<u>Question 3 – Bonus</u>

In this question we are required to upload a picture and noise to it. Our goal is to clean the image from the noise and reconstruct the original image with the highest possible quality.

*First, we dealt with the following RGB image because it is very "rich":

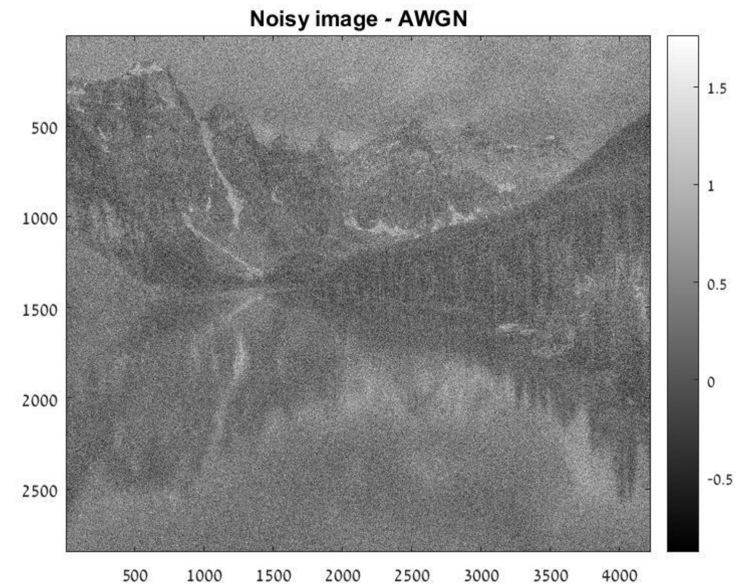
*After transforming the image to grayscale by rgb2gray and normalizing we got:

*We added a white gaussian noise to the image:

$$x_n(i,j) = x(i,j) + v(i,j)$$

Where $v(i,j) \sim \mathcal{N}(0, \sigma_v^2)$ and $v(i_0, j_0)$, $v(i_1, j_1)$ are i.i.d for $(i_0, j_0) \neq (i_1, j_1)$.

The fact that the noise samples are i.i.d makes it very difficult since we can't obtain any information about the noise added to a pixel from other pixels.
*After adding the noise, the image looks as follows:



It can bee seen very clearly that the quality of the noisy image is very low. We will explain how we have cleaned the image from the additive noise.
<u>Mathematical explanation</u>:
We have the following statistical model:

$$x_n(i,j) = x(i,j) + v(i,j)$$

Where the noise $v(i,j) \sim \mathcal{N}(0, \sigma_v^2)$ is i.i.d and independent of the image itself.
We choose to estimate the original pixel value by estimating the best linear estimator under the MSE criteria:

$$\hat{x}(i,j) = \alpha * x_n(i,j) + \beta$$

Our goal is to find α, β which minimizes the mean squared error cost:

$$J(\alpha,\beta) = \mathbb{E}\left[\left(x(i,j) - \hat{x}(i,j)\right)^2\right] = \mathbb{E}\left[\left(\alpha * x_n(i,j) + \beta - x(i,j)\right)^2\right]$$
$$= \mathbb{E}\left[\left(\alpha * (x(i,j) + v(i,j)) + \beta - x(i,j)\right)^2\right]$$

After using the orthogonality principle, we obtain two equations:

1. $\mathbb{E}[x(i,j) - \hat{x}(i,j)] = 0$
2. $\mathbb{E}[(x(i,j) - \hat{x}(i,j))\hat{x}(i,j)] = 0$

After solving those two equations we obtained the final form for our estimator:

$$\hat{x}(i,j) = \mu_x + \left(1 - \frac{\sigma_v^2}{\sigma_x^2}\right) * (x_n(i,j) - \mu_x)$$

We have some parameters to estimate:

1. $\mu_x = \mathbb{E}[x(i,j)]$

2. $\sigma_v^2 = Var(v(i,j))$
3. $\sigma_x^2 = Var(x(i,j))$

In order to estimate those parameters, around each pixel we take a k x k box denoted by $B_{i,j}$:

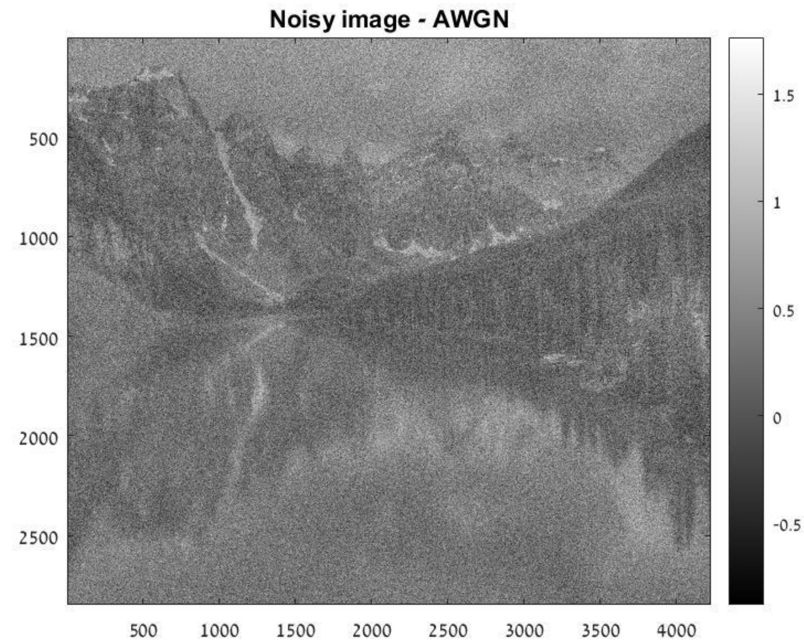
$$\mu_x \approx \frac{1}{k^2} \sum_{x \in B_{i,j}} x$$

$$\sigma_x^2 \approx \frac{1}{k^2} \sum_{x \in B_{i,j}} x^2 - \mu_x^2$$

To estimate the noise variance we subtracted the mean from each box and took the mean of squares:

$$\mathbb{E}[v(i,j)^2] = \sigma_v^2 \approx \frac{1}{k^2} \sum_{x \in B_{i,j}} (x - \mu_x)^2$$

Note that $x - \mu_x$ is a good estimation of the noise since we can regard the original image pixel around the box to be stationary. So, the pixels of the original image do not varies much so the average is close to the original pixel value itself. Note that the noise has 0 mean.



As can be seen the algorithm works very well. By comparing the noisy image to the filtered image there is s a huge difference and the quality is remarkably improved. We can still see some leftovers from the noise, but most of it have been cleaned well.
The best of it is that we have no assumptions about the noise. The algorithm calculates everything by itself so it can deal with any random image with an additive i.i.d noise. If the algorithm gets the noise variance the results will be even better.

We hope it worth 1 extra bonus point (: