PCV Practical-6

190030059
A.R.Snehita

Prelab

1. What is Image denoising?

A. Image denoising is to remove noise from a noisy image, so as to restore the true image. However, since noise, edge, and texture are high frequency components, it is difficult to distinguish them in the process of denoising and the denoised images could inevitably lose some details. Overall, recovering meaningful information from noisy images in the process of noise removal to obtain high quality images is an important problem nowadays

2. Mention some image denoising techniques ?

A. There are three basic approaches to image denoising – Spatial Filtering, Transform Domain Filtering and Wavelet Thresholding Method. Objectives of any filtering approach are:

To suppress the noise effectively in uniform regions.
To preserve edges and other similar image characteristics.
To providea visually natural appearance.

3. For Grayscale images which image denoising method is used?

A. fastNlMeansDenoising()

Perform image denoising using Non-local Means Denoising algorithm with several computational optimizations. Noise expected to be a gaussian white noise. This function expected to be applied to grayscale images.

4. For color image which image denoising method is used?

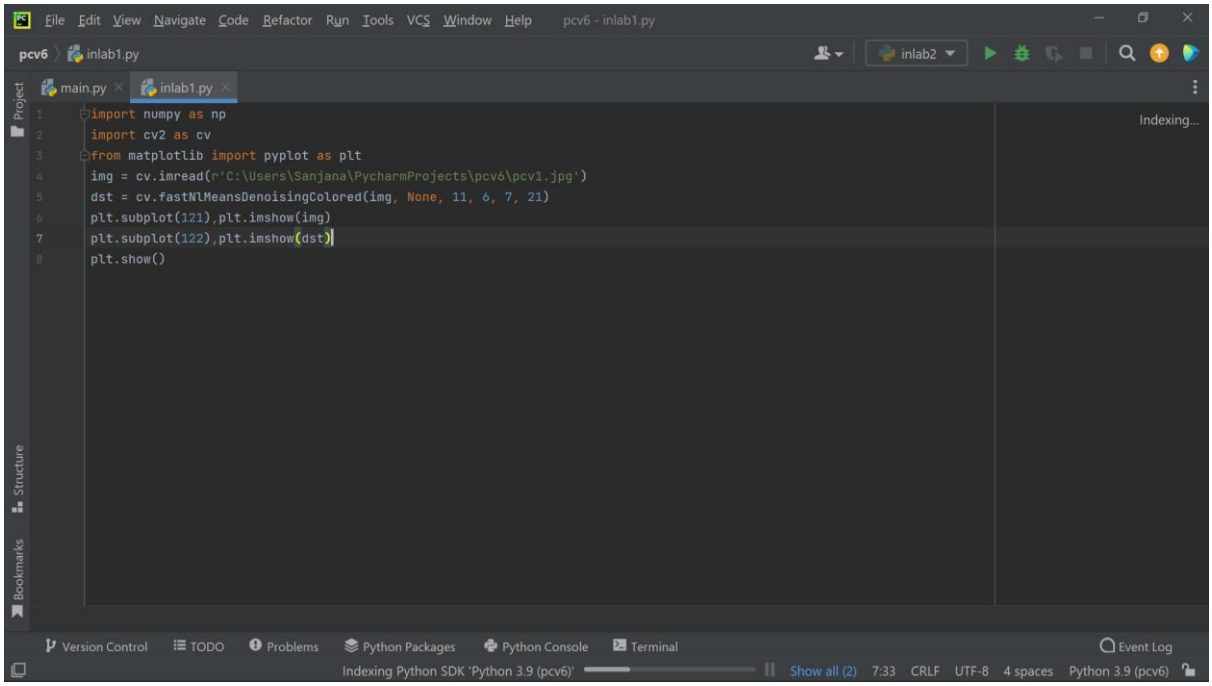
A. fastNlMeansDenoisingColored()

