

E9 241 Digital Image Processing

Assignment 4

November 9, 2025

Dwaipayan Haldar

1. Image Downsampling

Ans:

- (a) When we are downsampling the image by selecting the N^{th} pixel in the image without a low pass filter, then aliasing would occur, so there are dots when $N = 2,4$. But when $N = 5$, the whole image changes because there is some other image hidden in every 5th pixel of the `city.png` image. This is the typical case of aliasing. Fig.1 shows the results for downsampling with the custom function. Fig.2 gives the same interpolation results for library functions. Here the answer is different due to the implementation of the library function. It uses interpolation instead of selecting every N^{th} pixel so the answer is different.

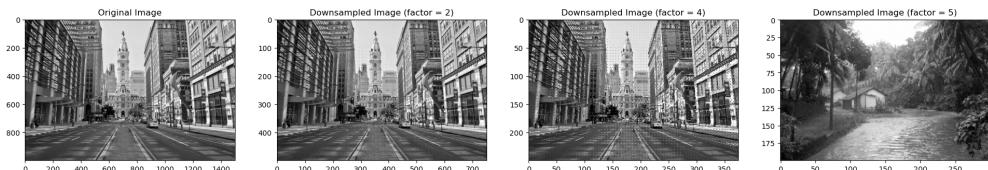


Figure 1: Original Image, Downsampled Image for $N = 2,4,5$ by custom function

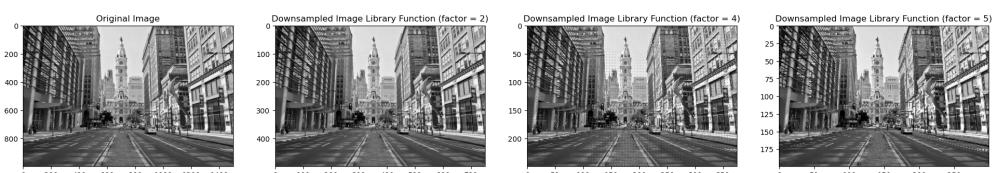


Figure 2: Original Image, Downsampled Image for $N = 2,4,5$ by library function

(b) There is a significant improvement in the result when we have applied a low pass filter, the aliasing effects are not there anymore. When we select the $N=5$, those severe aliasing effect is not there. Visually, the results are comparable to the output of the library functions. The MSE of the image for $N=2,4,5$ are 0.00051, 0.00092, 0.0013 respectively. The MSE is higher for $N=5$ as opposed to $N = 2$ and 4. Fig.3 and Fig.4 gives the results for the downsampling of the smoothed image via custom function and library function respectively.

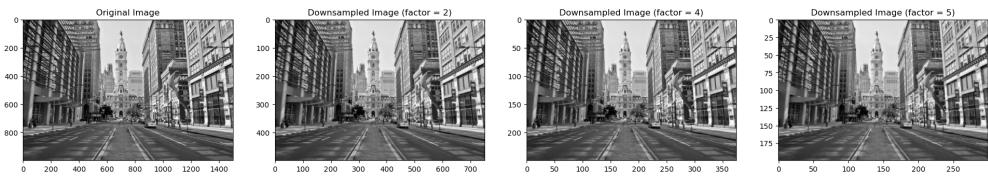


Figure 3: Original Image, Downsampled Smoothed Image for $N = 2,4,5$ by custom function

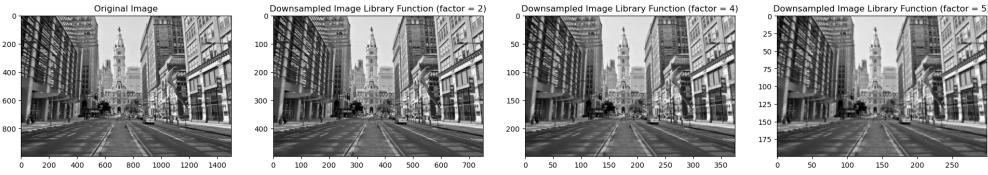


Figure 4: Original Image, Downsampled Smoothed Image for $N = 2,4,5$ by library function

(c) For $N = 5$, the optimal value of window size and sigma value is found out. The loop is ran over all window size in the list $[3,5,7,9,11,13,15,17]$ and for all sigma in the range of $[0.05, 4]$ with resolution of 0.05. The optimal parameters comes out to be:

$$\text{Window Size}_{\text{optimal}} = 13; \sigma_{\text{optimal}} = 3.05$$

For this parameters the optimal MSE comes out to be 0.000885.

