

INDEX

- 1. Code For Ps 1-2**
- 2. Image Outputs Of Circuit.Png**
 - a. Input Image**
 - b. Grayscale Image**
 - c. Binary Image**
 - d. Output Image**
- 3. Image Outputs Of Circuit.Png**
 - a. Input Image**
 - b. Grayscale Image**
 - c. Binary Image**
 - d. Output Image**
- 4. Readme.Txt For Ps 1-2**
- 5. Code For Ps 1-3**
- 6. Image Outputs Of Smiley.Jpg**
 - a. Input Image**
 - b. Gamma Corrected Image**
- 7. Image Outputs Of Carnival.Jpg**
 - a. Input Image**
 - b. Gamma Corrected Image**
- 8. Readme.Txt For Ps 1-3**

```
import cv2
import numpy as np
import os

#Taking input image from the user int the form of file name
#Taking input on whether to expose dark or bright regions
file_name = input("Please enter the location of the image:")
intensity = input("Do you want to emphasize brighter or darker regions enter b for brighter d for darker :")

#reading the input image
first_image = cv2.imread(file_name)

#printing the org image
cv2.imshow("(a) input color image", first_image)

#get the image name from the input filename
image_name = os.path.basename(file_name)
image_name = image_name.split(".",1)[0]

#converting into GRAY_SCALE image
gray_image = cv2.cvtColor(src=first_image,code=cv2.COLOR_BGR2GRAY)

#converting to binary image with a threshold of 90 for circuit and 150 for crack
if(image_name == "crack"):
    _,binary_image = cv2.threshold(gray_image,150,255,type=cv2.THRESH_BINARY)

elif(image_name == "circuit"):
    _,binary_image = cv2.threshold(gray_image,90,255,type=cv2.THRESH_BINARY)

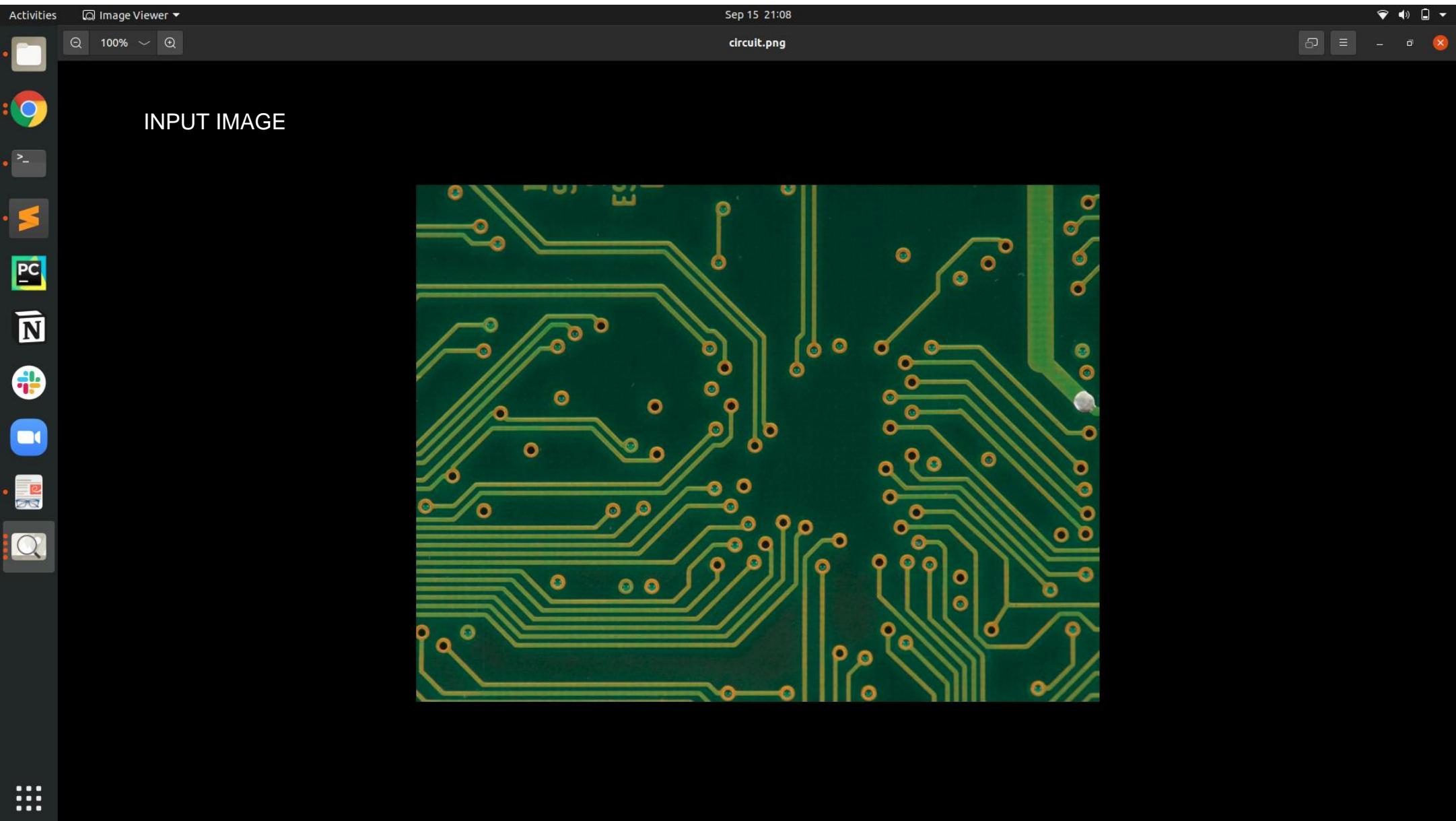
m, n = binary_image.shape
output = first_image
#if we want to convert dark regions to red, specifically for crack.png
if intensity == 'd':
    for i in range(m):
        for j in range(n):
            if(binary_image[i,j] == 0):
                output[i,j,2] = 255
