

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://programmingcomputervision.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:

- A. Abrir la imagen “**captura.jpg**”
- B. Imprimir dimensiones
- C. Convertirla a escala de grises
- D. Convertirla a modo binario
- E. 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(1/29.97) & 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%.

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