



Maestro: Mauricio Alejandro Cabrera Arellano

Alumno: Nene Miranda José Said 22110319

Materia: Visión Artificial

Tarea: Practica 9

Fecha: 01-06-2025

Practica 9:

Objetivo:

Crear un template (marcara del ROI a detectar).

Encontrar por lo menos 2 ROI's con un minimo de detección de .85

Codigo:

```
import cv2 # OpenCV para procesamiento de imágenes
import numpy as np # Para operaciones numéricas

# Cargar la imagen principal y la plantilla (ROI) en escala de grises
img = cv2.imread('Samus.png', cv2.IMREAD_GRAYSCALE)
template = cv2.imread('template.png', cv2.IMREAD_GRAYSCALE)
# 'Samus.png' es la imagen completa y 'template.png' es la región que queremos detectar

# Obtener las dimensiones del template
h, w = template.shape
# h = alto, w = ancho; se usan para dibujar los rectángulos del tamaño correcto

# Comparar la plantilla con la imagen usando correlación normalizada
result = cv2.matchTemplate(img, template, cv2.TM_CCOEFF_NORMED)
# Esta función devuelve una imagen de coincidencias con valores entre -1 y 1

# Definir el umbral mínimo de detección (valor de confianza)
threshold = 0.85
loc = np.where(result >= threshold)
# np.where devuelve las coordenadas donde la coincidencia fue mayor o igual al umbral

# Convertimos la imagen original a color para poder dibujar rectángulos de color
img_color = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
```

Contador de coincidencias detectadas

detect_count = 0

for pt in zip(*loc[::-1]): # Intercambiamos filas y columnas para usar como coordenadas (x, y)

detect_count += 1

cv2.rectangle(img_color, pt, (pt[0] + w, pt[1] + h), (0, 165, 255), 1)

Dibujamos un rectángulo naranja en cada coincidencia encontrada

Mostrar en consola cuántas coincidencias se detectaron

print(f"Regiones detectadas con confianza >= {threshold}: {detect_count}")

Mostrar la imagen con los recuadros de detección

cv2.imshow('Detecciones', img_color)

cv2.waitKey(0) # Esperar hasta que se presione una tecla

cv2.destroyAllWindows() # Cerrar la ventana de la imagen

Demostración:

