E9-241 DIGITAL IMAGE PROCESSING

Assignment #2

Sai Teja Kuchi (21317)
kuchisai@iisc.ac.in

Run the main.py file to generate the results file.

1 Image Display

By default, **plt.imshow** uses np.min(image) and np.max(image) of the image data for vmin and vmax parameters, which enhances the low-contrast images before displaying it. In this case the vmin and vmax values are 79 and 166 respectively.

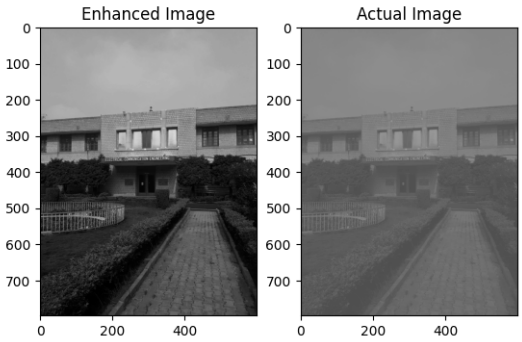
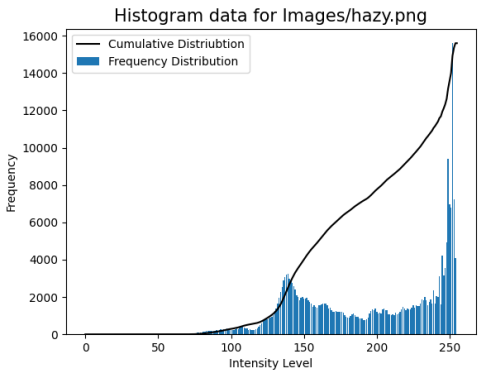


Figure 1

To fix the issue we need to manually override the vmin and vmax with 0 and 255 respectively which will show the actual image instead of enhancing it.

2 Contrast Stretching

(a) Image used for histogram computation

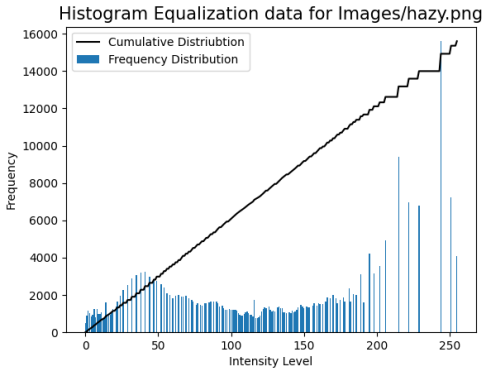


(b) Histogram Data

Figure 2: Image Data of hazy.png

From the histogram data (2b) of the image (2a), we can observe that most of the intensities of the image is above 100.

(a) Histogram Equalized Image



(b) Histogram Data of Equalized Image

Figure 3: Histogram Equalized Image Data of hazy.png

Applying histogram equalization on the image (2a), we get the image (3a) and it's equivalent histogram data (3b). We can see that the frequency distribution is now spread across different intensities of the image compared to (2b) where most frequency is above 100 and the cumulative distribution is linear but not quite towards the end.
On trying to find the optimal $\gamma \in (0, 5]$ value for the image (2a), by finding the minimal MSE value between the histogram equalized image (3a) and the gamma transformed image at that γ value, we obtain the below distribution.

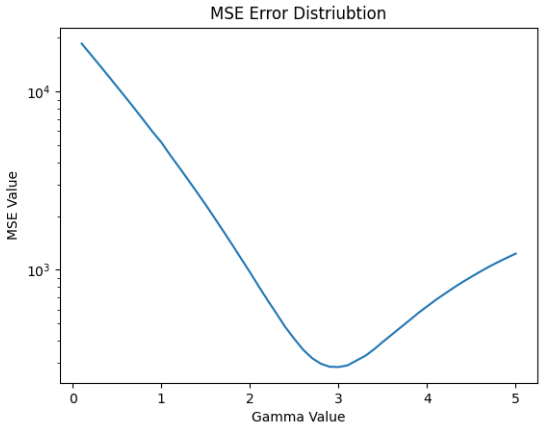
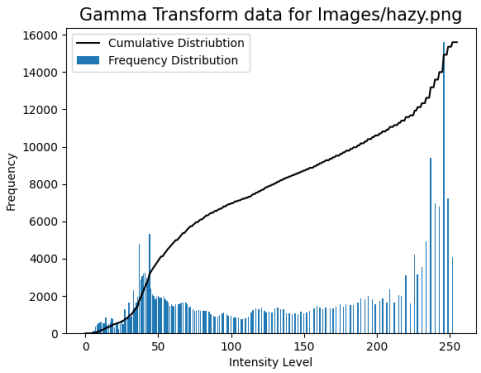


Figure 4: Mean Square Error Distriubtion

The Optimal γ value in this case is 3. Using this γ value (=3) to get the gamma transformed image, we will have the below results.

