

Adquisición de imagen digital e introducción a la fotografía

Objetivo: Utilizando Python se pretende que el alumno logre introducirse en el nuevo software implementando conceptos adquiridos en el teórico.

Metodología: Repaso de conceptos teóricos. Uso de material bibliográfico específico (Libro “Programming Computer Vision with Python”, <http://programmingcomputer vision.com/>). Discusiones pertinentes.

Higiene y Seguridad: Antes de realizar el laboratorio es necesario haber firmado la planilla que certifica la lectura y aceptación del documento “PROCEDIMIENTO DE SEGURIDAD GENERAL DEL LABORATORIO DE SEÑALES” y “NORMAS DE PROCEDIMIENTO PARA EL TRABAJO CON EQUIPOS ENERGIZADOS”.

Problema 1

Utilizando la librería OpenCv capturar una imagen sencilla desde la webcam y guardarla con el nombre “**captura.jpg**”

Código ejemplo

```
camera = cv2.VideoCapture(0)
return_value, image = camera.read()
cv2.imwrite('captura.png', image)
ahora = time.strftime("%c")
del(camera)
```

Utilizando la librería PIL realizar las siguientes funciones:

1. Abrir la imagen “**captura.png**”
2. Imprimir dimensiones
3. Convertirla a escala de grises
4. Convertirla a modo binario
5. Agregarle información de captura (Día, Hora y Año) y Guardarla como “**captura_ConInfo**”

Problema 1-1

Existen otros métodos de lectura de imagen (los vistos hasta ahora son webcam y desde archivo local). A continuación le adjuntamos 3 códigos.

Correr los mismos y explicar el funcionamiento.

Código 1

```
import numpy as np
import cv2
cap = cv2.VideoCapture('MATI.mp4')
ret, frame = cap.read()
```

```
while ret:
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame',gray)
    if cv2.waitKey(int((1/29.97)*1000)) & 0xFF == ord('q'):
        break
    ret, frame = cap.read()

cap.release()
cv2.destroyAllWindows()
```

Código 2

```
from PIL import Image
import requests
from matplotlib.pyplot import imshow, figure, subplot, show

url = "https://docs.opencv.org/3.4/lena.png"
im = Image.open(requests.get(url, stream=True).raw)
nx, ny = im.size
im2 = im.resize((int(nx*.10), int(ny*.10)),
Image.Resampling.BICUBIC)
figure()
subplot(1,2,1)
imshow(im)
print(im.size)
subplot(1,2,2)
imshow(im2)
print(im2.size)
show()
```

Código 3

```
import numpy as np
import cv2

# Open a sample video available in sample-videos
url="http://80.32.125.254:8080/cgi-bin/faststream.jpg?
stream=half&fps=15&rand=COUNTER"
vcap = cv2.VideoCapture(url)
#if not vcap.isOpened():
#    print "File Cannot be Opened"

while(True):
    # Capture frame-by-frame
    ret, frame = vcap.read()
    #print cap.isOpened(), ret
    if frame is not None:
```

```
# Display the resulting frame
cv2.imshow('Houston, Texas,Us',frame)
# Press q to close the video windows before it ends if you want
if cv2.waitKey(22) & 0xFF == ord('q'):
    break
else:
    print ("Frame is None")
    break

# When everything done, release the capture
vcap.release()
cv2.destroyAllWindows()
```

Problema 2

Crearle una **miniatura** de 128 x 128 a “**hoja.jpg**”, corriendo los códigos previstos a continuación, mostrar la original y la miniatura. Guardar esta última.

```
from PIL import Image
from PIL import ImageOps
from matplotlib.pyplot import imshow, show

pil_im = Image.open('hoja.jpg')
size=(128,128)

#USANDO RESIZE
imResize = pil_im.resize(size)
imResize.save('hojaResize.jpg')
imResize.show()

#USANDO FIT
imFit = ImageOps.fit(pil_im, size)
imFit.save('hojaFit.jpg')
imFit.show()

#USANDO THUMBNAIL
temp = pil_im.copy()
temp.thumbnail(size)
temp.save('hojaThumbnail.jpg')
temp.show()
imshow(temp)
show()
```

Explique las diferentes formas de la miniatura. Además, ¿qué diferencia hay entre función "imshow" (de matplotlib) y el método "show"(de PIL)?