                output[i,j,0], output[i,j,1] = 0, 0

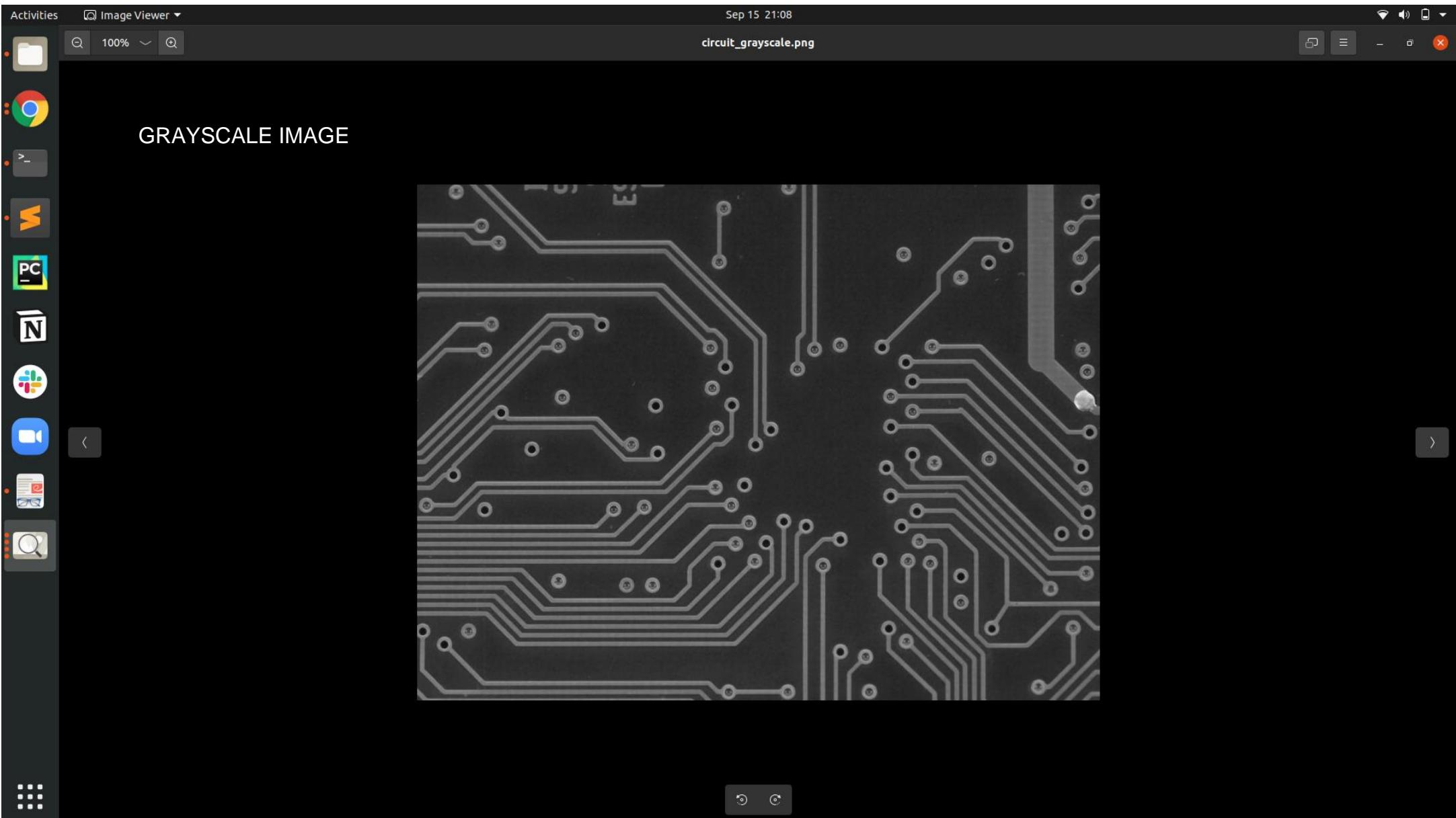
#if we want to convert bright regions to red, specifically for circuit.png
elif intensity == 'b':
    for i in range(m):
        for j in range(n):
            if(binary_image[i,j] == 255):
                output[i,j,2] = 255
                output[i,j,0], output[i,j,1] = 0, 0

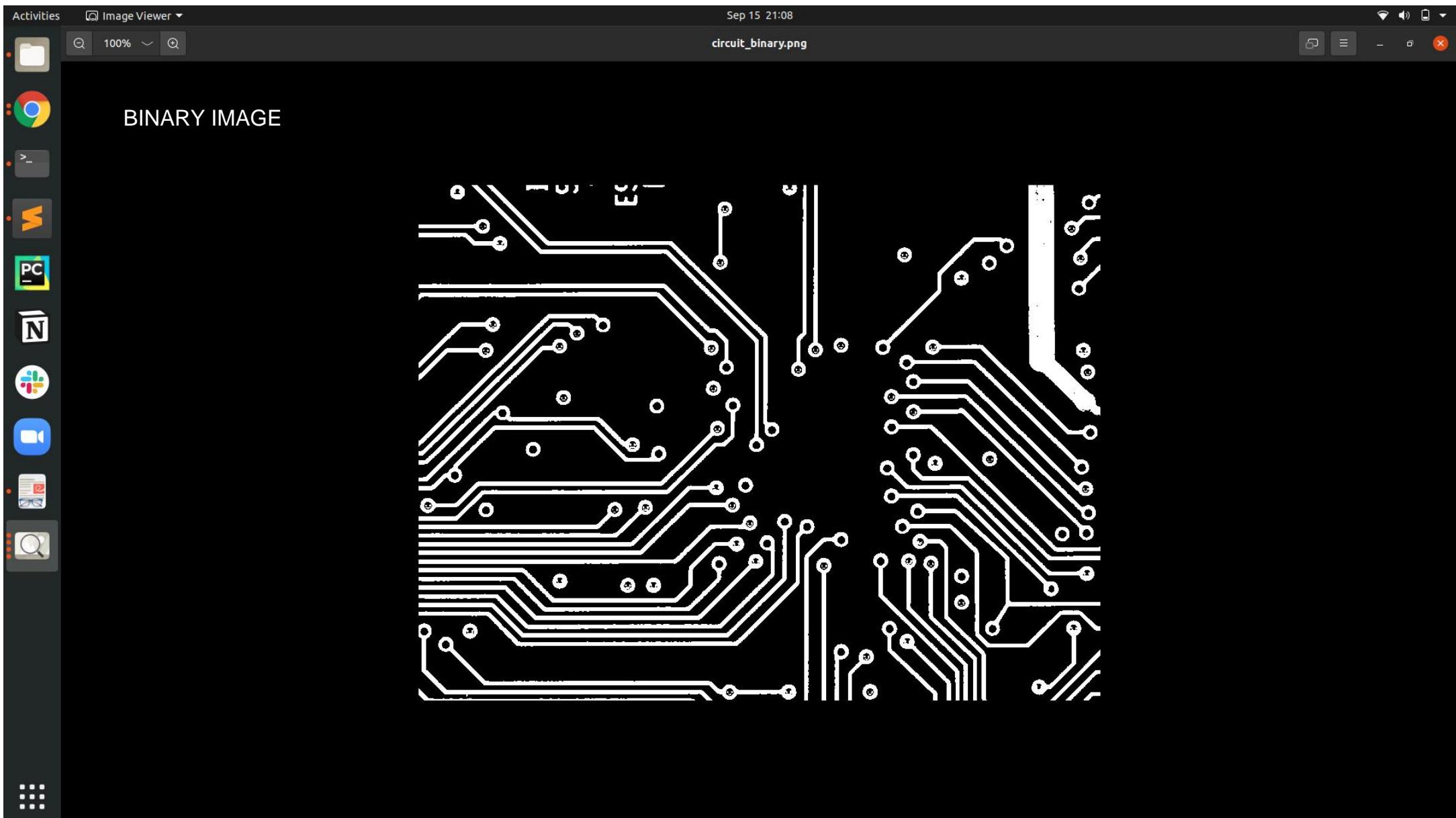
##Writing image to the files in current directory
path = "/home/akshay/Downloads/CV/ps1-sol/ps1-2/"
file = path+image_name+"_grayscale.png"
cv2.imwrite(file, gray_image)
file = path+image_name+"_binary.png"
cv2.imwrite(file, binary_image)
file = path+image_name+"_output.png"
cv2.imwrite(file, output)

#display all the images in 3 screens
cv2.imshow("(b) Grayscale image", gray_image)
cv2.imshow("(c) Black-and-White image", binary_image)
cv2.imshow("(d) Output color image", output)
cv2.waitKey(10000)
```

