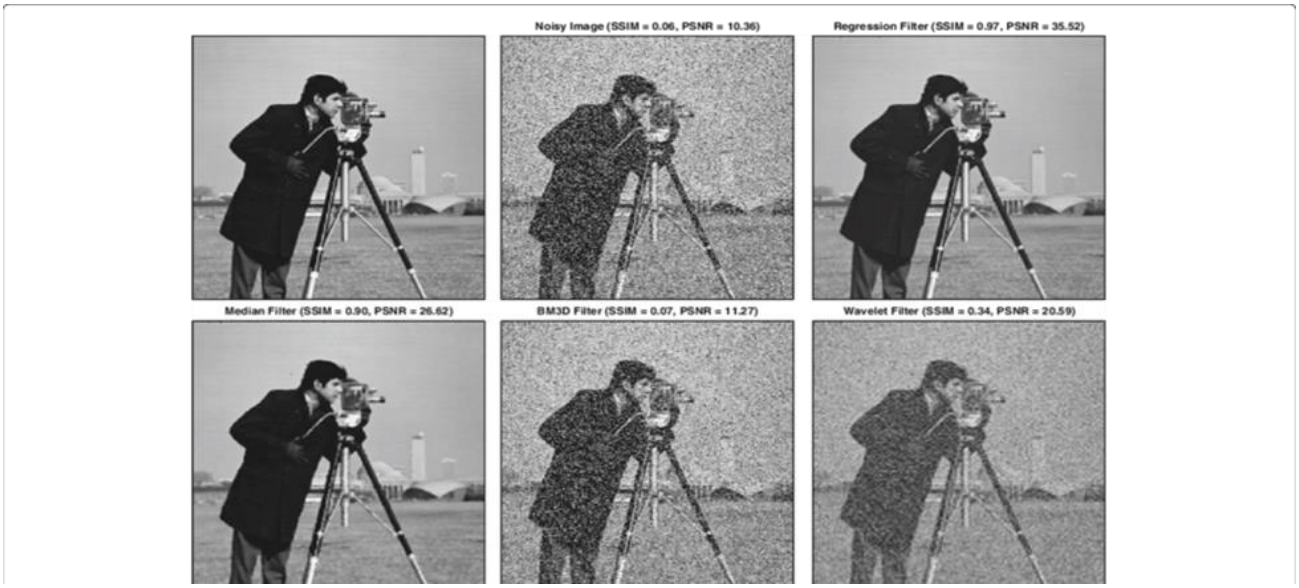
Project Report

Image Processing

—Introduction to Data, Signal and Image Analysis



Made By -

Jay Ajmera (J004)

Atharva Rode(J056)

Kallind Soni(J065)

Prsni Kanani(J073)

Introduction

Image processing refers to the use of algorithms and techniques to manipulate digital images. It involves a wide range of operations, including filtering, enhancement, restoration, segmentation, feature extraction, and recognition. Image processing has many applications, such as in medical imaging, satellite imagery, surveillance, and digital photography. The goal of image processing is to improve the quality of images, extract useful information, and make them more suitable for analysis and interpretation by humans or machines. Some common techniques used in image processing include convolution, Fourier transforms, edge detection, and image compression.

Objective

- To apply various image processing techniques like blurring , motion blur , sharpening , salt and pepper noise (noise adding) and Denoising (With mean and median).
- To improve the image quality , extract information and enhance human perception.

Applications

Image processing techniques have a wide range of applications across various fields. Here are a few examples:

- Medical Imaging: Image processing techniques are used to improve medical imaging, such as X-rays, MRI, CT, and ultrasound. These techniques can be used to enhance images, extract information, and detect abnormalities.
- Robotics: Image processing techniques can be used in robotics to enable robots to detect and recognize objects, navigate their environment, and perform tasks autonomously.
- Surveillance and Security: Image processing techniques are used in surveillance and security systems to detect and recognize people and objects, track their movements, and identify suspicious behavior.
- Agriculture: Image processing techniques can be used in agriculture to analyze crop health, detect pests and diseases, and optimize irrigation and fertilizer use.
- Entertainment: Image processing techniques are used in the entertainment industry to create special effects, animation, and virtual reality experiences.

Blurring

Blurring is a technique in digital image processing in which we perform a convolution operation between the given image and a predefined low-pass filter kernel. The image looks sharper or more detailed if we are able to perceive all the objects and their shapes correctly in it.

E.g. An image with a face looks clearer when we can identify eyes, ears, nose, lips, forehead, etc. very clearly. This shape of the object is due to its edges. So, in blurring, we simply reduce the edge content and make the transition from one color to the other very smooth. It is useful for removing noise.

