



[Practica 7]

[Sistemas de Visión Artificial]

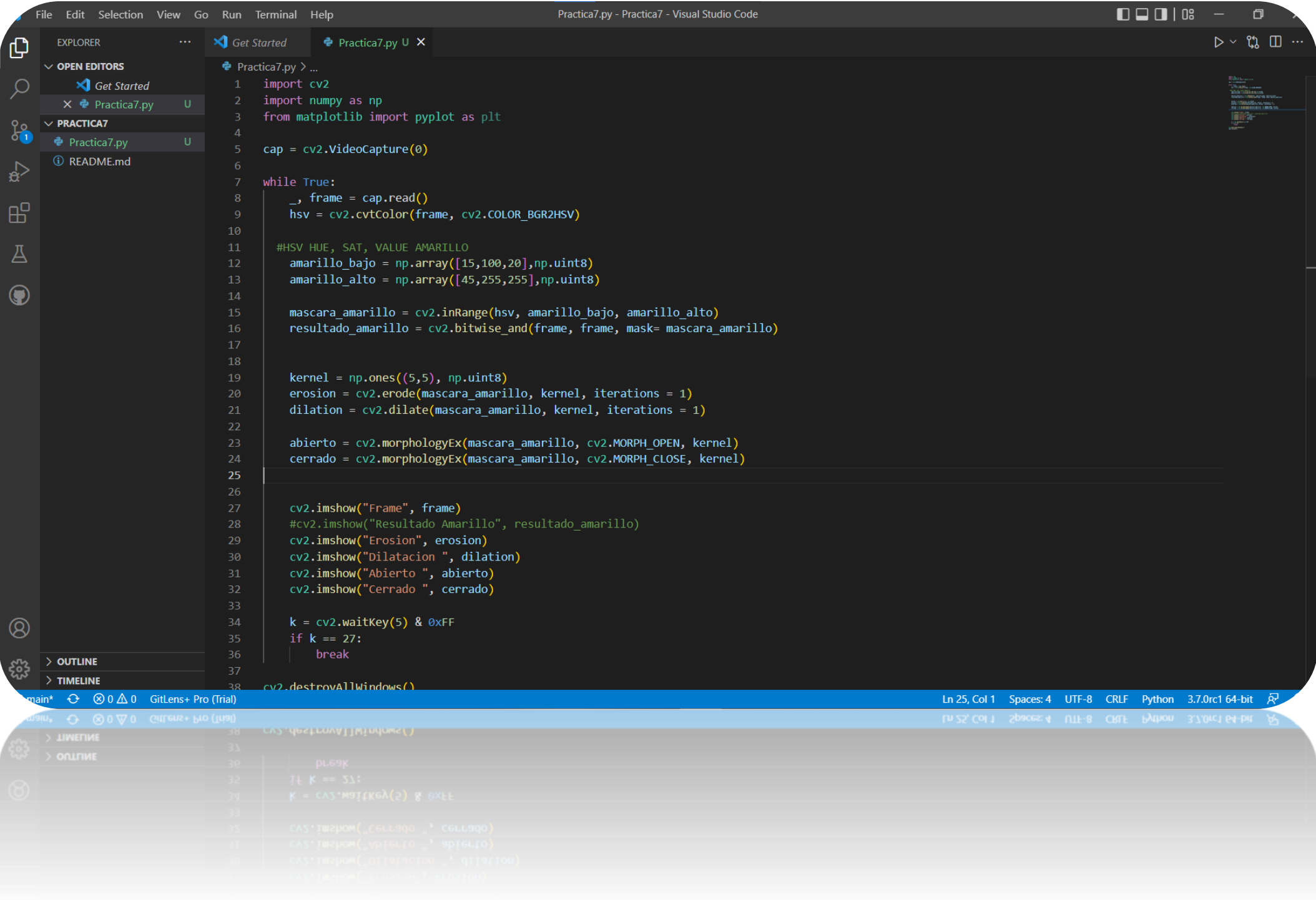
Loera Contreras Gildardo Yair – 19110230 – 7E
PROFESOR [MAURICIO ALEJANDRO CABRERA ARELLANO]

PROBLEMA

Remover Ruido – Lineal y morfológicamente en VIDEO.
Objetivo: Remover ruido de la detección F+ y F-

Falsos positivos - En el área negra lejos del objeto a detectar se ven pixeles azules porque estamos buscando el color azul
Falsos negativos - Dentro del área azul se ven pixeles negros