IMAGE OUTPUTS OF CIRCUIT.PNG







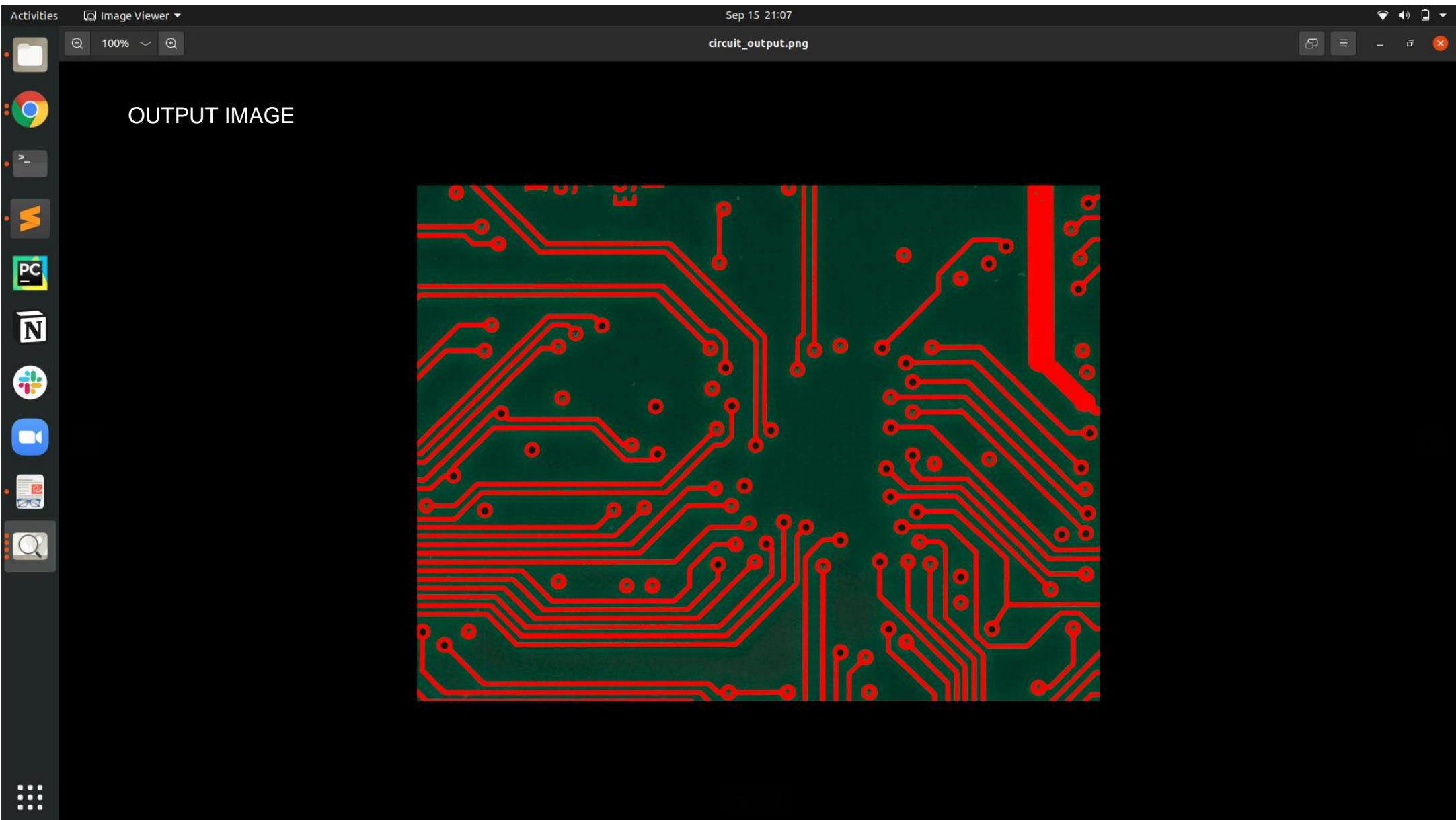