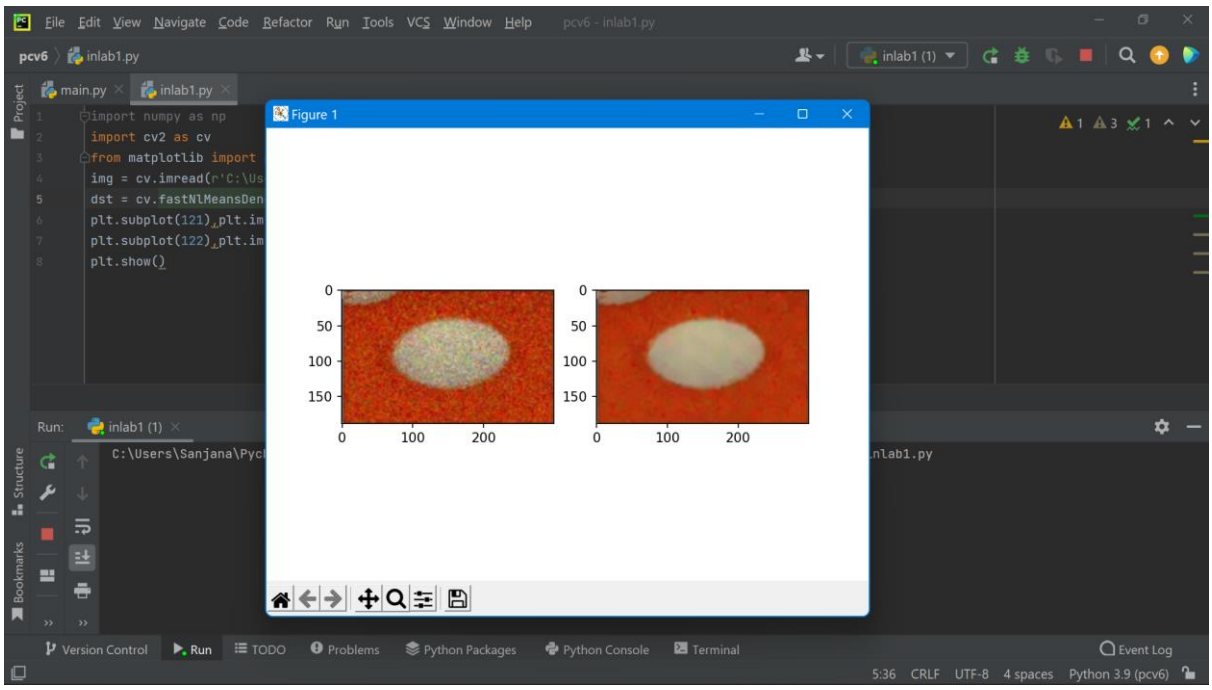
Modification of fastNlMeansDenoising function for colored images. The function converts image to CIELAB colorspace and then separately denoise L and AB components with given h parameters using fastNlMeansDenoising function.