2. Edge Detection

Ans:

- (a) Here, I have used Prewitt Operator. In case of the prewitt, thresholding is applied to make the edge map. So, the output is binary. The threshold is chosen to be equal to Otsu's threshold. For the Laplacian of Gaussian case, the zero crossings are detected. One observation in all the case is that LoG is more precise than the Prewitt, which is more widespread. The results are shown below.

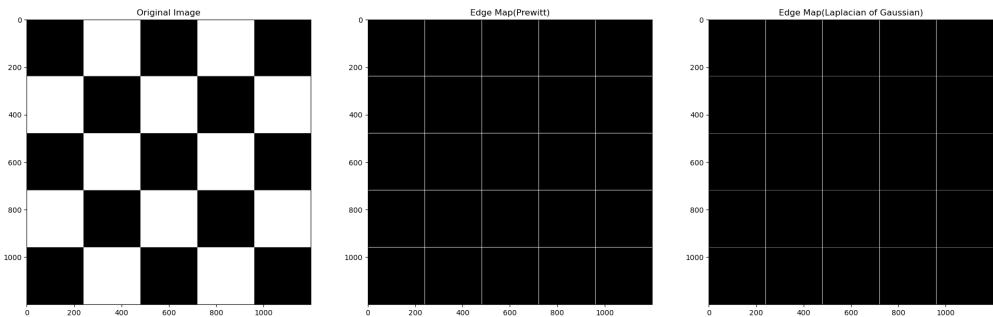


Figure 5: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for Checkerboard

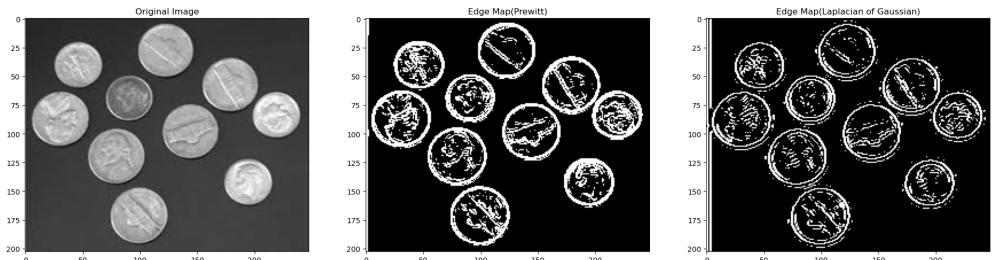


Figure 6: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for Coins

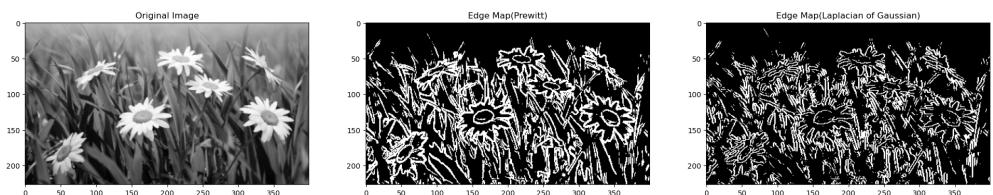


Figure 7: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for Flowers

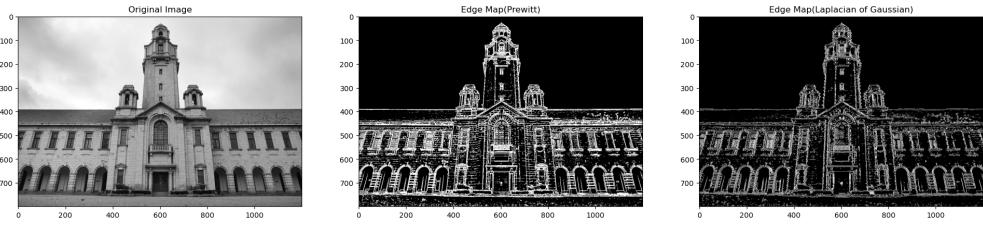


Figure 8: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for Main Building

- (b) Gaussian Noise is added. Gaussian smoothing is applied with window size 7×7 and $\sigma = 3$ and Prewitt and Laplacian of Gaussian is applied. The results are shown below.

