ADVANCED IMAGE PROCESSING ASSIGNMENT 3

NAME - JYOTISH RANJAN
M. Tech AI,2023

QNS 1:

Problem 1: MMSE estimation for Laplacian source (10 points)
Consider the noise model $Y = X + Z$, where X is distributed according to a Laplace distribution $f_X(x) = \frac{1}{2\sigma_X} \exp(-\frac{|x|}{\sigma_X})$ with $\sigma_X = 1$ and $Z \sim \mathcal{N}(0, \sigma_Z^2)$ with $\sigma_Z^2 = 0.1$. Compute the minimum mean squared estimate (MMSE) of x given y. If the MMSE estimate can not be written as a closed form expression of y, plot the estimate $\hat{x}$ as a function of y for the given parameters of the system (Ref: E. P. Simoncelli, and E. H. Adelson, "Noise removal via Bayesian wavelet coring," Proc. of IEEE International Conference on Image Processing, 1996).

ANSWER:
Steps:

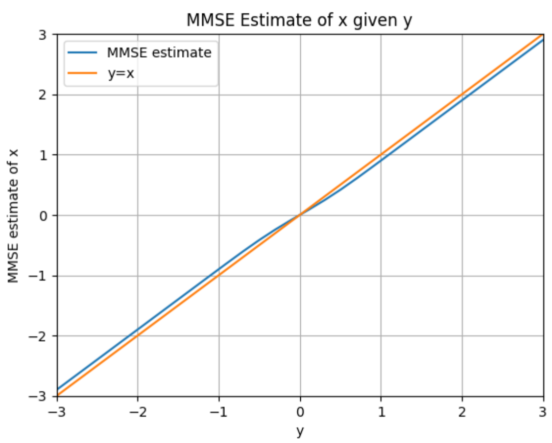
1. MMSE estimate is the conditional expectation of E [X | Y].

$$E[X|Y] = \int_{-\infty}^{\infty} x \cdot f_{X|Y}(x|y)\, dx$$

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x) \cdot f_X(x)}{f_Y(y)}$$

2. Since Laplacian distribution is not normalizable (the integral diverges), so there's no closed-form expression for E [X | Y]. Numerical methods are used to estimate it.
3. Generating a range of y values.
4. For each *y*, computing the conditional probability density function $f_{X|Y}(x|y)$.
5. Calculating the expectation *E[X|Y=y]* by integrating x. $f_{X|Y}(x|y)$. over x. Generating a range of y values.
6. Plotting estimated x as a function of y.

Results:



QNS 2.1:

Problem 2: Image denoising (30 points)
Take the lighthouse image provided to you, convert to greyscale and add white Gaussian noise with variance $\sigma_Z^2 = 100$ to it. Be sure to add noise in the grey scale domain where the range of pixel values is between 0 and 255. Compute and compare (subjectively and using mean squared error) the results of the following denoising methods

1. Low pass Gaussian filter: Vary the filter length in the set $\{3, 7, 11\}$ and standard deviations in the set $\{0.1, 1, 2, 4, 8\}$ to identify the filter with the best mean squared error (MSE).

ANSWER:

Steps:

1. Image Loading and noise augmentation.
2. Building a Gaussian Kernel of specified size and standard deviations.
3. Convolving noisy image with the kernels.
4. Calculating Mean Squared Error for each kernel on noisy image.

