BM3D Tutorial Results

Dhruv Meena

October 2024

Introduction

In this document I summarize the results found in the tutorial by comparing multiple images and drawing conclusions. Please refer to bm3d_tut.ipynb for the code.

BM3D: Increasing Standard Deviation

All Images below employ hard thresholding after the collaborative filtering is applied.

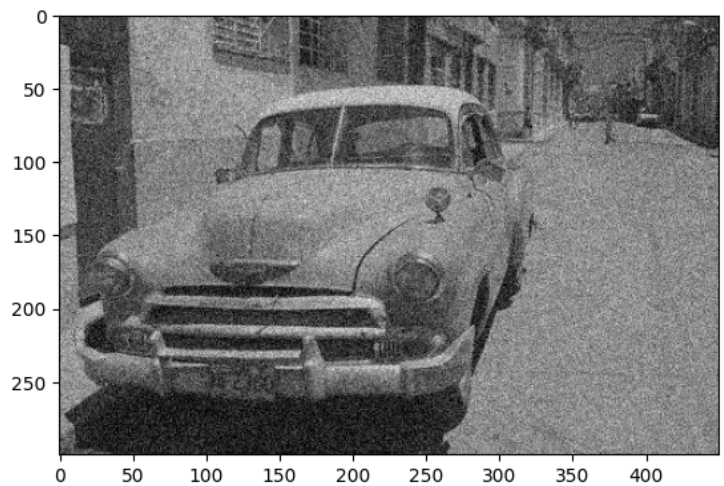

Figure 1: Noisy Image

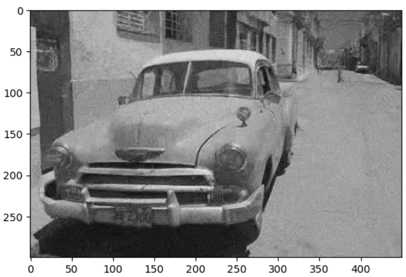

Figure 2: PSD = 0.05

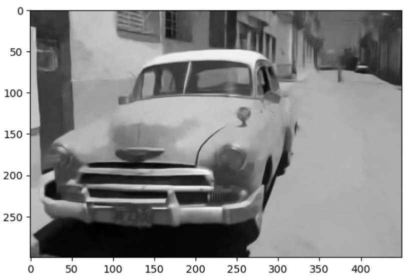

Figure 3: PSD = 0.1

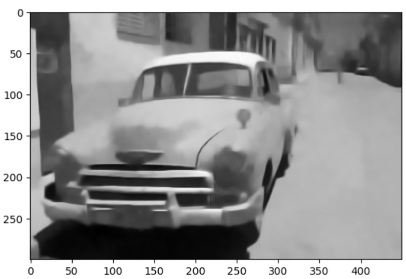

Figure 4: PSD = 0.2

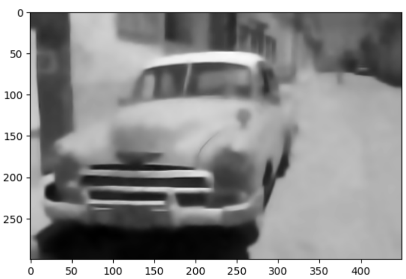

Figure 5: PSD = 0.4

Increasing the standard deviation to 0.4 has led to a lot of **smoothening** and **loss** of almost all the **structure of the image** including edges, shape of the car, windows, etc., A very low standard deviation of 0.05 still has a lot of noise although it looks considerably better than the noisy image. A PSD of 0.1 is perfect as all the noise is gone and we also have retained image structure.

Keeping the PSD at 0.1 we will alter the **stage_arg** and note the results.

Figure 6: Hard Thresholding

Figure 7: All Stages

All Stages is a more powerful method and also takes longer to run (AS took 10s while HT took 5s) but we can see that the results are much better when applying All Stages as the stage_arg. The most notable outcome is that the number plate is more clearly visible in Figure 7. With **All Stages** we have both **Hard Thresholding** and **Wiener Filtering**.