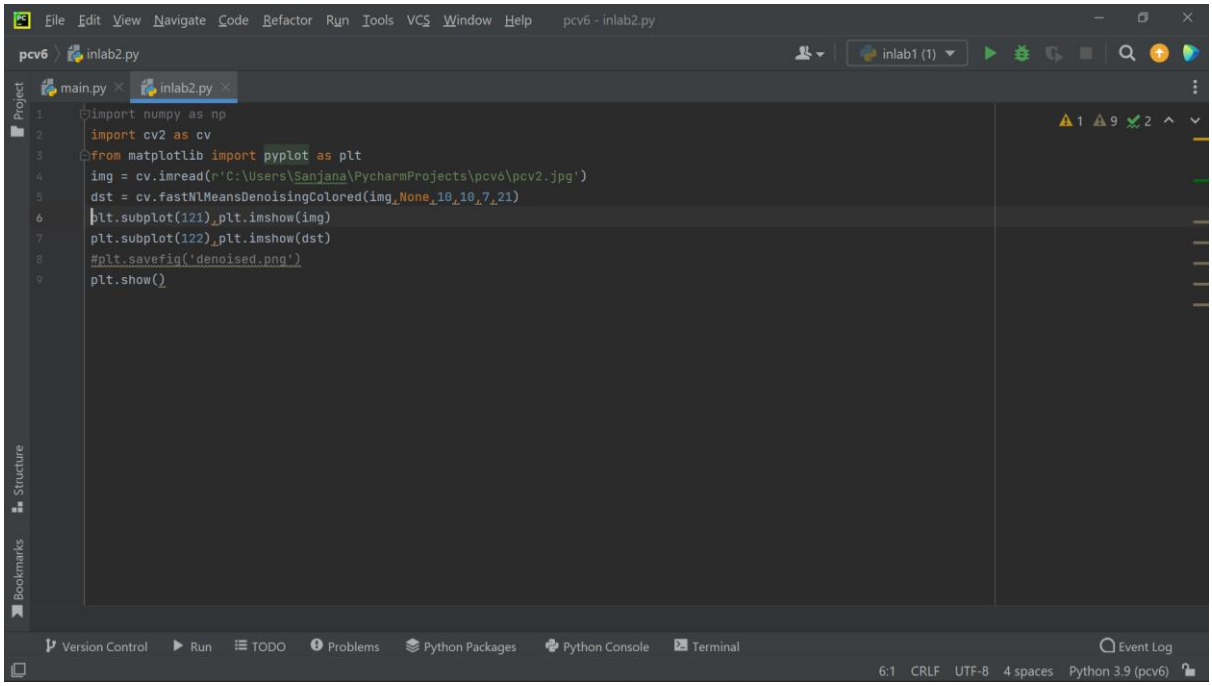
Inlab

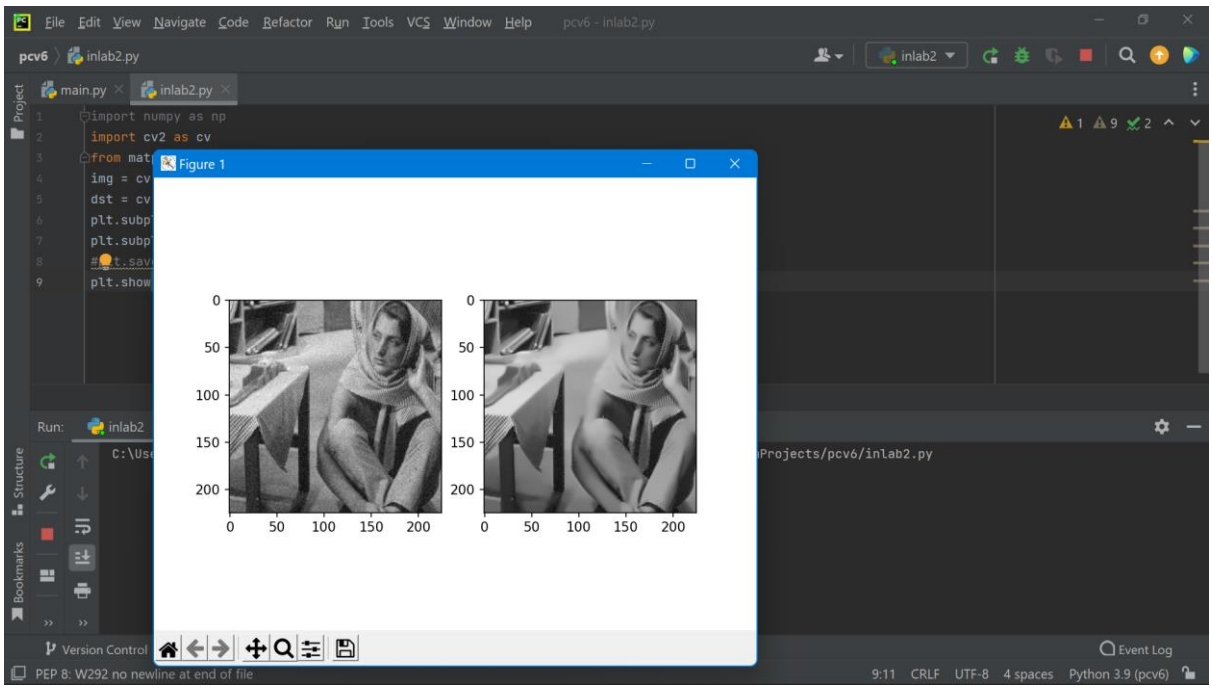
1. Harsha took pictures on his trip to mumbai in which few pictures were blurred, he wants to unblur the pictures using image denoising ,Guys can you help harsha in implementing image denoising.
   Harsha has four types of pictures.they are
   (i) Grayscale image
   (ii) Color image
   (iii)Grayscale image captured in a short period of time(use video capturing and take first 5
   frames)
   (iv) Color image captured in a short period of time(use video capturing and take first 5 frames)

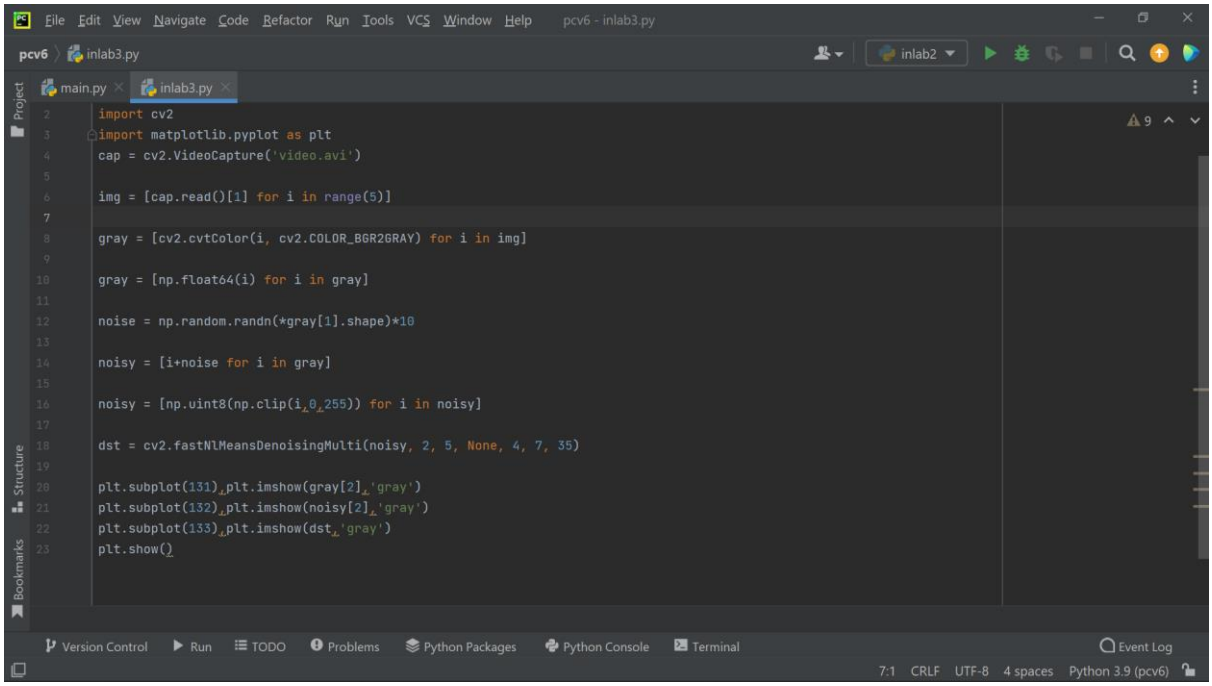
A.

