Report

**Question 2**

1)

**Code:**

import cv2

img = cv2.imread('C:/Users/Shubham/Desktop/Fall19/imagin 558/ECE558-HW01/ECE558-HW01/wolves.png')

cv2.imshow('image1', img)

cv2.waitKey(0)

**Screenshot:**

A picture containing tree, outdoor, ground, sky

Description automatically generated

2)

**Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Character | ASCII | # in Red channel | # in Green channel | # in Blue channel |
| s | 115 | 2534 | 3170 | 2096 |
| m | 109 | 2717 | 3247 | 2254 |
| i | 105 | 2764 | 3381 | 2647 |
| g | 103 | 2814 | 3465 | 2605 |
| l | 108 | 2771 | 3274 | 2316 |
| a | 97 | 3063 | 3472 | 2807 |
| n | 110 | 2683 | 3281 | 2332 |

**Code:**

import cv2

import numpy as np

img = cv2.imread('C:/Users/Shubham/Desktop/Fall19/imagin 558/ECE558-HW01/ECE558-HW01/wolves.png')

myid = 'smiglan'

l = list(myid)

converted = np.zeros(len(l))

for i in range(len(l)):

converted[i] = ord(l[i])

print(converted)

b,g,r = cv2.split(img)

blue, green, red = np.zeros(len(l)),np.zeros(len(l)),np.zeros(len(l))

listOfCoordinates1,listOfCoordinates2,listOfCoordinates3 = list(np.zeros(len(l))),list(np.zeros(len(l))),list(np.zeros(len(l)))

x1 = dict(zip(np.unique(b, return\_counts=True)[0],np.unique(b, return\_counts=True)[1]))

x2 = dict(zip(np.unique(g, return\_counts=True)[0],np.unique(g, return\_counts=True)[1]))

x3 = dict(zip(np.unique(r, return\_counts=True)[0],np.unique(r, return\_counts=True)[1]))

for i in range(len(l)):

blue[i] = x1[converted[i]]

green[i] = x2[converted[i]]

red[i] = x3[converted[i]]

bluecount, greencount, redcount = dict(zip(converted,blue)),dict(zip(converted,green)),dict(zip(converted,red))

print('bluecount',bluecount,'\n','greencount',greencount,'\n','redcount',redcount)

#SECOND PART ENDS HERE

for i in range(len(l)):

listOfCoordinates1[i]= list(zip(np.where(b == converted[i])[0],np.where(b == converted[i])[1]))

listOfCoordinates2[i]= list(zip(np.where(g == converted[i])[0],np.where(g == converted[i])[1]))

listOfCoordinates3[i]= list(zip(np.where(r == converted[i])[0],np.where(r == converted[i])[1]))

def Filter(coordinates,channel):

for i in range(len(l)):

for c1,c2 in coordinates[i]:

if c1>1 and c2>1 and c1<img.shape[0]-2 and c2<img.shape[1]-2:

arr1 = channel[c1-2:c1+3,c2-2:c2+3]

arr1[arr1>=0] = 255

elif c1<2:

arr1 = channel[0:c1+3,c2-2:c2+3]

arr1[arr1>=0] = 255

elif c2<2:

arr1 = channel[c1-2:c1+3,0:c2+3]

arr1[arr1>=0] = 255

elif c1>img.shape[0]-3:

arr1 = channel[c1-2:img.shape[0],c2-2:c2+3]

arr1[arr1>0] = 255

elif c2>img.shape[1]-3:

arr1 = channel[c1-2:c1+3,c2-2:img.shape[1]]

arr1[arr1>0] = 255

else: print(c1,c2)

Filter(listOfCoordinates1,b)

Filter(listOfCoordinates2,g)

Filter(listOfCoordinates3,r)

img1 = cv2.merge((b, g, r))

cv2.imwrite( "smiglan.png", img1 );

cv2.imshow('image1', img1)

cv2.waitKey(0)

**Picture:**

**A close up of a tree

Description automatically generated**