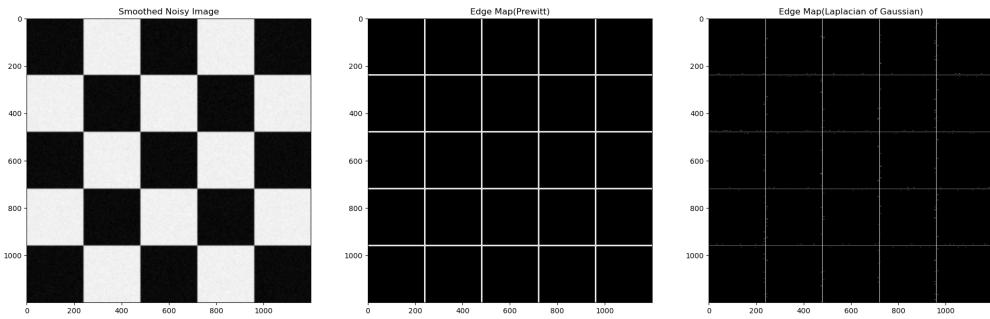


Figure 9: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for smoothed Checkerboard

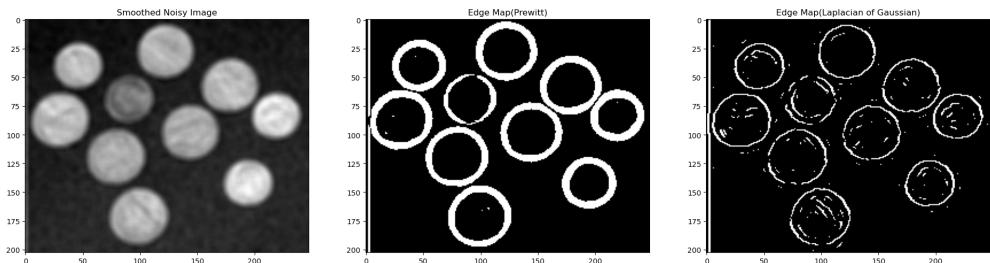


Figure 10: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for smoothed Coins

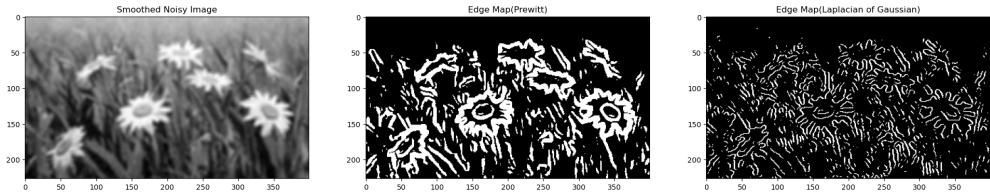


Figure 11: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for smoothed Flowers

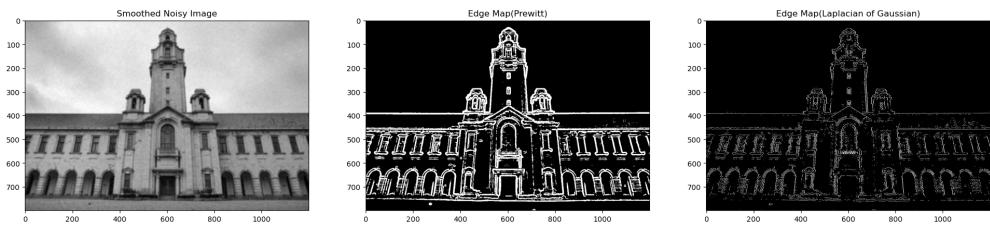


Figure 12: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for smoothed Main Building

The results with noise but without the gaussian smoothing is also calculated and shown below.

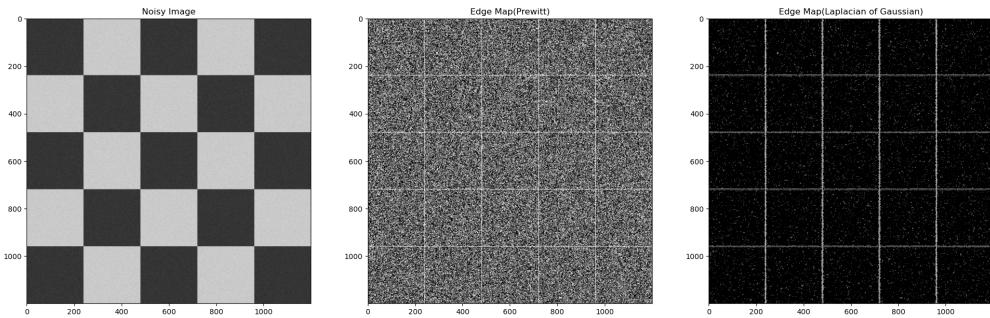


Figure 13: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for noisy Checkerboard