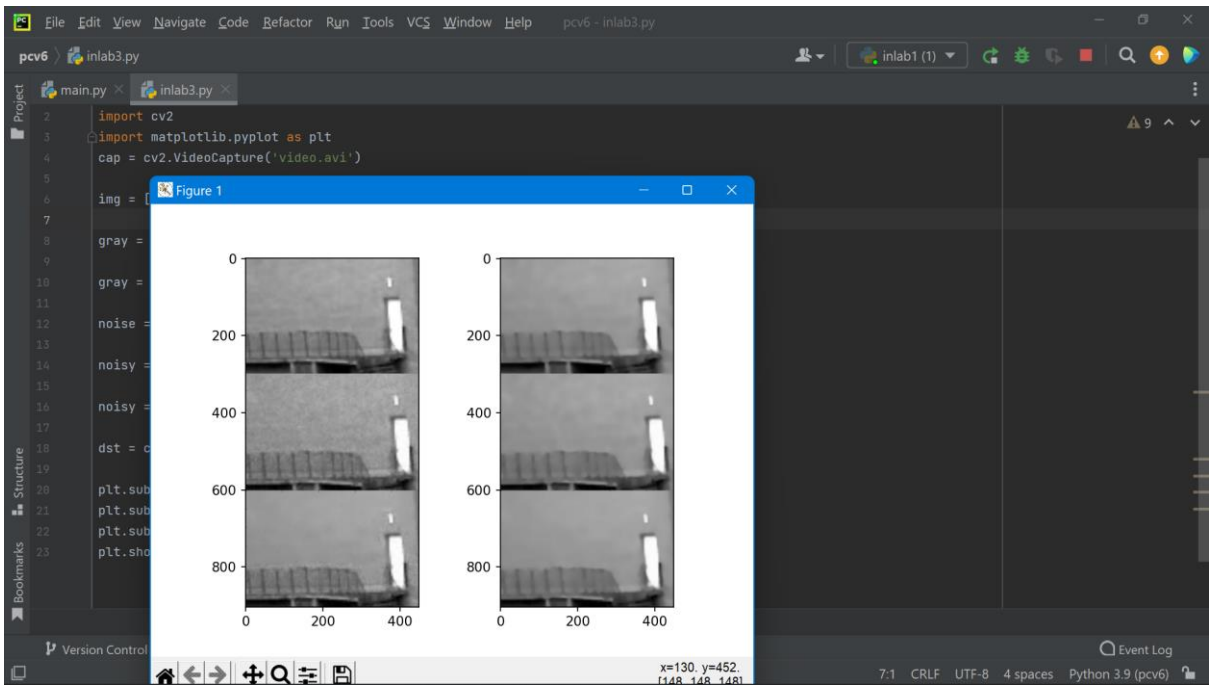

File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - inlab1.py
pcv6 > inlab1.py
inlab2
main.py
inlab1.py
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread(r'C:\Users\Sanjana\PycharmProjects\pcv6\pcv1.jpg')
dst = cv.fastNlMeansDenoisingColored(img, None, 11, 6, 7, 21)
plt.subplot(121),plt.imshow(img)
plt.subplot(122),plt.imshow(dst)
plt.show()
Indexing...
Version Control
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
Indexing Python SDK 'Python 3.9 (pcv6)'
Show all (2)
7:33
CRLF
UTF-8
4 spaces
Python 3.9 (pcv6)

