

**CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL**



**Template matching.**

Practica 9

Victor Alejandro Dominguez Cruz

21310228

6°G

Visión Artificial

10 de junio de 2025

**Objetivo:** Crear un template (marcara del ROI a detectar). Encontrar por lo menos 2 ROI's con un mínimo de detección de .85

**Código:**

```
import cv2
```

```
import numpy as np
```

```
img = cv2.imread('descarga.png', cv2.IMREAD_GRAYSCALE)
```

```
template = cv2.imread('template.png', cv2.IMREAD_GRAYSCALE) #región que  
queremos detectar
```

```
h, w = template.shape #dimensiones de la region h = alto, w = ancho se usan para  
dibujar los rectángulos del tamaño correcto
```

```
result = cv2.matchTemplate(img, template, cv2.TM_CCOEFF_NORMED) #  
Comparar la plantilla con la imagen usando correlación normalizada
```

```
threshold = 0.85 # Definir el umbral mínimo de detección (valor de confianza)
```

```
loc = np.where(result >= threshold) # np.where devuelve las coordenadas donde la  
coincidencia fue mayor o igual al umbral
```

```
img_color = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR) #Convierte la imagen a  
color
```

```
detect_count = 0 #Coincidencias detectadas
```

```
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, 0), 2) #Dibuja un  
rectángulo naranja en cada coincidencia encontrada
```

```
print(f"Regiones detectadas con confianza >= {threshold}: {detect_count}")#Cuenta las coincidencias
```

```
cv2.imshow('Detecciones', img_color) #Muestra recuadros de deteccion
```

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

```
cv2.destroyAllWindows()
```

### Demostración:

