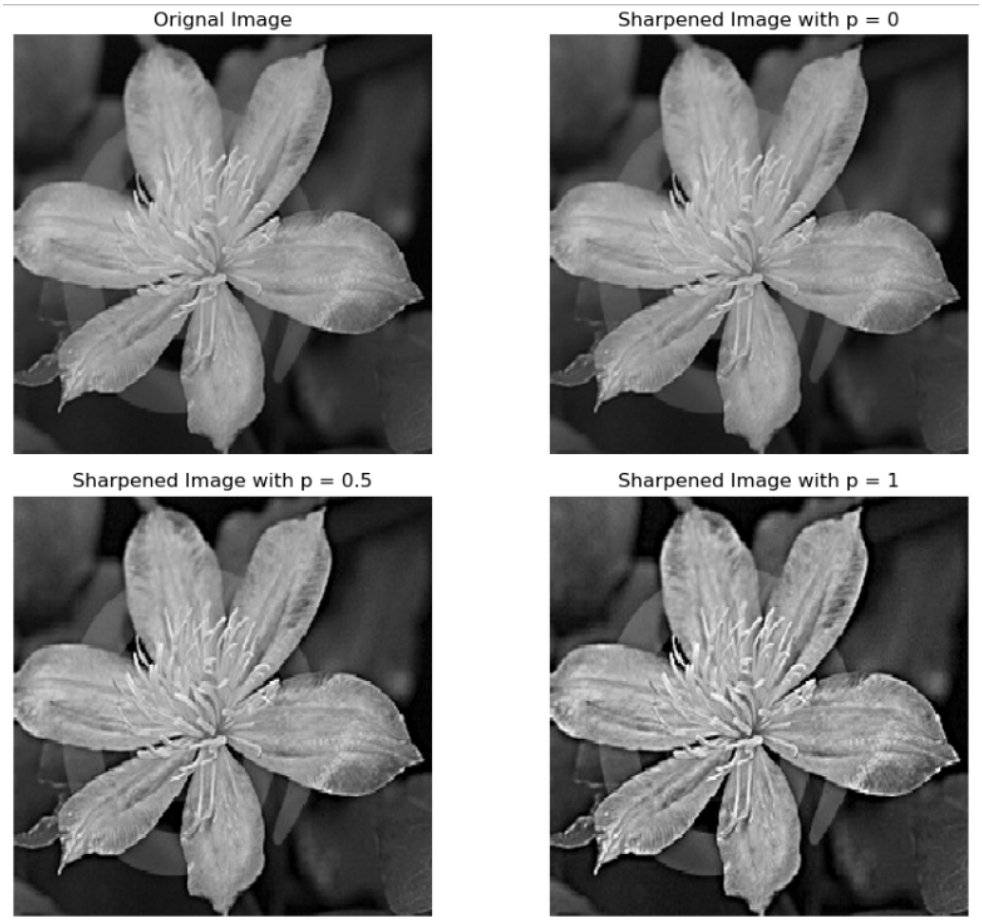
. Image is sharpened using Unsharp Masking with p values [0, 0.5, 1] to show the different levels of Sharping. Filter used is Box filter of 11x11 size.



Digital Image Processing
Assignment 2

Tejash More
M.Tech SP

Same has been done in some more images as well, for example.



When p = 1, Images are very much clearer than the Original Image. What sharpening does is, it increases the pixel values of the edges, making them look more visible. But, if extremely high-power sharpening is applied, then it will also highlight the noises present in the image, which in turn sacrifices the quality of the image.

Impact of filter is also there, the bigger the filter size is, more powerful the mask of the image will be, which in turn increases the sharpness of the image as well.

Some problems I faced were:

a. Mask is the difference between the original image and the blurred version of the image. If the datatype of the image is 8bits, then the difference was confined to the 8 bits as well, which resulted in very noisy mask of the image.
   To avoid this, datatype was changed to float32.
b. Since mask can have negative pixel values, due to which sharpened image also has some negative pixel values. Matplotlib imshow automatically rectify this by mapping the least pixel value to 0 and highest to 255, which resulted in grayish final image. This was corrected using the vmin and vmax attributes.