DAIICT

Assignment 4

Advanced Image Processing

Kishan Vaishnani - 202011004
Dhyanil Mehta - 202011032
3-21-2021

Table of Contents

Task 1

Problem Statement

Find out the difference between averaging operation and weighted averaging (higher weight to center pixel) operation on Fig. 1 by applying spatial filtering. For this purpose, convolve the image with 9 × 9 masks (averaging mask and weighted averaging mask).

Figure 1

Library Used: NumPy

Method

1. Convolution formula used

   We used the following formula for implementing convolution function:

$$\boldsymbol{x[n] * h[n] = \sum_{k=0}^{N-1} x[k]h[n-k]}$$

   where we took x[n] = Lena Image, and h[n] = Averaging/Weighted Averaging 9x9 mask kernel

2. For rotating matrix by 180°

   numpy.flipud(matrix) -> matrix

   numpy.fliplr(matrix) -> matrix

   Upper 2 library we have used before convolution as we need to rotate kernel matrix before apply to image.

Observation

1. Average 9*9 filter

2. Weighted average 9*9 filter with higher center pixel value

Conclusion

- Averaging filter blurs out the image and the blurriness is dependent on the size of the averaging filter.
- Weighted averaging filter also blurs the image but the intensity values are higher than that of averaging filter.

Task 2

Problem Statement

Consider the image of Figure 2 and apply unsharp masking and high boost filtering to enhance the edges of the image. Consider a row of the original image and plot the intensity values. This kind of plot is known as intensity profile of that row. Plot similar intensity profiles (consider the same row) for the blurred image, unsharp mask, sharpened image, and the high-boost filtered image. Explain your observation.

Figure 2

Library Used: OpenCV

Method

1. Equation used for unsharp & high boost filtering

   $$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$$

   $$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

   $$where, k = 1 : Unsharp\ masking$$
   $$k > 1 : Highboost\ filtering$$

   $\bar{f}(x,y)$: Is a smoothened/blurred actual image. We used Gaussian blurring to blur the image.

   cv2.GaussianBlur(image, kernel size, sigma) -> smoothen image

   - For high boost filtering, we have used k = 2

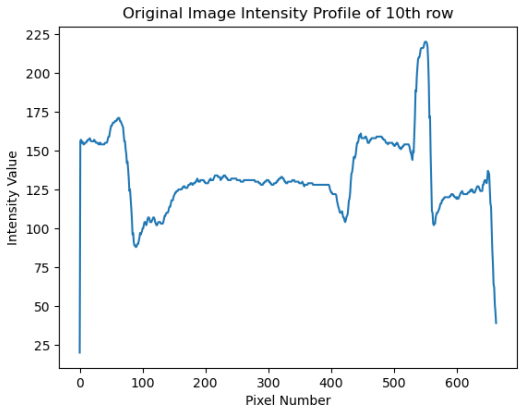
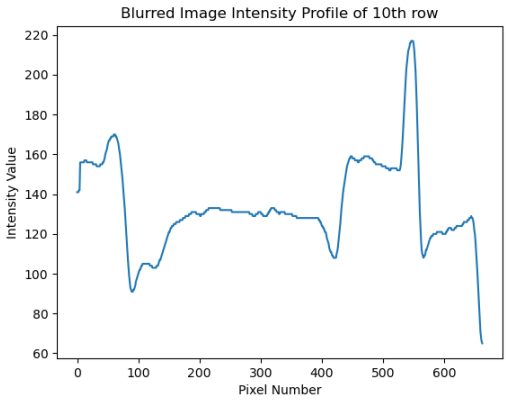
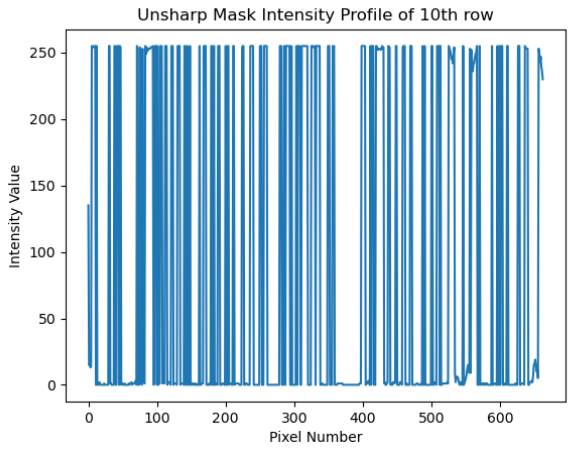
Observation

