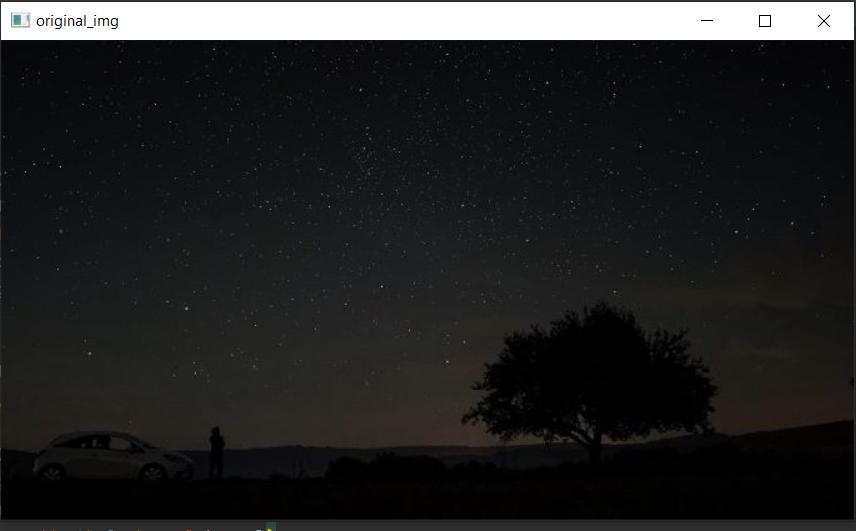
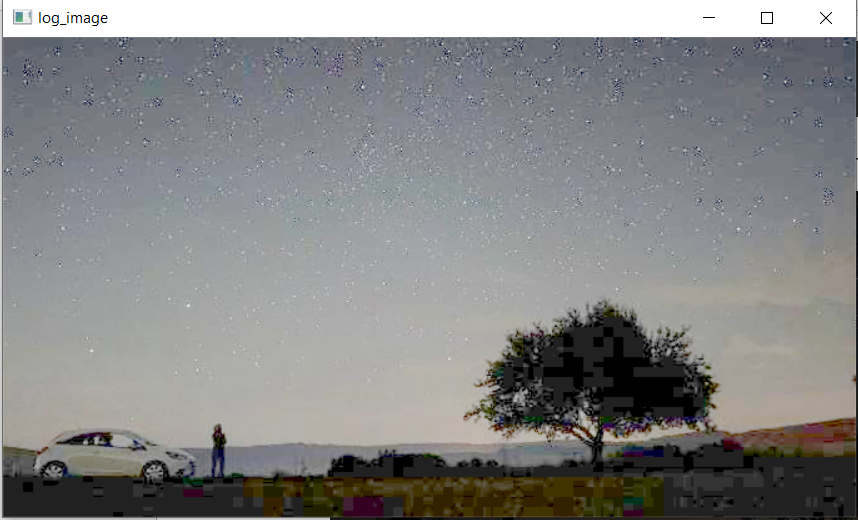
**Hàm logarith, hàm mũ**

import cv2  
import imutils  
import numpy as np  
  
# logarithmic, ham mu  
img = cv2.imread('logimg.jpg')  
img\_log = (np.log(img+1)/(np.log(1+np.max(img))))\*255  
img\_log = np.array(img\_log, dtype=np.uint8)  
cv2.imshow('log\_image', img\_log)  
cv2.imshow('original\_img', img)



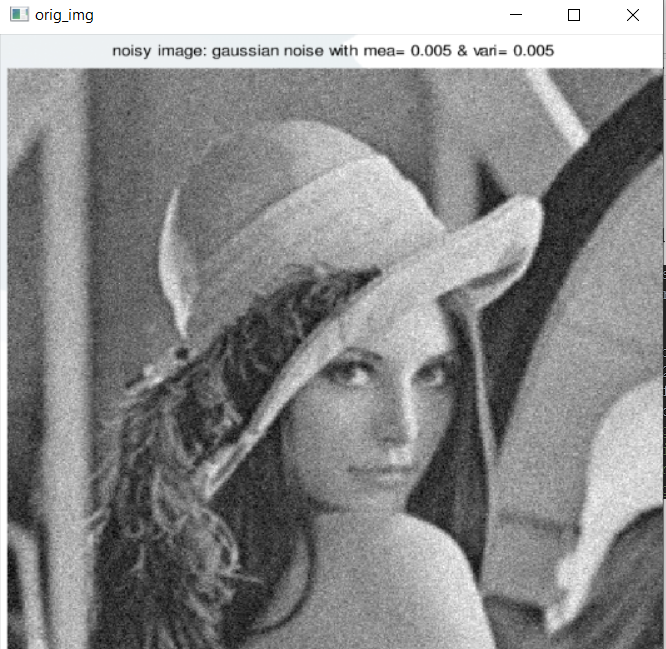
*Ảnh gốc*



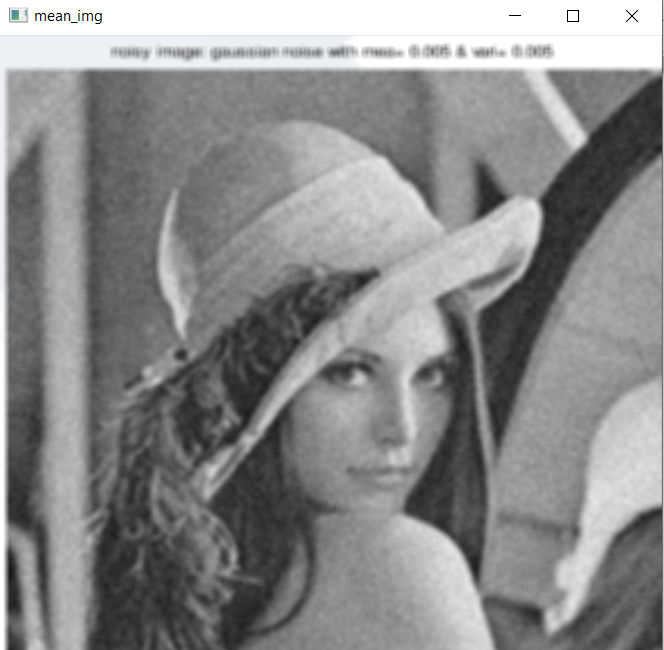
*Ảnh sau khi logarithmic transformation*

**Mean filter, median filter, weighted filter**

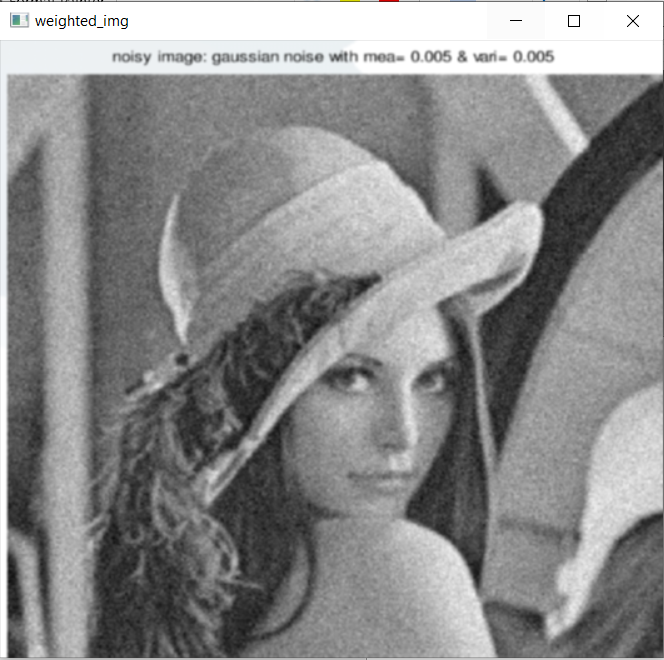
kernel = np.ones((5, 5), np.float32)/25  
kernel1 = np.array([[1, 2, 1],  
 [2, 4, 2],  
 [1, 2, 1]])/16  
img = cv2.imread('noiseimg.png')  
img\_median = cv2.medianBlur(img, 9)  
img\_mean = cv2.filter2D(src=img, ddepth=0, kernel=kernel)  
img\_weighted = cv2.filter2D(src=img, ddepth=0, kernel=kernel1)  
cv2.imshow('orig\_img', img)  
cv2.imshow('median\_image', img\_median)  
cv2.imshow('mean\_img', img\_mean)  
cv2.imshow('weighted\_img', img\_weighted)



