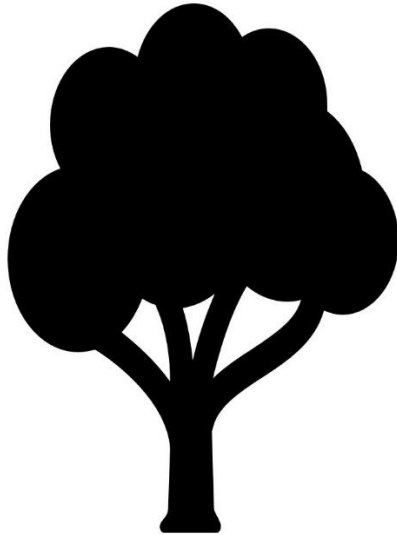


Análisis y Procesamiento de Imágenes

1.5 (1p) Características de la iluminación

RIVERA VARGAS JUAN

1. Imagen utilizada 'imagen 1'



2. Captura de imagen 2 [Código](#)

```
import cv2
#es usar un delay
import time
"""
En este caso, 0 quiere decir que queremos acceder
a la cámara 0. Si hay más cámaras, puedes ir probando
con 1, 2, 3...
"""
#se detiene 5s para posicionar la hoja
time.sleep(5)
cap = cv2.VideoCapture(0)

leido, frame = cap.read()

if leido == True:
    cv2.imwrite("imagen_2.jpg", frame)
    print("Foto tomada correctamente")
else:
    print("Error al acceder a la cámara")
"""
Finalmente liberamos o soltamos la cámara
"""
cap.release()
```



3. Comparación imagen 1 y 2 [Código](#)

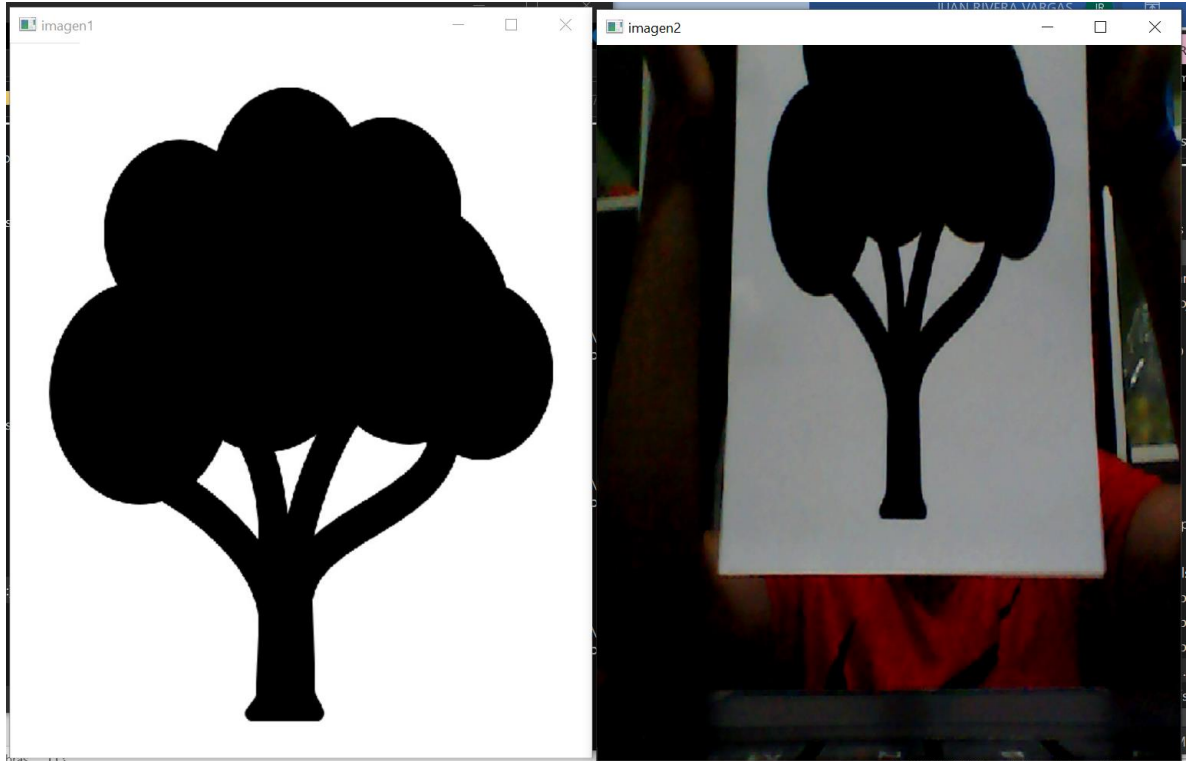
```
from __future__ import division
import cv2

img = cv2.imread('imagen_1.jpg')
img2 = cv2.imread('imagen_2.jpg')

screen_res = 1280, 720
scale_width = screen_res[0] / img.shape[1]
scale_height = screen_res[1] / img.shape[0]
scale = min(scale_width, scale_height)
window_width = int(img.shape[1] * scale)
window_height = int(img.shape[0] * scale)

#muestra imagen 1
cv2.namedWindow('imagen1', cv2.WINDOW_NORMAL)
cv2.resizeWindow('imagen1', window_width, window_height)
cv2.imshow('imagen1', img)
#muestra imagen 2
cv2.namedWindow('imagen2', cv2.WINDOW_NORMAL)
cv2.resizeWindow('imagen2', window_width, window_height)
cv2.imshow('imagen2', img2)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



4. Captura de imagen 3 [Código](#)

