Computer Vision

Assigment-2

Submitted By

Krishna Kumari Ravuri –M22AI567

Master of Technology

In

Data and Computational Science

Under Guidance of

Dr. Riby Abraham Boby

Department of Mechanical Engineering,

Indian Institute of Technology, Jodhpur

Code Files:

Question1:

Implementation of adding and removing "**salt and pepper (Impulse)"** and "**Gaussian noise**". ***M22AI567_Q1. ipynb***

Question2:

Implementation of blur and de-blur

"**Gaussian**" and "**Motion**". ***M22AI567_Q2. ipynb***

Question 3:

Implementation of Blending Two Images. ᴛake the image of a dog and place it on the road in the city image. ***M22AI567_Q3. ipynb***

Question 1

1.1 Adding and Removing Salt and pepper noise.

1.2 Adding and Removing Gaussian noise.

My Learning to know background of the problem statement:

I stared exploring about image capturing technique and reason for the noise occurrence and types of different noises and how to remove such noises.

Image: An image is a visual representation of an object, scene, or idea, often captured through art, photography, or technology. It conveys information and emotions through colors, shapes, and textures.

Noise in the Image: Noise in images refers to random or unwanted variations in pixel values that can degrade the quality and clarity of the image. It appears as irregular patterns of brightness or color that were not present in the original scene. Noise can arise from factors such as sensor limitations in cameras, low light conditions, or transmission errors in digital communication.

Types of image noise:

Gaussian Noise: Gaussian noise is random noise that follows a Gaussian or normal distribution. It's characterized by small variations in pixel values and can make an image appear grainy.

Salt and Pepper Noise: salt and pepper noise appears as randomly occurring white and black pixels scattered throughout the image, resembling salt and pepper. It can result from sensor errors or transmission issues.

Speckle Noise: Speckle noise creates small bright and dark regions in an image, resembling grainy patterns. It's common in ultrasound or radar images.

Quantization Noise: quantization noise is introduced when the continuous range of pixel values in an image is discretized into a limited set of levels, causing rounding errors and introducing small variations.

Chromatic Noise: chromatic noise is also known as color noise, this affects the color channels of an image and can result in discolored or speckled regions.

Temporal Noise: Temporal noise occurs over time in video sequences and can be caused by fluctuations in lighting conditions, sensor variations, or transmission errors.

Pattern Noise: Pattern noise is a consistent and repetitive variation in pixel values across an image. It can result from imperfections in the camera sensor or image processing pipelines.

Thermal Noise: This noise is caused by the random thermal motion of electrons in electronic components, such as camera sensors. It tends to be more pronounced in low-light conditions.

These are some of the different types of noise, can have different origins and effects on image quality, we can apply noise reduction techniques to enhance the clarity and visual appeal of images.

Filtering techniques to remove noises in the image:

There are several filtering techniques commonly used to remove noise from images. Here are some of the prominent filtering techniques:

Mean Filter: This filter replaces each pixel with the average value of its neighboring pixels within a specified window. It's effective at reducing high-frequency noise but may blur edges and fine details.

Median Filter: The median filter replaces each pixel with the median value of its neighboring pixels within a window. It's particularly effective at removing salt-and-pepper noise while preserving edges and fine features.

Gaussian Filter: The Gaussian filter convolves the image with a Gaussian kernel. It reduces noise by averaging pixel values within the kernel window while giving more weight to closer pixels. It's widely used and can be adjusted to control the amount of smoothing.

Bilateral Filter: Bilateral filter considers both spatial and intensity differences while filtering. It smooths the image while preserving edges by weighting neighboring pixels based on their spatial and intensity similarity.

Wiener Filter: The Wiener filter is a frequency domain filter that estimates the power spectral density of noise and signal. It can be effective when the noise characteristics are known.

The choice of filtering technique depends on the specific type of noise in the image and the desired level of noise reduction. Some techniques work better for certain types of noise while others are more versatile. combination of techniques and parameter tuning might help to achieve the desired noise reduction to maintaining image quality.

My Approach on Salt and pepper noise and de-noise:

Salt and pepper is the one of the type of impulse noise.

Salt and pepper noise observed on black and white (grayscale) images. As the name says salt means white spots and pepper means black spots on the image.

an image with salt-and-pepper noise means isolated bright pixels (salt) in dark areas and isolated dark pixels (pepper) in bright areas. This noise is sometimes also referred to as impulse noise due to its scattered and sporadic nature.

function to add salt and pepper noise:

To achieve this implemented a function to randomly selects pixels in the input image and changes their intensity values to either 255 (white) or 0 (black), simulating the addition of salt and pepper noise. The number of pixels to be modified is determined randomly within a specified range. The function then returns the modified image with the added noise.

function to remove salt and pepper noise:

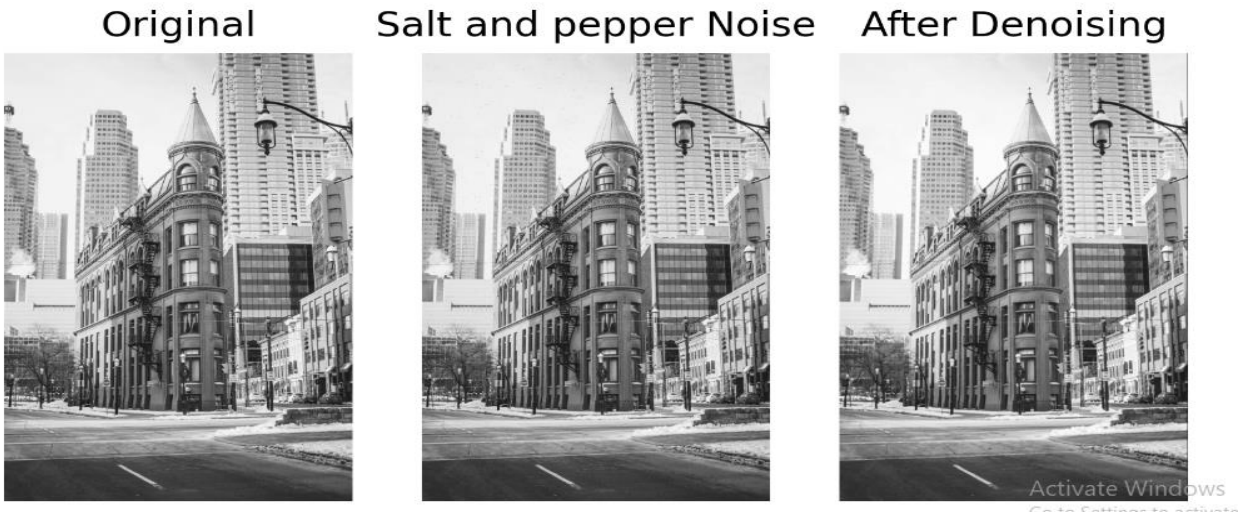
To achieve this, I have used median filter.

median filter is a popular choice for removing salt-and-pepper noise from images while preserving edges and details. It replaces each pixel's value with the median value of its local neighborhood, effectively removing extreme values caused by the noise.

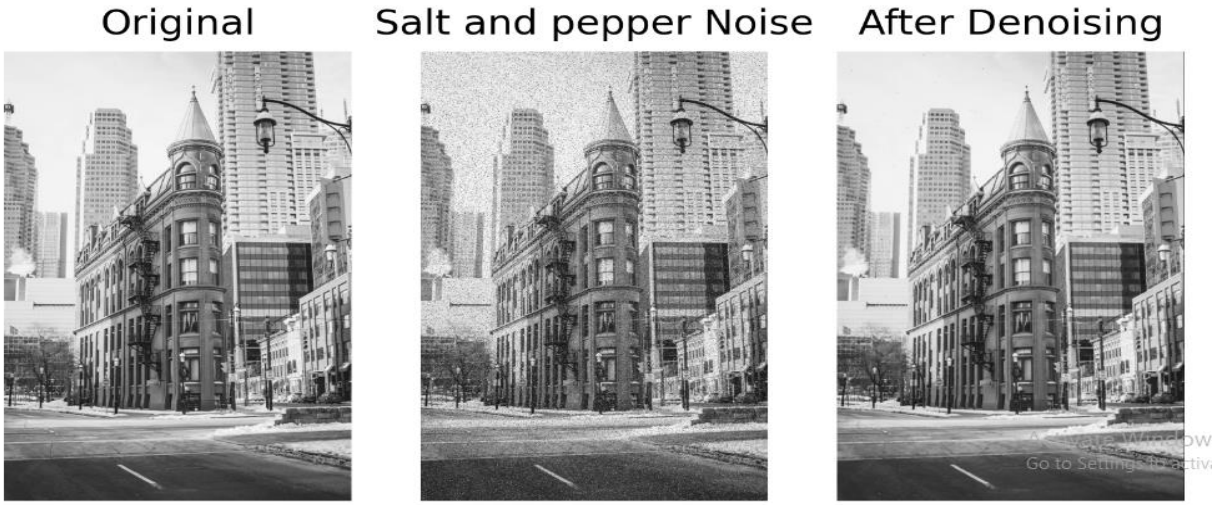
Salt and pepper noise and de-noise for image1

Output images for different random ranges to add black and white spots and different kennel size to de-noise.

