Graphical user interface, text

Description automatically generated

**Department of Computing**

**Digital Image Processing**

**Class: BSCS-9ABC**

**Lab 3**

**Submitted By: Fatima Seemab**

**CMS ID: 291310**

**BSCS 9B**

Lab 3

Basic Image Processing

**Task #1: Image Binarization using a predefined global threshold.**

Take an RGB image (preferably from the provided ones) and convert it to binarized form (in 0/1 form) by defining a single global threshold. Repeat the experiment with the three provided images and identify why a single global binarization threshold may not be applicable in a wide variety of application scenarios.

Note :\_ you have to perform task 1 on following umages B1,B2,B3

Code for B1

***RGB To binary using threshold***

from PIL import Image

#For processing image class PIL is imported from library PIL

im=Image.open("B1.png")

#For opening the image

x,y=im.size

Text

Description automatically generated#This will return no of rows and no of cols in #an image

for i in range(x):

#To iterate over all the rows

for j in range(y):

#To iterate over all the cols in a row

cordinate=i,j

# finding the intensity of current

#pixel

a,b,c=(im.getpixel((cordinate)))

#condition when red intensity level is

# above 160 new image will map to #lightest shade otherwise darker

if a>160:

a=265

else:

a=0

#condition when green

# intensity level is above 160 new #image will map to lightest shade otherwise darker

if b>160:

b=265

else:

b=0

#condition when blue intensity level is

#above 160 new image will

#map to lightest shade otherwise darker

if c>160:

c=265

else:

c=0

threshold=a,b,c

#for assigning new values to current pixels

im.putpixel((cordinate),(threshold))

#displays image in Jupiter notebook

im

***GrayScale to Binary using threshold***

Text

Description automatically generatedfrom PIL import Image,ImageOps

#For processing image class PIL is imported from library PIL

im=Image.open("B1.png")

#For opening the image

x,y=im.size

#This will return no of rows and no of cols in an imag

im=ImageOps.grayscale(im)

#Converting RGB to grayscale image

for i in range(x):

#To iterate over all the rows

for j in range(y):

#To iterate over all the columns in a row

cordinate=i,j

# finding the intensity of current

#pixel

a=(im.getpixel((cordinate)))

#condition when intensity level is

#above 180 new image will

#map to lightest shade otherwise darker

if a>180:

a=265

else:

a=0

threshold=a

#for assigning new values to current pixels

im.putpixel((cordinate),(threshold))

im

Code for B2

***RGB image to binary using threshold***

Text

Description automatically generated

from PIL import Image

#For processing image class PIL is imported from library PIL

im=Image.open("B2.jpg")

#For opening the image

x,y=im.size

#This will return no of rows and no of cols #in an image

for i in range(x):

#To iterate over all the rows

for j in range(y):

#To iterate over all the cols in a row

cordinate=i,j

# finding the intensity of current

pixel

a,b,c=(im.getpixel((cordinate)))

#condition when red intensity

#level is above 60 new image will

#map to lightest shade otherwise darker

if a>60:

a=265

else:

a=0

#condition when green

# intensity level is

#above 60 new image will

#map to lightest shade otherwise darker

if b>60:

b=265

else:

b=0

#condition when blue intensity level is

#above 60 new image will

#map to lightest shade otherwise darker

if c>60:

c=265

else:

c=0

threshold=a,b,c

#for assigning new values to current pixels

im.putpixel((cordinate),(threshold))

#displays image in Jupiter notebook

im

***GrayScale to binary using threshold***

#For processing image class PIL is imported from library PIL

from PIL import Image,ImageOps

#For opening the image

im=Image.open("B2.jpg")

#This will return no of rows and no of cols in an imag

x,y=im.size

#Converting RGB to grayscale image

im=ImageOps.grayscale(im)

#To iterate over all the rows

for i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current pixel

a=(im.getpixel((cordinate)))

#condition when intensity level is

#above 115 new image will

#map to lightest shade otherwise darker

if a>115:

a=265

else:

a=0

threshold=a

#for assigning new values to current pixels

im.putpixel((cordinate),(threshold))

Text, letter

Description automatically generated

Code for B3

***Grayscale to binary using threshold***

#For processing image class PIL is imported from library PIL

from PIL import Image

#For opening the image

im=Image.open("B3.jpg")

#This will return no of rows and no of cols in an image

x,y=im.size

#To iterate over all the rows

A picture containing text

Description automatically generatedfor i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current

pixel

a,b,c=(im.getpixel((cordinate)))

#condition when red intensity level is

#above 100 new image will

#map to lightest shade otherwise darker

if a>100:

a=265

else:

a=0

#condition when green

# intensity level is

#above 100 new image will

#map to lightest shade otherwise darker

if b>100:

b=265

else:

b=0

#condition when blue intensity level is

#above 100 new image will

#map to lightest shade otherwise darker

if c>100:

c=265

else:

c=0

#for assigning new values to current pixels

threshold=a,b,c

im.putpixel((cordinate),(threshold))

#displays image in Jupiter notebook

im

***Grayscale to binary using threshold***

from PIL import Image,ImageOps

#For processing image class PIL is imported from library PIL

im=Image.open("B3.jpg")

#For opening the image

x,y=im.size

#This will return no of rows and no of cols in an imag

#Converting RGB to grayscale image

im=ImageOps.grayscale(im)

#To iterate over all the rows

for i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current

#pixel

a=(im.getpixel((cordinate)))

#condition when intensity level is

#above 150 new image will

#map to lightest shade otherwise darker

if a>150:

a=265

else:

a=0

#for assigning new values to current pixels

threshold=a

im.putpixel((cordinate),(threshold))

Task #2: Create Intensity Histogram from a Greyscale image

CODE

#For processing image class PIL is imported from library PIL

from PIL import Image,ImageOps

#For plotting histogram class PIL is imported from library PIL

from matplotlib import pyplot as plt

# numpy libraray for dealing with array is imported

import numpy

#initializes array with zeros size of an array is equal to size of image

arr=numpy.zeros((98915))

#For opening the image

im=Image.open("B1.png")

#Converting RGB to grayscale image

greyscale=ImageOps.grayscale(im)

#This will return no of rows and no of cols in an imag

x,y=greyscale.size

count=0

#To iterate over all the rows

for i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current pixel

f=greyscale.getpixel((cordinate))

# array particular index will be assigned value equal to intensity

arr[count]=f

count=count+1

# hist function works in a way that it calculates frequency of an data point and displays the graph according so I give it an array having intensity value according to index

plt.hist (arr,bins=256)

plt.xlabel('pixel value')

plt.ylabel('no of points on that pixel')

.

Graphical user interface

Description automatically generated with medium confidence

Task #3: Recursive XY-cut algorithm

CODE:

from PIL import Image,ImageOps

#For processing image class PIL is imported from library PIL

from matplotlib import pyplot as plt

#For plotting histogram class PIL is imported from library PIL

import numpy

# numpy library for dealing with array is imported

plt.style.use('fivethirtyeight')

#For opening the image

im=Image.open("XY-cuts.jpg")

#Converting RGB to grayscale image

greyscale=ImageOps.grayscale(im)

#This will return no of rows and no of cols in an imag

x,y=greyscale.size

#array is initialized with zeros size equals no of rows in image

arr=numpy.zeros((x))

count=0

#To iterate over all the rows

for i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current

pixel

f=greyscale.getpixel((cordinate))

# if intensity is towards black shade particular array index will be incremented. In this way we maintains a

# number for black pixels in an array

if f < 50 :

arr[i]=arr[i]+1

else:

arr[i]=arr[i]

#after getting idea from graph one lowest peak shows white area is at 133 row so all cols will be assigned #black for this row

for j in range(x):

cordinate=x,j=j,133

greyscale.putpixel((cordinate),0)

#similarly other one lowest peak showing white area is at 270 row so all cols will be assigned black for this row

for j in range(x):

cordinate=x,j=j,270

greyscale.putpixel((cordinate),0)

#similarly other lowest peak showing white area is at 390 row so all cols will be assigned black for this row

for j in range(x):

cordinate=x,j=j,390

greyscale.putpixel((cordinate),0)

#for displaying image

Greyscale

# for plotting graph

plt.plot (arr)

#Labeling graph axis

plt.xlabel('no of rows')

plt.ylabel('no of black points ')

**PLOT**

Chart, line chart

Description automatically generated

**OUTPUT**

Text, letter

Description automatically generated

Hand in

Submit a lab report containing both the code and screenshots of output.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.
2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.

**Due date:**

**Check on lms**