```
import cv2
#es usar un delay
import time
"""
En este caso, 0 quiere decir que queremos acceder
a la cámara 0. Si hay más cámaras, puedes ir probando
con 1, 2, 3...
"""
#se detiene 5s para posicionar la hoja
time.sleep(5)
cap = cv2.VideoCapture(0)

leido, frame = cap.read()

if leido == True:
    cv2.imwrite("imagen_3.jpg", frame)
    print("Foto tomada correctamente")
else:
    print("Error al acceder a la cámara")
"""
Finalmente liberamos o soltamos la cámara
"""
cap.release()
```



5. Comparación imagen 1 y 3 [Código](#)

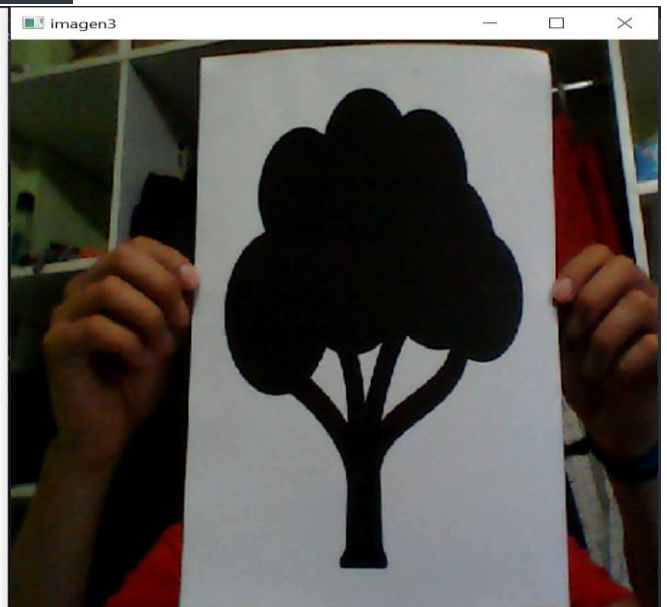
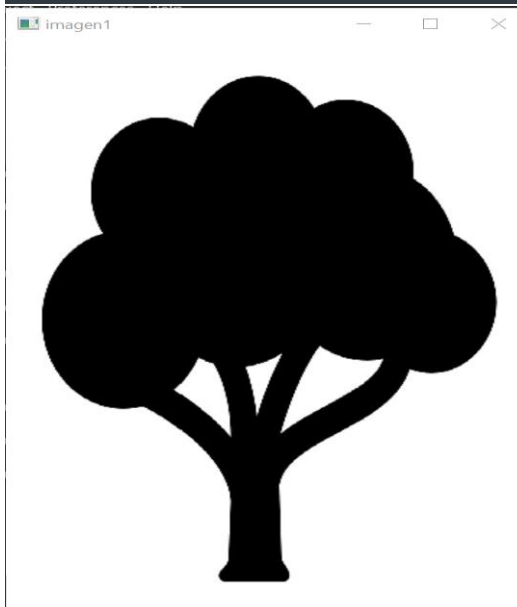
```
from __future__ import division
import cv2

img = cv2.imread('imagen_1.jpg')
img3 = cv2.imread('imagen_3.jpg')

screen_res = 1280, 720
scale_width = screen_res[0] / img.shape[1]
scale_height = screen_res[1] / img.shape[0]
scale = min(scale_width, scale_height)
window_width = int(img.shape[1] * scale)
window_height = int(img.shape[0] * scale)

#muestra imagen 1
cv2.namedWindow('imagen1', cv2.WINDOW_NORMAL)
cv2.resizeWindow('imagen1', window_width, window_height)
cv2.imshow('imagen1', img)
#muestra imagen 2
cv2.namedWindow('imagen3', cv2.WINDOW_NORMAL)
cv2.resizeWindow('imagen3', window_width, window_height)
cv2.imshow('imagen3', img3)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



6. Conclusión

Lo primero que ajusté para tener una buena comparación fue la posición de la hoja porque se salía del rango de la cámara, también agregué una pausa en la ejecución del código para tener tiempo suficiente y posicionar la hoja, la iluminación no la ajusté porque considero que ya me encontraba en la mejor posición sin embargo creo que el resultado no fue el mejor.