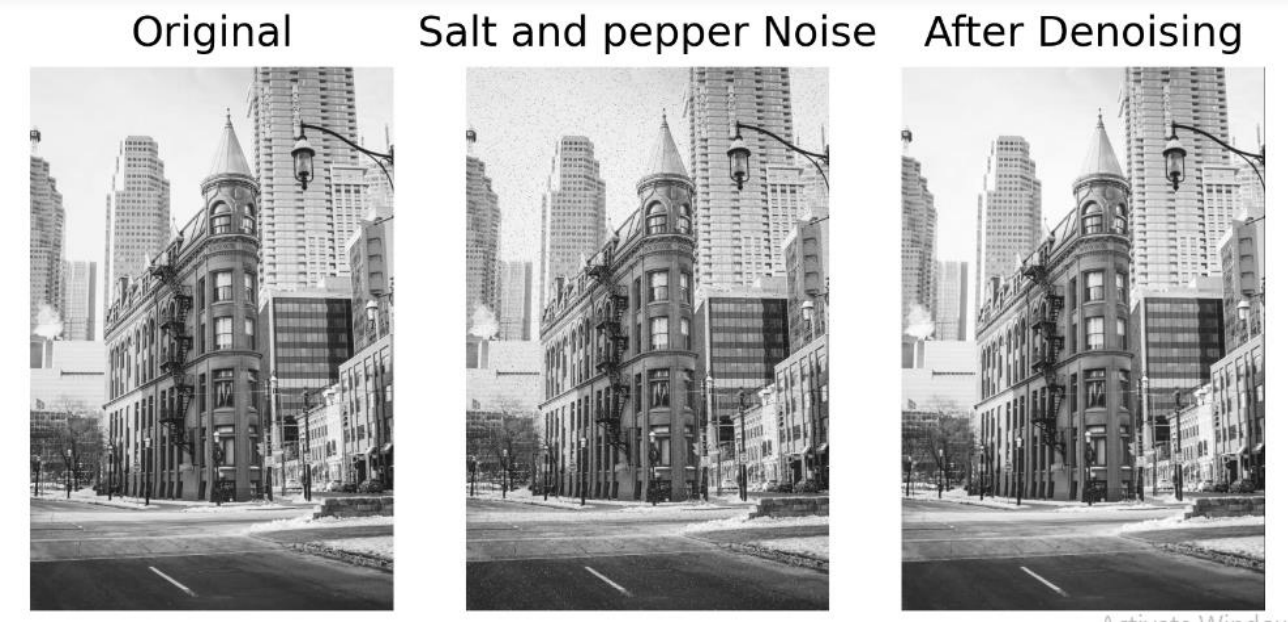
Salt and pepper noise with random pixels range from (300, 1000) De-noise with kennel size 3



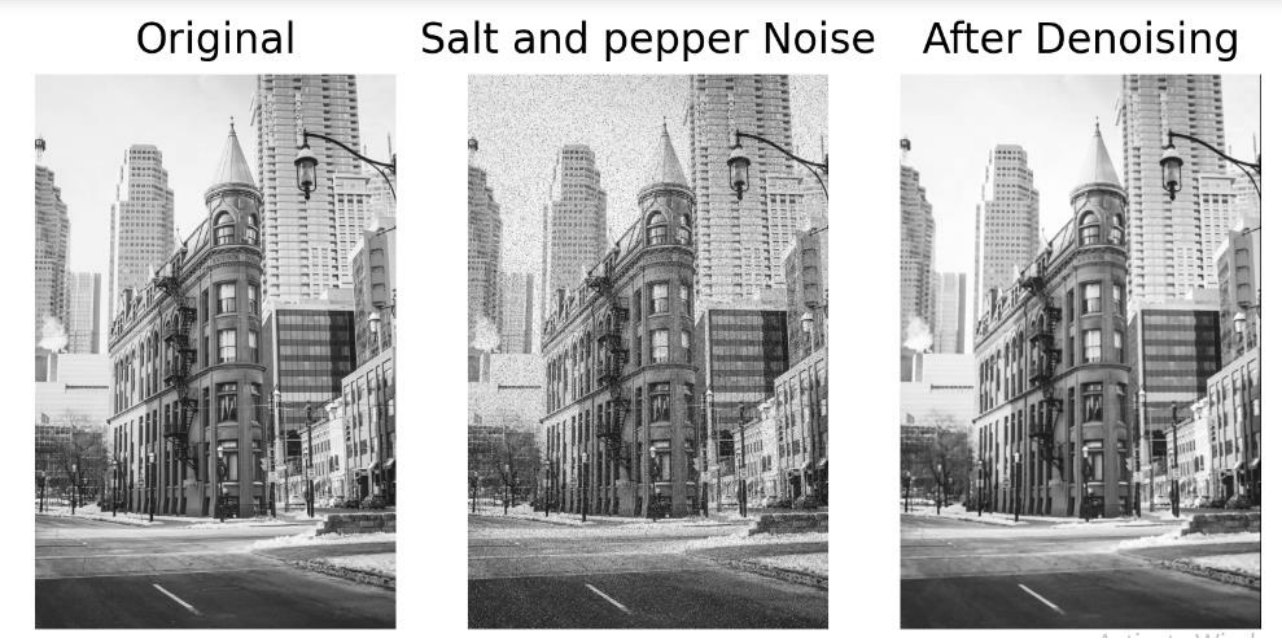
Salt and pepper noise with random pixels range from (300, 100000) De-noise with kennel size 3



Salt and pepper noise with random pixels range from (1000, 100000) De-noise with kennel size 3

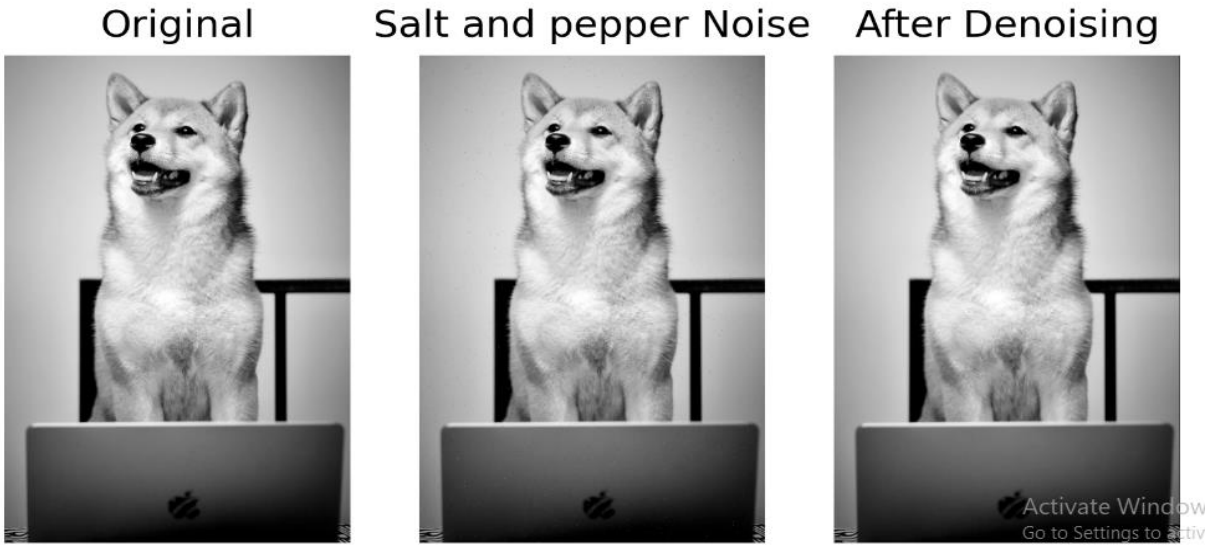


Salt and pepper noise with random pixels range from (100, 100000) De-noise with kennel size 5

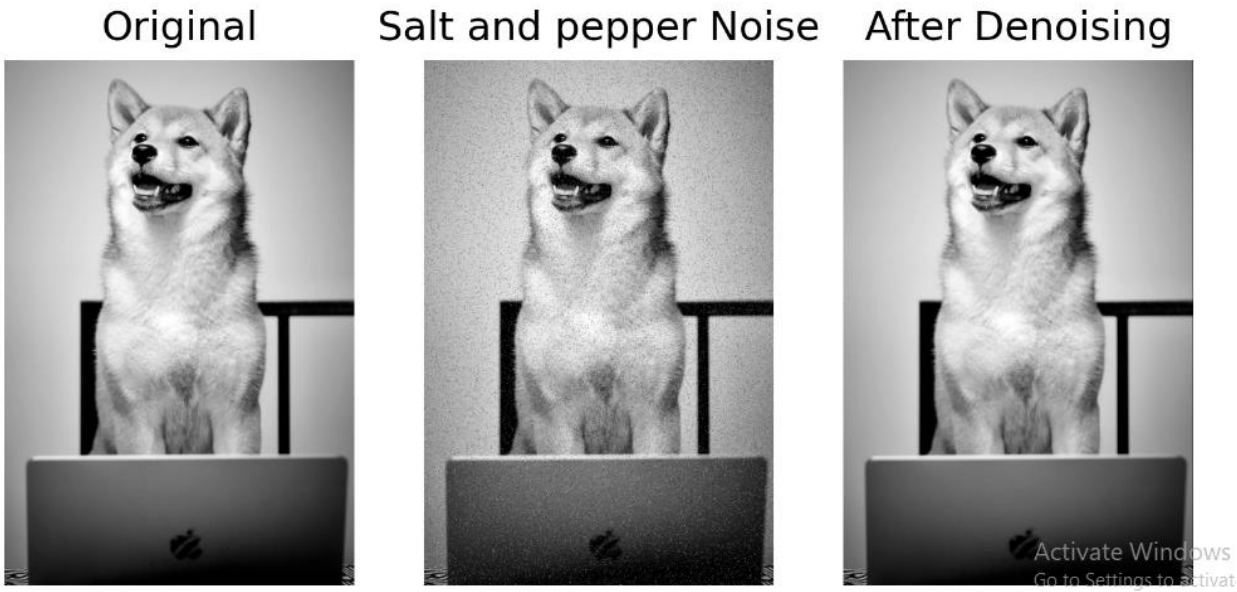


Salt and pepper noise and de-noise for image2:

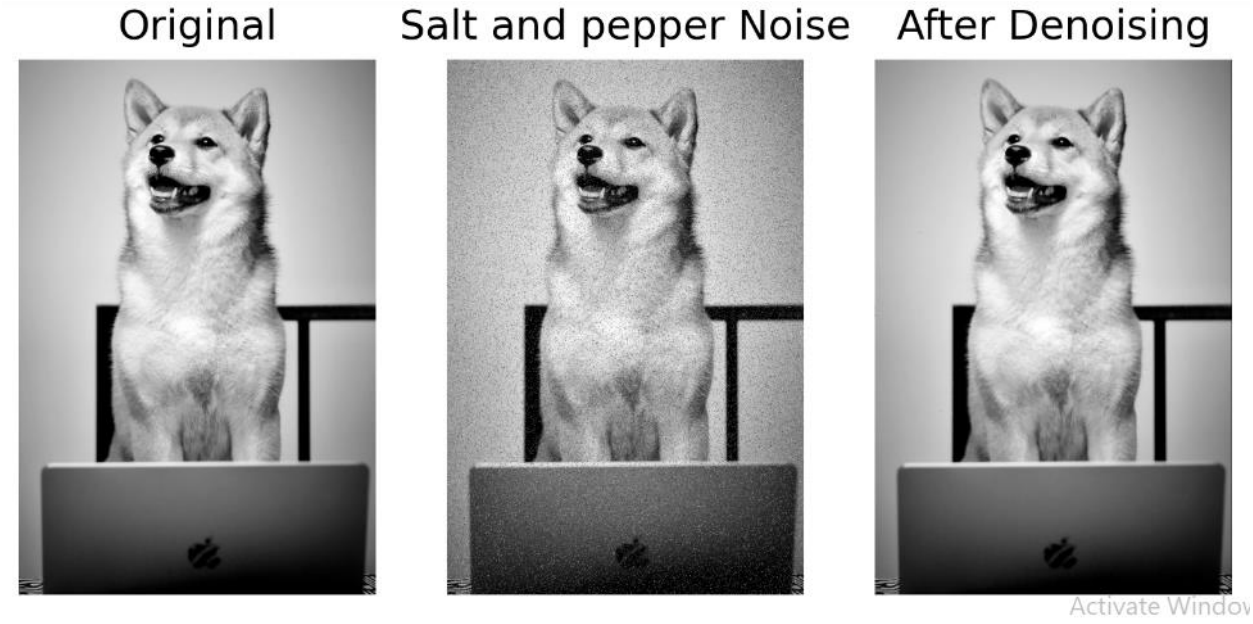
Salt and pepper noise with random pixels range from (300, 1000) De-noise with kennel size 3



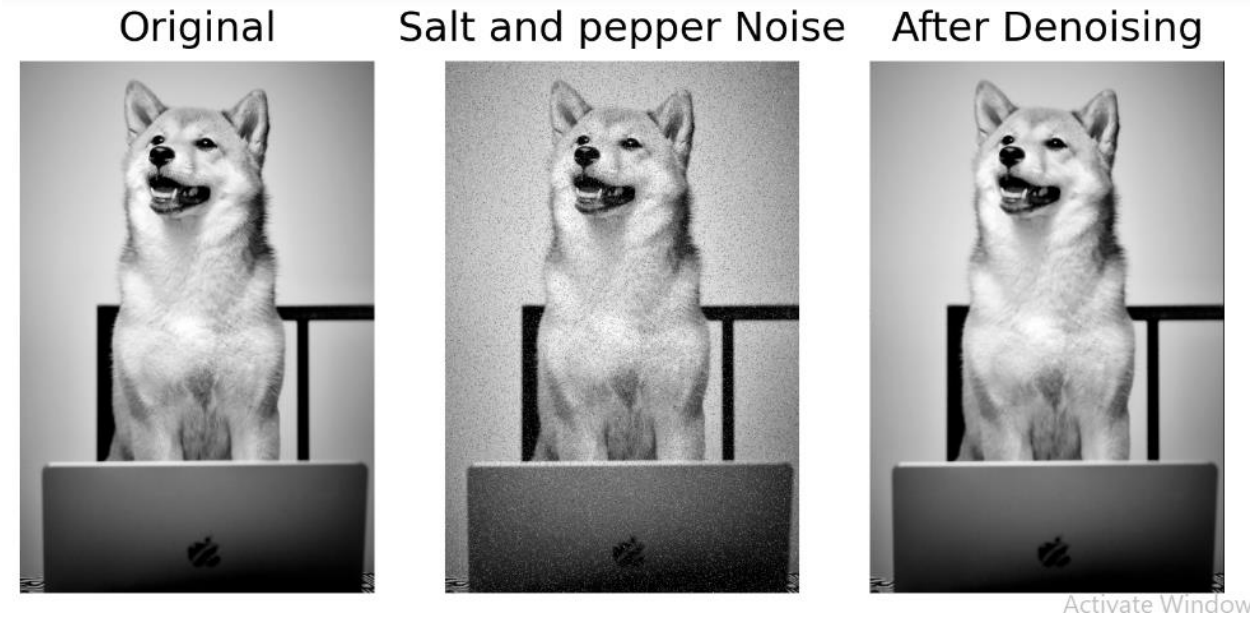
Salt and pepper noise with random pixels range from (300, 100000) De-noise with kennel size 3



Salt and pepper noise with random pixels range from (1000, 100000) De-noise with kennel size 3



Salt and pepper noise with random pixels range from (100, 100000) De-noise with kennel size 5



My Observations while applying noise with different random pixel range:

Different levels of salt-and-pepper noise (i.e., different probabilities of white and black pixels) effected the severity of noise in the image. A higher density of noisy pixels will result in more density in noise.

My Observations while applying de- noise with different kennel size:

The kernel size of the de-noising filter determines the neighborhood over which the filtering operation takes place.

A larger kernel size provided better noise reduction, However, very large kernels can also blur the image and lead to loss of fine details. Smaller kernel sizes are less effective in removing noise.

My Approach on Gaussian noise and de-noise:

Gaussian noise is a continuous noise that follows a Gaussian (normal) distribution. It manifests as a random variation in pixel values with a mean of zero and a certain standard deviation. The effect of Gaussian noise on an image is similar to other types of noise.it reduces the clarity and quality of the image. Gaussian noise will make an image appear grainy or speckled.

Gaussian noise is generally more suited for de-noising using linear filters like the mean filter.

function to add Gaussian noise:

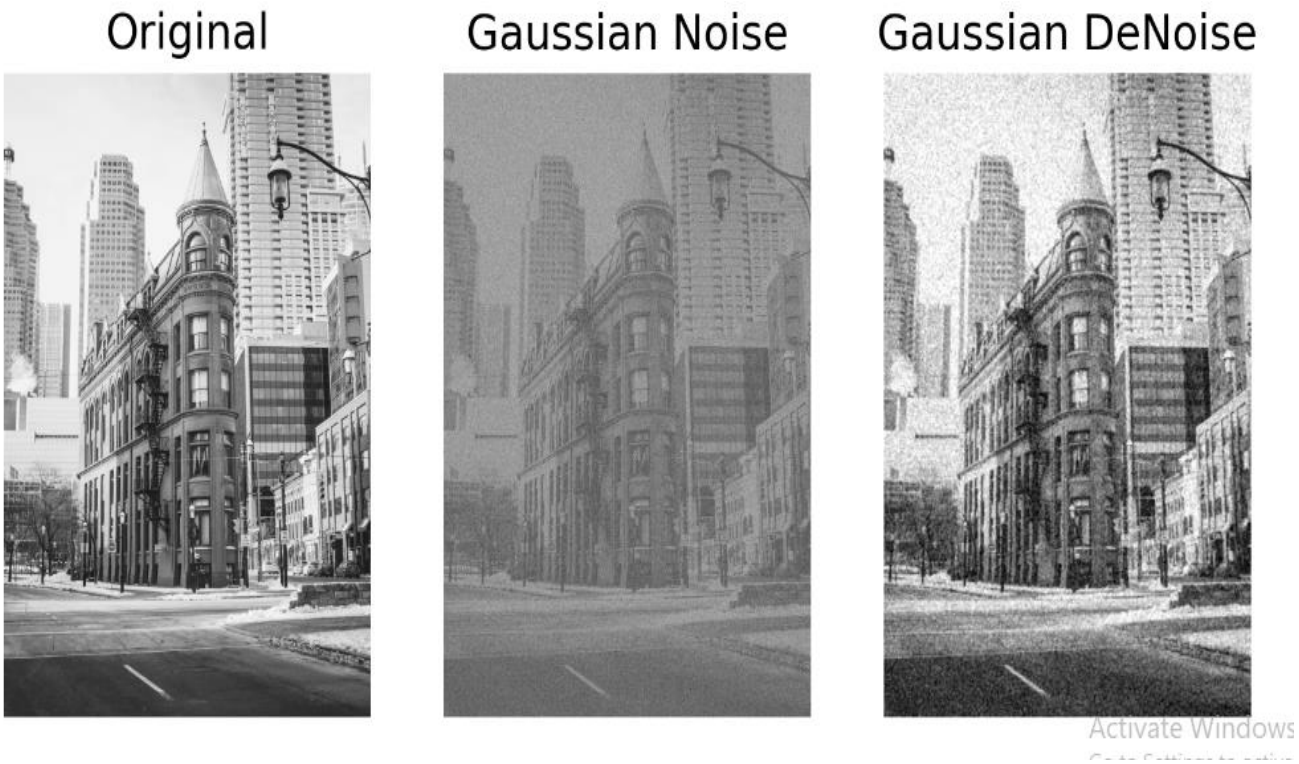
To achieve this implemented a function while works as follows gets the dimensions (rows and columns) of the input image using the shape attribute. The mean (average) of the Gaussian distribution to 0, since Gaussian noise is centered around its mean. The variance of the Gaussian distribution to 0.01. Variance controls the spread of the distribution, and a smaller value results in less intense noise. calculated the standard deviation (sigma) from the variance (var). The standard deviation is the square root of the variance and defines the width of the Gaussian distribution. Generated random samples from a normal distribution with