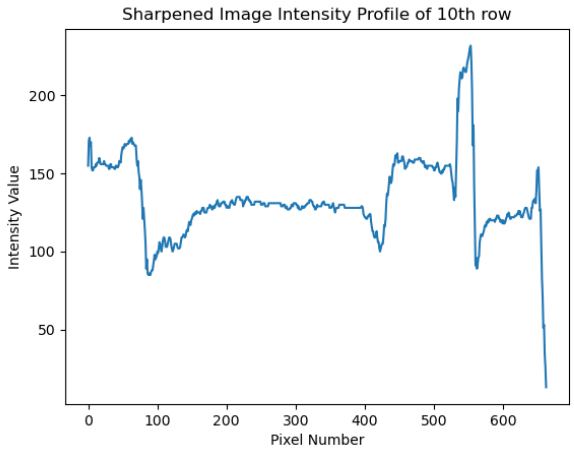
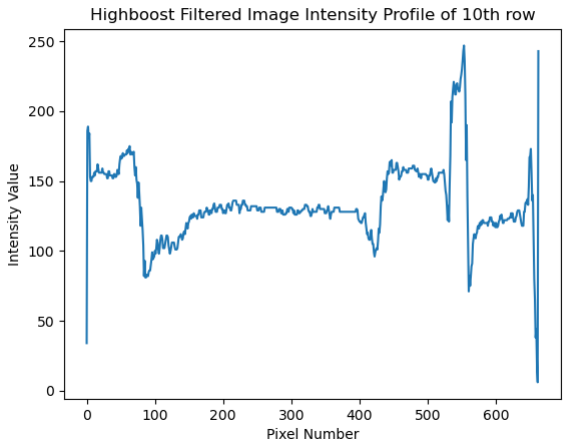
1. Gaussian blurred image with kernel size of (10,10) and sigma 10

2. Unsharp masked image

3. High boost filtered image with k = 2

4. Plot intensity profile of 10th row

	Original • Intensity values of the 10th row of original Lena image. • Many rough and sawtooth edges can be seen. • Significant drops and highs in intensity values.
	Blur image • Rough edges are smoothed out. • All intensity values dropped by small amount. • Magnitude of drops and highs in intensity values increased.
	Unsharp mask • Mask has significantly more and bigger heights and drops. • Most intensity values lie either at the maximum value 255 or minimum value 0.

	Sharpened image • Roughness of edges and plateaus increased. • All intensity values increased by a small amount.
	High-boost filtered image • Increase in roughness of the graph more than the sharpened image with unsharp mask. • Magnitude of heights and drops increased.

Conclusion

- Gaussian blurring reduces details of the image overall. Edges are not preserved. Overall image quality is reduced.
- Unsharp masking enhances the edges of the image and increases the overall quality of the image.
- High-boost filtering also enhances the edges but they are enhanced much that the quality of the image is degraded as seen from the observation.

Task 3

Problem Statement

Smooth the image of Figure 3 by using Bilateral filtering. Compare the result with average filtering and weighted average filtering. Which result is better? –Explain your understanding.

Figure 3

Library used: OpenCV

Method

1. Bilateral filtering

   cv2.bilateralFilter(image, kernel size, σs, σr)

$$\boldsymbol{BF[I]p = \frac{1}{W} \sum_{q \in S} G_{\sigma s}\left(||p - q||\right) G_{\sigma r}\left(|I_p - I_q|\right) I_q}$$

$$\textbf{Where } G_\sigma(x) = \frac{1}{\sqrt{2\Pi\sigma}} e^{\left(\frac{-x^2}{2\sigma^2}\right)}$$

   It will return a smooth image while also preserving the edges/edge information.

Observation

1. Avg. filter v/s weighted Avg. filter v/s Gaussian Blur v/s Bilateral Image

			
Average Filter • Normal 9x9 averaging filter. • Blurs the image. • Edges not preserved.	Weighted Avg. Filter • Weighted 9x9 averaging filter. • Blurs the image. • Image is slightly bright. • Edges not preserved.	Gaussian Blur • Gaussian 10x10 filter with sigma = 10. • Blurs the image. • Edges not preserved.	Bilateral Image • Bilateral 15x15 filter with space region = 75 and range region = 75. • Smoothens the image. • Edges preserved.

Conclusion

- Bilateral filter proves out to be the best in preserving the edges while blurring or smoothing out the image as compared to averaging or gaussian filter.

Task 4

Problem Statement

Consider the image of Fig.2 and process it to bring out more skeletal detail. (You may use combination of sharpening and smoothing operations).