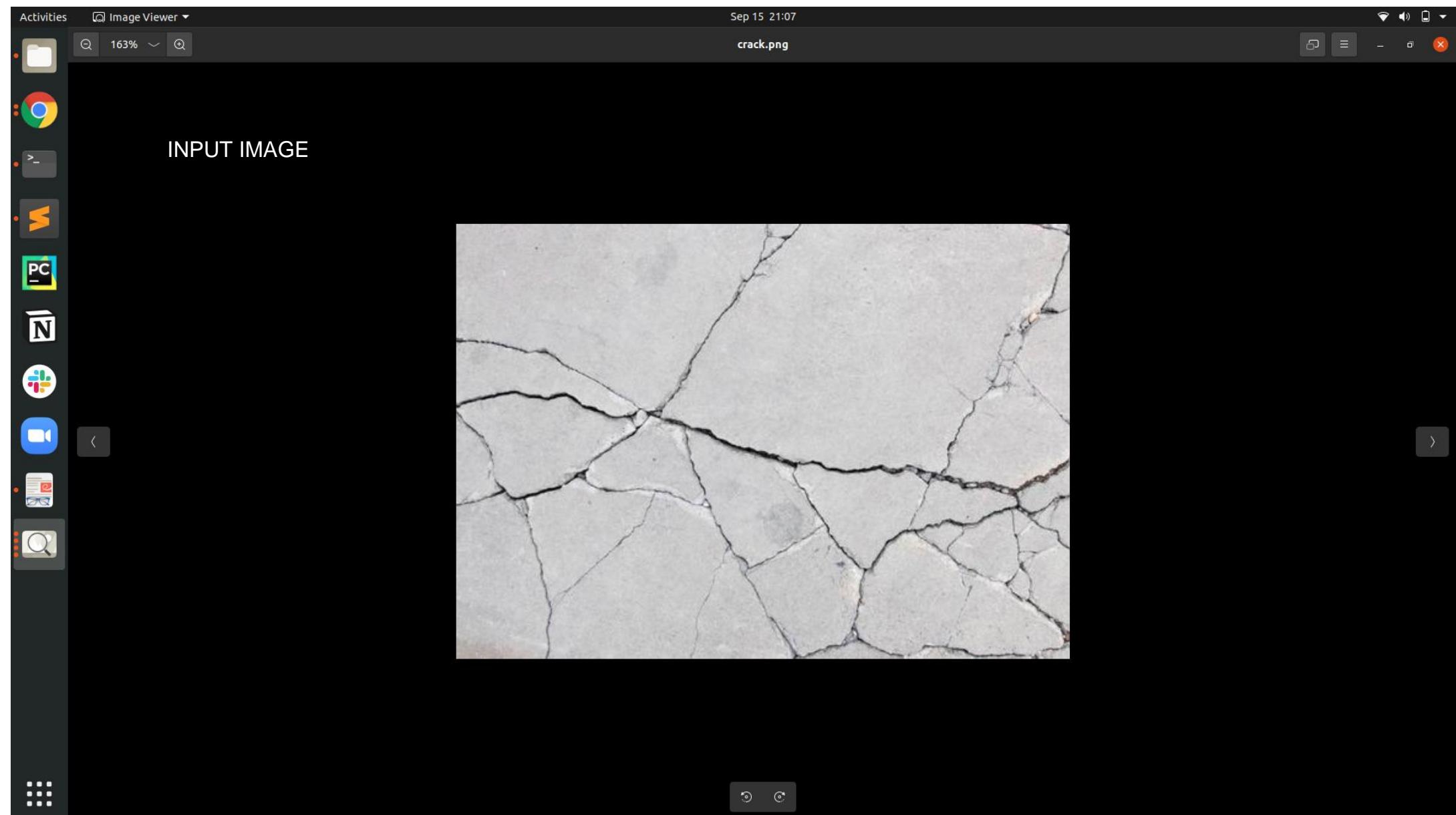