Blurring Image Using Python

```
#full pic blur
id_kern = np.array([[0,0,0],[0,1,0],[0,0,0]])
id_kern
```

```
blurring_kern3 = np.ones((3,3),dtype = np.float32)/9.0
blurring_kern11 = np.ones((11,11),dtype = np.float32)/121.0
blurring_kern11
```

```
out1 = cv2.filter2D(img,-1,id_kern)
out2 = cv2.filter2D(img,-1,blurring_kern3)
out3 = cv2.filter2D(img,-1,blurring_kern11)
```

```
cv2.imshow('same',out1)
cv2.imshow('3',out2)
cv2.imshow('11',out3)
cv2.waitKey(0)
```

Output

Original Image

Blurred Image

Sharpening

Image sharpening is an effect applied to digital images to give them a sharper appearance. Sharpening enhances the definition of edges in an image. The dull images are those which are poor at the edges. There is not much difference in background and edges. On the contrary, the sharpened image is that in which the edges are clearly distinguishable by the viewer. Almost all lenses can benefit from at least a small amount of sharpening. Sharpening is applied in-camera to JPEG images at a level specified by the photographer or at the default set by the camera manufacturer. Lightroom automatically applies some sharpening to images unless it is instructed not to. For further sharpening, there are Photoshop techniques and filters like Unsharp Mask and Smart Sharpen to do the job. Sharpening works best on images whose blur did not stem from camera shake or drastically missed focus, though minor camera shake or slightly out-of-focus shots can also be fixed with sharpening.

Sharpening Image Using Python

```
#Sharpening
#subtract blurred image from og image
#Gauusian kernel for sharpening
gaussian_blur = cv2.GaussianBlur(img,(7,7), 2)
#Sharpening using addweighted( )
sharpened1 = cv2.addWeighted(img, 1.5,gaussian_blur,-0.5,0)
sharpened2 = cv2.addWeighted(img, 4.5,gaussian_blur,-3.5,0)
sharpened3 = cv2.addWeighted(img, 7.5,gaussian_blur,-6.5,0)
#
```

```
#Showing the sharpened Images
cv2.imshow('Sharpened3' , sharpened3)
cv2.imshow('Sharpened2' , sharpened2)
cv2.imshow('Sharpened1' , sharpened1)
cv2.waitKey(0)
```

Output

Motion Blur

Motion blur is a common form of camera shake which can occur when the shutter speed on your camera is too slow for you to hand hold. It can also happen when you have shaky hands and cannot mount the camera on a tripod.

Motion blur is when an object appears blurry because it is moving too quickly for the camera to capture a clear image. To help understand motion blur we need to take a step back and look at how cameras work.

Cameras are like our eyes in that they record the light that reflects off of objects in our environment. There are two key differences, though:

1. Cameras have adjustable lenses that control the amount of light that reaches the film or sensor.

2. Cameras can capture light over longer periods of time than our eyes can.

Motion Blur Using Python

```
#motion blur
```

```
size = 15
kernel = np.zeros((size,size))
kernel[:,7]= np.ones(size)
#for horizontal motion blurring make the middle column = 1
kernel = kernel/size
output= cv2.filter2D(img, -1, kernel)
kernel
```

```
cv2.imshow( 'motion blur' , output)
cv2.imshow('og', img)
cv2.waitKey(0)
```

Output

Salt And Pepper Noise

Noise means random disturbance in a signal in a computer version. In our case, the signal is an image. Random disturbance in the brightness and color of an image is called Image noise. It is found only in grayscale images (black and white image). As the name suggests salt (white) in pepper (black)–white spots in the dark regions or pepper (black) in salt (white)–black spots in the white regions. In other words, an image having salt-and-pepper noise will have a few dark pixels in bright regions and a few bright pixels in dark regions. Salt-and-pepper noise is also called impulse noise. It can be caused by several reasons like dead pixels, analog-to-digital conversion error, bit transmission error, etc.