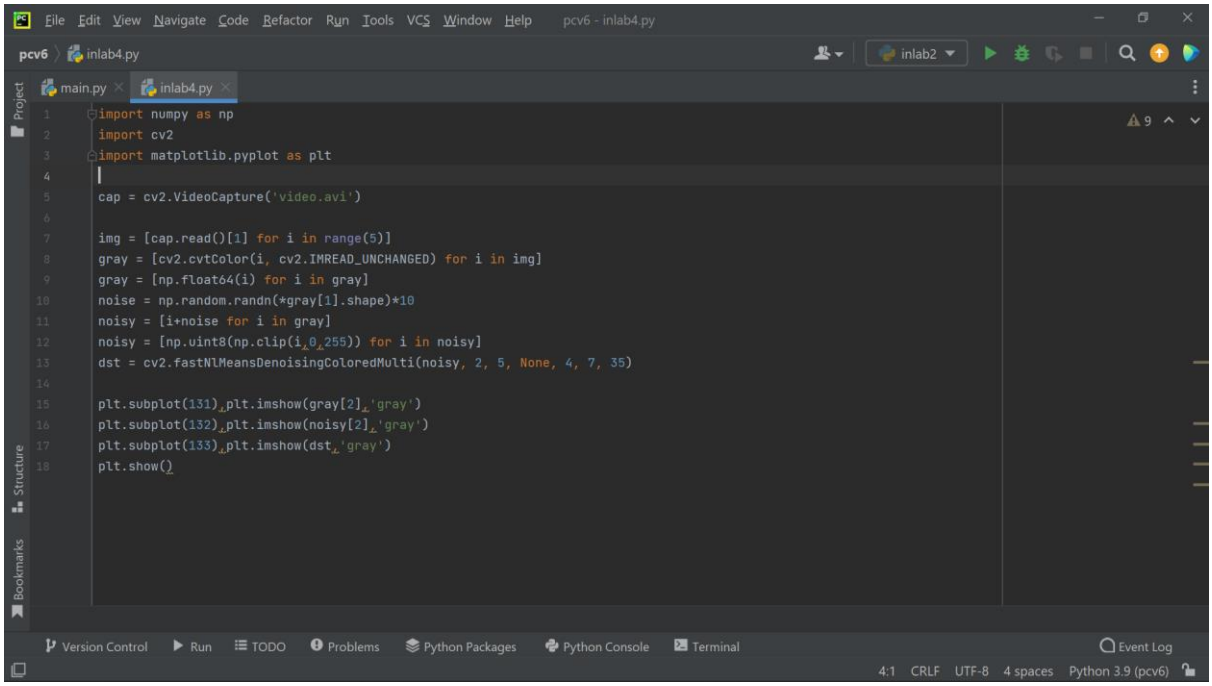

File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - inlab1.py
pcv6 > inlab1.py
inlab1 (1)
Figure 1
0
50
100
150
0
100
200
Run:
inlab1 (1)
Version Control
Run
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
5:36
CRLF
UTF-8
4 spaces
Python 3.9 (pcv6)