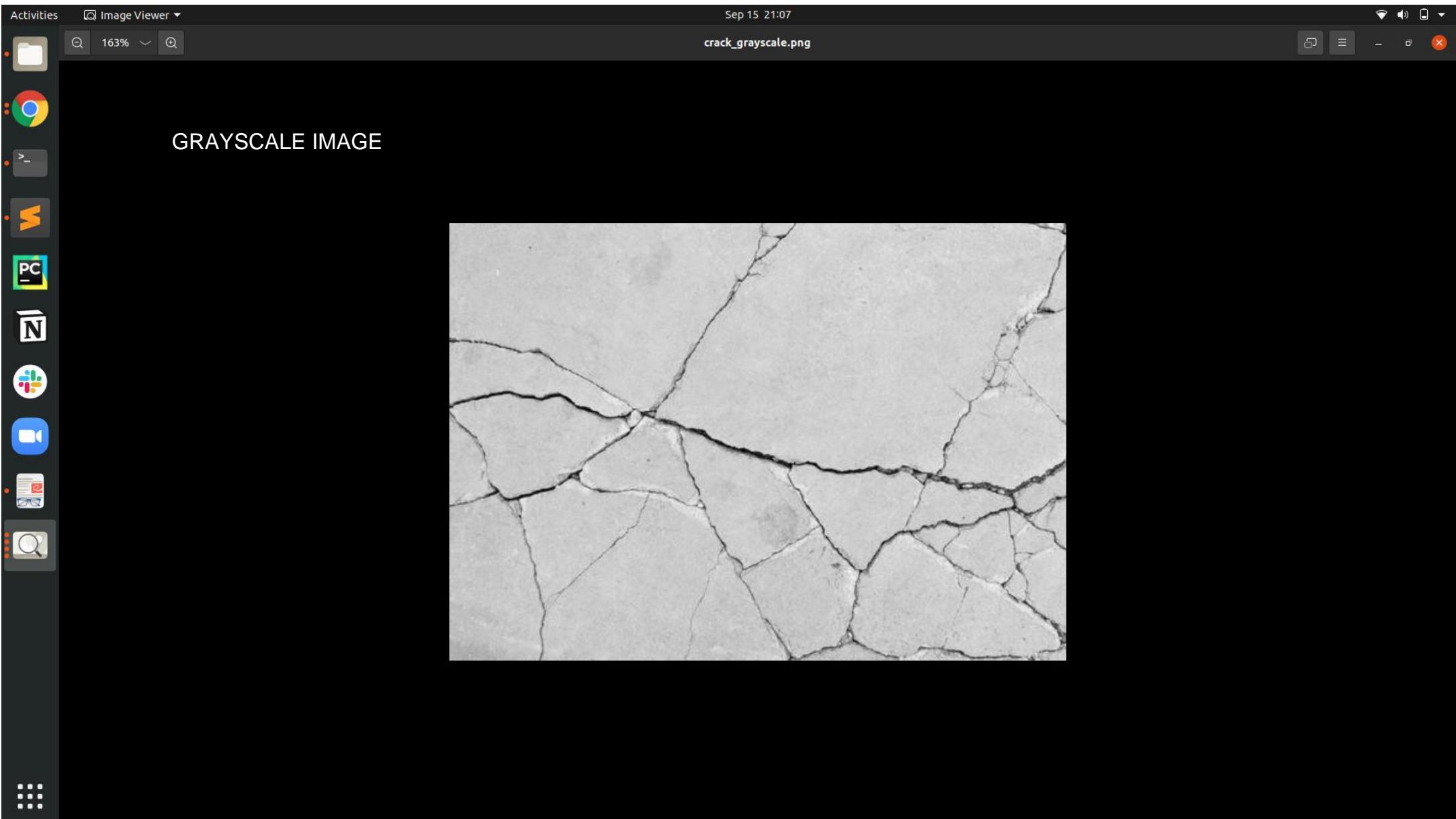
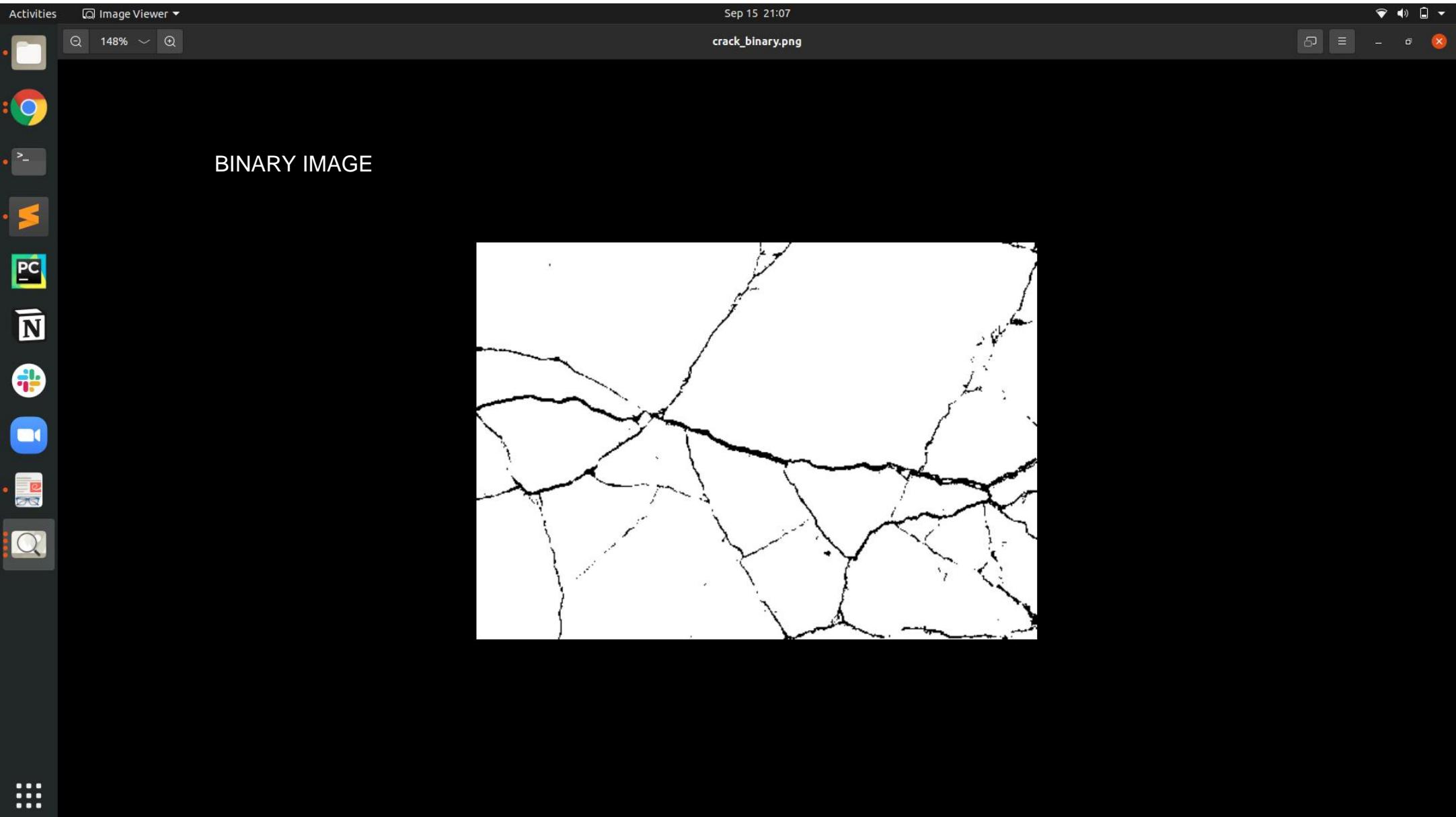
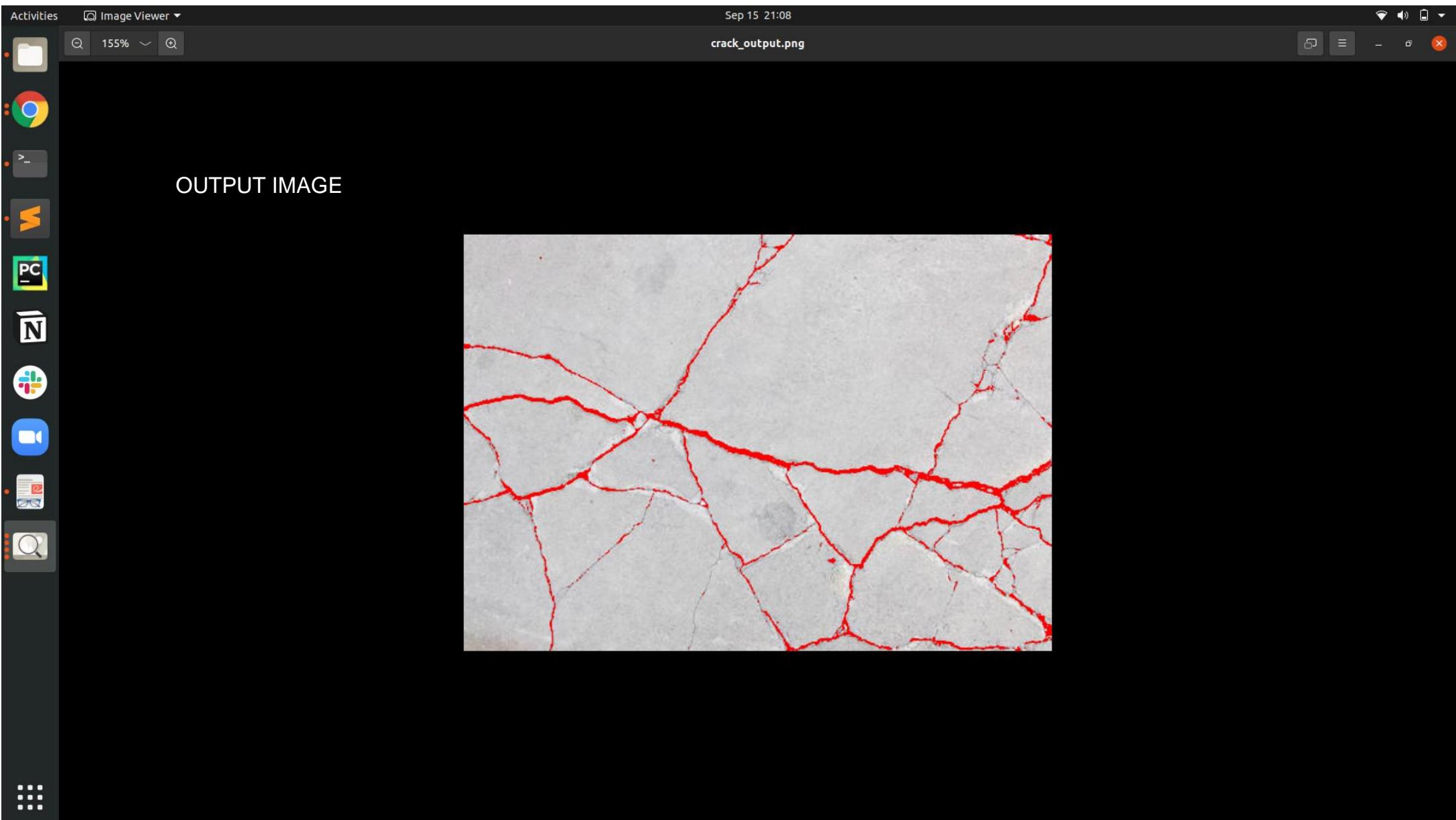


IMAGE OUTPUTS OF CRACK.PNG









README TXT

File: /home/akshay/Downloads/CV/ps1-sol/ps1-2/readme.txt

Page 1 of 1

Python version: Python 3.9.6

OpenCV Version: 4.5.2

Operating System: Linux 20.02

IDE: Sublime text, run via terminal

Almost spent 2-3 hours for this problem

```
import cv2
import numpy as np
import os

#Taking the filename and gamma from the user
file_name = input("Please input the location of the image: ")
#gamma for the dark image is around 2.0 and for the carnival image is around .5
gamma = input("Please input the gamma value: ")

#extracting the image name
image_name = os.path.basename(file_name)
image_name = image_name.split('.',1)[0]

#Reading the input image
image_input = cv2.imread(file_name)

#Making the image between 0-1
n_image = image_input/255.0
#Applying gamma correction by raising to the power of
n_image = np.power(n_image,1/float(gamma))
#Converting back to 0-255
n_image *= 255
#Making the data type 8-bit
n_image = np.asarray(n_image, dtype="uint8")

#Location of the current directory
path = "/home/akshay/Downloads/CV/ps1-sol/ps1-3"
file = path+image_name+"_gcorrected.jpg"
#Writing the image
cv2.imwrite(file,n_image)

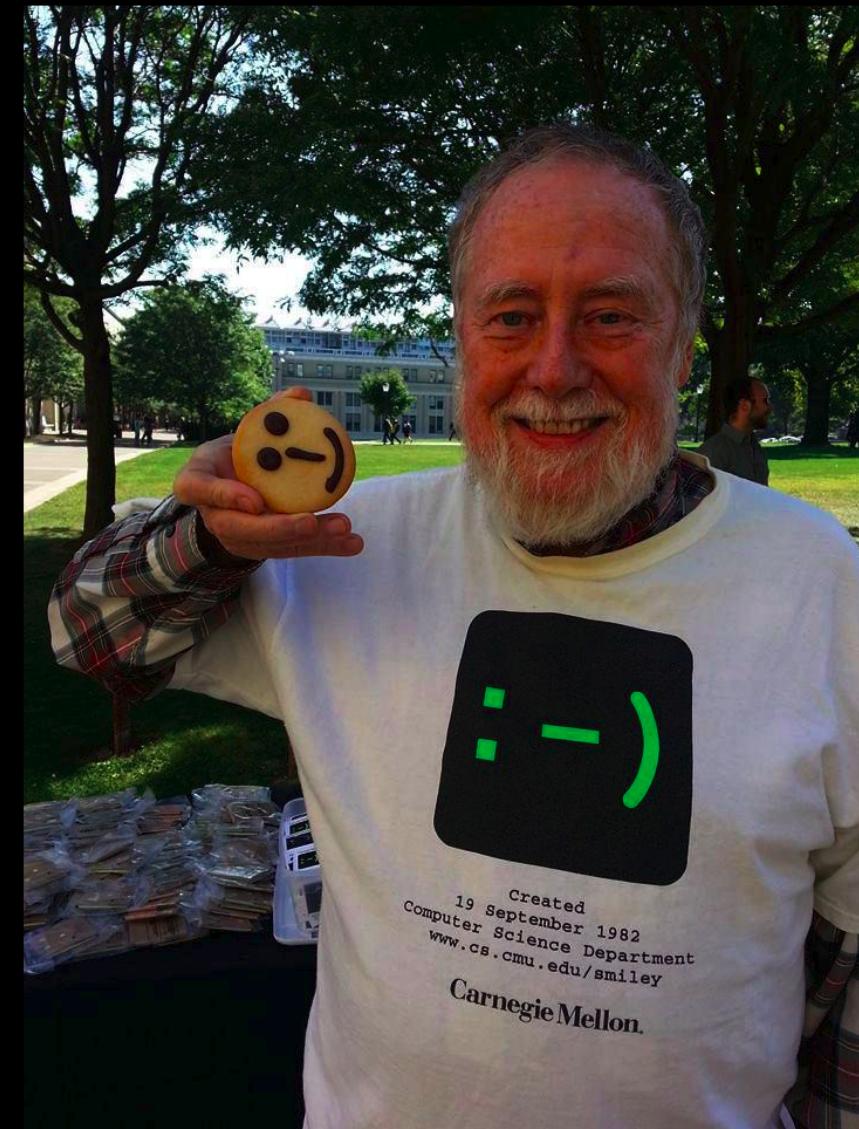
#Showing the orginal input and gamma corrected
cv2.imshow("(a)Input image",image_input)
cv2.imshow("(b) Output image after gamma correction",n_image)
cv2.waitKey(0)
```

100%

smiley.jpg

IMAGE OUTPUTS OF SMILEY.JPG

INPUT IMAGE



Activities

Image Viewer

Sep 15 21:09

Activities

100%

smiley_gcorrected.jpg

GAMMA CORRECTED IMAGE



INPUT IMAGE

GAMMA CORRECTED IMAGE



README TXT

File: /home/akshay/Downloads/CV/ps1-sol/ps1-3/readme.txt

Page 1 of 1

Python version: Python 3.9.6
OpenCV Version: 4.5.2
Operating System: Linux 20.02
IDE: Sublime text, run via terminal
Almost spend 3-3.5 hours for this problem