Adding salt-and-pepper noise in an image –

```
# orginal image
img = cv2.imread("C:/Users/Jay Ajmera/Downloads/imgg.jpg",0)
img = img/255

cv2.imshow('original image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

# blank image
x,y = img.shape
g = np.zeros((x,y), dtype=np.float32)

# salt and pepper amount
pepper = 0.1
salt = 0.95

# create salt and peper noise image
for i in range(x):
    for j in range(y):
        rdn = np.random.random()
        if rdn < pepper:
            g[i][j] = 0
        elif rdn > salt:
            g[i][j] = 1
        else:
            g[i][j] = img[i][j]

cv2.imshow('image with noise', g)
cv2.waitKey(0)
```

Output

Original image

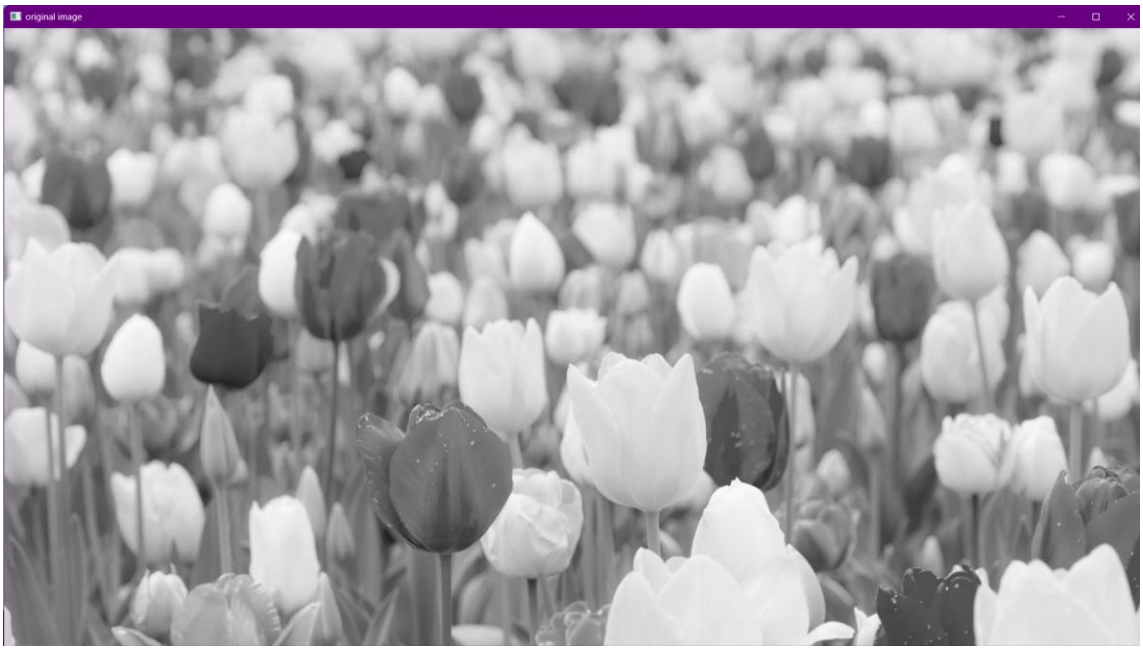


Image with noise

Denoising

Denoising is a task in image processing and computer vision that aims to remove or reduce noise from an image. Noise can be introduced into an image due to various reasons, such as camera sensor limitations, lighting conditions, and compression artifacts. The goal of denoising is to recover the original image, which is considered to be noise-free, from a noisy observation.

Mean Filter

The mean filter can be expressed with a convolution product. Indeed, consider the case where the neighborhood is a square of size N X N pixels.

