CSC3831 Image Informatics:

Introduction:

The task of denoising images is seen as quite a common challenge and is usually done to pre-process images before analysis. I had to develop an algorithm capable of denoising images with varying levels of noise. Using the theory and practical material I implemented the Fourier transforms and low-pass filters and used them as tools to separate signal from noise. Through experimentation I found that low-pass filter worked the best for my algorithm.

Algorithm Description and Flowchart:

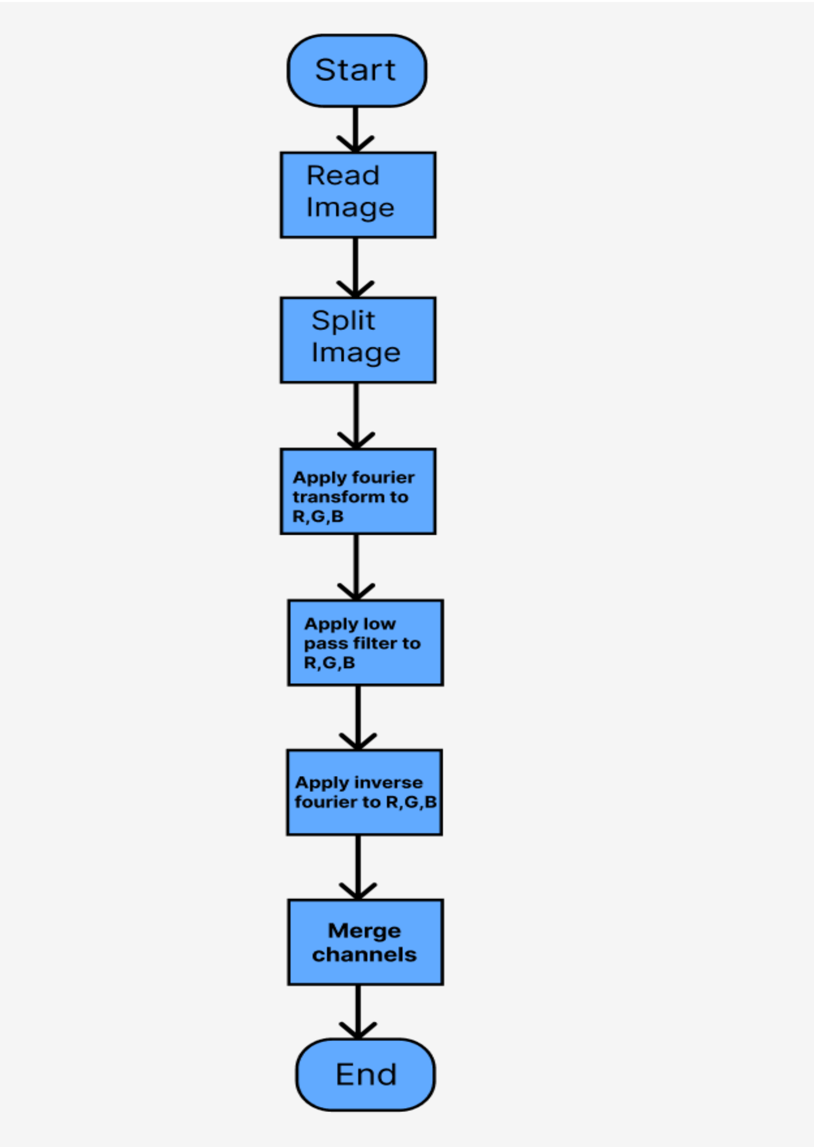

Figure 1 - Flowchart for the denoising algorithm

Initially, for the denoising algorithm I split the images into red, green, blue channels and then for each channel I move it from image space to the frequency space using Fourier transform. I use the low pass filter to filter out the higher frequencies, which is usually the noise, while still preserving the lower frequencies that have the actual image details. Then when I transform back using inverse Fourier transform, the image will have less noise. These Fourier techniques I used in the algorithm helped to simplify the separation and filtering of noise from signal.