the specified mean (loc), standard deviation (scale), and size (row rows and col columns). This noise will be added to the image.

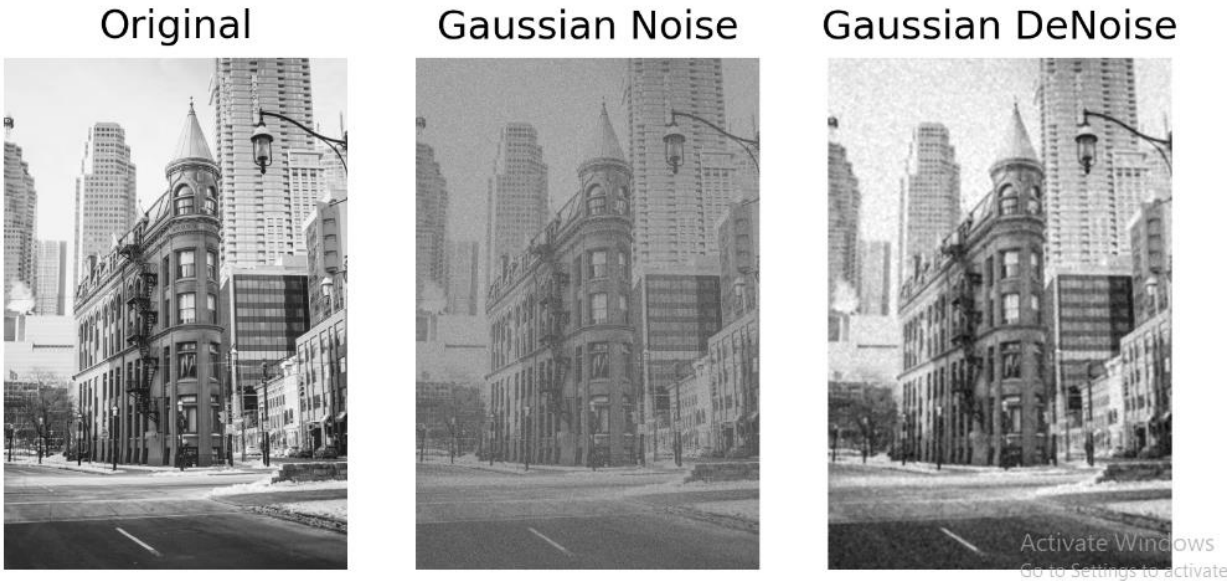
function to remove Gaussian noise:

Set the kernel size for the mean filter. Applied a mean filter using cv2.blurThe mean filter calculates the mean value within the kernel window and replaces the center pixel value with the calculated mean. If the pixel values in the noisy image might have exceeded the valid range of [0, 1]. The np.clip function is used to ensure that pixel values are within the valid range. Converted uint8 format using img_as_ubyte because cv2.medianBlur expects input in uint8 format. Applied median filter the median value within the kernel window and replaces the center pixel value with the calculated median.

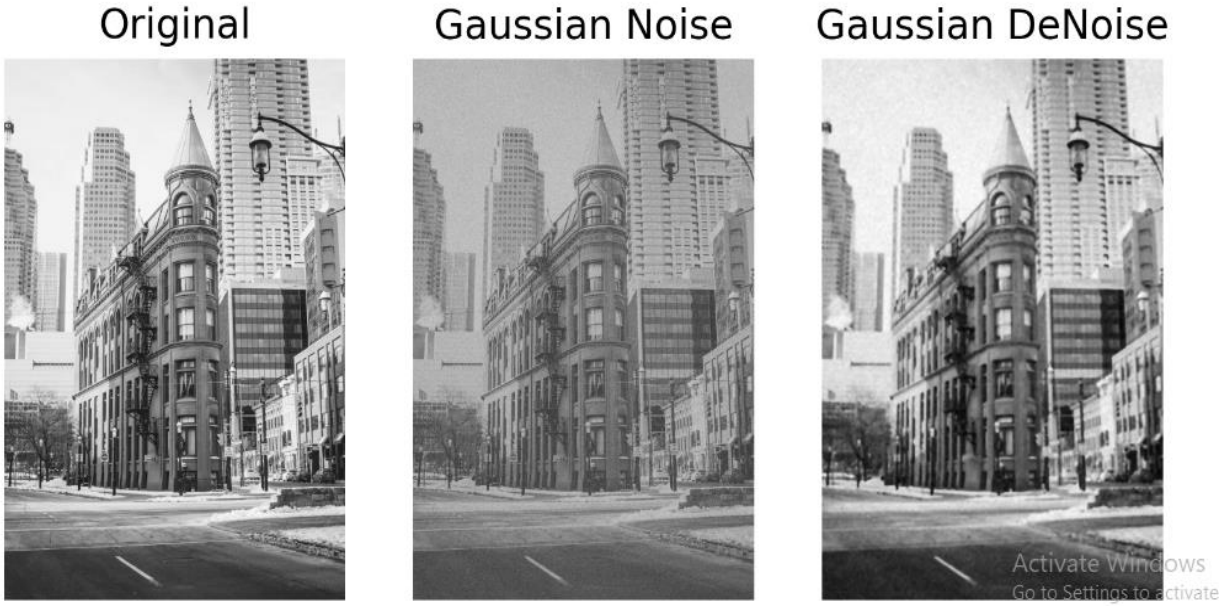
Gaussian noise with mean=0, variance = 0.08 De-noise with kennel size 3



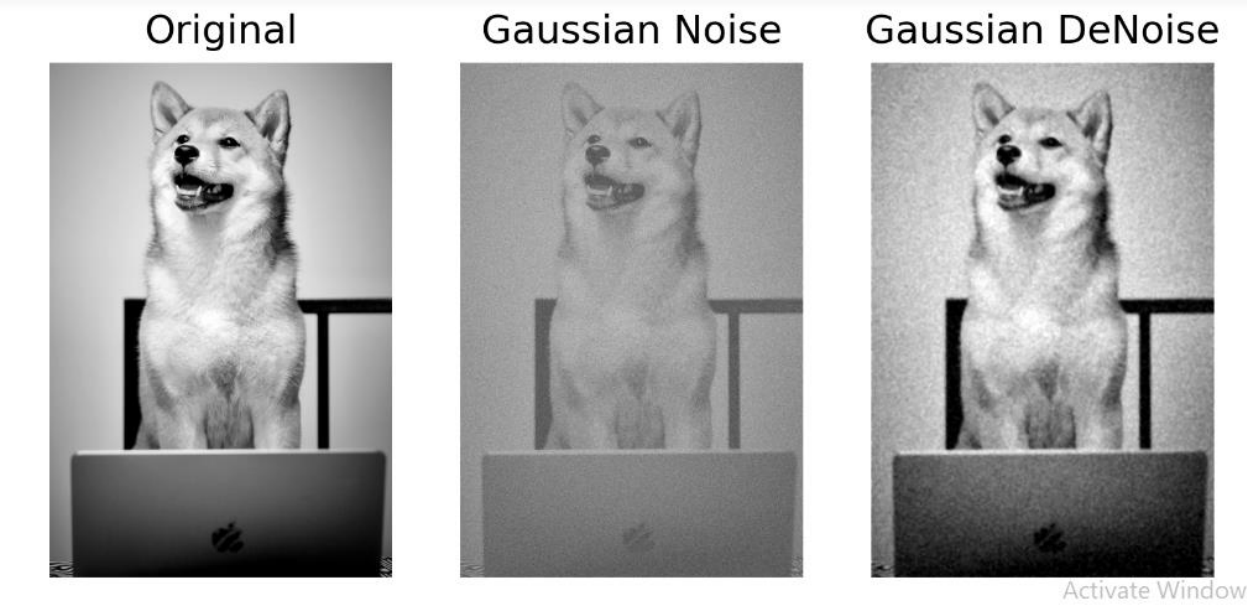
Gaussian noise with mean=0, variance = 0.05 De-noise with kennel size 9