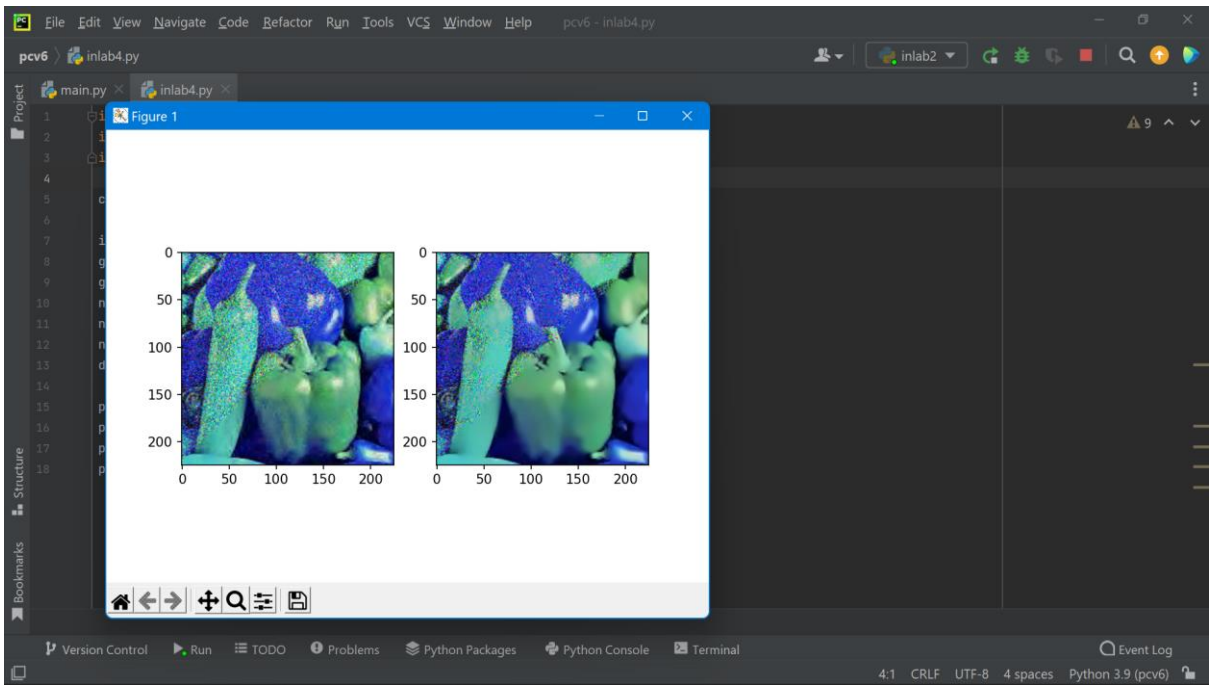

File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - inlab2.py
pcv6 > inlab2.py
inlab1 (1)
main.py
inlab2.py
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread(r'C:\Users\Sanjana\PycharmProjects\pcv6\pcv2.jpg')
dst = cv.fastNlMeansDenoisingColored(img,None,10,10,7,21)
plt.subplot(121),plt.imshow(img)
plt.subplot(122),plt.imshow(dst)
#plt.savefig('denoised.png')
plt.show()
Version Control
Run
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
6:1 CRLF UTF-8 4 spaces Python 3.9 (pcv6)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - inlab2.py
pcv6 > inlab2.py
inlab2
main.py
inlab2.py
Figure 1
0
50
100
150
200
Run:
inlab2
Projects/pcv6/inlab2.py
Version Control
Event Log
PEP 8: W292 no newline at end of file
9:11 CRLF UTF-8 4 spaces Python 3.9 (pcv6)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - inlab3.py
pcv6 > inlab3.py
inlab2
main.py
inlab3.py
import cv2
import matplotlib.pyplot as plt
cap = cv2.VideoCapture('video.avi')
img = [cap.read()[1] for i in range(5)]
gray = [cv2.cvtColor(i, cv2.COLOR_BGR2GRAY) for i in img]
gray = [np.float64(i) for i in gray]
noise = np.random.randn(*gray[1].shape)*10
noisy = [i+noise for i in gray]
noisy = [np.uint8(np.clip(i,0,255)) for i in noisy]
dst = cv2.fastNlMeansDenoisingMulti(noisy, 2, 5, None, 4, 7, 35)
plt.subplot(131),plt.imshow(gray[2],'gray')
plt.subplot(132),plt.imshow(noisy[2],'gray')
plt.subplot(133),plt.imshow(dst,'gray')
plt.show()
Version Control
Run
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
7:1 CRLF UTF-8 4 spaces Python 3.9 (pcv6)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - inlab3.py
pcv6 > inlab3.py
inlab1 (1)
main.py
inlab3.py
Figure 1
0
200
400
600
800
0
200
400
0
200
400
x=130, y=452.
Version Control
Event Log
7:1 CRLF UTF-8 4 spaces Python 3.9 (pcv6)





Postlab

1.State and explain the applications of denoising.

A. Data classification algorithms are often used in the engineering field, but the data measured in the actual engineering often contains different types and degrees of noise, such as vibration noise caused by water flow when measuring the natural frequencies of aqueducts or other hydraulic structures, which will affect the accuracy of

classification. In reality, these noises often appear disorganized and stochastic and some existing algorithms exhibit poor performance in the face of these non-Gaussian noise. Therefore, the classification algorithms with excellent performance are needed. To address this issue, a hybrid algorithm of robust principal component analysis (RPCA) combined multigroup random walk random forest (MRWRF) is proposed in this paper. On the one hand RPCA can effectively remove part of non-Gaussian noise, and on the other hand MRWRF can select a better number of decision trees (DTs), which can effectively improve random forest (RF) robustness and classification performance, and the combination of RPCA and MRWRF can effectively classify data with non-Gaussian distribution noise. Compared with other existing algorithms, this hybrid algorithm has strong robustness and preferable classification performance and can thus provide a new approach for data classification problems in engineering.

