CS 663, Fall 2023

Assignment 5

Jahanvi Rajput 23D0378
Badri Vishal Kasuba 22M2119
Abhishek Kumar Singh 22M2104

Question 1:

1. In this part, we will apply the PCA technique for the task of image denoising. Consider the images `barbara256.png` and `stream.png` present in the corresponding data/ subfolder - this image has gray-levels in the range from 0 to 255. For the latter image, you should extract the top-left portion of size 256 by 256. Add zero mean Gaussian noise of $\sigma = 20$ to one of these images using the MATLAB code `im1 = im + randn(size(im))*20`. Note that this noise is image-independent. (If during the course of your implementation, your program takes too long, you can instead work with the file `barbara256-part.png` which has size 128 by 128 instead of 256 by 256. You can likewise extract the top-left 128 by 128 part of the `stream.png` image. You will not be penalized for working on these image parts.)

   (a) In the first part, you will divide the entire noisy image 'im1' into overlapping patches of size 7 by 7, and create a matrix $\mathbf{P}$ of size $49 \times N$ where N is the total number of image patches. Each column of $\mathbf{P}$ is a single patch reshaped to form a vector. Compute eigenvectors of the matrix $\mathbf{P}\mathbf{P}^T$, and the eigen-coefficients of each noisy patch. Let us denote the j^{th} eigen-coefficient of the i^{th} (noisy) patch (i.e. $\mathbf{P}_i$) by α_{ij}. Define $\bar{\alpha}_j^2 = \max(0, \frac{1}{N}[\sum_{i=1}^{N} \alpha_{ij}^2] - \sigma^2)$, which is basically an estimate of the average squared eigen-coefficients of the 'original (clean) patches'. Now, your task is to manipulate the noisy coefficients $\{\alpha_{ij}\}$ using the following rule, which is along the lines of the Wiener filter update that we studied in class: $\alpha_{ij}^{\text{denoised}} = \dfrac{\alpha_{ij}}{1 + \frac{\sigma^2}{\bar{\alpha}_j^2}}$. Here, $\alpha_{ij}^{\text{denoised}}$ stands for the j^{th} eigencoefficient of the i^{th} denoised patch. Note that $\frac{\sigma^2}{\bar{\alpha}_j^2}$ is an estimate of the ISNR, which we absolutely need for any practical implementation of a Wiener filter update. After updating the coefficients by the Wiener filter rule, you should reconstruct the denoised patches and re-assemble them to produce the final denoised image which we will call 'im2'. Since you chose overlapping patches, there will be multiple values that appear at any pixel. You take care of this situation using simple averaging. Write a function `myPCADenoising1.m` to implement this. Display the final image 'im2' in your report and state its RMSE computed as $\dfrac{\|\text{im2}_{\text{denoised}} - \text{im2}_{\text{orig}}\|_2}{\|\text{im2}_{\text{orig}}\|_2}$.

Solution:
The **RMSE** value calculated using above formula and the value obtained for Barbara image is **0.137189** and for Stream image is **0.126986**.

