

# Practica 10

## Github

<https://github.com/Leogaltre/Practica-10---Uso-de-ROI-y-deteccion-de-esquinas.git>

Tipo de letra de escritura en Word para mostrar la programación  
-> 3ds tamaño 11

Tipo de Letra de Python -> Courier New tamaño 12

Objetivo dibujar sobre una imagen creada en fondo negro, posteriormente a la hora de entrega se agregará la segmentación y ROI

### ¿Qué es ROI?

Es una extracción de una región de interés que forma parte de una imagen

### Sintaxis de ROI

```
ROI = cv2.selectROI(Pizarra)
print("Recorte es de: ", ROI)
Pizarra_recortada = Pizarra[int(ROI[1]):int(ROI[1]+ROI[3]),
int(ROI[0]):int(ROI[0]+ROI[2])]
```

### Ejemplo de Creación de un Circulo

Circulo (nombre\_imagen, (a,b), radio, (B, G, R), Grosor)

Centro (a, b)

Para rellenar el área es con un -1

```
cv2.circle(Pizarra, (250,250), 140, (215, 0, 0), 5)
```

### Nota:

# Maneras de usar font

```
#cv2.putText(Pizarra, 'LGT', (200,270), #fondo, 2, (0,255,0), 2, cv2.LINE_AA)
```

# El número de fondo -> #fondo significa distintos tipos de letra

# 0 = cv2.FONT\_HERSHEY\_SIMPLEX

# 1 = cv2.FONT\_HERSHEY\_PLAIN

# 2 = cv2.FONT\_HERSHEY\_DUPLEX

# 3 = cv2.FONT\_HERSHEY\_COMPLEX

# 4 = cv2.FONT\_HERSHEY\_TRIPLEX

# 5 = cv2.FONT\_HERSHEY\_COMPLEX\_SMALL

# 6 = cv2.FONT\_HERSHEY\_SCRIPT\_SIMPLEX

```
# 7 = cv2.FONT_HERSHEY_SCRIPT_COMPLEX
```

```
# Extra
```

```
# Tamaño de la imagen 500 x 500
```

```
# Analiza 36 pixeles y sus colores de HSV
```

```
# for fila in range (20):
```

```
    # for columna in range (20):
```

```
        # print("Color", ": Fila:", + fila, "; Columna:", + columna, " -> ", str(hsv[fila, columna]))
```

```
#Conocer el valor del pixel (0 ,0) y (8,8)
```

```
# Va desde 0 hasta el tamaño de la imagen asi que
```

```
# Pienza en un numero menos 1
```

```
# Colores de Línea "Azul" -> Amarillo verde
```

```
#print("Color posición 0,0: " + str(Pizarra[250,250]))
```