Problema 3



Escalar al 75% la imagen **“hoja.jpg”** y rotarla al 90%. Mostrar resultados. Guardar la imagen resultante en formato .jpg con calidad del 90%.

1. ¿Qué sucede si se la guarda con el 25% de calidad?
2. Comparar las mismas observando diferencias.

Problema 4

Recortar la cara de Mickey de la imagen **“Mickey.jpg”** y pegarla sobre la imagen **“captura.jpg”**.
Mostrar resultados.

Problema 5

Cargue la imagen **“empire.jpg”** y dibuje una línea que pase sobre la cúpula del edificio (ver código ejemplo en página 4 del cap. 1 del libro supracitado)

Problema 6

Dada la imagen **“twobox.jpg”** convertirla a escala de grises, mostrarla y luego obtener el perfil horizontal de la línea (fila) 100.

- 1- ¿Puede obtener el tamaño relativo en pixeles de los cuadrados? (aproximado)
- 2- utilizando la función “pickle” ¿Cómo haría para volver a cargar dichos datos en un programa nuevo? Mostrarlos.
- 3- Comparar modulo “pickle” de python con las funciones “loadtxt” y “savetxt” de la librería Numpy.

Problema 7

Utilizando la imagen **“telefe.jpg”** separar la misma por canal de color R, G, B. Mostrar la original, la descomposición por canales y la resultante de la unión de los mismos.

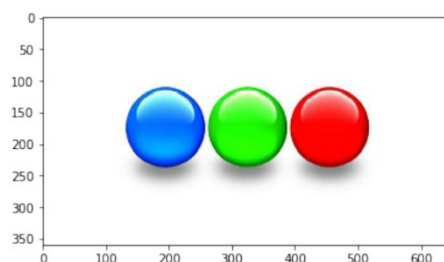


Ilustración 1_ Imagen RGB

Problema 8

Usando la imagen “**albatros.png**” realice un programa que la pueda convertir a otros espacios de color, ejemplo: HSV, YCrCb, CIE LUV, CMYK. Explique en qué se diferencian los mismos.

Nota: (“**Los esquemas de colores son distintas formas de parametrizar una imagen dependiendo de la aplicación para la cual se necesitan. Por ejemplo, el esquema RGB se usa mucho en electrónica y pantallas para mostrar una imagen al usuario, y HSV se utiliza para detección de cierto color en particular ya que su valor H determina el tinte.**”)



Ilustración 2_ Imagen RGB

Problema 9

La imagen “**fruta.tif**” fue obtenida de un sensor electrónico como parte de un sistema de control de calidad de fruta. Sin embargo, como puede apreciarse, la visualización de esta imagen no es buena. ¿Puede Ud. como Ingeniero detectar la razón de dicha dificultad para visualizar la imagen y corregirla?

Problema 10

Un sistema automático de almacenamiento de envíos postales debe reconocer el tipo de paquete que debe acomodar, siendo “A”, “B” y “C” los tres tipos existentes. El sistema lee directamente de una calcomanía pegada en el paquete la clasificación del mismo. Idee el algoritmo del sistema e implementelo.

Nota: RESOLVER sin utilizar machine learning, template machine, solamente usando geometría de la imagen.

Problema 11

Realizar 3 fotos con la misma exposición con distintos parámetros (configuraciones) del triángulo de exposición.



Problema 12

Tomar una imagen bien expuesta. Realizar una compresión. Posteriormente realizar una comparación de calidad de imagen.

**El informe debe contener: caratula, número de página, títulos debajo de las imágenes, etc.
También lo pueden realizar en Jupyter Notebook, entregando los .ipynb y .pdf.**

**Fecha de entrega
30 de agosto – subir al Aula virtual**

Un archivo .rar (con los apellidos de los integrantes) que contenga los .py mas el informe en formato PDF.