CODIGO



```
import cv2
import numpy as np
from matplotlib import pyplot as plt

cap = cv2.VideoCapture(0)

while True:
    _, frame = cap.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    #HSV HUE, SAT, VALUE AMARILLO
    amarillo_bajo = np.array([15,100,20],np.uint8)
    amarillo_alto = np.array([45,255,255],np.uint8)

    mascara_amarillo = cv2.inRange(hsv, amarillo_bajo, amarillo_alto)
    resultado_amarillo = cv2.bitwise_and(frame, frame, mask= mascara_amarillo)

    kernel = np.ones((5,5), np.uint8)
    erosion = cv2.erode(mascara_amarillo, kernel, iterations = 1)
    dilation = cv2.dilate(mascara_amarillo, kernel, iterations = 1)

    abierto = cv2.morphologyEx(mascara_amarillo, cv2.MORPH_OPEN, kernel)
```

```
cerrado = cv2.morphologyEx(mascara_amarillo, cv2.MORPH_CLOSE, kernel)

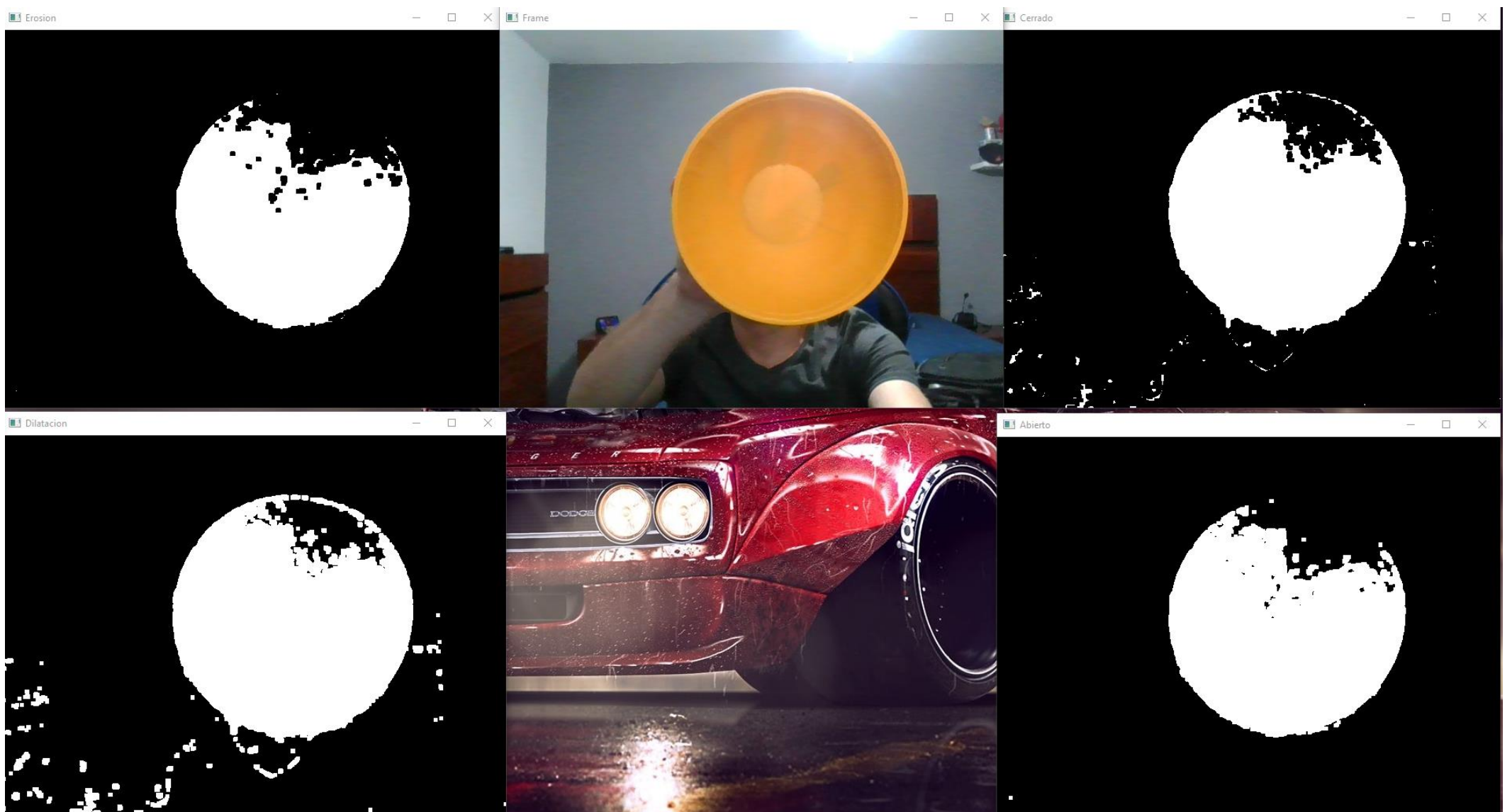
cv2.imshow("Frame", frame)
#cv2.imshow("Resultado Amarillo", resultado_amarillo)
cv2.imshow("Erosion", erosion)
cv2.imshow("Dilatacion ", dilation)
cv2.imshow("Abierto ", abierto)
cv2.imshow("Cerrado ", cerrado)

k = cv2.waitKey(5) & 0xFF
if k == 27:
    break

cv2.destroyAllWindows()
cap.release()
```

EVIDENCIAS

Al correr el programa mostramos el frame del video en tiempo real, en este caso detectando el color amarillo el cual aplicándole transformaciones morfológicas, corrosión, dilatación y después un closed y opened a la imagen para poder quitar el ruido y poder definir mejor solo lo que nos interesa de la misma, el color amarillo del video.



GITHUB:

<https://github.com/Gylc87/Practica7.git>