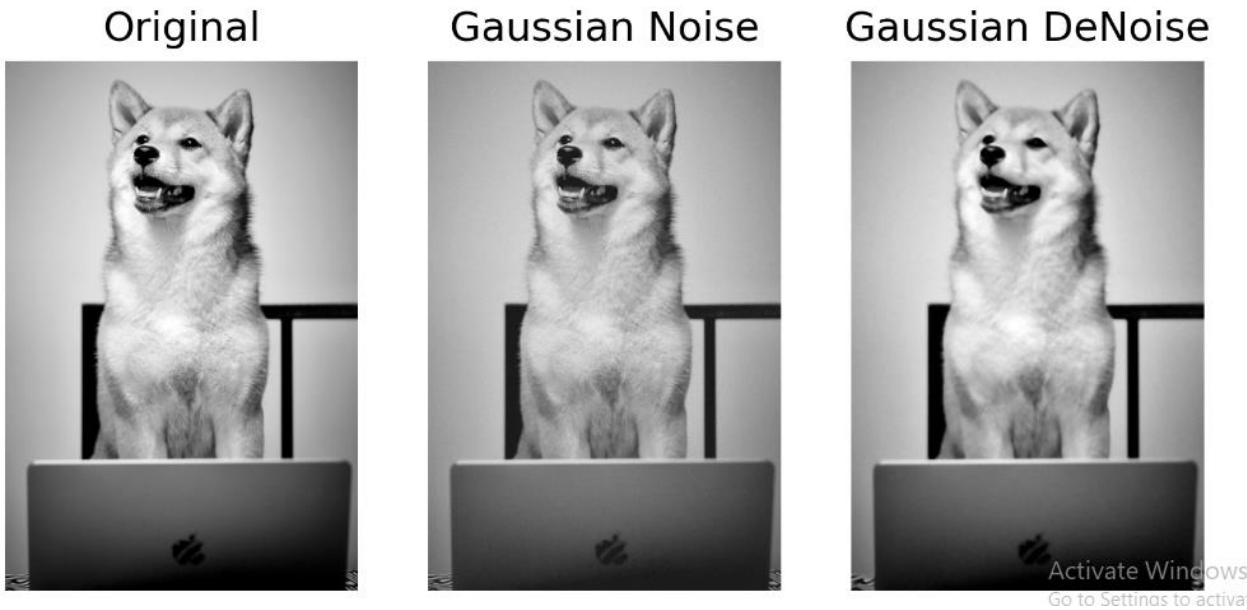
Gaussian noise with mean=0, variance = 0.08 De-noise with kennel size 5



Gaussian noise with mean=0, variance = 0.05 De-noise with kennel size 9



Gaussian noise with mean=0, variance = 0.001 De-noise with kennel size 5



My Observations while applying Gaussian noise

When dealing with Gaussian noise, its impact on the image varied based on the mean and variance of the noise distribution. Gaussian noise is centered around the mean and its intensity is controlled by the variance. Higher variance resulted in more intense noise. De-noising techniques, such as a median filter, can help reduce the impact of the noise on the image.

Gaussian Noise Parameters:

Mean = 0: The noise is centered around zero, which means it added an equal amount of positive and negative values to the pixel intensities. Variance = 0.05: controls the spread or intensity of the noise. A higher variance means more intense noise.

Impact of Gaussian Noise: A variance of 0.05 indicates moderate noise intensity. The noise is visible in the image, and its effect is noticeable, especially in regions with lower contrast. The noise is appeared as a subtle graininess across the image.

De-noising with Median Filter (Kernel Size = 9):

A median filter is effective at reducing impulse noise, such as salt-and-pepper noise and Gaussian noise. A kernel size of 9 means that for each pixel, the median value is calculated from a 9x9 neighborhood around that pixel. The median filter is particularly good at preserving edges and fine details while effectively removing noise spikes. The median filter with a kernel size of 9 will likely smooth out the noise, making the image appear less grainy.

Challenges faced while applying salt and pepper noise and de-noise:

Salt-and-pepper noise introduces very bright and very dark pixels into the image, which has an extreme impact on pixel values. These extreme values make it challenging to remove the noise without affecting the image's details.

Salt-and-pepper noise is not being uniformly distributed across the entire image. It could be more prominent in certain regions, making it necessary to use adaptive de-noising techniques that vary their behavior based on local noise characteristics.

Difference between Salt and pepper noise and Gaussian noisy images:

Salt-and-pepper noise appears as randomly occurring bright and dark pixels in an image, similar to grains of salt and pepper. These pixels have extreme values, typically maximum (white) and minimum (black).

Gaussian noise appears as random variations in pixel values across the image. It doesn't exhibit the extreme, bright-and-dark pixel values seen in salt-and-pepper noise.

Question 2

2.1 Adding and Removing Gaussian Blur.

2.2 Adding and Removing Motion Blur.

Adding and Removing Gaussian Blur:

Gaussian blur is a form of image degradation that smooths out an image by averaging the pixel values in a way that simulates the effect of an out-of-focus camera or a defocused lens. To add Gaussian blur to an image, I followed these steps:

Import Necessary Libraries: Typically, you would use image processing libraries like OpenCV.

Load the Image: Loaded the image to apply blur.

Apply Gaussian Blur: Used cv2.GaussianBlur in OpenCV to apply Gaussian blur to the image. specified the input image, the kernel size (which determines the amount of blur), and the standard deviation (which controls the spread of the blur).

Save or Display the Blurred Image

Adding and Removing Motion Blur:

Motion blur simulates the effect of an object in motion or camera movement while capturing an image. To add motion blur to an image, I followed these steps:

Load the Image. Created a Kernel: Defined a kernel which represent the direction and intensity of the motion blur. This kernel is typically a rectangular matrix with values that define the blur direction and length. Save or Display the Blurred Image

many advanced de-blurring techniques available, the effectiveness of de-blurring depends on factors like the blur severity, noise, and the accuracy of the estimated blur kernel. In practice, perfect removal of motion blur might not always be achievable.

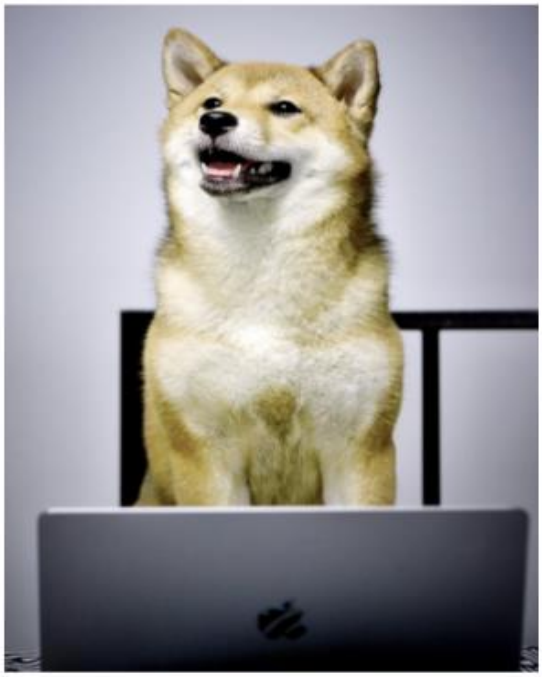
Gaussian Blur on image1 with kernel_size = (5, 5) sigma = 0