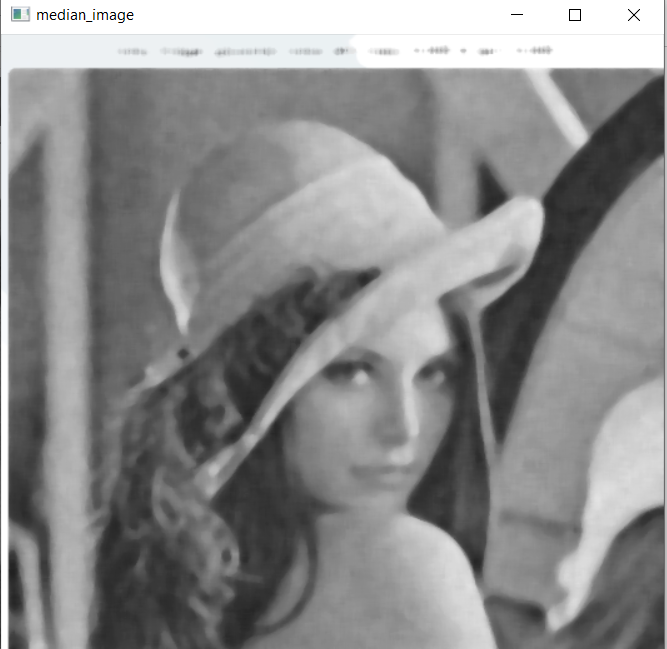
*Ảnh gốc*



*Ảnh lọc trung bình*



*Ảnh lọc trung bình có trọng số*



*Ảnh lọc trung vị*

**Sharpening, segmentation**

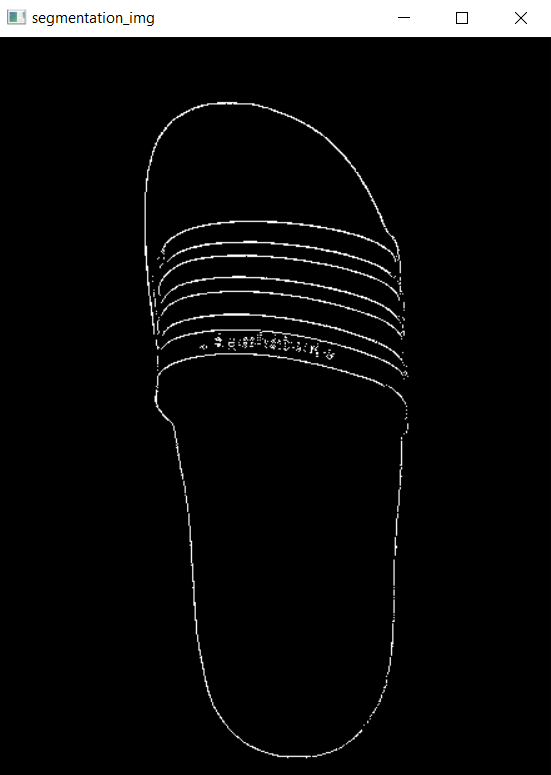
kernel2 = np.array([[0, 1, 0],  
 [1, -4, 1],  
 [0, 1, 0]])  
img = cv2.imread('dep1.jpg')  
img = imutils.resize(img, width=443, height=591)  
img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
cv2.imshow('origi\_img', img)  
img = cv2.medianBlur(img, 7)  
image\_sharp = cv2.filter2D(src=img, ddepth=0, kernel=kernel2)  
s, thresh\_sharp = cv2.threshold(image\_sharp, 20, 255, cv2.THRESH\_BINARY)  
cv2.imshow('sharp\_img', image\_sharp)  
cv2.imshow('segmentation\_img', thresh\_sharp)



*Ảnh gốc*



*Ảnh sau khi sharpening*



*Ảnh sau khi thersholding từ ảnh sharpening*