Práctica 0

Realizado por: Braulio Castro

Procedimiento:

- Tiempo que toma procesar todas las imágenes: 1093.45700002 s.

- Tiempo promedio Conversión Escala de grises: 17.7689647675 s.

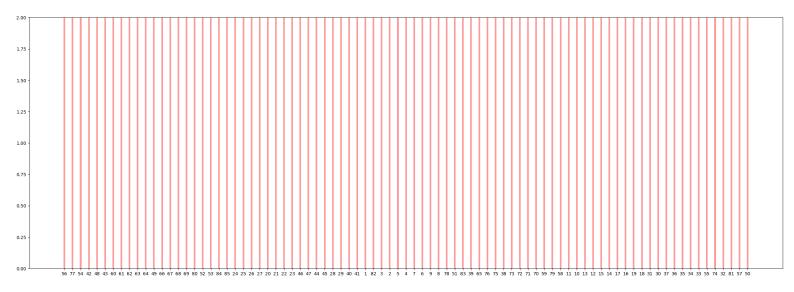
- Tiempo promedio Histograma Imágenes a color: 1055.40121245 s.

Tiempo promedio Histograma Imágenes grises: 20.0916996002 s.

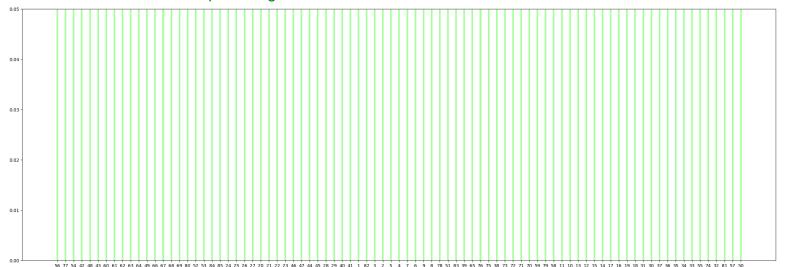
- Tamaño inicial de las imágenes (color): 1

- Tamaño final (escala de grises): 1

Gráficos



Tiempo Histograma de 1 canal



Tiempo Histograma de 3 canales



```
# -*- coding: utf-8 -*-
##Librerias
from reportlab.lib.utils import ImageReader
from time import sleep
from reportlab.pdfgen import canvas
import os
import time
from reportlab.lib.pagesizes import A4
import pylab as pl
import numpy as np
inicioT = time.time()##Devuelve el tiempo en segundos desde la época como un número de coma flotante.
##Este método devuelve una lista que contiene los nombres de las entradas en el directorio dado por la ruta.
sunflowers = os.listdir('sunflower/')
os.system('mkdir -p sunflowerGris')
os.system('mkdir histogramas1')
os.system('chmod 777 histogramas1')
sunflowersGris = os.listdir('sunflowerGris/')
##Variables.
gris_tiempo = 0
histograma3tiempo = 0
histograma1tiempo = 0
##Diccionarios
dic_colores = {}
dic_grises = {}
##
                         Metodos
##Metodo para obtener el Tamaño
def peso(ruta):
  print('Metodo peso= ',ruta)
```

```
"""Get size of a directory tree in bytes."""
  t = 0
  for path, dirs, files in os.walk(ruta):
     for archivo in files:
       t += os.path.getsize(os.path.join(path, archivo))
       print('peso...: ',t)
  return t
##Metodo para convertir a escala de Grises
def aGris(sunflowers, gris_tiempo):
  ## Se leen Archivos
  print('_____
  print("_ Convertir a Gris _")
  print('----')
  n = 0:
  for i in sunflowers:
     inicioG = time.time()
     orden = 'convert \'sunflower/' + i + '\' -set colorspace Gray -separate -average \'sun
     os.system(orden)
     gris_tiempo += (time.time()-inicioG)
     n += 1
     print('**************************T.P Grises ',n, ':::::::'\
        ,str(gris_tiempo/n), 'segundos.')
  return gris_tiempo
##Metodo par4a Obtener Histogramas de 3 canales a color
def histograma3(sunflowers, histograma3tiempo, dic_colores):
  print('_____
  print("_ Histogramas Color _")
  print('----')
  n = 1
  for j in sunflowers:
     inicio = time.time()
     carpeta = j[0: len(j)-4: 1]
     existe = os.path.exists('histogramas3/Histogramas-'+ carpeta)
     if existe != True:
Github ====: https://github.com/braulio1996/IAPractica0
```

```
os.system('mkdir -p histogramas3/Histogramas-'+ carpeta)
    orden = 'convert sunflower/'+ j +' -define histogram:unique-colors=true -format %c
    os.system(orden)
    orden = 'convert histogramas3/Histogramas-'+carpeta+'/histograma.gif -strip -resiz
    os.system(orden)
    dic_colores[str(n)] = round((time.time()-inicio),2)
    histograma3tiempo += (time.time()-inicio)
    n += 1
    print('T.P histogramas a color',n ,'::::'\
       ,str(histograma3tiempo/n), 'segundos.')
  print('_____')
  print("_ Histogramas obtenidos _")
  print('----')
  return histograma3tiempo
##Metodo para Obtener Histogramas de 1 canales a Escala de grises
def histograma1(sunflowers, histograma1tiempo, dic_grises):
  print('*******************************)
  print("* Histogramas Escala de grises *")
  print('************************')
  n = 1
  for sunflower in sunflowers:
    inicio_gris = time.time()
    print('-----',inicio_gris)
    carpeta = sunflower[0: len(sunflower)-4: 1]
    orden = 'convert sunflower/'+ sunflower\
       +' -colorspace Gray -define histogram:unique-colors=false histogram:histograma
         + carpeta +'.gif'
    os.system(orden)
    os.system('mv histograma-'+ carpeta +'.gif histogramas1/')
    dic_grises[str(n)] = round((time.time()-inicio_gris),2)
    histograma1tiempo += (time.time()-inicio_gris)
    n += 1
    print('********T.P Histogramas en Gris ',n ,'::::::'\
       ,str(histograma1tiempo/n), ' segundos.')
  print('_____
  print("* Histogramas obtenidos
  print('----')
```

