Converting a grayscale image to a pseudo-color image

CODE:

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 24 08:53:07 2021

@author: Parth Malpathak
"""
#C:/Users/parth/OneDrive/Desktop//cracks.png
#C:/Users/parth/OneDrive/Desktop//night-vision.png
#C:/Users/parth/OneDrive/Desktop//x-ray.png
#C:/Users/parth/OneDrive/Desktop//thermal.png
#C:/Users/parth/OneDrive/Desktop//topography.png
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
#Enter an image as input
inp = input("Enter a grayscale image: ")
img1 = cv.imread(inp)
img1 = cv.cvtColor(img1, cv.COLOR_BGR2RGB) #convert to RGB Scale
cv.imshow('Original Grayscale Image', img1)
gray = cv.cvtColor(img1, cv.COLOR_RGB2GRAY) #convert to Gray Scale single channel
shape = np.shape(gray)
print(shape)
width = shape[0]
height = shape[1]

#determine the max and min pixel size
small = np.amin(gray)
large = np.amax(gray)

num, a,b = 0,0,0

#Identify the coordinates where max pixel is located
for i in range(0,width):
    for j in range(0,height):
        if gray[i,j] == large:
            num += 1
            a += i
            b += j
```

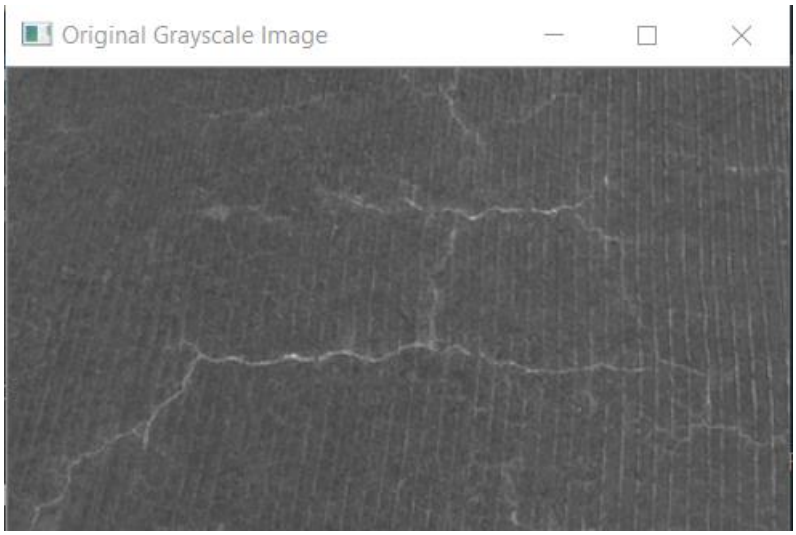
```
center_y = int(a/num)
center_x = int(b/num)
print(center_x , center_y)
#Differnece of largest and Smallest Pixel
diff = large - small

#Empty arry for r, g, b
r = []
g = []
b = []
inpt = []

for i in range (0,256):
    inpt.append(i)
'''' The tone curve was divided in 4 parts and calculation of each part was
     done separately like for example in the first part (255/4), the Red and Blue
     lines are horizontal with Green line traversing from 0 to 255. Same approach
     has been used for the other parts of the tone curve.'''
for i in range ((64)):
    r.append(0)
    g.append((255* ((i)/(255/4))))
    b.append(255)

for i in range ((64), (128)):
    r.append(0)
    g.append(255)

    b.append((-255* (4*i- 255)/255)+ 255)
for i in range ((128), (192)):
    g.append(255)
    b.append(0)
    r.append((255* (4*i - 2*255)/(255)))
    #r.append(int(diff*i)/255)

for i in range ((192) , 256):
    r.append(255)
    b.append(0)
```

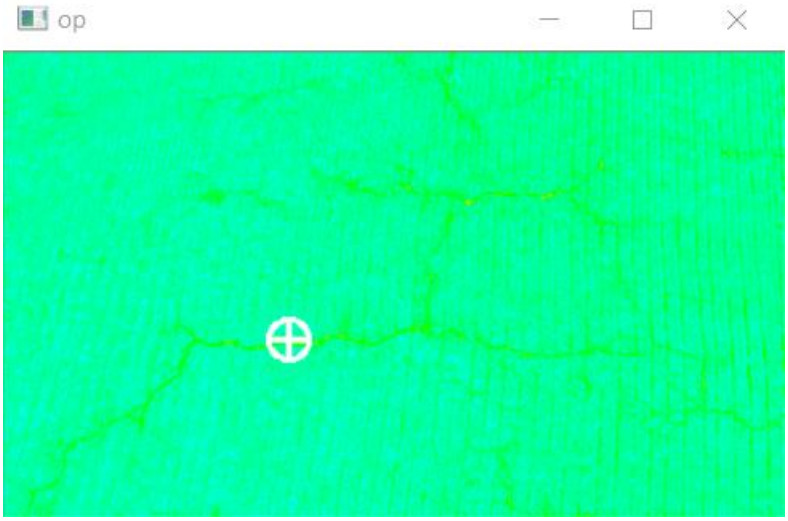
```
    g.append((-255*(4*i - 3*255)/255) + 255)


r= np.uint8((np.array(r)))
g= np.uint8((np.array(g)))
b= np.uint8((np.array(b)))
'''' Output Image is being made by adding the blue, green and the red colou
     to the corresponding spaces'''
output = np.zeros((width,height ,3), np.uint8)
output[:,:,0] = cv.LUT(img1[:,:,0], b)
output[:,:,1] = cv.LUT(img1[:,:,1], g)
output[:,:,2] = cv.LUT(img1[:,:,2], r)



white = (255, 255, 255)

#Initializing the Circle and the Marker to show the max pixel values
cv.circle(output, (center_x, center_y), 10, white, thickness = 2)
cv.drawMarker(output, (center_x, center_y), white, thickness = 2)


scale_percent = 50 #Resize the image to fit the screen
width = int(output.shape[1] * scale_percent / 100)
height = int(output.shape[0] * scale_percent / 100)
dim = (width, height)
output1 = cv.resize(output, dim, interpolation = cv.INTER_AREA)



cv.imshow("output", output) #Display the original Image

#Plotting the tone curve for reconfirmation of execution.
plt.plot(inpt, r, color ="red")
plt.plot(inpt,g, color ="green")
plt.plot(inpt, b, color ="blue")
plt.show()
```

```
cv.waitKey(0)
cv.destroyAllWindows
```

OUTPUT:

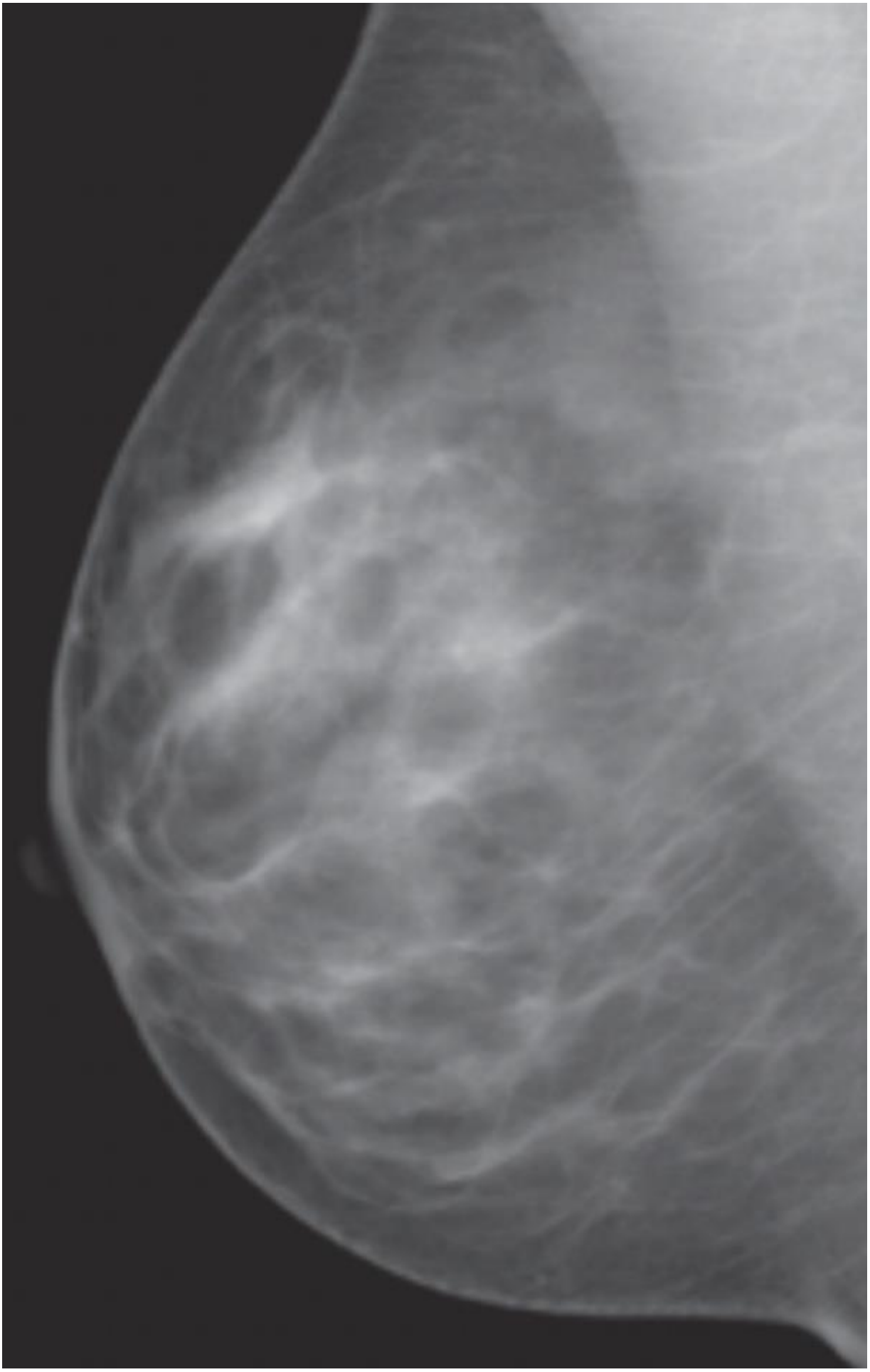
Cracks.png

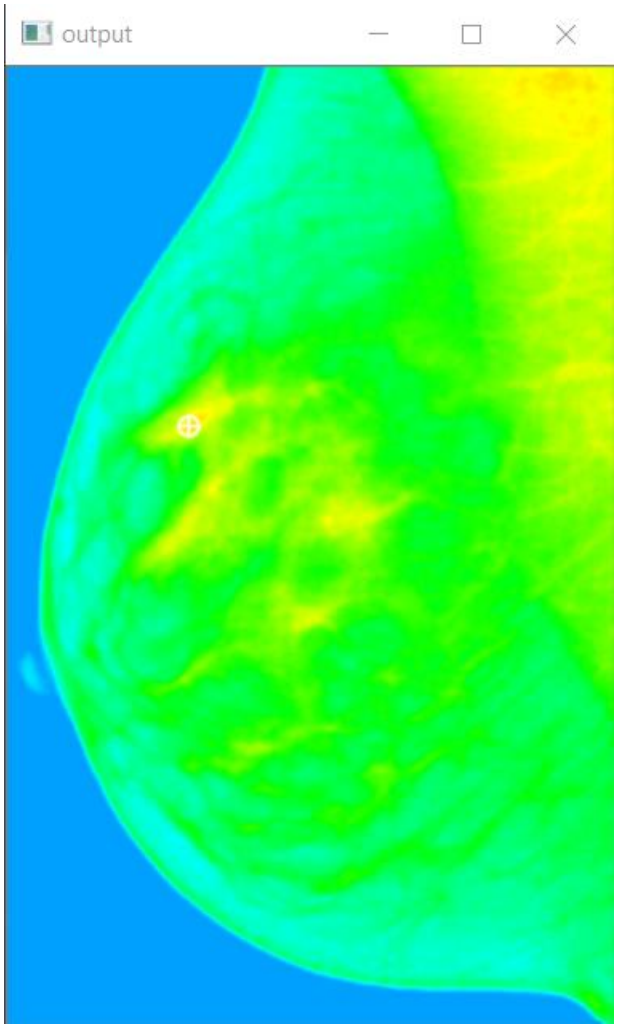




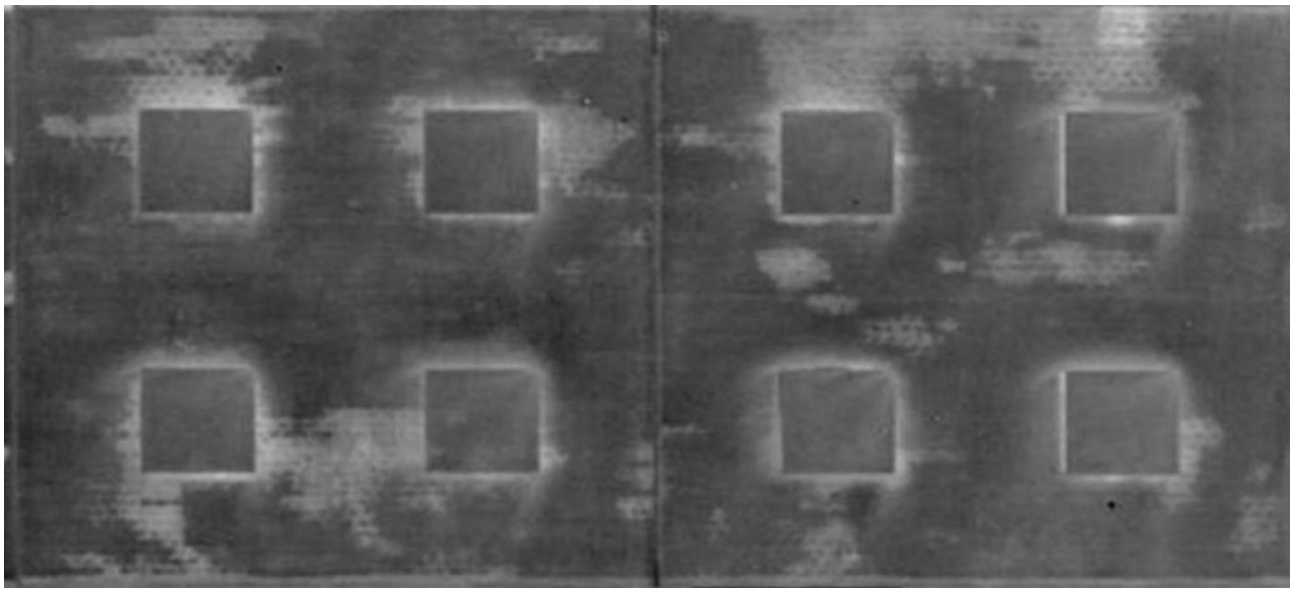
Night Vision. Png

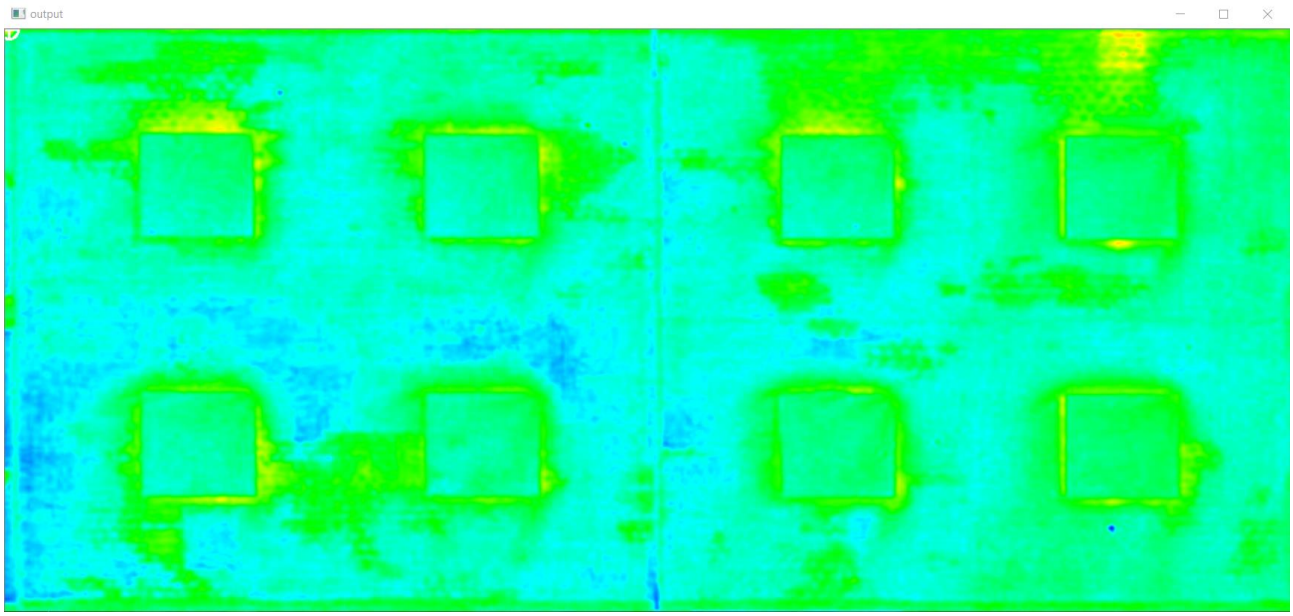
X-RAY.png



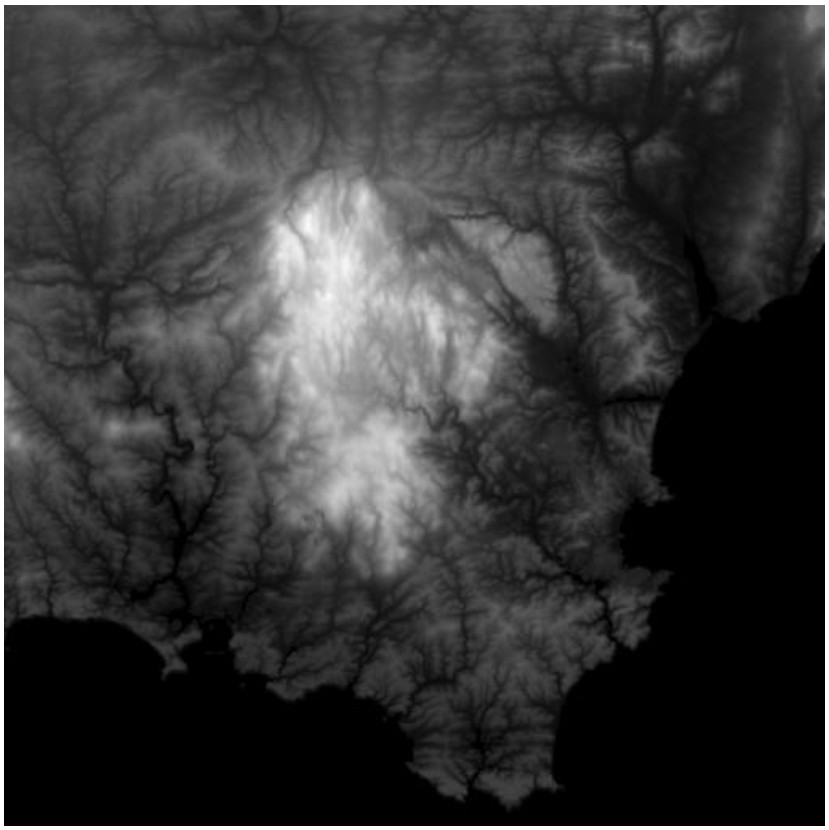


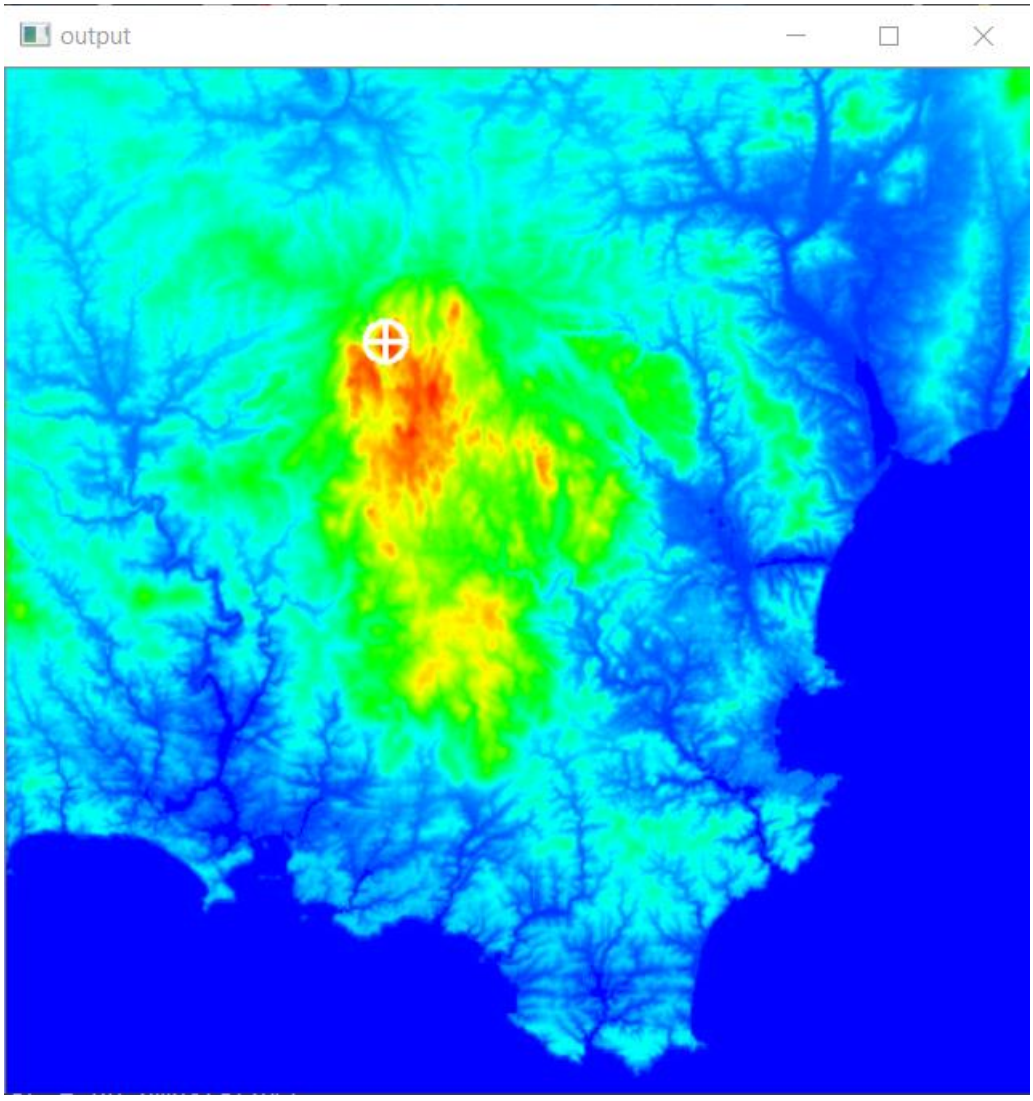
Thermal.png



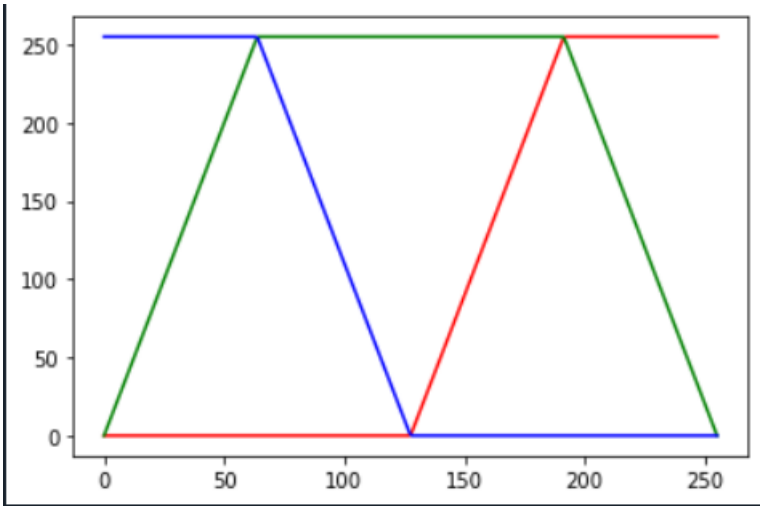


Topography.png





TONE CURVE OUTPUT:



Equation used to solve the tone curve was a standard line equation in two point form.

Operating System: Windows 10
IDE used: Spyder
Number of Hours spent: 7.5-8 Hours