Original

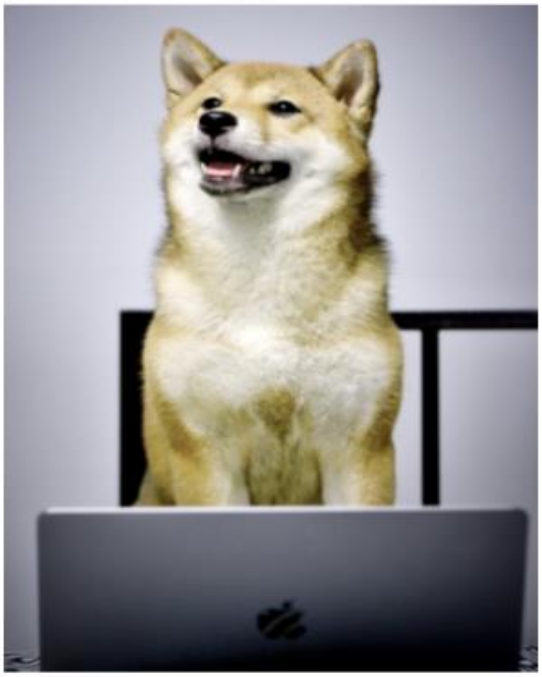
Gaussian Blur

Gaussian Blur on image2 with kernel_size = (5, 5) sigma = 0

Original

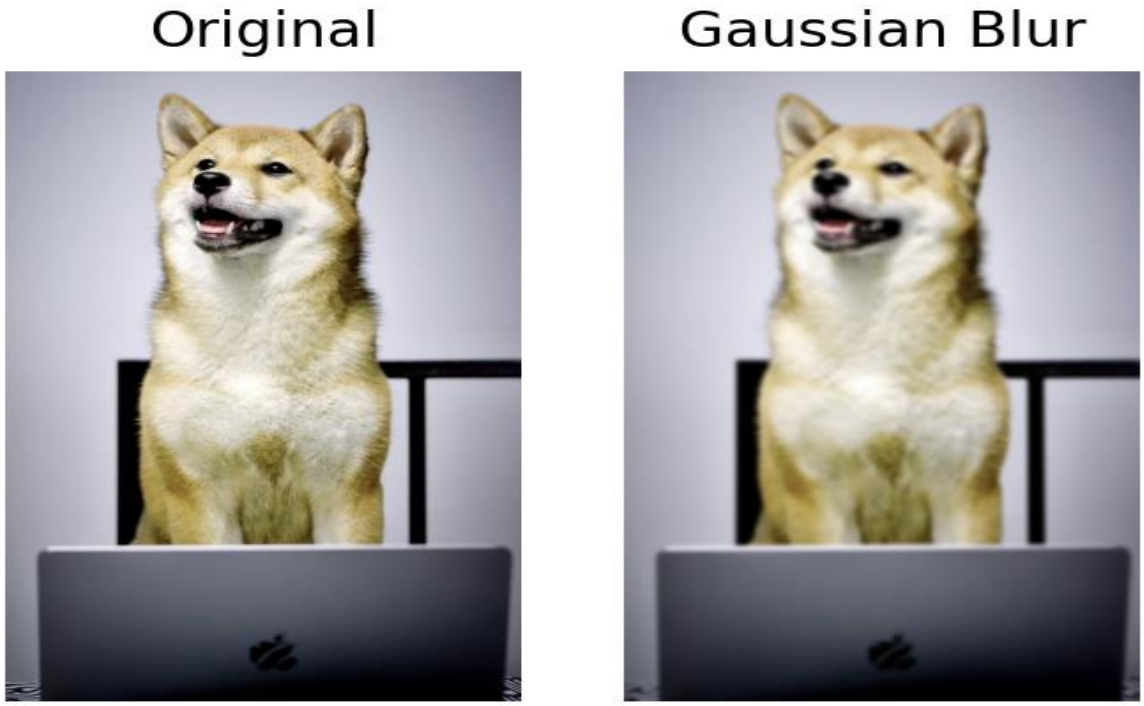


Gaussian Blur

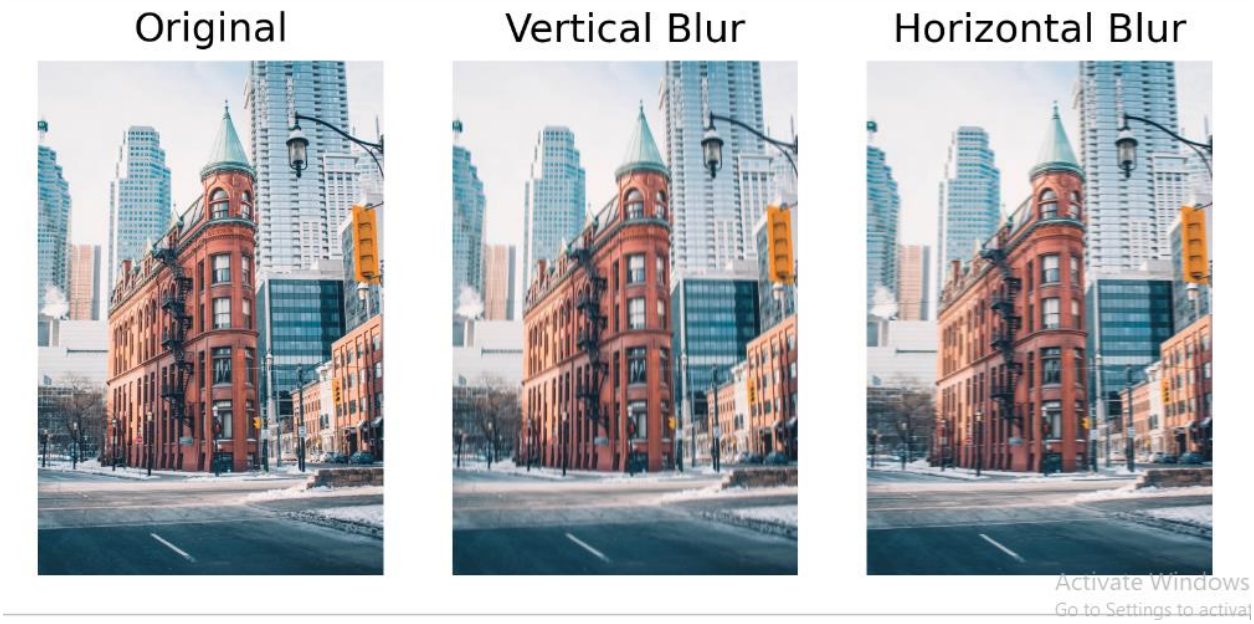


Gaussian Blur on image1 with kernel_size = (15, 15) ,sigma =5.5

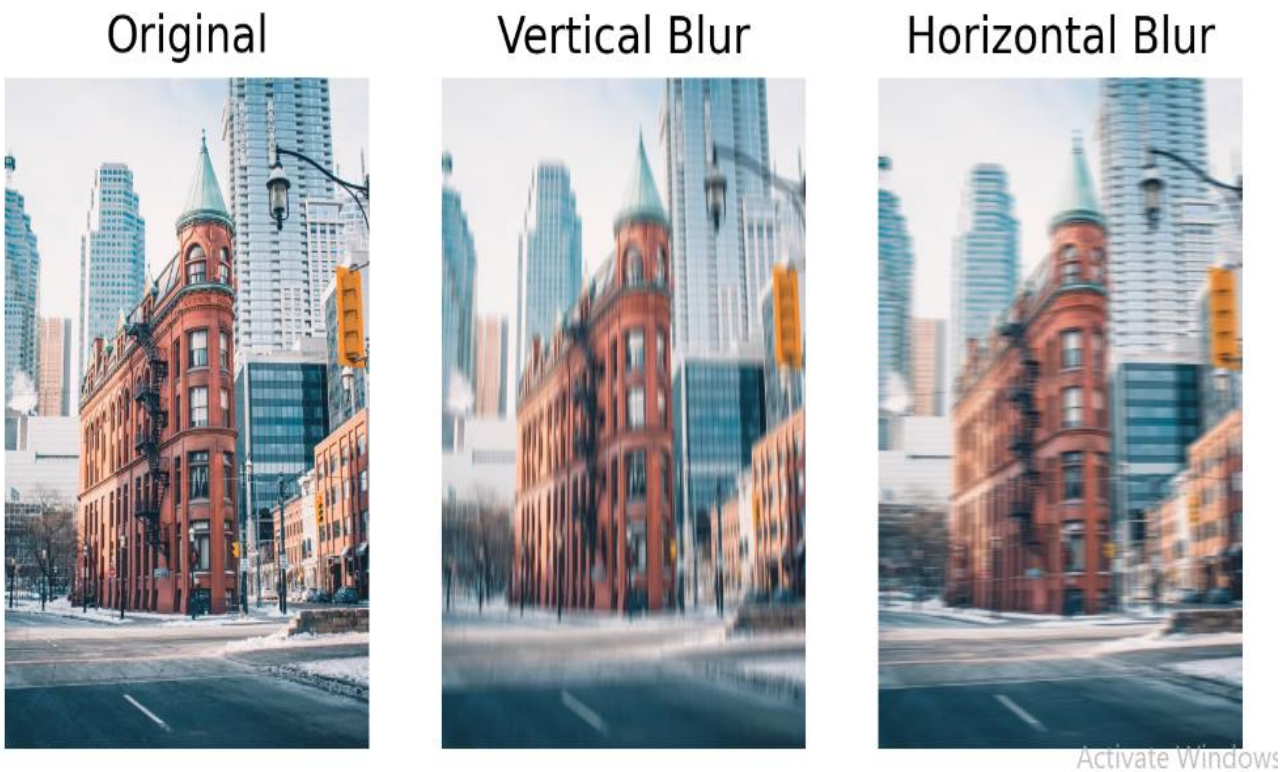
Gaussian Blur on image2 with kernel_size = (15, 15) ,sigma =5.5