Testing Out Filters and Results:

As stated in the introduction, I tested the algorithm across the various images with different noise levels i.e. noisy 10, noisy 25, and noisy 50. I used the mean squared error (MSE) and structural similarity index measure (SSIM) to analyse the performance. Results for bilateral, gaussian, and butterworth filtering are as follows:

Results:
Bilateral MSE (Noisy 10): 15139.147096364963
Bilateral SSIM (Noisy 10): 1.9881372202085116e-05
Bilateral MSE (Noisy 25): 15139.166574134071
Bilateral SSIM (Noisy 25): 3.0177963505870 53e-05
Bilateral MSE (Noisy 50): 15139.386707095808
Bilateral SSIM (Noisy 50): 4.671078091544182e-05

Gaussian MSE (Noisy 10): 15139.223353625075
Gaussian SSIM (Noisy 10): 3.2955519855355044e-05
Gaussian MSE (Noisy 25): 15139.244983489241
Gaussian SSIM (Noisy 25): 3.270218062881072e-05
Gaussian MSE (Noisy 50): 15139.186669924888
Gaussian SSIM (Noisy 50): 3.1833873049317246e-05

Butterworth MSE (Noisy 10): 15141.58235637807
Butterworth SSIM (Noisy 10): 8.02012828441427e-06
Butterworth MSE (Noisy 25): 15141.547667983947
Butterworth SSIM (Noisy 25): 8.110812505974001e-06
Butterworth MSE (Noisy 50): 15141.319831332075
Butterworth SSIM (Noisy 50): 7.98924944118342e-06

Figure 2 - Bilateral, Gaussian, Butterworth filters results

As you can see the SSIM score for all filtering techniques were not good, with Gaussian being the best out of the bunch but not by a lot.

<u>Key Findings and Discussion:</u>

I thought that the gaussian would perform a lot better since I identified the noise was Gaussian using histograms. The histograms (figure 3, 4 and 5) showed me the pixel intensities of the noisy images and for each noisy image it had a bell-shaped curve which is indicative of gaussian noise. Additionally, I used the bilateral filter as I read an article [1] given to us which said bilateral filtering is a common technique. But after trying it out bilateral filtering had poor SSIM scores, showing large structural differences from the original images. This was the same for the butterworth filter.