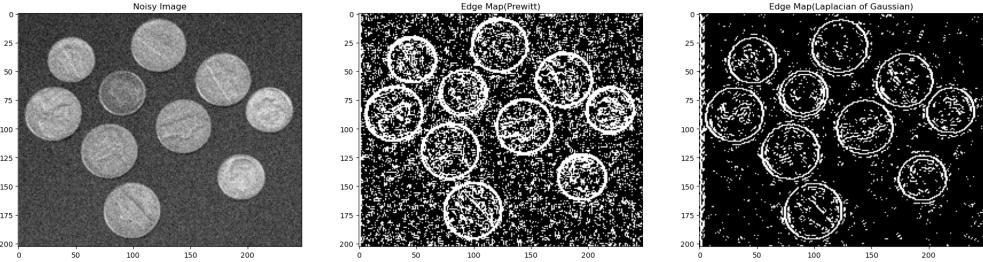


Figure 14: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for noisy Coins

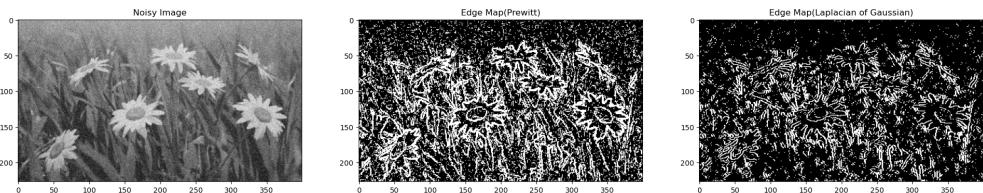


Figure 15: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for noisy Flowers

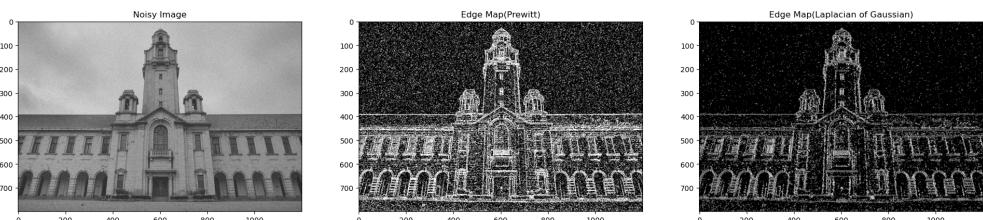


Figure 16: Original Image, Edge Map by Prewitt, Edge Map by Laplacian of Gaussian for noisy Main Building

(c)

- **Clean vs. Noisy Images**

Fig.5 - Fig.8 shows the results for clean images and Fig.13 - Fig.16 shows the results for the noisy image. With the clean image, both the algorithms give fine results, LoG gives a better precision producing finer edges than the Prewitt because it detects the zero crossings, while the Prewitt detects the local maxima and minima. Zero crossings is more precise. But for noisy images, the results of both the algorithms falters. Relatively, the LoG performs better than the first order filter. Such results are expected. The 2nd order filter is

less sensitive to noises. Since a Gaussian smoothing is done internally before the Laplacian, so that smooths out the noises and so the effect of noise is a bit less than the first order filter.

- **Gradient vs LoG Edge Detection**

The difference between the gradient and LoG based detector is pretty clear from the results. LoG based method has better edge localization than the Gradient based method. LoG based method are less sensitive to noises than the gradient based method. After smoothing of the noise, performance of both the algorithm on noisy image improves, while the prewitt has wider edges and LoG has finer edges.

- **With and without Gaussian smoothing**

Fig.9 - Fig.12 shows the results for noisy images with Gaussian smoothing and Fig.13 - Fig.16 shows the results for the noisy image without Gaussian smoothing. Overall the results for both of the algorithms with per gaussian smoothing. That is expected, since both the filters are sensitive to noise. Noise reduction using Gaussian smoothing helps to improve the edge map. In this case also, LoG gives finer precision than the Prewitt detector.