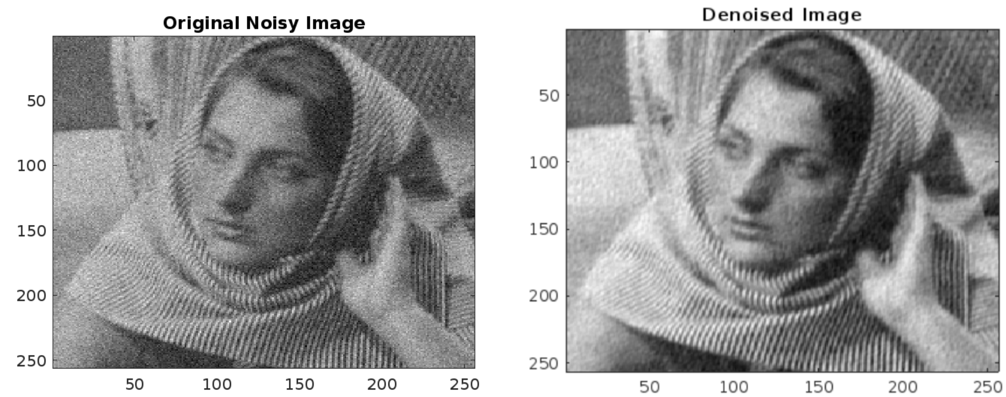

Figure 1: Barbara Image

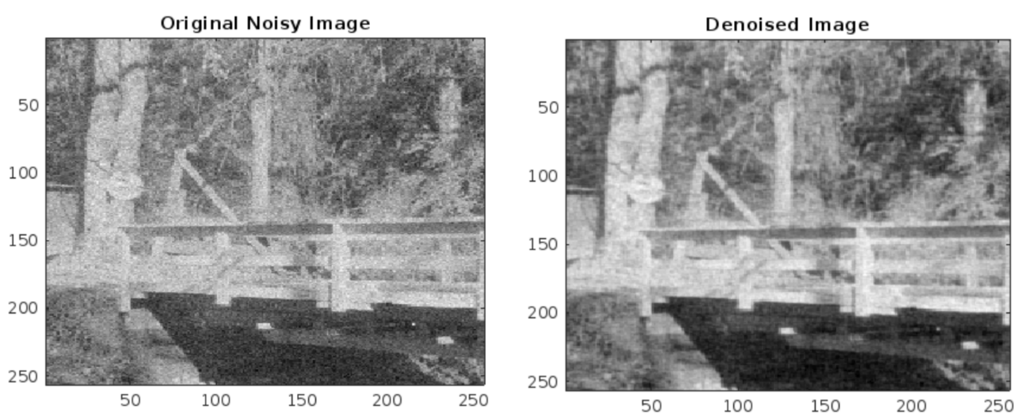

Figure 2: Stream Image

(b) In the second part, you will modify this technique. Given any patch $\mathbf{P}_i$ in the noisy image, you should collect $K = 200$ most similar patches (in a mean-squared error sense) from within a 31×31 neighborhood centered at the top left corner of $\mathbf{P}_i$. We will call this set of similar patches as Q_i (this set will of course include $\mathbf{P}_i$). Build an eigen-space given Q_i and denoise the eigen-coefficients corresponding to **only** P_i using the Wiener update mentioned earlier. The only change will be that $\bar{\alpha}_j^2$ will now be defined using only the patches from Q_i (as opposed to patches from all over the image). Reconstruct the denoised version of P_i. Repeat this for every patch from the noisy image (i.e. create a fresh eigen-space each time). At any pixel, there will be multiple values due to overlapping patches - simply average them. Write a function `myPCADenoising2.m` to implement this. Reconstruct the final denoised image, display it in your report and state the RMSE value. *Do so for both barbara as well as stream.*

Solution:

The **RMSE** value calculated using and the value obtained for Barbara image is **0.122856** and for Stream image is **0.150592**.