Results:

```
Filter lenth = 3
mse for std_dev: 0.1 is 99.151207708725
mse for std_dev: 1 is 89.65665454151114
mse for std_dev: 2 is 110.38623593856805
mse for std_dev: 4 is 115.76317882244582
mse for std_dev: 8 is 117.10916595255689
Filter lenth = 7
mse for std_dev: 0.1 is 99.151207708725
mse for std_dev: 1 is 114.96468731213692
mse for std_dev: 2 is 218.3453055592682
mse for std_dev: 4 is 266.75296238695773
mse for std_dev: 8 is 280.322695236847
Filter lenth = 11
mse for std_dev: 0.1 is 99.1512077087249
mse for std_dev: 1 is 115.0145536294038
mse for std_dev: 2 is 234.22880960540203
mse for std_dev: 4 is 327.56453361541554
mse for std_dev: 8 is 361.62846873847224
Best filter length: 3
Best standard deviation: 1
Best MSE: 89.65665454151114
```

BEST FILTER : 3 X 3 with standard deviation = 1

Analysis:

1. Generally increasing kernel size increases mean squared error.
   Reason:
   - Increased Smoothing leads to loss of high pass information.
   - Smoothing filters effectively averages image details along with noise. With increased kernel size means more noise will be considered in averaging leading to amplification of noise.
2. Generally increasing standard deviation of Gaussian filter increases mean squared error.
   Reasons:
   - With increasing standard deviation blur becomes more pronounced which results in loss of fine details and edges in the image.
   - With a larger standard deviation, it effectively spreads this noise over a larger area, amplifying its effect.

QNS 2.2:

2. Adaptive MMSE: Compute an adaptive version of the MMSE filter where the estimates are computed for patches of size 32×32 with overlap of 16 in the high pass domain. All pixels belonging to multiple patches need to be assigned the average values arising from the output due to multiple patches. You need to calculate the variance of the high pass coefficients of the original image and the variance of noise in the high pass image given the noise variance $\sigma_Z^2 = 100$ in the pixel domain. You cannot simulate noise in the high pass domain separately and estimate its variance.

Answer:

Steps:

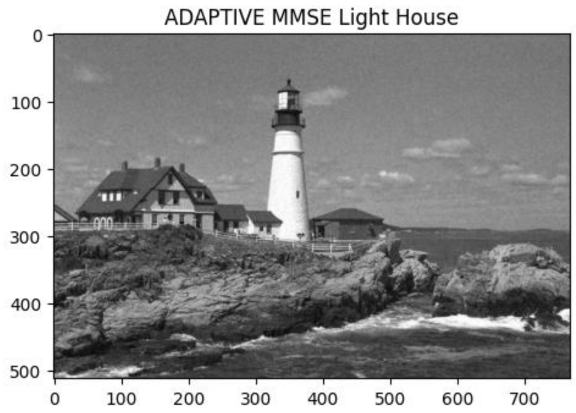
1. Image Loading and noise augmentation.
2. Extract high pass component of noisy image (y1).
3. Now divide y1 into patches and for each patch calculate it's variance (var_y1).
4. Calculate variance of z1 (var_z1). Here noise variance = 100

$$\text{var\_z1} = \text{noise\_variance} \times \left[(1 - w_{0,0})^2 + \left(\sum_{i=1,j=1}^{m,n} w_{i,j}^2 - w_{0,0}^2\right)\right]$$

5. Calculate var_x1 = var_y1 – var_z1
6. Best estimated patch (var_x1 / var_x1 + var_z1) * patch + (patch of (y – y1))
7. Do this for each patch and overlapping patch should be averaged.

<u>Results:</u>

MSE for Adaptive MMSE: 41.78



<u>QNS 2.3:</u>

3. Adaptive Shrinkage: Shrinkage estimator on the high pass coefficients of the noisy image with the threshold optimized using *SureShrink* for patches of size 32×32. Here it is implicit that you need to the determine the threshold parameter t for every patch. (Ref: D. L. Donoho, and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," Journal of the American Statistical Association, vol. 90, no. 432, 1995)

<u>Answer:</u>

Steps:

1. For each patch we have to minimise SURE (y1) because minimising loss of denoising using SURE shrinkage is equal to minimising SURE function.