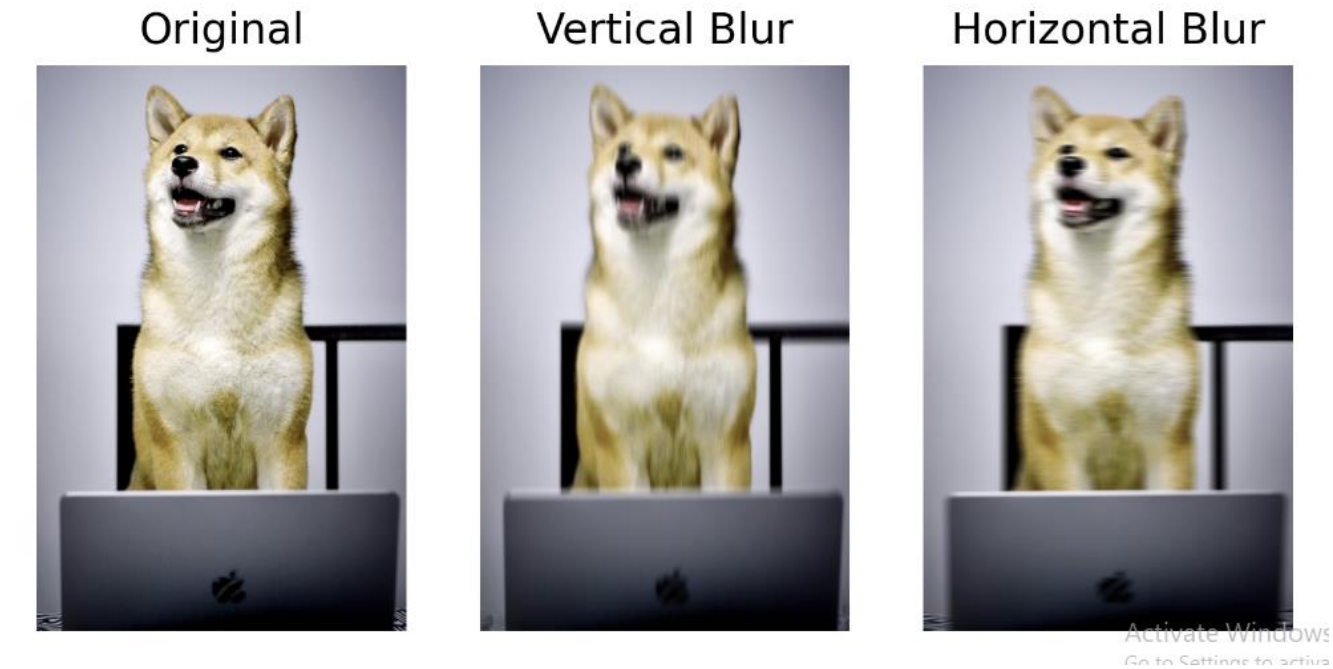
Motion Blur on image1 with kernel_size = 10



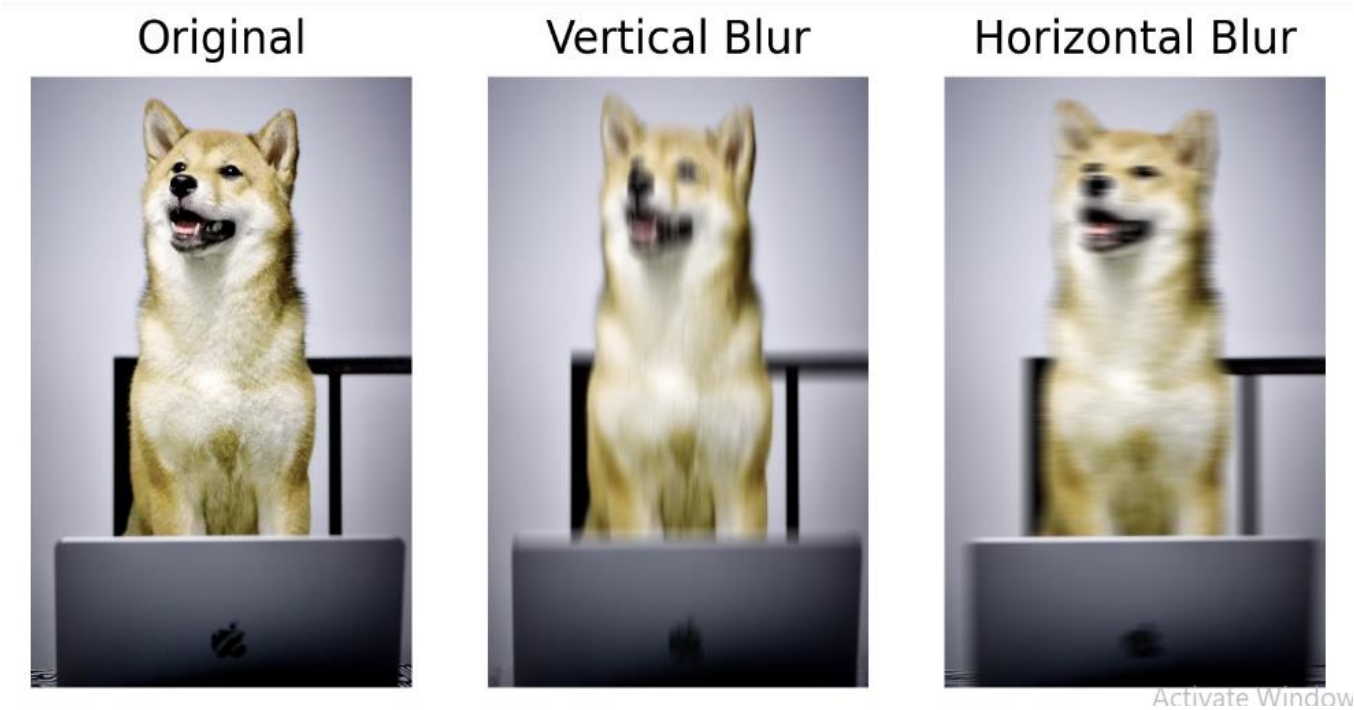
Motion Blur on image1 with kernel_size = 30



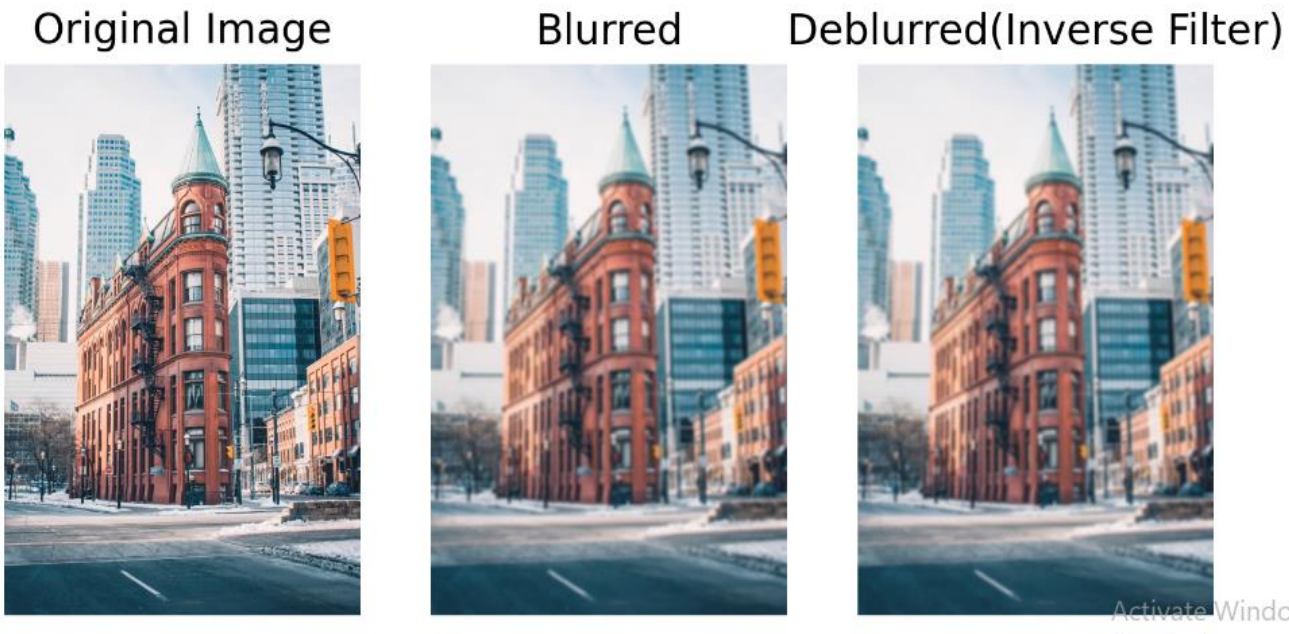
Motion Blur on image2 with kernel_size = 30