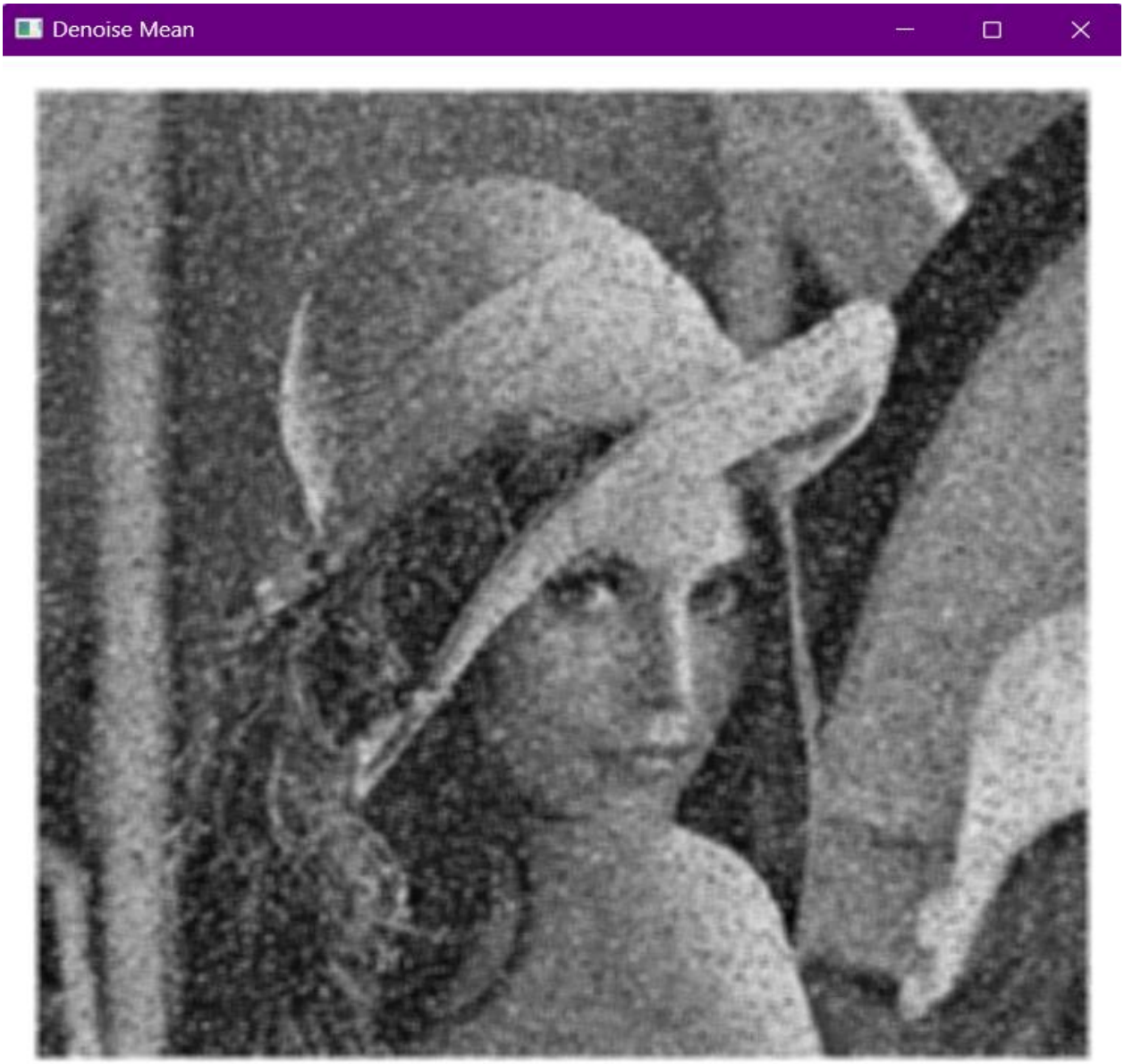
- The mean filter calculates the average of the pixels in a neighbourhood.
- The mean filter can be written as a convolution.
- The noise is reduced by averaging the intensities but the image is blurred.

Denoising Using Mean (Code)

```
from skimage import img_as_ubyte

## original image
img = cv2.imread("C:/Users/Jay Ajmera/Downloads/rode_wali.jpg")
img = img/img.max() # normalize the pixel value (0~1)

m = 5
n = 5
denoise_mean = cv2.blur(img, (m,n))

# median filter
img_noise_median = np.clip(img, -1, 1) #pixel value range
img_noise_median = img_as_ubyte(img_noise_median) #convert to uint8
denoise_median = cv2.medianBlur(img_noise_median, 5)

# preview the images
cv2.imshow('Original Image', img)
cv2.imshow('Image + Noise', img)
cv2.imshow('Denoise Mean', denoise_mean)

cv2.waitKey(0)
```


