INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 1

Aim:

Task1–¿Read a color image and convert it into gray scale image without using inbulit function(i.e do it by average method(R+G+B)/3)). Task2–¿Convert the pixel of gray scale image to either 1 or 0. Task3–¿Add gray image and image with pixels either 1 or 0 and add 20 to gray scale image. Perform the task and display the output images.

Results:

Code:

```
import math
import numpy as np
import cv2
image = cv2.imread('R.jpg')
m,n,v = image.shape
gray_image = np.zeros((m,n),np.uint8)
image_onezeroes = np.zeros((m,n),np.uint8)
image_onezeroes2 = np.zeros((m,n),np.uint8)
print(m,n,v)
dummy = 0
for i in range(m):
    for j in range(n):
        for k in range(v):
            dummy = dummy + image[i][j][k]
```

```
            gray_image[i][j] = math.floor(dummy/3)
            dummy = 0
for t in range(m):
    for u in range(n):
        image_onezeroes[t][u] = (gray_image[t][u])/255
for v in range(m):
    for w in range(n):
        if(gray_image[v][w] >= 128):
            image_onezeroes2[v][w] = 1
        else:
            image_onezeroes2[v][w] = 0
print("color image pixels")
print(image)
print("gray image pixels")
print(gray_image)
print("one/zero image pixels")
print(image_onezeroes)
print("one/zero image2 pixels")
print(image_onezeroes2)

cv2.imshow("original",image)
cv2.waitKey(0)
cv2.imshow('gray',gray_image)
cv2.waitKey(0)
cv2.imshow('zerosones',image_onezeroes)
cv2.waitKey(0)
cv2.imshow('zerosones2',image_onezeroes)
cv2.waitKey(0)
cv2.imshow('grayimage + zerosoneimage ',gray_image +
  image_onezeroes)
cv2.waitKey(0)
cv2.imshow('grayimage + zerosoneimage2 ',gray_image +
  image_onezeroes2)
cv2.waitKey(0)
cv2.imshow('grayimage + 20 ',gray_image + 20)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 2

Aim:

Task–¿Read a color image,convert the color image to gray scale and dispaly both images. Make some part of that gray scale image total black and display it.Now subtract this to images andd display output image.

Results:

Code:

```
import math
import numpy as np
import cv2
picture = cv2.imread(r"C:\Users\udayn\OneDrive\Desktop\
  PYTHONFILES\DIP\prabhas1.jpeg")
p,q,r = picture.shape
gray_img = np.zeros((p,q),np.uint8)
gray_img2 = np.zeros((p,q),np.uint8)
picture_onezeroes2 = np.zeros((p,q),np.uint8)
print("dimensions of picture is {}x{}x{}".format(p,q,r))

total = 0
for i in range(p):
    for j in range(q):
        for k in range(r):
            total = total + picture[i][j][k]
```

```
            gray_img[i][j] = math.floor(total/3)
            total = 0

for x in range(p):
    for y in range(q):
            gray_img2[x][y] = gray_img[x][y]

for x in range(20,720,1):
   for y in range(300,980,1):
           gray_img2[x][y] = 0

cv2.imshow("original picture",picture)
cv2.waitKey(0)
cv2.imshow('Grayscale picture',gray_img)
cv2.waitKey(0)
cv2.imshow('Grayscale picture2',gray_img2)
cv2.waitKey(0)
cv2.imshow('Grayscale picture - Grayscale picture2',abs(
   gray_img - gray_img2))
cv2.waitKey(0)
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : **D.Uday Kiran**
Roll.No : **BT19ECE007**
Course : **Digital Image Processing**
Instructor : **Dr.Tapan Jain**

LAB ASSIGNMENT 3

Aim:

Create two images one with a white circle at center and another with a white rentangle at center and perform all logical gate operations on both images and display the output images.

Results:

Code:

```
import math
import numpy as np
import cv2

pic1 = np.zeros((512,512),np.uint8)
pic2 = np.zeros((512,512),np.uint8)

pic1 = cv2.circle(pic1,(235,250),80,(255,255),-1)
pic2 = cv2.rectangle(pic2,(100,300),(330,190),(255,255,255)
   ,-1)


cv2.imshow("Image with white rectangle at center",pic2)
cv2.waitKey(0)
cv2.imshow("Image with white circle at center",pic1)
cv2.waitKey(0)
```

```
cv2.imshow("AND operation on images",cv2.bitwise_and(pic1,
    pic2))
cv2.waitKey(0)
cv2.imshow("NAND operation on images",cv2.bitwise_not(cv2.
    bitwise_and(pic1,pic2)))
cv2.waitKey(0)
cv2.imshow("OR operation on images",cv2.bitwise_or(pic1,pic2
    ))
cv2.waitKey(0)
cv2.imshow("NOR operation on images",cv2.bitwise_not(cv2.
    bitwise_or(pic1,pic2)))
cv2.waitKey(0)
cv2.imshow("EXOR operation on images",cv2.bitwise_xor(pic1,
    pic2))
cv2.waitKey(0)
cv2.imshow("EXNOR operation on images",cv2.bitwise_not(cv2.
    bitwise_xor(pic1,pic2)))
cv2.waitKey(0)
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : **D.Uday Kiran**
Roll.No : **BT19ECE007**
Course : **Digital Image Processing**
Instructor : **Dr.Tapan Jain**

LAB ASSIGNMENT 4

Aim:

Read a color image and display its reddish,greenish and bluish image.

Results:

Code:

```
import cv2

image_original = cv2.imread(r"C:\Users\udayn\OneDrive\
    Desktop\PYTHONFILES\DIP\prabhas1.jpeg")
img_red = cv2.imread(r"C:\Users\udayn\OneDrive\Desktop\
    PYTHONFILES\DIP\prabhas1.jpeg")
img_green = cv2.imread(r"C:\Users\udayn\OneDrive\Desktop\
    PYTHONFILES\DIP\prabhas1.jpeg")
img_blue = cv2.imread(r"C:\Users\udayn\OneDrive\Desktop\
    PYTHONFILES\DIP\prabhas1.jpeg")

img_blue[:,:,1],img_blue[:,:,2] = 0,0
img_green[:,:,0],img_green[:,:,2] = 0,0
img_red[:,:,0],img_red[:,:,1] = 0,0

cv2.imshow("Original Image",image_original)
cv2.waitKey(0)
cv2.imshow("Reddish Image",img_red)
```

```
cv2.waitKey(0)
cv2.imshow("Greenish Image",img_green)
cv2.waitKey(0)
cv2.imshow("Blueish Image",img_blue)
cv2.waitKey(0)
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 5

Aim:

Read a color image,convert the color image to gray scale and perform histogram equalization by algorithm discussed in class(i.e of wikipedia site). also check with direct function avaialable.

Results:

Code:

```
#Assignment5:
#Task-->Read a color image,convert the color image to gray
   scale and performe histogram equalization by algorithm
   discussed in class.
#also check with direct function avaialable.
#note: to get the histogram results of wikipedia site that
   dissussed in class, uncomment the commented part and
   comment the part that is mentioned to comment,
# you cant able to visualize image of wikipedia site
   becacause it is small but you can see the graph.
import math
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

```
image = cv2.imread(r"C:\Users\udayn\OneDrive\Desktop\
  PYTHONFILES\DIP\prabhas1.jpeg")  #comment this for
   wikipedia results
a,b,c= np.shape(image)                                  #comment
    this for wikipedia results

#grayscaleimage = np.array([[52,55,61,59,79,61,76,61],
#[62,59,55,104,94,85,59,71],
#[63,65,66,113,144,104,63,72],
#[64,70,70,126,154,109,71,69],
#[67,73,68,106,122,88,68,68],
#[68,79,60,70,77,66,58,75],
#[69,85,64,58,55,61,65,83],
#[70,87,69,68,65,73,78,90]],dtype=np.uint8)
#m,n = np.shape(grayscaleimage)
#grayscaleimage_1 = np.array([[52,55,61,59,79,61,76,61],
#[62,59,55,104,94,85,59,71],
#[63,65,66,113,144,104,63,72],
#[64,70,70,126,154,109,71,69],
#[67,73,68,106,122,88,68,68],
#[68,79,60,70,77,66,58,75],
#[69,85,64,58,55,61,65,83],
#[70,87,69,68,65,73,78,90]],dtype=np.uint8)

grayscaleimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
          #comment this for wikipedia results
grayscaleimage_1 = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
          #comment this for wikipedia results
unique_array = np.unique(grayscaleimage)
count_array = np.zeros(len(unique_array),int)
cdf_array = np.zeros(len(unique_array),int)
hv_array = np.zeros(len(unique_array),int)

for i in range(0,len(unique_array)):
    count_array[i] = np.count_nonzero(grayscaleimage ==
  unique_array[i])

cdf_array[0] = count_array[0]

for i in range(1,len(unique_array)):
```

```
    cdf_array[i] = cdf_array[i-1] + count_array[i]

cdf_min = min(cdf_array)

for i in range(0,len(unique_array)):
    hv_array[i] = round(((cdf_array[i] - cdf_min)*255)/((a*b
  ) - cdf_min))

for i in range(0,len(unique_array)):
    for j in range(a):
        for k in range(b):
            if(grayscaleimage_1[j][k] == unique_array[i]):
                grayscaleimage_1[j][k] = hv_array[i]
            else:
                continue

print("a x b x c  = %d x %d x %d"%(a,b,c))              #
  comment this for wikipedia results
#print("a x b = %d x %d"%(a,b))

print("grayscaleimage =")
print(grayscaleimage)
print("unique_array =")
print(unique_array)
print("count_array =")
print(count_array)
print("cdf_array =")
print(cdf_array)
print("cdf_min = %d"%(cdf_min))
print("hv_array =")
print(hv_array)
print("grayscaleimage_1 =")
print(grayscaleimage_1)

cv2.imshow('Image before histogram equalization',
  grayscaleimage)
cv2.waitKey(0)
cv2.imshow('Image after histogram equalization with
  algorithm discussed in class',grayscaleimage_1)
cv2.waitKey(0)
```

```
cv2.imshow('Image after histogram equalization with direct
    function',cv2.equalizeHist(grayscaleimage))
cv2.waitKey(0)
cv2.destroyAllWindows()
plt.subplot(3,1,1)
plt.hist(grayscaleimage.ravel(),256,[0,256])
plt.subplot(3,1,2)
plt.hist(grayscaleimage_1.ravel(),256,[0,256])
plt.subplot(3,1,3)
plt.hist(cv2.equalizeHist(grayscaleimage).ravel()
    ,256,[0,256])
plt.show()

#note:Enter space or any key on keyboard after display of
    first image to see other images....
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 6

Aim:

Task–¿Read a color image,convert the color image to gray scale and do contrast maniplution. display output images. ————————Contrast Manipulation————————- to increase contrast we multiple image with a constant, greater than one. to decrease contrast we multiple image with a constant, lesser than one

Results:

Code:

```
#Assignment6:
#Task-->Read a color image,convert the color image to gray
    scale and do contrast maniplution.
#display output images.
import math                                                    #
    import the required libraries.
import numpy as np
import cv2
image = cv2.imread(r"C:\Users\udayn\OneDrive\Desktop\
    PYTHONFILES\DIP\prabhas1.jpeg")          #reading the image
    .
grayscaleimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)    #
    Converting it into grayscale.

#----------------------Contrast Manipulation----------------------
```

```
#to increase contrast we multiple image with a constant,
   greater than one.
#to decrease contrast we multiple image with a constant,
   lesser than one.

cv2.imshow("original Image",grayscaleimage)
cv2.waitKey(0)
cv2.imshow("increase contrast Image",grayscaleimage*1.1)
cv2.waitKey(0)
cv2.imshow("decrease contrast Image",grayscaleimage*0.0009)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 7

Aim:

————————Bit Plane Slicing———————— Bit plane slicing is a method of representing an image with one or more bits of the byte used for each pixel. One can use only MSB to represent the pixel, which reduces the original gray level to a binary image. The three main goals of bit plane slicing is: –Converting a gray level image to a binary image. –Representing an image with fewer bits and corresponding the image to a smaller size –Enhancing the image by focussing.

Results:

Code:

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

Image = cv2.imread(r'C:\Users\udayn\OneDrive\Desktop\
  PYTHONFILES\DIP\prabhas1.jpeg',0)

Peak_SNR=10*np.log10((255*255)/(1/(225*225)*np.sum(Image)*np
  .sum(Image)))
print('Peak_SNR is: ',Peak_SNR)

def Convert_to_binary(num):
    binary_num = [int(i) for i in list('{0:0b}'.format(num))
  ]
```

```
    for j in range(8 - len(binary_num)):
        binary_num.insert(0,0)
    return binary_num
def Convert_to_decimal(listt):
    x = 0
    for i in range(8):
        x = x + int(listt[i])*(2**(7-i))
    return x
def discriminate_bit(bit,Image):
    z = np.zeros([225,225])
    for i in range(225):
        for j in range(225):
            x = Convert_to_binary(Image[i][j])
            for k in range(8):
                if k == bit:
                    x[k] = x[k]
                else:
                    x[k] = 0
            x1 = Convert_to_decimal(x)
            z[i][j] = x1
    return z
    # set up side-by-side image display

fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

for i in range(1,9):
    fig.add_subplot(4,2,i)
    plt.imshow(discriminate_bit(i-1,Image), cmap='gray')


plt.show(block=True)
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 8

Aim:

Contrast maniplulation of the Color image

Results:

Code:

```
#Assignment8:Read a color image and do contrast
   maniplulation of the image

import math
import numpy as np
import cv2

image = cv2.imread(r'C:\Users\udayn\OneDrive\Desktop\
   PYTHONFILES\DIP\prabhas1.jpeg')
          #Reading the image
a,b,c  = image.shape

Increased_contrast = np.zeros(image.shape, image.dtype)
        # To display image of increased contrast then
   original
Decreased_contrast = np.zeros(image.shape, image.dtype)
        # To display image of increased contrast then
   original
```

```
High_Contrast_Control = 1.6   # control for High Contrast
Low_Contrast_Control =  0.7      # control for Low Contrast
Brightness_Control = 0          # brightness control

for y in range(image.shape[0]):
    for x in range(image.shape[1]):
        for c in range(image.shape[2]):
            Decreased_contrast[y,x,c] = np.clip(
  Low_Contrast_Control*image[y,x,c] + Brightness_Control, 0,
   255)
            Increased_contrast[y,x,c] = np.clip(
  High_Contrast_Control*image[y,x,c] + Brightness_Control,
  0, 255)  #Clipping values out of range to into range

#Dispalying Images
cv2.imshow("Original Image",image)
cv2.waitKey(0)
cv2.imshow("Contrast Increased Image", Increased_contrast)
cv2.waitKey(0)
cv2.imshow("Contrast Decreased Image", Decreased_contrast)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 9

Aim:

————————-Histogram Streching———————————————— –¿There are two methods of enhancing contrast. The first one is called Histogram stretching that increase contrast.

Results:

Code:

```
################-----------Histogram Strecthing
    ----------------------##############

import matplotlib.pyplot as plt
import numpy as np
import cv2

image = cv2.imread(r'C:\Users\udayn\OneDrive\Desktop\
   PYTHONFILES\DIP\prabhas1.jpeg',0)

m,n = image.shape
print(m,n)

Strechted_image = np.zeros((m,n),np.uint8)

f_max = np.max(image)
f_min = np.min(image)
```

```
for i in range(m):
    for j in range(n):
        Strechted_image[i][j] = round((((image[i][j] - f_min
  )/(f_max - f_min))*255))
plt.subplot(2,1,1)
plt.hist(image.ravel(),256,[0,256])
plt.subplot(2,1,2)
plt.hist(Strechted_image.ravel(),256,[0,256])

plt.show()
cv2.imshow("Original Gray Scaled Image",image)
cv2.waitKey(0)
cv2.imshow("Histogram strechted Image",Strechted_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
NAGPUR

Name : D.Uday Kiran
Roll.No : BT19ECE007
Course : Digital Image Processing
Instructor : Dr.Tapan Jain

LAB ASSIGNMENT 10

Aim:

——————Histogram Specification————————————————–¿In image processing, histogram matching or histogram specification is the transformation of an image so that its histogram matches a specified histogram.

Results:

Code:

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
image = cv2.imread(r'C:\Users\udayn\OneDrive\Desktop\
   PYTHONFILES\DIP\sss1.jpg')
reference_image = cv2.imread(r'C:\Users\udayn\OneDrive\
   Desktop\PYTHONFILES\DIP\ss2.jpg')
gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
refer_img = cv2.cvtColor(reference_image, cv2.COLOR_BGR2GRAY
   )



m_img,n_img = img.shape
m_refer_img,n_refer_img = refer_img.shape
print(m_img,n_img)
print(m_refer_img,n_refer_img)
```

```
unique_array_img = np.unique(img)
unique_array_refer_img = np.unique(refer_img)

count_array_img = np.zeros(len(unique_array_img),int)
count_array_refer_img = np.zeros(len(unique_array_refer_img)
  ,int)

cdf_array_img = np.zeros(len(unique_array_img),int)
cdf_array_refer_img = np.zeros(len(unique_array_refer_img),
  int)

for i in range(0,len(unique_array_img)):
    count_array_img[i] = np.count_nonzero(img ==
  unique_array_img[i])

for i in range(0,len(unique_array_refer_img)):
    count_array_refer_img[i] = np.count_nonzero(refer_img ==
   unique_array_refer_img[i])

cdf_array_img[0] = count_array_img[0]

for i in range(1,len(unique_array_img)):
    cdf_array_img[i] = cdf_array_img[i-1] + count_array_img[
  i]

cdf_array_refer_img[0] = count_array_refer_img[0]

for i in range(1,len(unique_array_refer_img)):
    cdf_array_refer_img[i] = cdf_array_refer_img[i-1] +
  count_array_refer_img[i]

for i in range(m_img):
    for j in range(n_img):
        index = np.where(unique_array_img == img[i][j])
        for k in range(0,len(cdf_array_refer_img)):
            if(cdf_array_img[index] == cdf_array_refer_img[k
  ]):
                img[i][j] = unique_array_refer_img[k]
            else:
```

```
                continue

cv2.imshow("original Image",image)
cv2.waitKey(0)
cv2.imshow('reference Image',reference_image)
cv2.waitKey(0)
cv2.imshow("Original image's  Gray Scale image",gray_img)
cv2.waitKey(0)
cv2.imshow("Reference Image's Gray Scale Image",refer_img)
cv2.waitKey(0)
cv2.imshow('Histogram Matched Image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```