## Código

```
import cv2
```

```
import numpy as np
```

```
img = cv2.imread("conto.png")
```

```
# Para seleccionar el Template
```

```
ROI = cv2.selectROI(img)
```

```
print("Recorte es de: ", ROI)
```

```
# Pizarra_cropped
```

```
recorte = img[int(ROI[1]):int(ROI[1]+ROI[3]),int(ROI[0]):int(ROI[0]+ROI[2])]
```

```
# cv2.imshow('Imagen Recortada', recorte)
```

```
# cv2.waitKey(0)
```

```
img_recorte_gris = cv2.cvtColor(recorte, cv2.COLOR_BGR2GRAY)
```

```
img_recorte_gray = np.float32(img_recorte_gris)
```

```
# Corner Harris
```

```
dst = cv2.cornerHarris(img_recorte_gray, 2, 3, 0.04)
```

```
height, width = dst.shape
```

```
color = (0, 200, 0)
```

```
for y in range(0, height):
```

```
for x in range(0, width):
    if dst.item(y, x) > 0.01 * dst.max():
        cv2.circle(recorte, (x, y), 2, color, cv2.FILLED, cv2.LINE_AA)

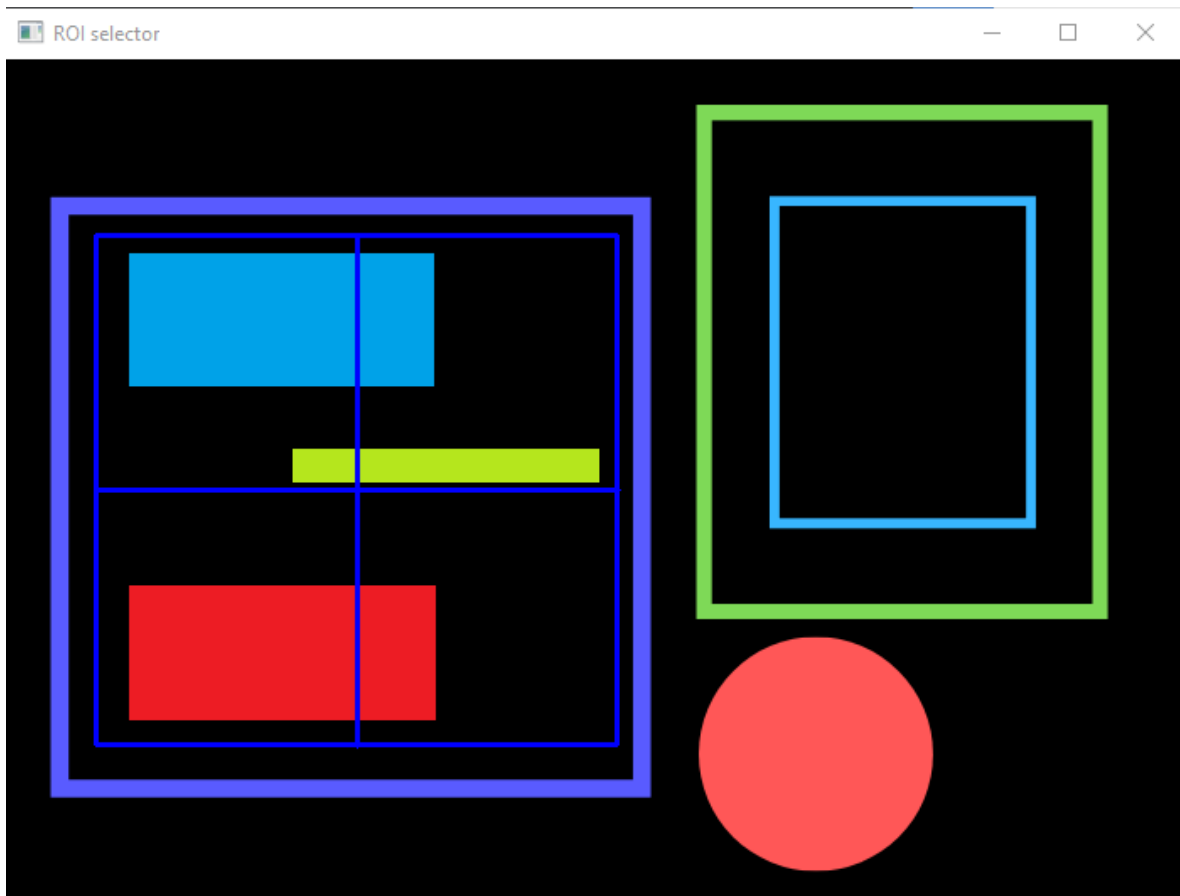
# Corner Harris
#cv2.imshow('Esquinas con Harris', dst)
cv2.imshow('Esquinas con Harris sobre el ROI', recorte)
cv2.waitKey(0)

# Good Features to track
esquinas = cv2.goodFeaturesToTrack(img_recorte_gris, 40, 0.01, 10)
esquinas = np.int0(esquinas)

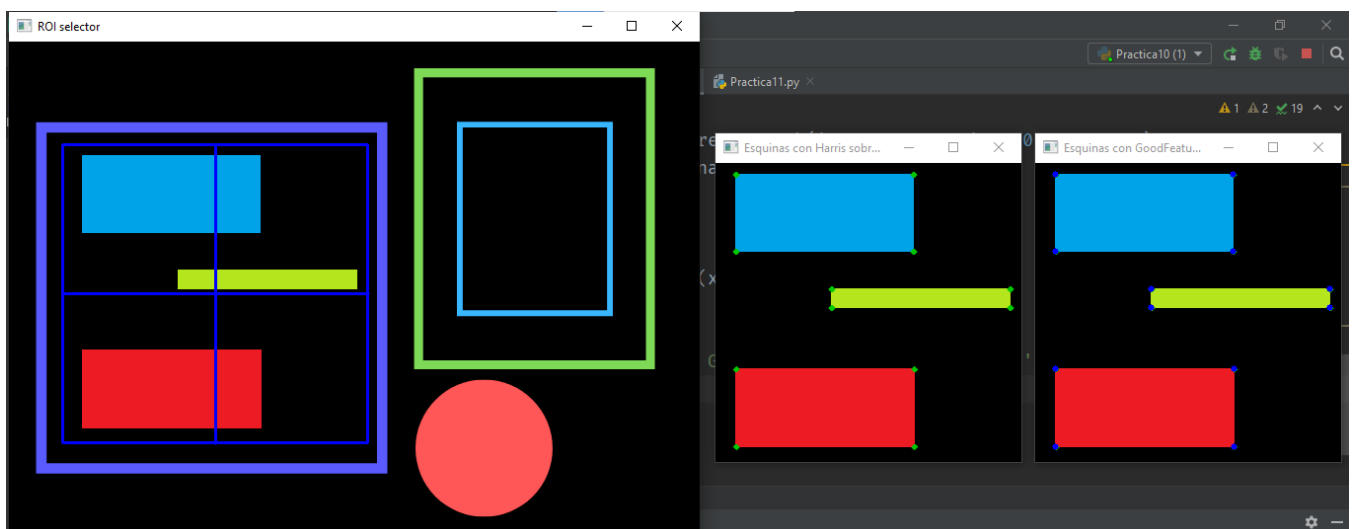
for i in esquinas:
    x, y = i.ravel()
    cv2.circle(recorte, (x, y), 3, 255, -1)