$$\mathrm{SURE}(y_1) = MN\sigma_Z^2 + \sum_{m,n} \left(g(y_{1_{m,n}})^2\right) + 2\sigma_Z^2 \sum_{m,n} \left(\frac{d}{dy_{1_{m,n}}} g(y_{1_{m,n}})\right)$$

$$\mathrm{SURE}(t; y_1) = MN\sigma_Z^2 + \sum_{m,n} \left(\min(y_{1_{m,n}}, t)\right)^2 - 2\sigma_Z^2 \cdot |\{(m,n) : |y_{1_{m,n}}| < t\}|$$

2. Make all intensities values of the patch as possible thresholds. Check which value of t which minimises SURE (t ; y1).
3. Using this t, apply below formulae to create best estimated patch.

$$\hat{x}_1(m,n) = \mathrm{sign}(y_1(m,n)) \times \max(|y_1(m,n)| - t, 0)$$

4. Do this all patches and then add this to low pass component of the noisy image.

<u>Results:</u>

MSE FOR SURE SHRINKAGE: 49.54

```
Filter lenth = 3
MSE for SURE MMSE for std_dev = 0.1 is --> : 99.02157632962651
MSE for SURE MMSE for std_dev = 1 is --> : 49.55428387127734
MSE for SURE MMSE for std_dev = 2 is --> : 54.494135388315755
MSE for SURE MMSE for std_dev = 4 is --> : 55.48111789484051
MSE for SURE MMSE for std_dev = 8 is --> : 55.9415495273205
Filter lenth = 7
MSE for SURE MMSE for std_dev = 0.1 is --> : 99.02157632962651
MSE for SURE MMSE for std_dev = 1 is --> : 50.79796677234115
MSE for SURE MMSE for std_dev = 2 is --> : 58.30005949509687
MSE for SURE MMSE for std_dev = 4 is --> : 60.79681724238475
MSE for SURE MMSE for std_dev = 8 is --> : 61.487134466714885
Filter lenth = 11
MSE for SURE MMSE for std_dev = 0.1 is --> : 99.02157632962651
MSE for SURE MMSE for std_dev = 1 is --> : 50.86828668873372
MSE for SURE MMSE for std_dev = 2 is --> : 59.13898586600871
MSE for SURE MMSE for std_dev = 4 is --> : 62.82874270786447
MSE for SURE MMSE for std_dev = 8 is --> : 64.20284024074674
Best filter length: 3
Best standard deviation: 1
Best MSE: 49.55428387127734
```

COMPARITIVE STUDY :

Filter Length	Std_Dev	LOW_PS_MSE	ADP_MSE	SURE_MSE
3	0.1	99.021576	NaN	99.021576
3	1.0	89.429065	41.696719	49.554284
3	2.0	110.144445	41.655622	54.494135
3	4.0	115.519718	41.809636	55.481118
3	8.0	116.865388	41.856411	55.941550
7	0.1	99.021576	NaN	99.021576
7	1.0	114.741849	41.748560	50.797967
7	2.0	218.146761	44.084133	58.300059
7	4.0	266.569696	45.549010	60.796817
7	8.0	280.144730	45.967781	61.487134
11	0.1	99.021576	99.021576	99.021576
11	1.0	114.791521	41.748585	50.868287
11	2.0	234.005719	44.537547	59.138986
11	4.0	327.376151	47.239905	62.828743
11	8.0	361.471930	48.158560	64.202840

- MSE for low pass of filter is more than other two because it is neglecting high pass details.
- Theoretically, Adaptive shrinkage estimation should produce less MSE than adaptive MMSE but in my case it turns out that adaptive MMSE is slightly better. This can be due to

  Reasons:
  1. Parameter Selection: The performance of the method can be highly dependent on the chosen parameters. For instance, the size and standard deviation of the Gaussian filter can significantly impact the outcome.
  2. Noise Characteristics: The type and level of noise in the image can also influence the effectiveness of the denoising methods. Some methods might be more suitable for particular types of noise.
  3. Image Characteristics: The specific characteristics of the image, such as the amount and type of detail, can influence the performance of the denoising methods as well.