009_P9_Template Matching

- Inicie sesión en mi Github

-Creé un nuevo archivo para la práctica 9 en Python

-Vi el tutorial del video de la práctica 9 de Python “Template Matching OpenCV Python Tutorial”

-Puse el código que venía de ejemplo y vi el video, en el video decía que podía intentar con mi propia imagen y tome una foto de mis carritos y la use de ejemplo.

-Lo ejecuté y ya me dio lo que aparecía en el video de cómo detectar las regiones idénticas de una imagen y a su vez que nos de una plantilla que le proporcionamos nosotros.

-Ya con mi imagen lo ejecuté y pues no salió del todo bien porque las casillas de los carritos no las detectaba del todo igual y la imagen de donde detectaba el rectángulo era de toda la imagen.

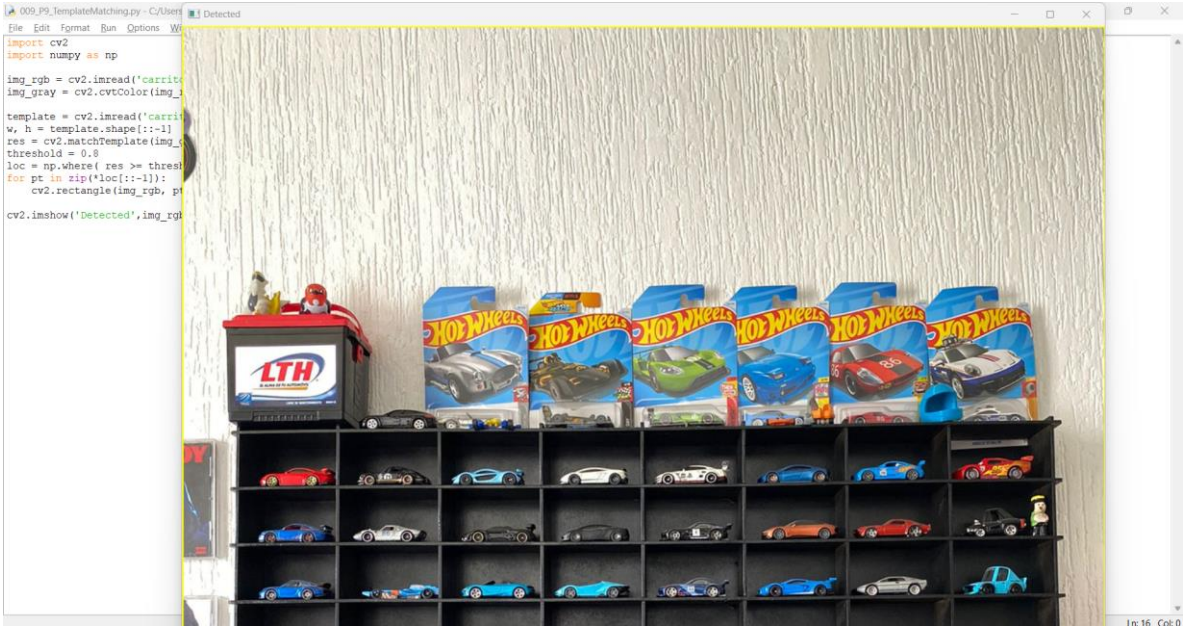
-Le pedí ayuda a chat y me mejoró que me detectara por lo menos 2 casillas y ya con esto hice lo que indicaba el profe de crear un template (marcara del ROI a detectar) y encontrar por lo menos 2 ROI's con un mínimo de detección de .85

-Lo ejecuté ya la versión mejorada y pues primero tuve que pedir a chat que con el mouse eligiera la parte que fuera mi plantilla del template y pues me cargo la zona, pero el mínimo de detección no fue del todo preciso con el 0.85 y tome mis respectivas pruebas.

