

VAA_NC_Reportweek1_Thuong

Một số lệnh Linux cơ bản

https://ubuntu.com/tutorials/command-line-for-beginners?fbclid=IwAR2JurY8qGj6yedoYT6PpwkNX_zFfI4-NnX8Yi1HVRmhJfHS4CJtnsINhU#4-creating-folders-and-files

Các lệnh cơ bản :

- Pwd thư mục đang làm việc hiện tại
- Mkdir tạo thư mục mới
- Cd đường dẫn
- Cd / thư mục gốc
- Cd ~ trở về thư mục mặc định

Vd từ root trở về desktop cd ~/Desktop

- Cd .. trở về thư mục mẹ
- Liệt kê : ls

vd ls > output.txt xuất ra những cái có trên desktop đưa vào output.txt

- Rm -r xóa

Vd 123.txt trên desktop

- Cp copy

vd Cp 123.txt 123 (123 folder)

- +Mv di chuyển

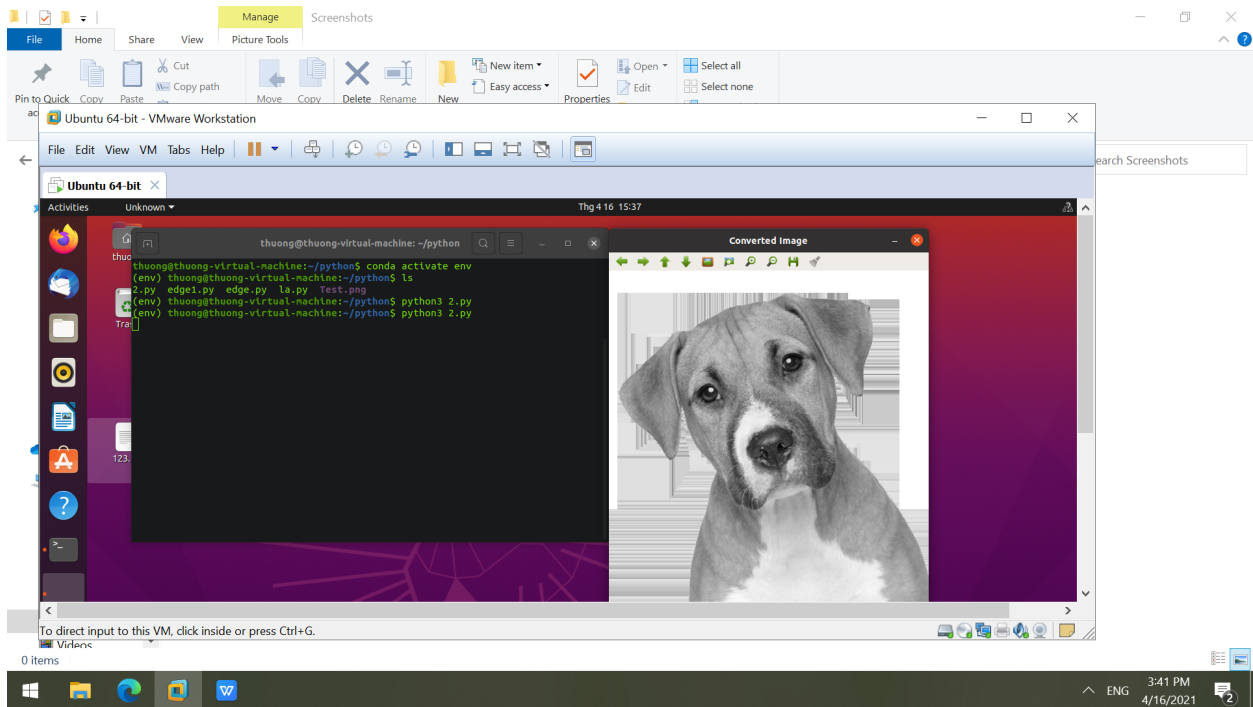
Vd mv 123.txt 123

Sour code chuyển ảnh thành ảnh xám bằng open cv

Link tham khảo:

<https://dev-akash.github.io/posts/how-to-convert-color-image-to-grayscale-in-opencv-with-python.html>

```
import cv2
# Reading color image
img = cv2.imread("Test.png")
# Converting color image to grayscale image
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Showing the converted image
cv2.imshow("Converted Image",gray)
# waiting for key event
cv2.waitKey(0)
# destroying all windows
cv2.destroyAllWindows()
Output
```



Source code sử dụng sobel và laplacian để phát hiện cạnh của ảnh

Link tham khảo:

<https://www.includehelp.com/python/edge-detection-of-image-using-opencv-cv2.aspx>

https://bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Gradient_Sobel_Laplacian_Derivatives_Edge_Dete

Sử dụng sobel

```

# importing the module
import cv2

# read the image and store the data in a variable
image=cv2.imread("Test.png")

# make it grayscale
Gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

# Make it with the help of sobel
# make the sobel_horizontal
# For horizontal x axis=1 and yaxis=0
# for vertical x axis=0 and y axis=1
Horizontal=cv2.Sobel(Gray,0,1,0,cv2.CV_64F)

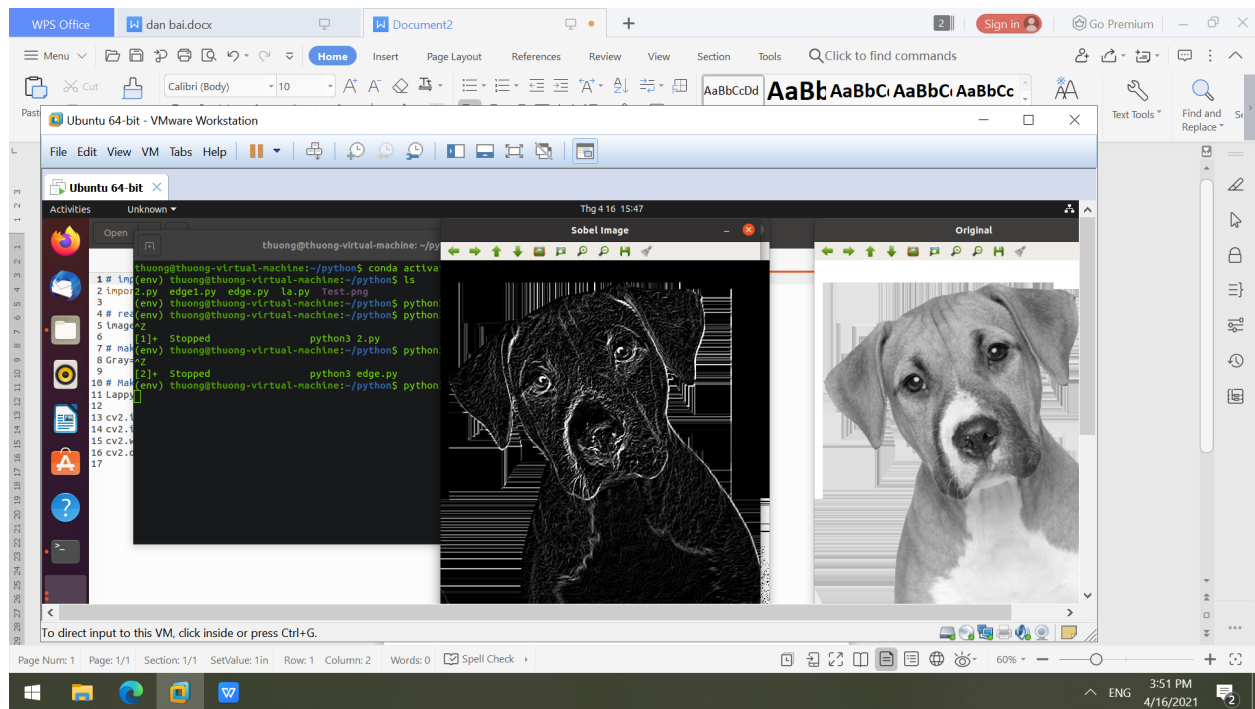
# the thresholds are like
# (variable,0,<x axis>,<y axis>,cv2.CV_64F)
Vertical=cv2.Sobel(Gray,0,0,1,cv2.CV_64F)

# DO the Bitwise operation
Bitwise_Or=cv2.bitwise_or(Horizontal,Vertical)

# Show the Edged Image
cv2.imshow("Sobel Image",Bitwise_Or)
cv2.imshow("Original Image",Gray)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Output



Sử dụng laplacian