(a) Gamma Transformed Image



(b) Histogram Data of Gamma Transformed Image

Figure 5: Gamma Transformed Image Data of hazy.png

Comparing (3b) and (5b), we see that the cumulative distribution is more evenly distributed in (3b) obtained using histogram equalized compared to the (5b) obtained using gamma-transformation and we are able to see more intensities in both (3a) and (5a) compared to the actual image (2a). Also in the gamma-transformed image (5a) we can see more trees at the bottom of the image, which are due to the lower intensities frequency contribution. It can be seen through the histogram data, where there is more frequency towards the lower intensity side of gamma-transformed image (5b) when compared to that of the histogram equalized image (3b).

3 Image rotation

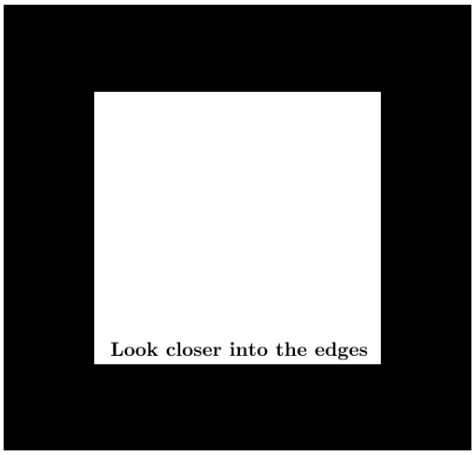
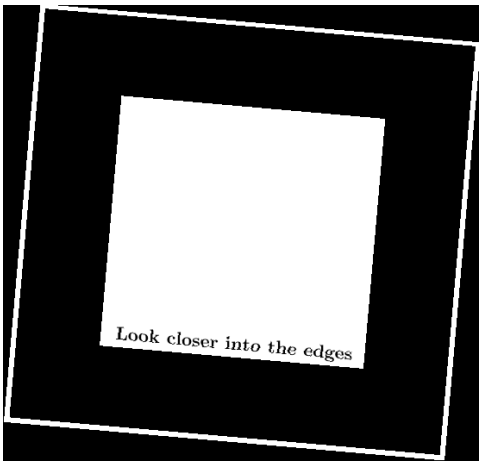


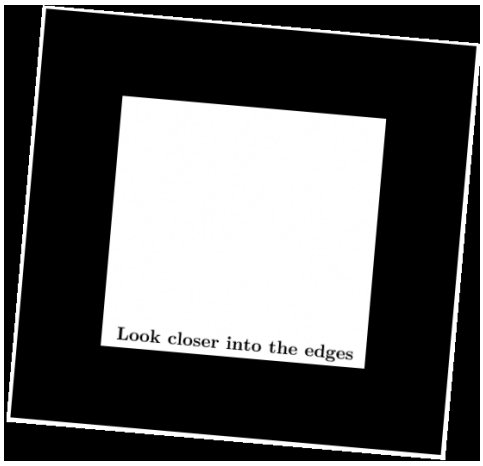
Figure 6: Image 'box.png'

Rotating the image (6), 5° clockwise using both nearest and bi-linear interpolation methods we have the below results

If we looks closely at the edges of (7a) and (7b), we can observe that the bi-linear interpo-



(a) Rotated Image with Nearest Neighbour Interpolation

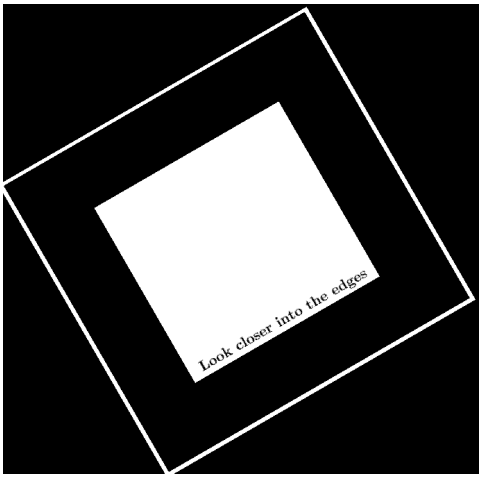


(b) Rotated Image with Bi-Linear Interpolation

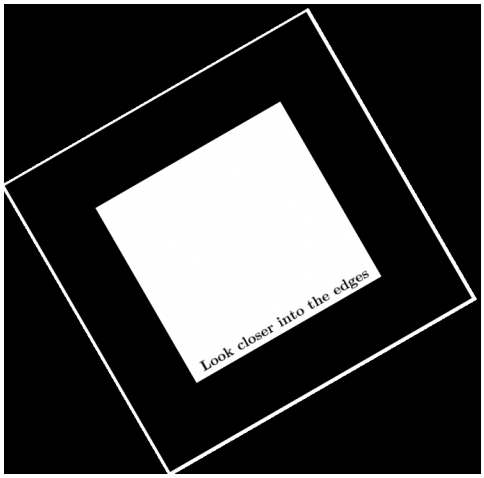
Figure 7: 5° clockwise rotated Images of 'box.png'

lation has smooth edge except for the outer one, while the nearest neighbour interpolation has zig-zag thing going for the edges which is not present in the original image (6). Also the text in the (7b) obtained using bi-linear interpolation is a bit clear than that of the rotated image (7a) obtained using nearest neighbour.