Original Image

Output



Median Filter

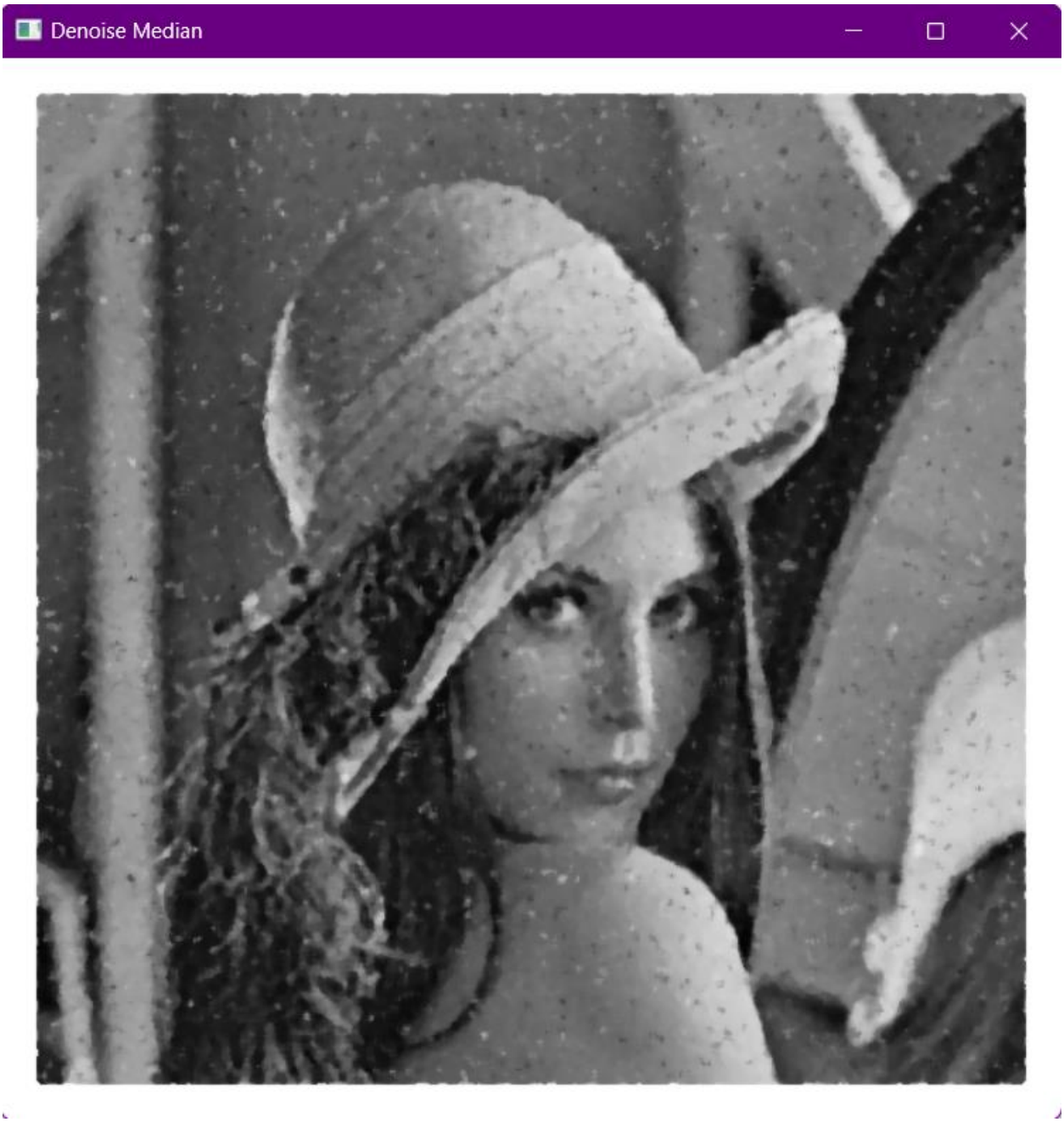
The median of a set of numbers is the element m of the set such that there are as many numbers smaller than m as there are numbers larger than m. For example, the median of {1,2,4,8,16} is 4.

The median filter is excellent for denoising an image in the case of salt-and-pepper noise because it does not blur the image, as a mean filter would do. Despite its name, the median filter is not a filter because it does not respect the linearity property. Therefore it cannot be written as a convolution.

Denoising Using Median (Code)

```
cv2.imshow('Denoise Median', denoise_median)
cv2.waitKey(0)
```

Output



Conclusion

In conclusion, image processing plays a vital role in the enhancement and restoration of digital images. The techniques such as blurring, motion blur, sharpening, salt and pepper noise, and denoising are essential for achieving the desired image quality. Blurring and motion blur can be used creatively to produce aesthetically pleasing effects, while sharpening can enhance the image's details and sharpness. On the other hand, salt and pepper noise can negatively impact the image's quality, but denoising can remove the noise and restore the image's clarity. A deep understanding of these techniques and their appropriate application can significantly improve the overall quality and usability of digital images in various fields, including photography, medical imaging, and security systems.

References

1. https://datacarpentry.org/image-processing/06-blurring/
2. https://www.analyticsvidhya.com/blog/2021/08/sharpening-an-image-using-opencv-library-in-python/
3. https://www.geeksforgeeks.org/add-a-salt-and-pepper-noise-to-an-image-with-python/amp/
4. https://medium.com/analytics-vidhya/remove-salt-and-pepper-noise-with-median-filtering-b739614fe9db
5. https://www.geeksforgeeks.org/noise-removal-using-median-filter-in-c/