MAIS 202 Deliverable 1

1. Dataset

https://www.eecs.yorku.ca/~kamel/sidd/dataset.php. This is a good dataset that I found. It includes the noisy images, as well as the "ground truth" images (i.e. denoisified images).

To create a dataset, I was also thinking of taking any set of images, and adding noise to them, this would be an easy way to create a dataset containing both the noisy and denoisified images.

While looking online, I found that there exist many types of noise (gaussian noise, salt and pepper,…). I think that I'll mainly focus on gaussian noise (noisy pixel will be related to their neighbors pixels) since it appears to be the easiest. More research will need to be done for this though.

2. Methodology

For preprocessing, we will need to split the training images into smaller $n \times n$ images (n is around 20 to 40 pixel) and work on the split-up images. Then we will need to merge all the images back. This will help since we're breaking our problem into smaller sub problems which make solving the general problem easier.

Also, we might need to split an image into the 3 colors channels sub-images. A channel is either red, green, or blue since all images are RGB. Hence, we can now represent each image as a 3-d matrix, where each component of this 3d matrix will be a color channel for the image. Each entry of each component of the 3d matrix would be the red, green, or blue number (ranging from 0-255 since we have 8bit colors). In other words, each picture consists of 3 matrices, where the dimension of the 3 matrices or the dimension of the images, and the entries in those 3 matrices are the red, green, or blue values of each pixel.

I'm not entirely sure how this might help. I saw that denoising is usually easier on single channel images (gray images for example). Hence splitting the images might help in optimizing the algorithm. This will need further investigation (especially since my knowledge of image processing is limited). Also, if working on RGB images appears to be too hard, I might have to work on only gray images.

Now, we can think of noise as a corrupted pixel. Our goal is to correct those pixels. We will also need to quantify the level of noise. I found the PSNR (peak signal-to-noise ratio) metric to be the most used metric to measure the quality of an image. We will need to maximize the PSNR. We will also use the mean squared error (comparing our image with its ground truth counterpart), and try minimizing it.

To correct the corrupted pixels, we can look at the majority of the pixels in a subimage ($n \times n$), and then replace the odd pixels in the subimage by what the majority is, or by taking the average of the pixel values (the average of the RGB numbers of each pixels).

I found many deep learning algorithms about denoising, but my knowledge of deep learning is limited. I don't know how my approach will work, but I'm willing on getting to know more about deep learning algorithms involving image processing to make my project better.