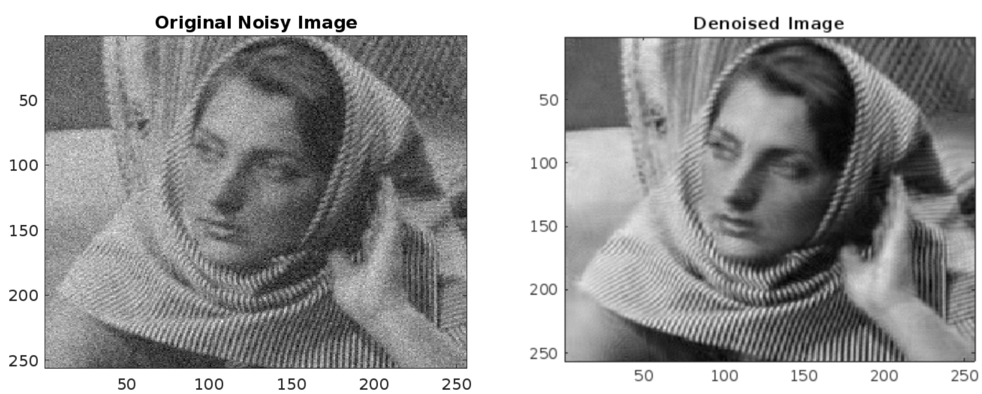

Figure 3: Barbara Image

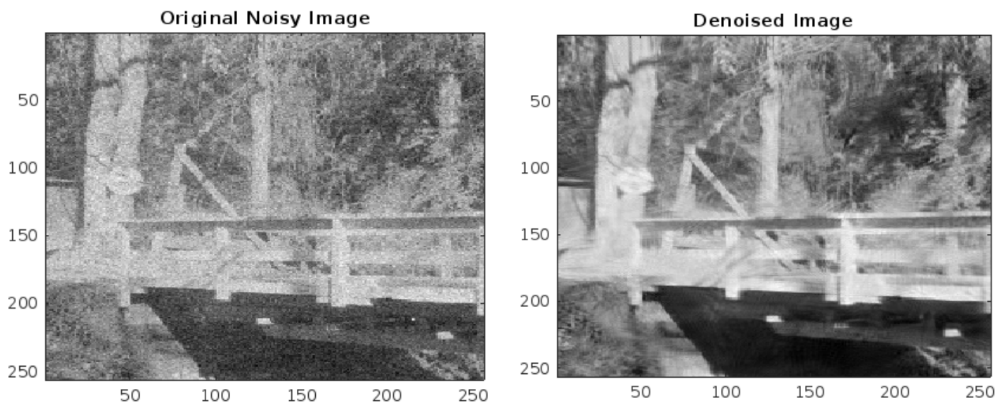

Figure 4: Stream Image

(c) Now run your bilateral filter code from Homework 2 on the noisy version of the barbara image. Compare the denoised result with the result of the previous two steps for both images. What differences do you observe? What are the differences between this PCA based approach and the bilateral filter?

Solution:

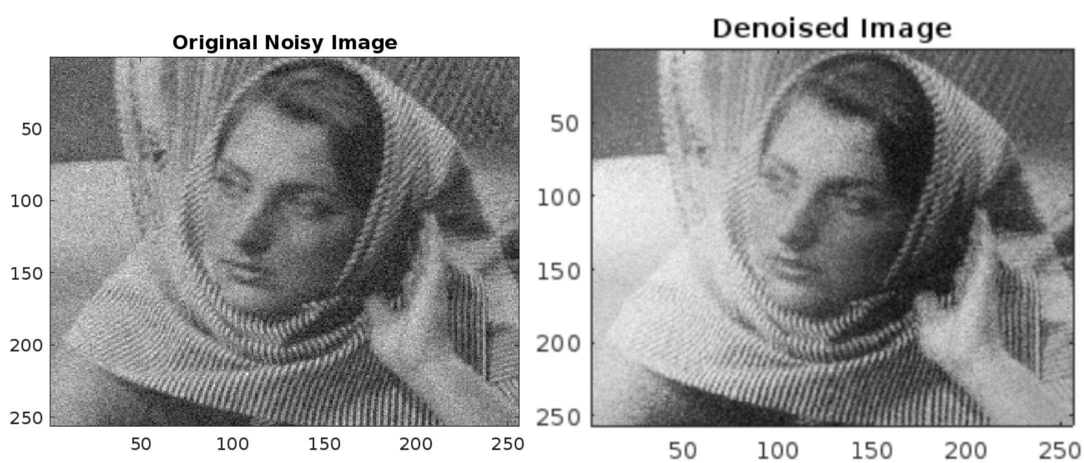

Figure 5: Barbara Image with $\sigma_s = 25$ and $\sigma_r = 50$

Here $\sigma_s = 25$ and $\sigma_r = 50$ are used. The **RMSE** value calculated

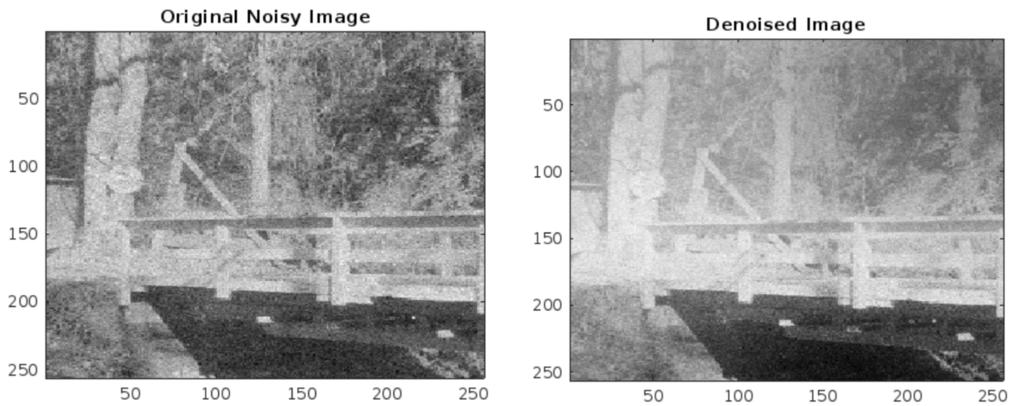

Figure 6: Stream Image with $\sigma_s = 25$ and $\sigma_r = 50$

and the value obtained for Barbara image is **0.147874** and for Stream image is **0.231485**.