Figure 4

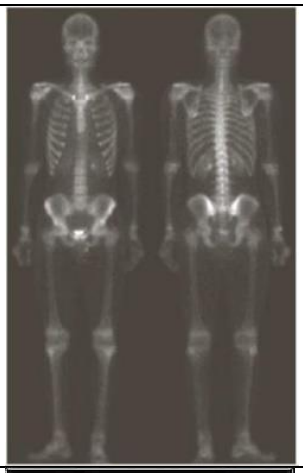
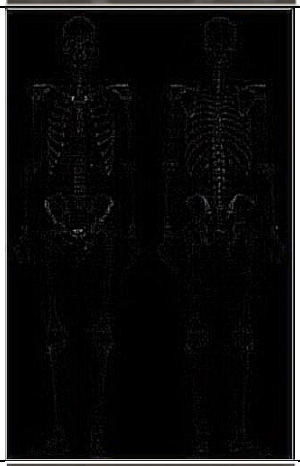
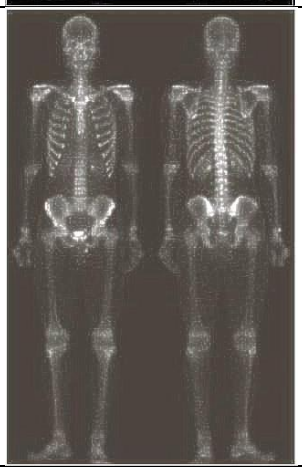
Library used: OpenCV

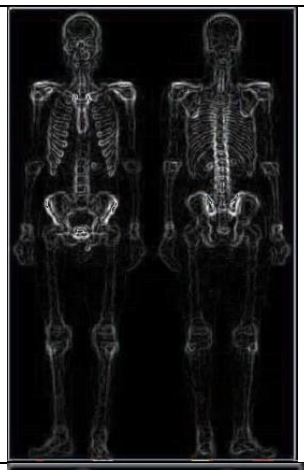
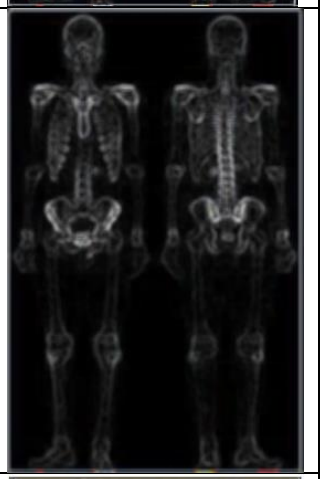
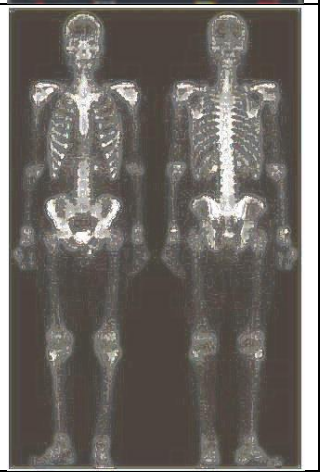
Method

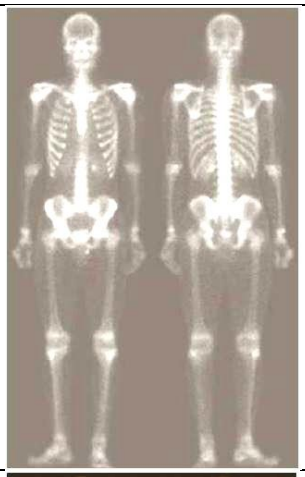
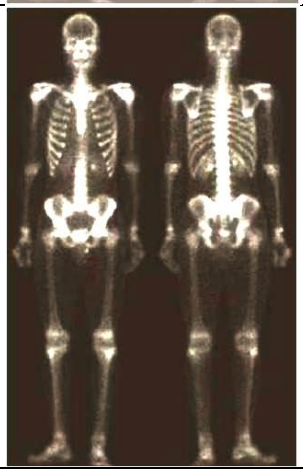
1. Applying/Convolving kernel filter on/with image

   cv2.filter2D(image, depth of image, kernel) -> image after applying kernel on it

Observation & steps:

1.	Input Image	
2	3x3 Laplacian filter of image Kernel= [[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]	
3	Sharpened image obtained by adding Laplacian filter with input image	

4	Sobel gradient of image with some gamma transformation applied to enhance it X Kernel= [[-1, -2, -1], [0, 0, 0], [1, 2, 1]] Y Kernel= [[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]]	
5	Sobel gradient smoothed by 5x5 averaging filter mask	
6	Generated mask by the product of Laplacian filter sharpened image and Smooth Sobel gradient	

7	Applying generated mask on Laplacian sharpened applied image	
8	Applying gamma transformation ($\gamma = 3.0$) on the image generated on the 7th step to adjust intensity	

Conclusion

- By applying a series of operations involving Laplacian filter mask and Sobel gradient mask along with some gamma transformation, skeletal details of the skeleton in the image were highlighted.