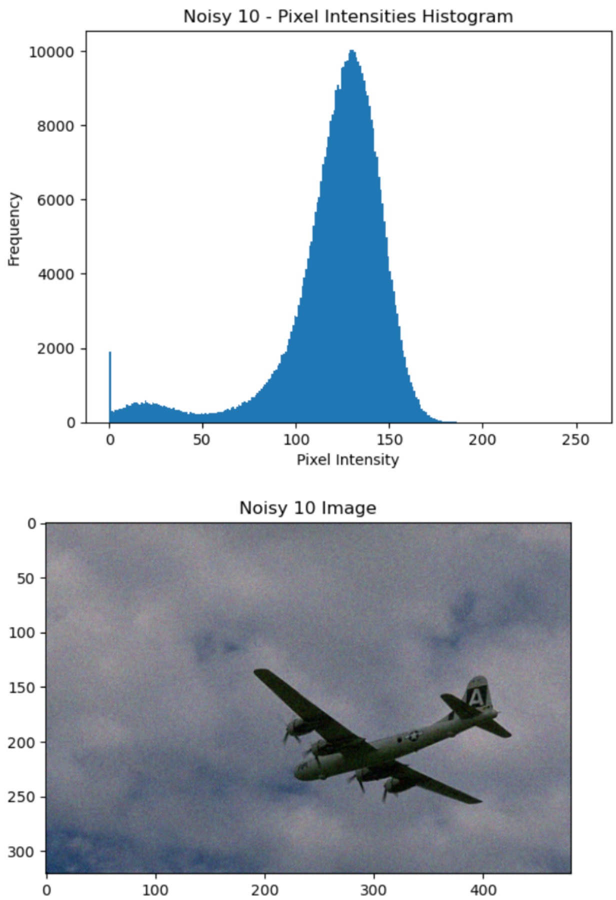

Figure 3 - Noisy 10 Histogram

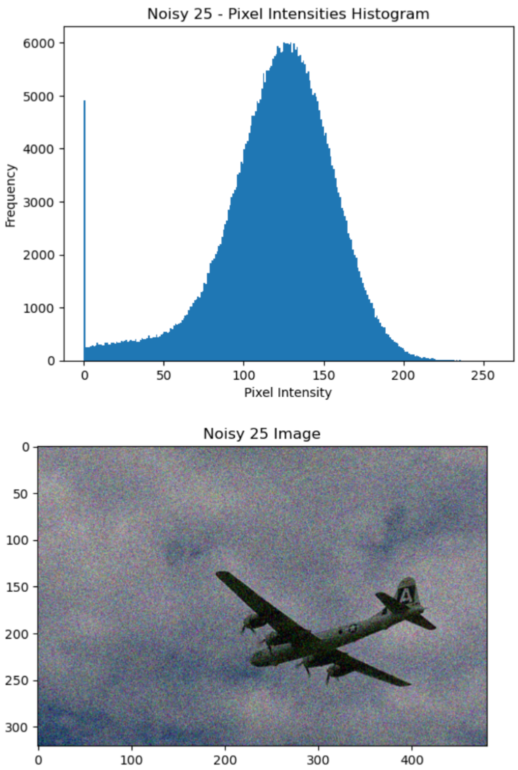

Figure 4 - Noisy 25 histogram

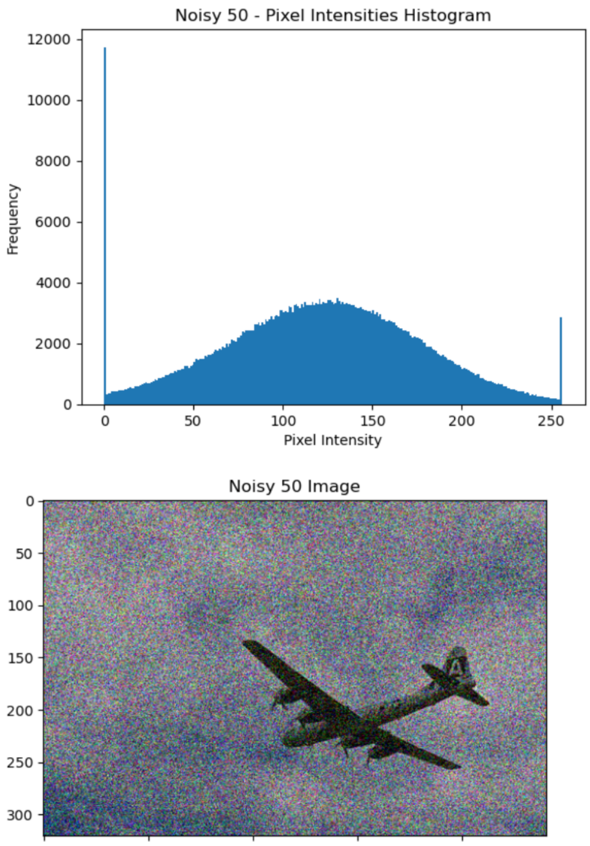

Figure 5 - Noisy 50 histogram

After testing out these filters I tried the low-pass filter for my actual denoising algorithm and there was some success but only for certain noise levels. In particular, the images with higher levels of noise, i.e. noisy 50, had significant improvement in the MSE and SSIM scores when the algorithm was applied. For noisy 10 it had the opposite impact and for some of the noisy 25 images it improved the MSE and SSIM scores for the denoised image.

Conclusion:

Overall, I realised the practical application of these filters may not align with the literature and the actual level of noise in the image needs to be accounted for when picking filters and methods for denoising. For images with higher noise, I found the low-pass filter worked the best. If I was to improve the algorithm, I would have implemented multiple different techniques that specifically denoise the image depending on its noise level.

References:

[1] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1, Jul. 2019, doi: https://vciba.springeropen.com/articles/10.1186/s42492-019-0016-7#Abs1