Rotating the image (6), 30° counter-clockwise using both nearest and bi-linear interpolation methods we have the below results



(a) Rotated Image with Nearest Neighbour Interpolation

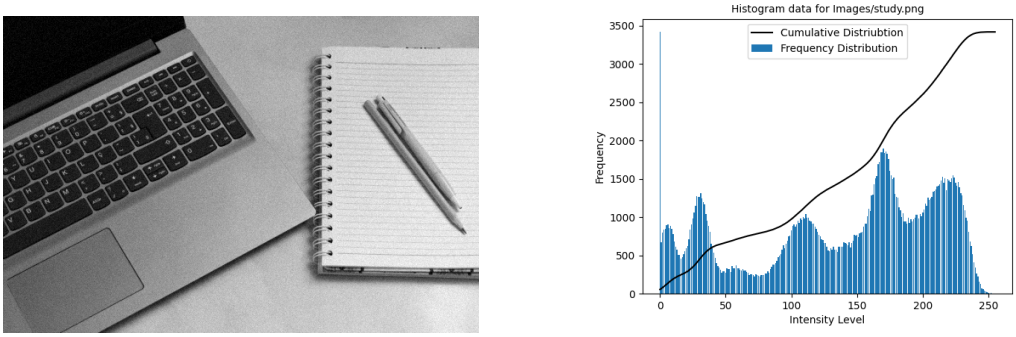


(b) Rotated Image with Bi-Linear Interpolation

Figure 8: 30° counter-clockwise rotated images of 'box.png'

The above observations of smooth text and edges of (7a) and (7b) can be observed in (8a) and (9b)

4 Spatial Filtering



(a) Image of 'study.png' (b) Histogram Data of 'study.png'

Figure 9: Image Data

On applying average filter with size 3 x 3 on the image (9a), we get the below blur image.

Figure 10: Blur Image after applying average 3 x 3 filter

Now using this blur image (10), we calculate the mask and use $k = 2.5$, to apply the high boost filtering on the original image (9a) and obtain the below result.

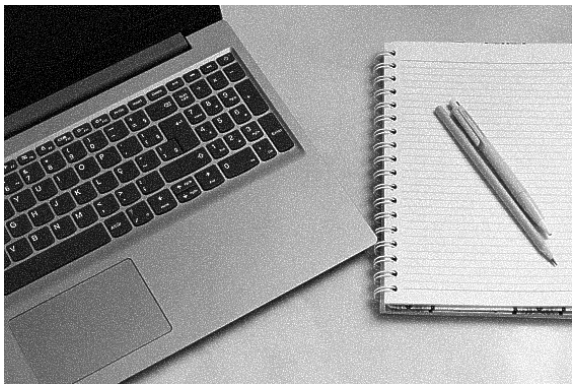


Figure 11: High Boost Filtered Image using average filter of size 3 x 3.

We observe that salt-pepper noise has been introduced in the image (11) when trying to sharpen the image using high-boost filtering.
Similarly applying average filter with size 5 x 5 on the image (9a), we get the below blur image.

Figure 12: Blur Image after applying average 5 x 5 filter

Now using this blur image (12), we calculate the mask and use $k = 2.5$, to apply the high boost filtering on the original image (9a) and obtain the below result.

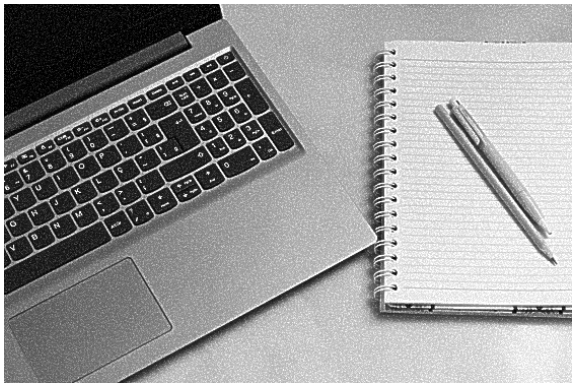


Figure 13: High Boost Filtered Image using average filter of size 5 x 5.

Similar to (11), we observe that the salt-pepper noise has been introduced in the image (13) and is even more because of using a bigger size square-filter when trying to sharpen the image using high-boost filtering,