# Good Features to track
cv2.imshow('Esquinas con GoodFeaturesToTrack sobre ROI', recorte)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## ROI



## Como luce la práctica



**Referencias:**

- ROI

<https://www.youtube.com/watch?v=bx96VKJ4IM>

- Esquinas

[http://acodigo.blogspot.com/2017/07/deteccion-de-esquinas-con-opencv.html?m=1#:~:text=Uno%20de%20los%20m%C3%A9todos%20m%C3%A1s,cornerHarris\(\)%20](http://acodigo.blogspot.com/2017/07/deteccion-de-esquinas-con-opencv.html?m=1#:~:text=Uno%20de%20los%20m%C3%A9todos%20m%C3%A1s,cornerHarris()%20)

<https://es.acervolima.com/python-detectar-la-esquina-de-una-imagen-usando-opencv/>

- Harris Opencv

<https://unipython.com/deteccion-esquinas-harris/>

[https://opencv24-python-tutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_feature2d/py\\_features\\_harris/py\\_features\\_harris.html](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html)

[https://docs.opencv.org/4.x/dc/d0d/tutorial\\_py\\_features\\_harris.html](https://docs.opencv.org/4.x/dc/d0d/tutorial_py_features_harris.html)

- goodFeaturesToTrack

[https://docs.opencv.org/3.4/d4/d8c/tutorial\\_py\\_shi\\_tomasi.html](https://docs.opencv.org/3.4/d4/d8c/tutorial_py_shi_tomasi.html)

<https://theailearner.com/tag/cv2-goodfeaturestotrack/>