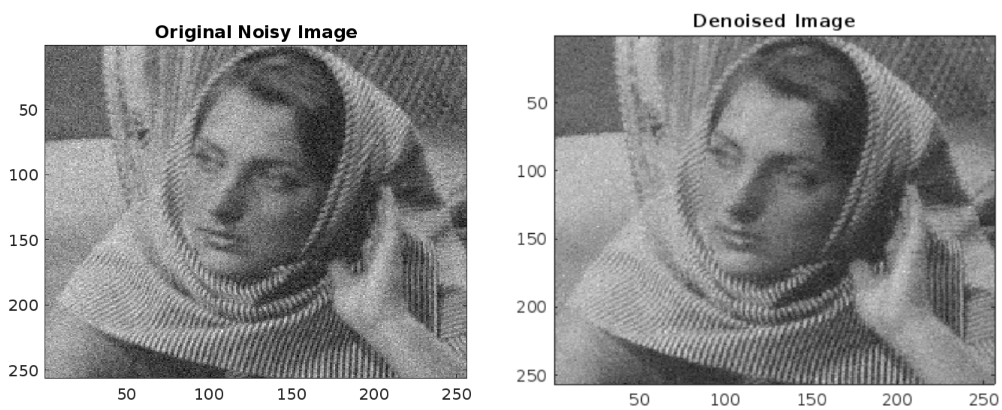

Figure 7: Barbara Image with $\sigma_s = 5$ and $\sigma_r = 15$

Here $\sigma_s = 5$ and $\sigma_r = 15$ are used. The **RMSE** value calculated and the value obtained for Barbara image is **0.038429** and for Stream image is **0.032105**.

Observation:

- Biletral filter is giving quite better results than when the value of σ_s and σ_r is high, i.e. 25 and 50 respectively. The results

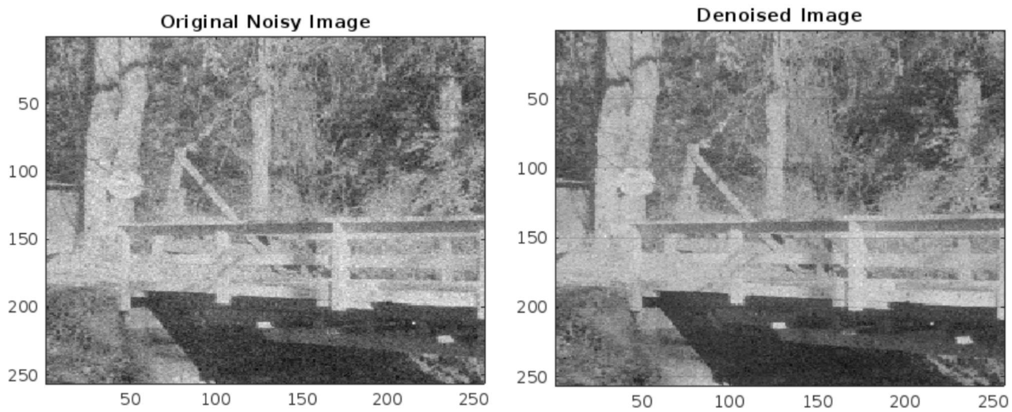

Figure 8: Stream Image with $\sigma_s = 5$ and $\sigma_r = 15$

obtained are shown in figure 5 and 6. But in this case time taken for calculating these results are quite more than the other two methods. That is why they can't be compared.

- Again the results obtained by smaller values of σ_s and σ_r. Figure 7 and 8 shows the results obtained for $\sigma_s = 5$ and $\sigma_r = 15$. Hence now we make comparision with part a and b as the time consuption is quite same.
- Part a and part b are giving quite better than part c, i.e. PCA is giving better results than bilateral filter. We can easily see that the results obtained by part b based denoising are far better than the other two methods. However, we also observe slight loss in the sharpness of the denoised image obtained.
- The method of PCA based denoising is based on feature extraction, however, in bilateral filtering we are simply performing a non-linear spatial averaging.
- The Bilateral Filter basically blurs the given image which helps in denoising, however, in PCA based denoising, feature extraction is done using eigenvectors and it utilises the fact that natural images have a great deal of redundancy, i.e. several patches in distant regions of the image can be very similar.

(d) Consider that a student clamps the values in the noisy image'im1' to the [0,255] range, and then denoises it using the aforementioned

PCA-based filtering technique which assumes Gaussian noise. Is this approach correct? Why (not)?

Solution:

No, the approach is not correct. Clamping the noisy image to $[0, 255]$ changes the noise pattern and we may not be able to model it as Gaussian noise. The wiener like update in PCA requires noise power spectrum, clamping changes the noise spectrum and σ^2 cannot be used for ISNR.