Github ====: https://github.com/braulio1996/IAPractica0

return histograma1tiempo

```
##Metodo para obtener Graficas Generales
def graficaG(dic_colores,dic_grises):
  f, ax = pl.subplots(figsize=(30,10))
  x = np.arange(len(dic_colores))
  pl.bar(x, dic_colores.values(), align='center', width=0.2, facecolor='#ff9999')
  pl.bar(x, dic_grises.values(), align='center', width=0.2, facecolor='#a5ff99')
  pl.xticks(x, dic_colores.keys())
  ymax = 2
  pl.ylim(0, ymax)
  pl.savefig('General', bbox_inches='tight',pad_inches=0.1)
##Metodo para obtener Graficas en Colores
def graficaColores(dic_colores):
  f, ax = pl.subplots(figsize=(30,10))
  x = np.arange(len(dic_colores))
  pl.bar(x, dic_colores.values(), align='center', width=0.2, facecolor='#ff9999')
  pl.xticks(x, dic_colores.keys())
 ##pl.title('Imágenes en Colores')
  ymax = 2
  pl.ylim(0, ymax)
  pl.savefig('Color', bbox_inches='tight',pad_inches=0.1)
##Metodo para obtener Graficas en escala de Grises
def graficaGrises(dic_grises):
  f, ax = pl.subplots(figsize=(30,10))
  x = np.arange(len(dic_grises))
  pl.bar(x, dic_grises.values(), align='center', width=0.2, facecolor='#a5ff99')
  pl.xticks(x, dic_grises.keys())
  ##pl.title('Imágenes en Escala de Grises ')
  ymax = 0.05
  pl.ylim(0, ymax)
  pl.savefig('Gris', bbox_inches='tight',pad_inches=0.1)
```

```
gris_tiempo = aGris(sunflowers, gris_tiempo)
histograma1tiempo = histograma1(sunflowersGris, histograma1tiempo, dic_grises)
histograma3tiempo = histograma3(sunflowers, histograma3tiempo, dic_colores)
peso_color = peso('sunflower/')/1048576
peso_gris = peso('sunflowerGris/')/1048576
tiempo = (time.time()-inicioT)
##PASO Parametros
graficaGrises(dic_grises)
graficaColores(dic_colores)
                                                          Informe PDF
##
print('Generando PDF')
logo = ImageReader('https://www.ups.edu.ec/ups_portal-theme/images/ups/home/logo-
documento = canvas.Canvas('Informe', pagesize=A4)
documento.setFont("Helvetica", 23)
documento.setFillColor('red')
documento.drawlmage(logo, 10, 10, mask='auto')
documento.drawString(100,800,'
                                Práctica 0')
documento.setFont("Helvetica", 12)
documento.setFillColor('black')
x = 15
y = 770
documento.setFillColor('green')
documento.drawString(x,y,'Realizado por: Braulio Castro')
documento.setFillColor('blue')
documento.drawString(x,y-20,'Procedimiento:')
documento.setFont("Helvetica", 12)
documento.setFillColor('black')
documento.drawString(x,y-40,'- Tiempo que toma procesar todas las imágenes: '+str(tie
documento.drawString(x,y-60,'- Tiempo promedio Conversión Escala de grises: '+str(gr
documento.drawString(x,y-80,'- Tiempo promedio Histograma Imágenes a color: '+str(h
documento.drawString(x,y-100,'- Tiempo promedio Histograma Imágenes grises: '+str(l
documento.drawString(x,y-120,'- Tamaño inicial de las imágenes (color): '+str(peso_co
```

Github ====: https://github.com/braulio1996/IAPractica0

```
documento.drawString(x,y-140,'- Tamaño final (escala de grises): '+str(peso_gris))
documento.setFont("Helvetica", 20)
documento.setFillColor('red')
documento.drawString(x+240,y-180,'Gráficos')
documento.drawImage("Color.png", 10, y-400, width=580, height=200)
documento.setFont("Helvetica", 12)
documento.setFillColor('green')
documento.drawString(x+120,y-425,'Tiempo Histograma de 1 canal')
documento.drawlmage("Gris.png", 10, y-630, width=580, height=200)
documento.drawString(x+120,y-650,'Tiempo Histograma de 3 canales')
documento.showPage()
f = open("script.py","r")
lineas = f.readlines()
f.close()
for linea in lineas:
  y = y-20
  if y < 40:
    documento.showPage()
    documento.setFont("Helvetica", 15)
    documento.drawString(x,20,'Github ====: https://github.com/braulio1996/IAPract
    y = 770
  documento.drawString(x,y,linea[0:len(linea)-1])
documento.save()
```

print('Proceso Terminado')