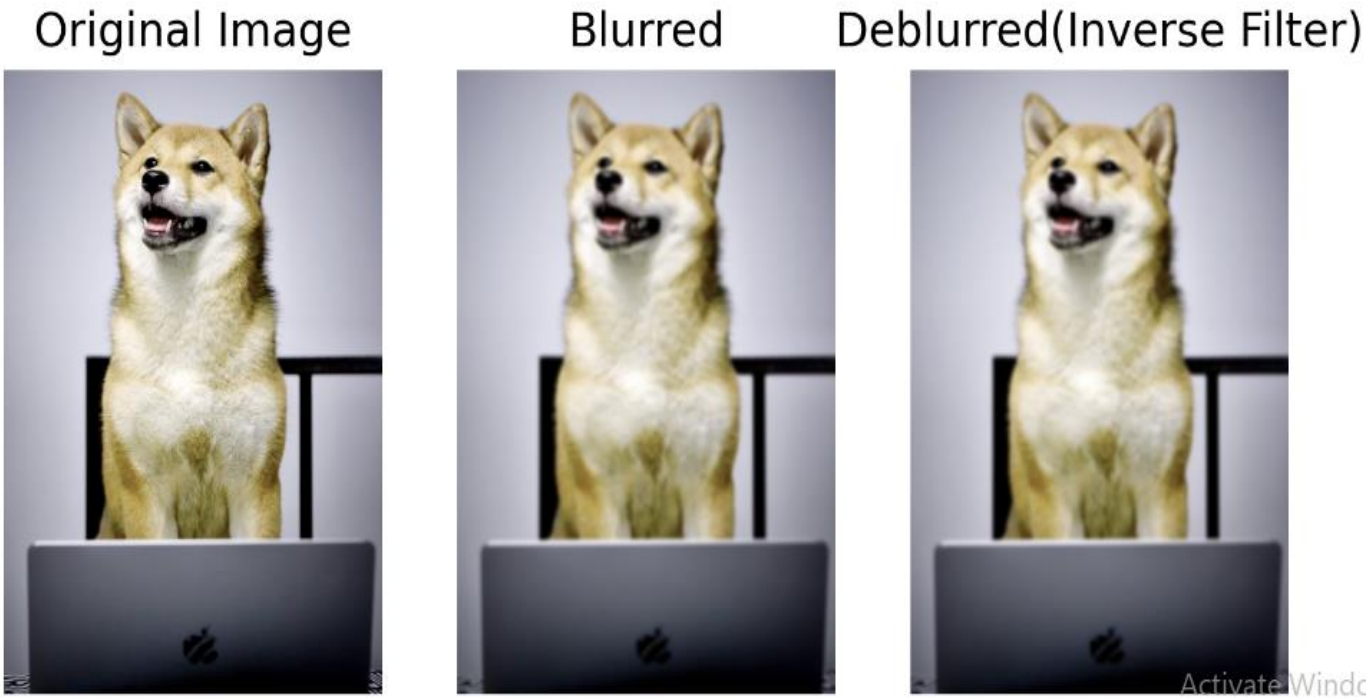
Motion Blur on image2 with kernel_size = 50



Gaussian Blur and deblur on image1

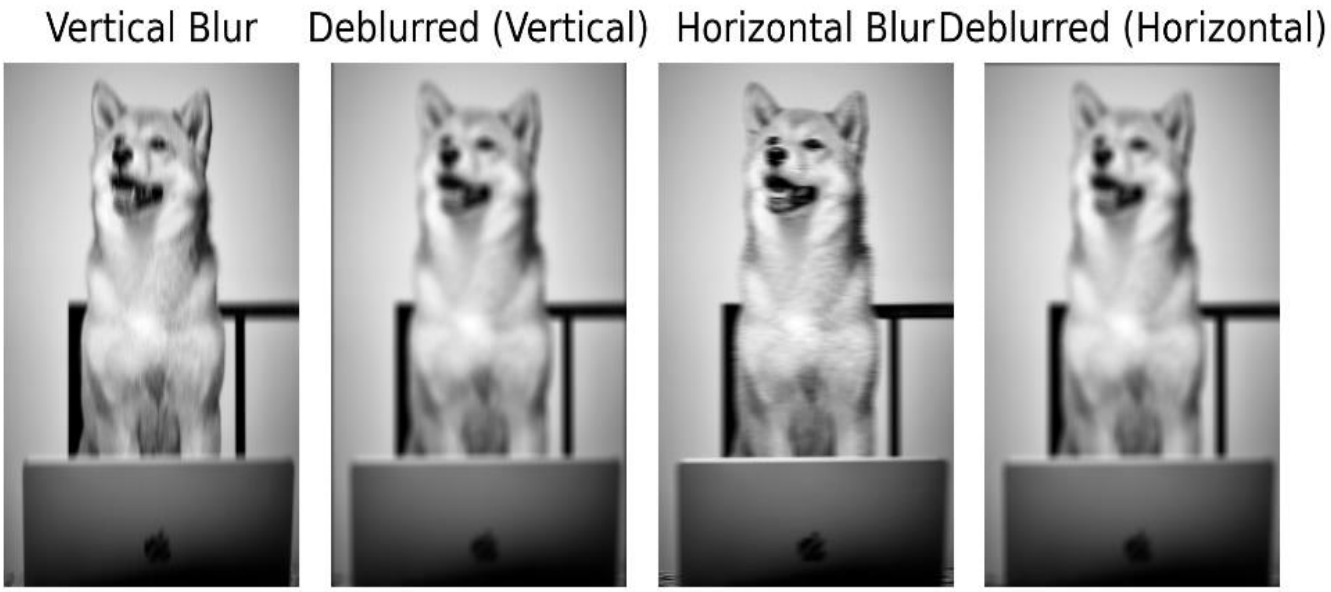


Gaussian Blur and deblur on image2



Motion Blur and deblur on image1

Motion Blur and deblur on image2



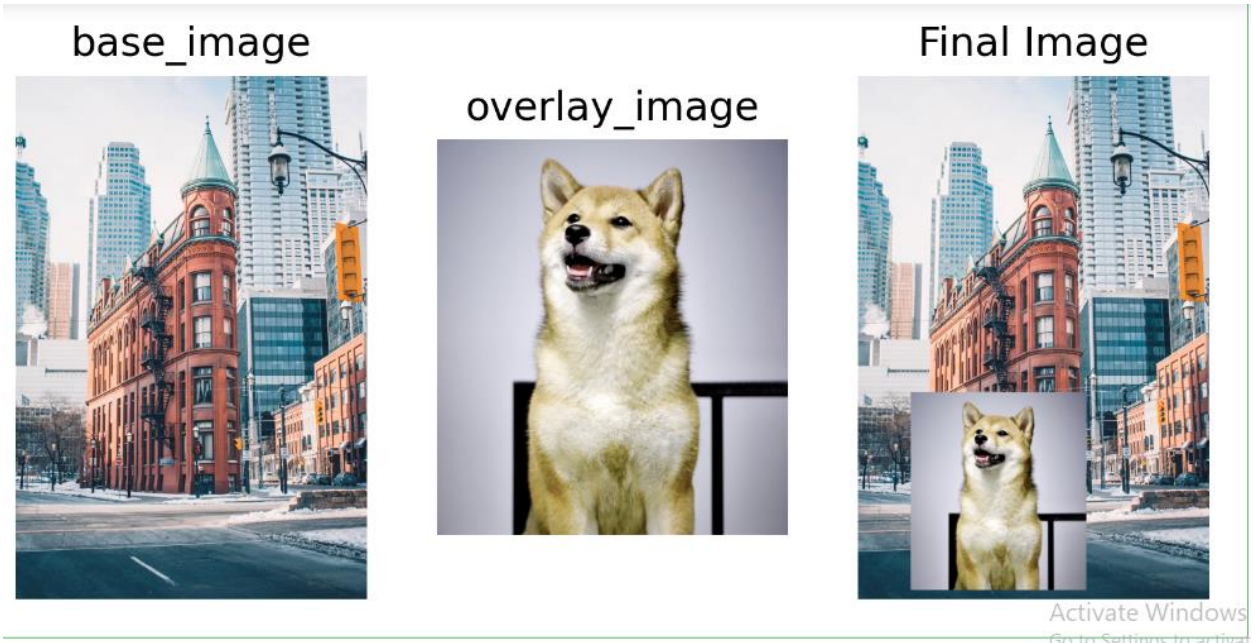
Question 3

Blending Two Images using OpenCV.

placing the image of a dog onto a road in a city image

To place an image of a dog onto a road in a city image, I have followed the following steps, those includes scaling, geometric transformations, and compositing.

- loaded the base image (image1.jpg) and the overlay image (croped_image2.jpg) using OpenCV.
- defined the position (position_x and position_y) where I want to place the overlay image on the base image. I have set to the bottom-center of the base image.
- I have resized the dog image (resized_overlay) to a smaller size (50% of its original size) to fit it onto the base image.
- displayed the base image, overlay image, and the final resulting image side by side using Matplotlib.



References:

- OpenCV Documentation
  https://docs.opencv.org/4.x/d5/d98/tutorial_mat_operations.html
- Basics of noise in image processing
  https://medium.com/@anishaswain/noise-in-digital-image-processing-55357c9fab71
- Add a “salt and pepper” noise to an image with Python
  https://www.geeksforgeeks.org/add-a-salt-and-pepper-noise-to-an-image-with-python/
- A Guide to Learn OpenCV in Python
  https://www.mygreatlearning.com/blog/opencv-tutorial-in-python/
- Salt and pepper noise
  https://www.projectrhea.org/rhea/index.php/How_to_Create_Salt_and_Pepper_Noise_in_an_Image
- Remove Salt and Pepper noise with Median Filtering
  https://medium.com/analytics-vidhya/remove-salt-and-pepper-noise-with-median-filtering-b739614fe9db
- Python OpenCV Tutorial
  https://pythonexamples.org/python-opencv/
- How to Add a Gaussian Noise to Image in Python
  https://www.youtube.com/watch?v=-Vk23ye2o_I
- OPENCV-PYTHON | Image Sharpening, Noise Reduction, Blur | Gaussian, Median, Bilateral FILTERING
  https://www.youtube.com/watch?v=SEj1imXHjqM
- Motion Blur in Image
  https://www.youtube.com/watch?v=8Oi7bfE3Cvg
- Blending images - OpenCV 3.4 with python 3 Tutorial 6
  https://www.youtube.com/watch?v=OwtqwpHJ4Bk
- OpenCV Python Image Blending
  https://www.youtube.com/watch?v=5jONfLRz4uk

Thank You