```
# importing the module
import cv2

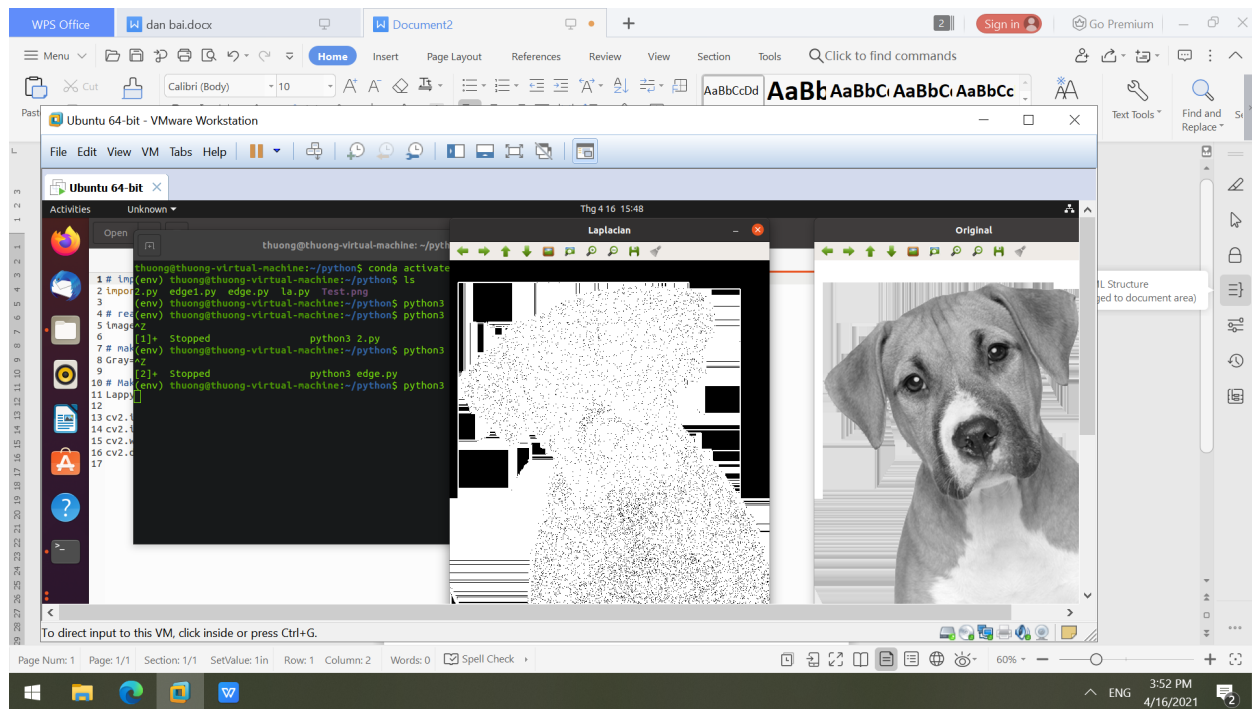
# read the image and store the data in a variable
image=cv2.imread("Test.png")

# make it grayscale
Gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

# Make Laplacian Function
Lappy=cv2.Laplacian(Gray,cv2.CV_64F)

cv2.imshow("Laplacian",Lappy)
cv2.imshow("Original",Gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output



Dùng sobel và laplacian vs matplotlib

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# loading image
img0 = cv2.imread("Test.png")

# converting to gray scale
gray = cv2.cvtColor(img0, cv2.COLOR_BGR2GRAY)

# remove noise
img = cv2.GaussianBlur(gray, (3,3),0)

# convolute with proper kernels
laplacian = cv2.Laplacian(img,cv2.CV_64F)
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0, ksize=5) # x
sobely = cv2.Sobel(img,cv2.CV_64F,0,1, ksize=5) # y

plt.subplot(2,2,1),plt.imshow(img,cmap = 'gray')
plt.title('Original'), plt.xticks([], plt.yticks([]))
plt.subplot(2,2,2),plt.imshow(laplacian,cmap = 'gray')
plt.title('Laplacian'), plt.xticks([], plt.yticks([]))
plt.subplot(2,2,3),plt.imshow(sobelx,cmap = 'gray')
plt.title('Sobel X'), plt.xticks([], plt.yticks([]))
plt.subplot(2,2,4),plt.imshow(sobely,cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([], plt.yticks([]))

plt.show()
```

Output