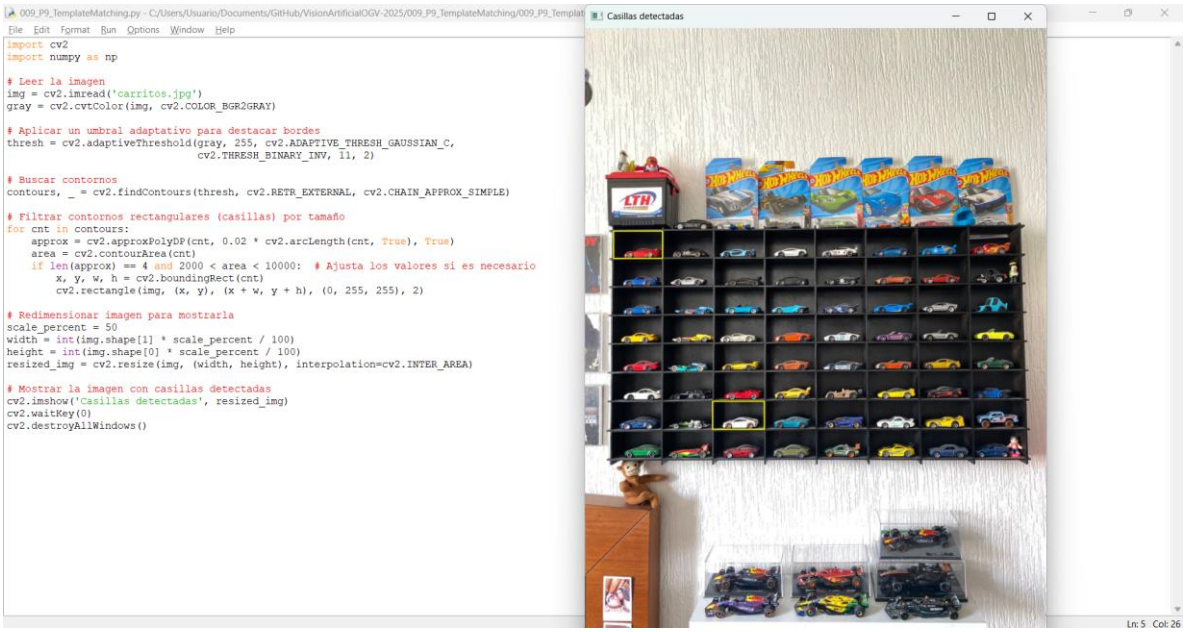
-Ya para finalizar, creé el reporte. -Subí todos los archivos a mi GithubDesktop sobre la práctica 9 “Template Matching”.

IMAGEN DE EJEMPLO:

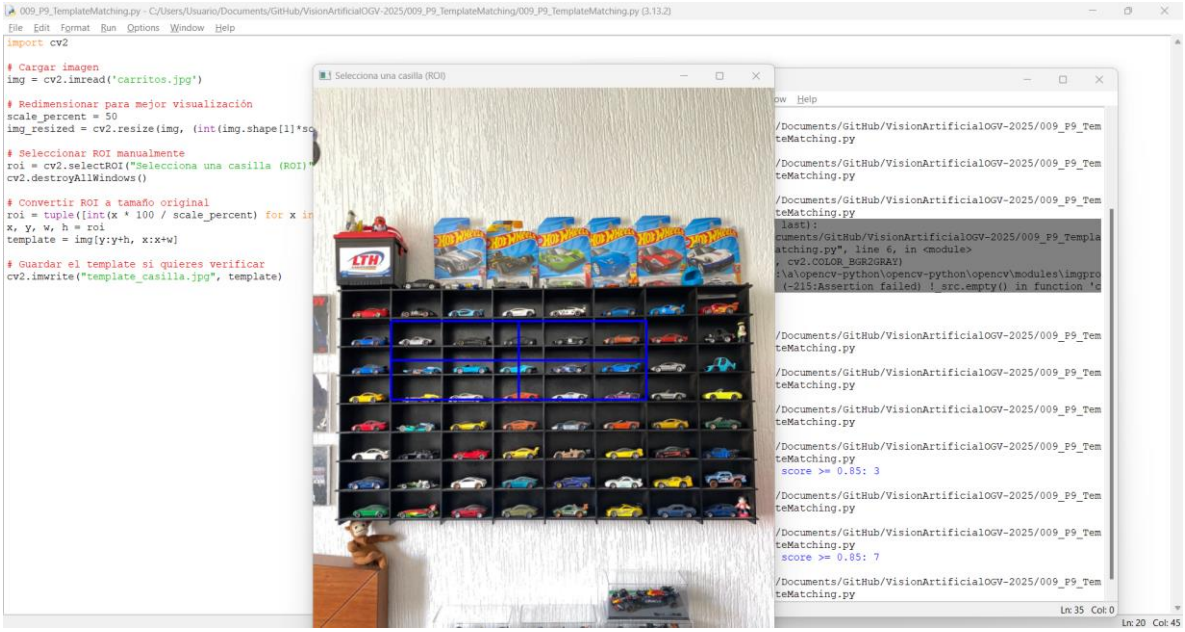
PRUEBA 1:



PRUEBA 2:

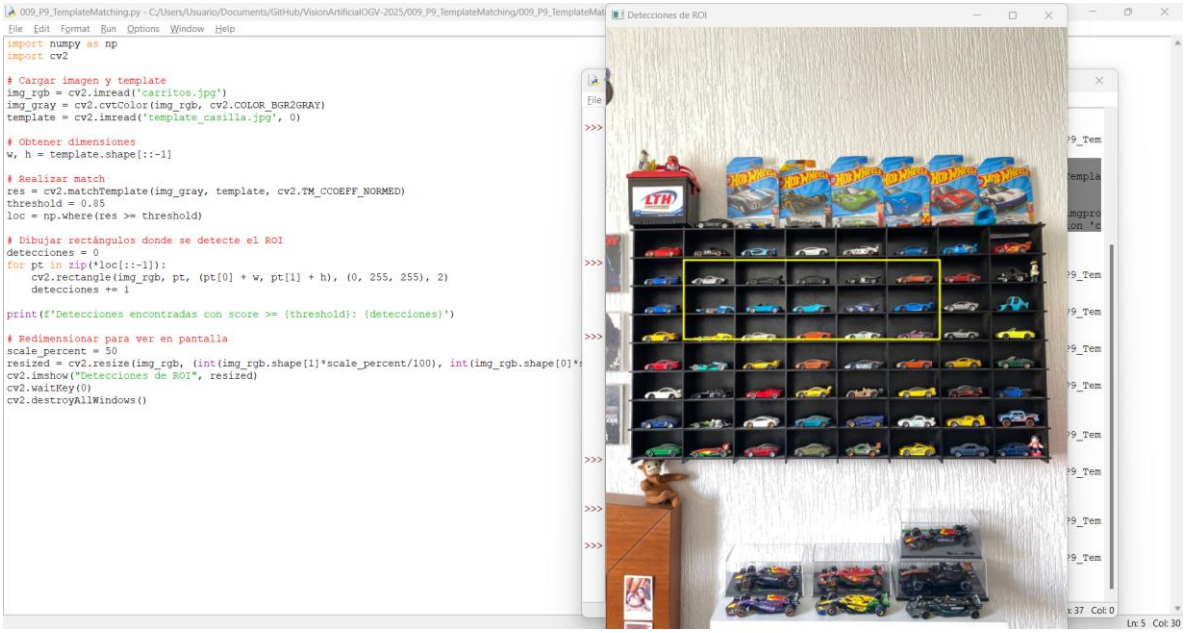


PRUEBA 3:



MI PLANTILLA DEL TEMPLATE:

PRUEBA 4:



MI CÓDIGO:

```
import numpy as np

import cv2


# Cargar imagen y template

img_rgb = cv2.imread('carritos.jpg')

img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)

template = cv2.imread('template_casilla.jpg', 0)


# Obtener dimensiones

w, h = template.shape[::-1]


# Realizar match

res = cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED)

threshold = 0.85
```

```
loc = np.where(res >= threshold)


# Dibujar rectángulos donde se detecte el ROI
detecciones = 0
for pt in zip(*loc[::-1]):
    cv2.rectangle(img_rgb, pt, (pt[0] + w, pt[1] + h), (0, 255, 255), 2)
    detecciones += 1


print(f'Detecciones encontradas con score >= {threshold}: {detecciones}')


# Redimensionar para ver en pantalla
scale_percent = 50
resized = cv2.resize(img_rgb, (int(img_rgb.shape[1]*scale_percent/100),
int(img_rgb.shape[0]*scale_percent/100)))
cv2.imshow("Detecciones de ROI", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

ENLACE DE MI REPOSITORIO DE GITHUB:

https://github.com/Ing-OscarValencia/VisionArtificialOGV-2025.git