Other Filters

Gaussian Filter

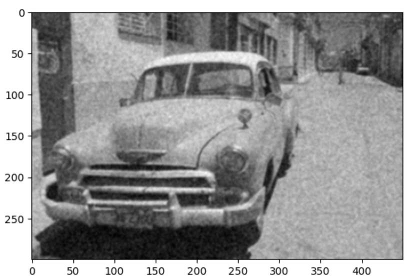

Figure 8: Gaussian of size (5,5) and SD = 0

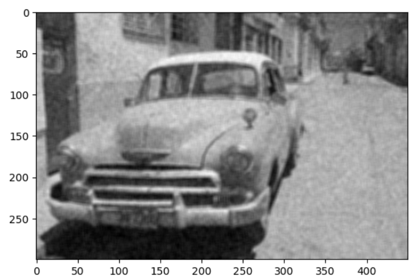

Figure 9: Gaussian of size (5,5) and SD = 100

Gaussian filter is used for the 'gaussian blur' and is not the best at denoising, definitely not when structural integrity is of utmost importance. But we can see that at a higher standard deviation value, some of the noise does go away.

Mean Filter

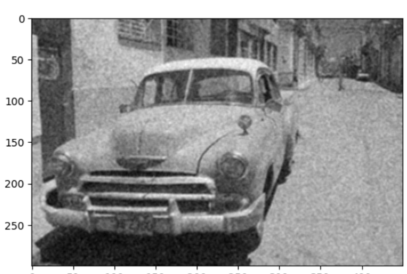

Figure 10: Mean Filter of size 3

Figure 11: Mean Filter of size 5

Mean filters have a similar effect as that of the gaussian filter but the blurring is more pronounced at higher sizes for the mean filter.

Median Filter

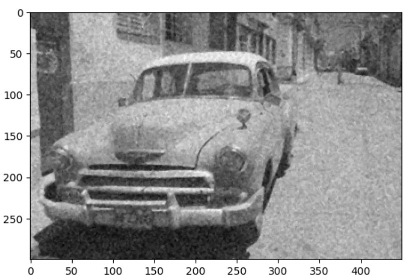

Figure 12: Median Filter of size 3

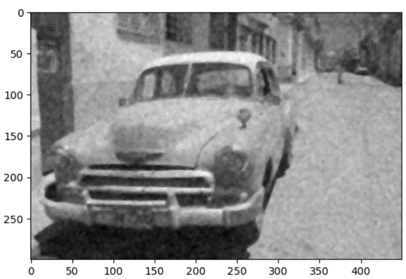

Figure 13: Median Filter of size 5

Median filters are **not** used to denoise gaussian noise images. But from the output we can see that when compared to a mean filter of the same size, the median filter **loses more information** about the structure such as edges, corners etc.,

Bilateral Filter

Unlike the previous filters and just like the bm3d filter, bilateral filter also preserves edges while smoothening the image. It replaces each pixel with the **weighted average** of its neighbors while **penalizing** pixels **far away** based on a spatial parameter and also penalizing pixels with **different intensity** based on a intensity parameter.

Increasing Filter Size

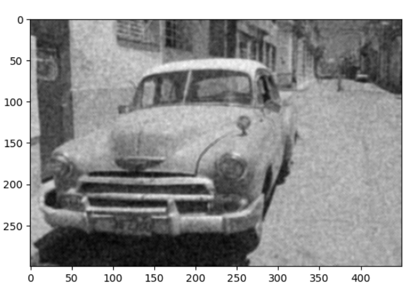

Figure 14: Size = 5, Sigma_S = 50, Sigma_R = 50

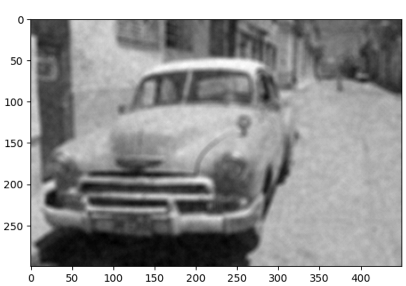

Figure 15: Size = 8, Sigma_S = 50, Sigma_R = 50

As is expected, increasing filter size causes a large amount of smoothening to happen. If we apply an edge detector on it, for example, a sobel edge detector using CV2, the outputs do not differ much. So we obtain a slightly denoised image with edges preserved.

Increasing Sigma_S

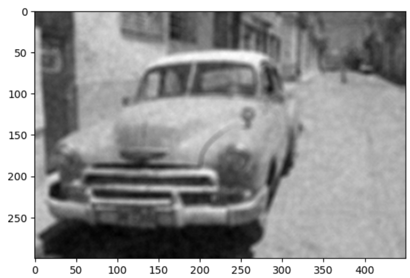

Figure 16: Size = 8, Sigma_S = 50, Sigma_R = 50

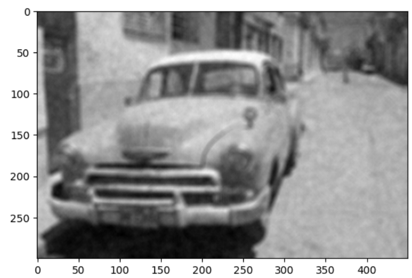

Figure 17: Size = 8, Sigma_S = 50, Sigma_R = 100

There is no visible change here. On applying sobel edge detectors we get the same image as output. The outcome is very different when we change the intensity component though.

Increasing Sigma_R

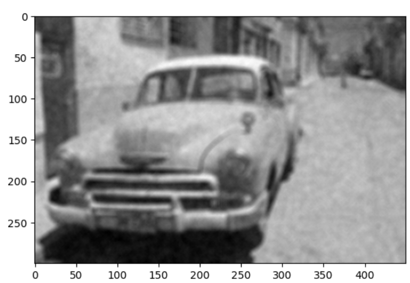

Figure 18: Size = 8, Sigma_S = 25, Sigma_R = 50

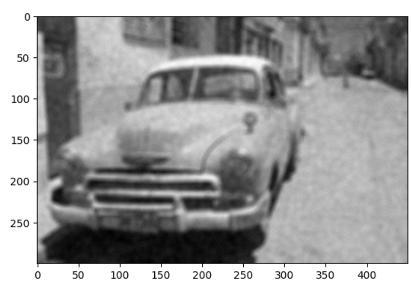

Figure 19: Size = 8, Sigma_S = 125, Sigma_R = 100

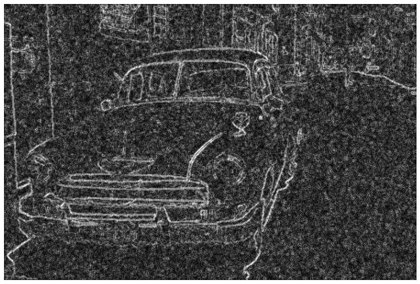

Figure 20: Size = 8, Sigma_S = 25, Sigma_R = 50

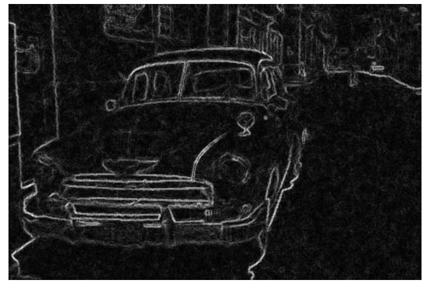

Figure 21: Size = 8, Sigma_S = 125, Sigma_R = 100

Increasing Sigma_R clearly helped in this case. We can see that the higher intensity parameter resulted in greater denoising while preserving structure. Figures 20 and 21 are the output of the sobel filters.

Mean Shift Filter

Mean Shift, although used to calculate cluster modes, can also be used for denoising. It is very similar to bilateral filtering in the sense that it is an iterated version of bilateral filtering where the number of iterations for smoothening varies from pixel to pixel and is decided by the algorithm itself.

Results of Mean Shift

Figure 22: Size = 8, Sigma_S = 50, Sigma_R = 50

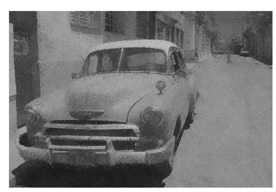

Figure 23: Size = 6, Sigma_S = 35, Sigma_R = 35

The level of denoising here is much higher compared to all other filters till now but we lose a lot of information in the process. Structure is not well preserved.

Conclusion

After thorough analysis and comparison, it is safe to say that BM3d filter is the best denoising candidate here as it preserves edges and removes all gaussian noise from the image. Mean Shift comes close to removing the noise but loses the structure unlike Bilateral Filtering, which preserved edges but did not sufficiently remove noise. Perhaps a version of iterated bilateral filtering which is not as severe as mean shift could be a good candidate but that is not explored further as we are satisfied with BM3d. BM4d, which is used in the paper works on the same principles.