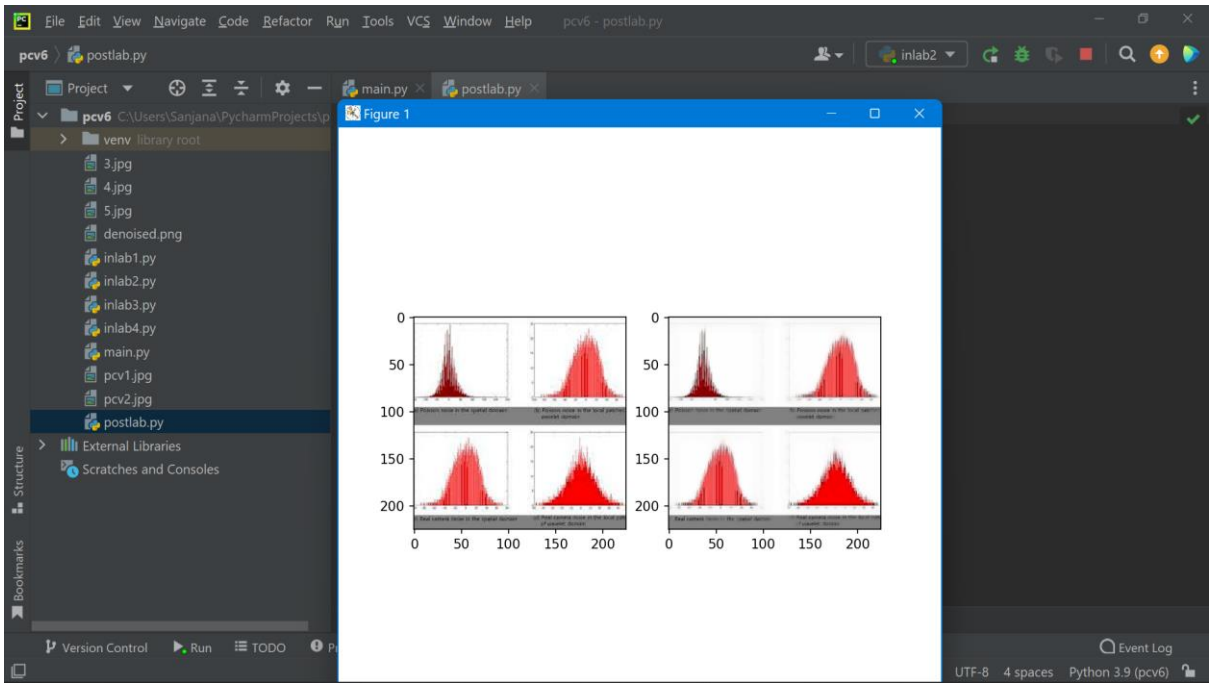
2.Compare gaussian noise to salt and pepper noise.

A. Salt and pepper noise definitely occurs in real life applications. Its presence indicates some hardware issues - problems with the camera sensors that make up the pixels, memory cell failure or errors in the digitization and transmission of data.

Random valued impulse noise, distributed as a Gaussian probability density function, occurs independent of the image intensity at the particular pixel. It is the standard model for many natural processes which introduce noise in images. The source of this noise can be thermal, or it can be electronic circuit noise caused during acquisition, amplification, or transmission, and introduced by capacitors during reset.

3.Plot and compare the histograms of the original noisy and the denoised images.

A.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pcv6 - postlab.py
pcv6 postlab.py
inlab2
Project
main.py
postlab.py
pcv6 C:\Users\Sanjana\PycharmProjects\p
venv library root
3.jpg
4.jpg
5.jpg
denoised.png
inlab1.py
inlab2.py
inlab3.py
inlab4.py
main.py
pcv1.jpg
pcv2.jpg
postlab.py
External Libraries
Scratches and Consoles
Figure 1
0
50
100
150
200
Version Control
Run
TODO
Event Log
UTF-8
4 spaces
Python 3.9 (pcv6)