

Practica 9

Centro de Enseñanza Técnica Industrial

Mecatrónica

Juan Alberto Carreon Merin

1F2

Vision Artificial

12/06/25

<https://github.com/yembo-34/Vision-Artificial>

Objetivo:

Template matching.

Crear un template (marcara del ROI a detectar).

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

Código:

```
import cv2
```

```
import numpy as np
```

```
imagen = cv2.imread('hotwheels.png', cv2.IMREAD_GRAYSCALE)
```

```
if imagen is None:
```

```
    raise ValueError("No se pudo cargar la imagen. Verifica el nombre y la ruta.")
```

```
# Coordinadas de recorte: (x, y, ancho, alto) — AJUSTA ESTAS SI ES NECESARIO
```

```
x, y, w, h = 70, 70, 50, 50 # Puedes ajustar para tu imagen
```

```
template = imagen[y:y+h, x:x+w]
```

```
cv2.imwrite('template_generado.png', template)
```

```
resultado = cv2.matchTemplate(imagen, template, cv2.TM_CCOEFF_NORMED)
```

```
umbral = 0.85
```

```
loc = np.where(resultado >= umbral)
```

```
imagen_color = cv2.cvtColor(imagen, cv2.COLOR_GRAY2BGR)
```

```
conteo = 0
```

```
for pt in zip(*loc[:, :-1]):
```

```
    cv2.rectangle(imagen_color, pt, (pt[0] + w, pt[1] + h), (255, 2, 45), 0)
```

```
    conteo += 1
```

```
print(f"Se encontraron {conteo} coincidencias con umbral  $\geq$  {umbral}")  
cv2.imshow('Detecciones de Template Matching', imagen_color)  
cv2.imshow('Template utilizado', template)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

Descripción:

Evidencia:

