

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

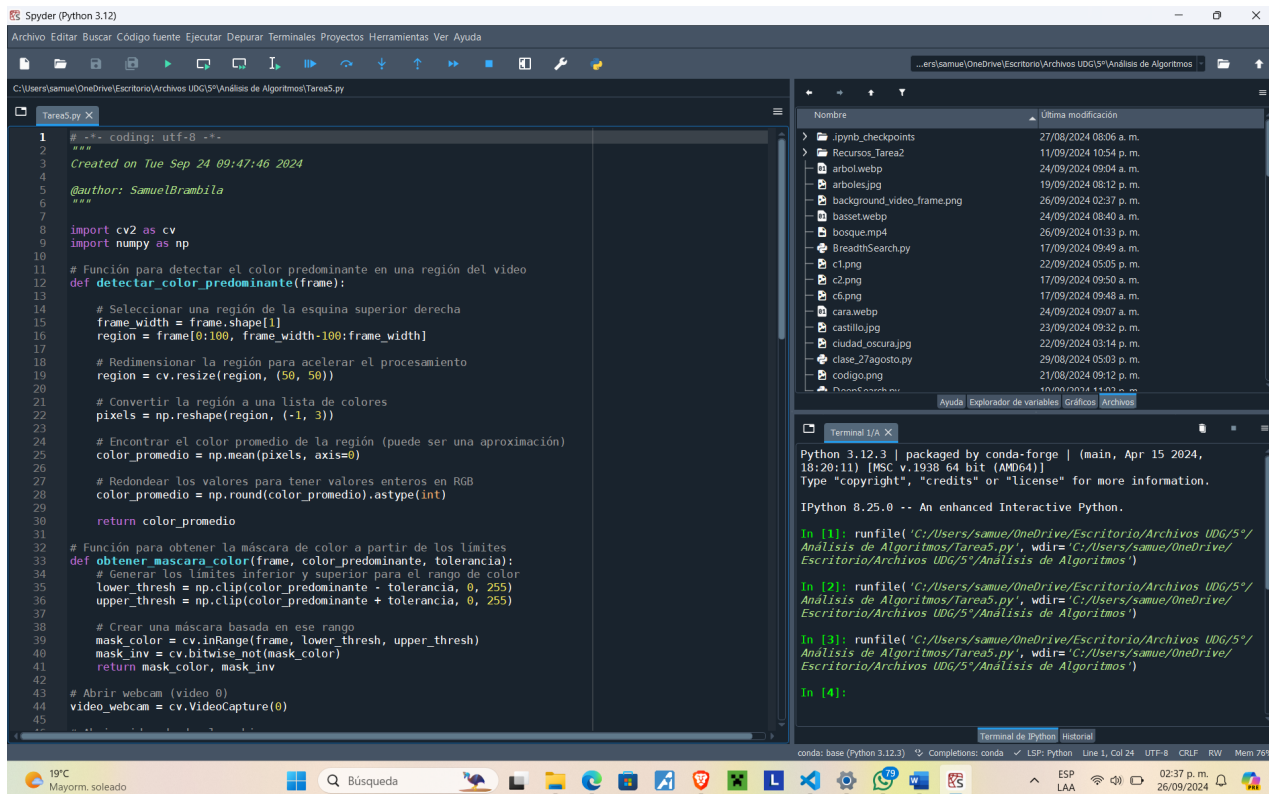
Análisis de Algoritmos

Reporte de práctica

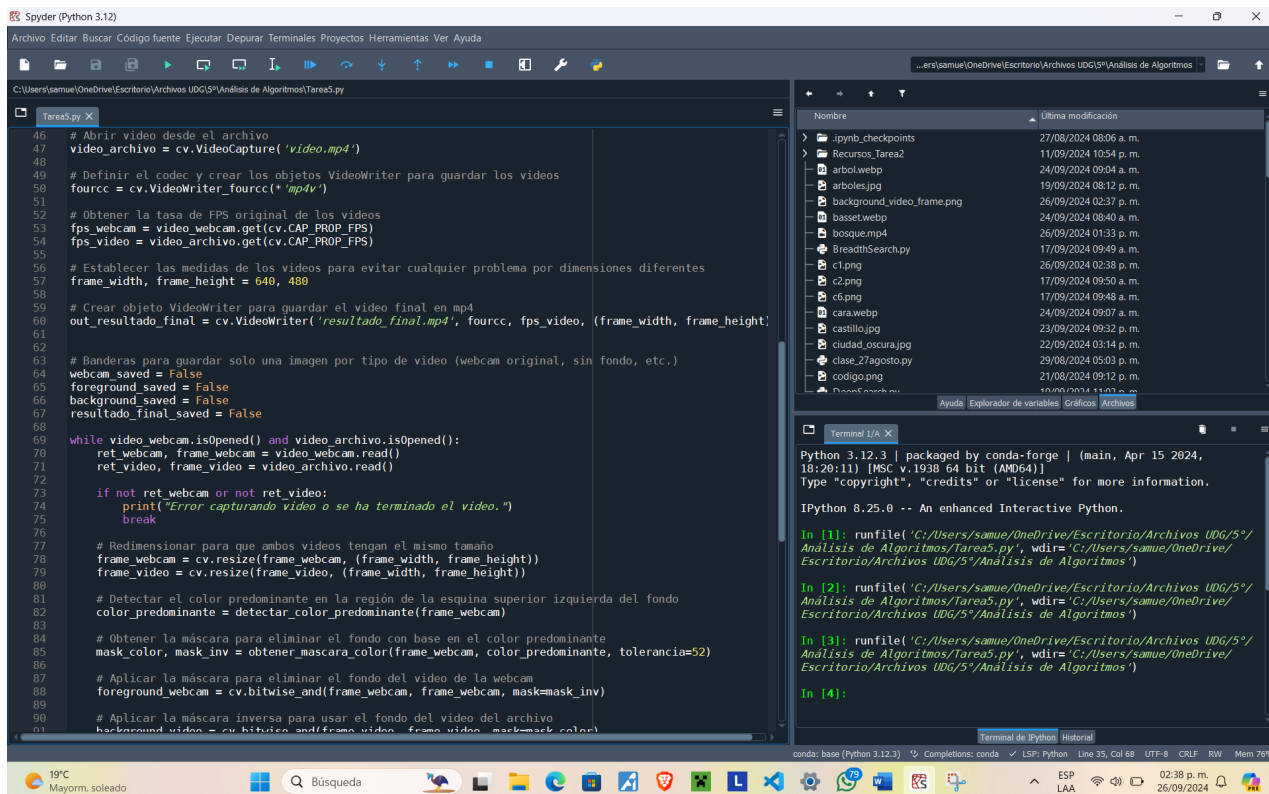
Nombre del alumno:	Samuel David Pérez Brambila
Código del alumno:	222966286
Profesor:	Erasmus Gabriel Martínez Soltero
Título de la práctica:	"Detección de color"
Fecha:	02 de Octubre de 2024

Metodología

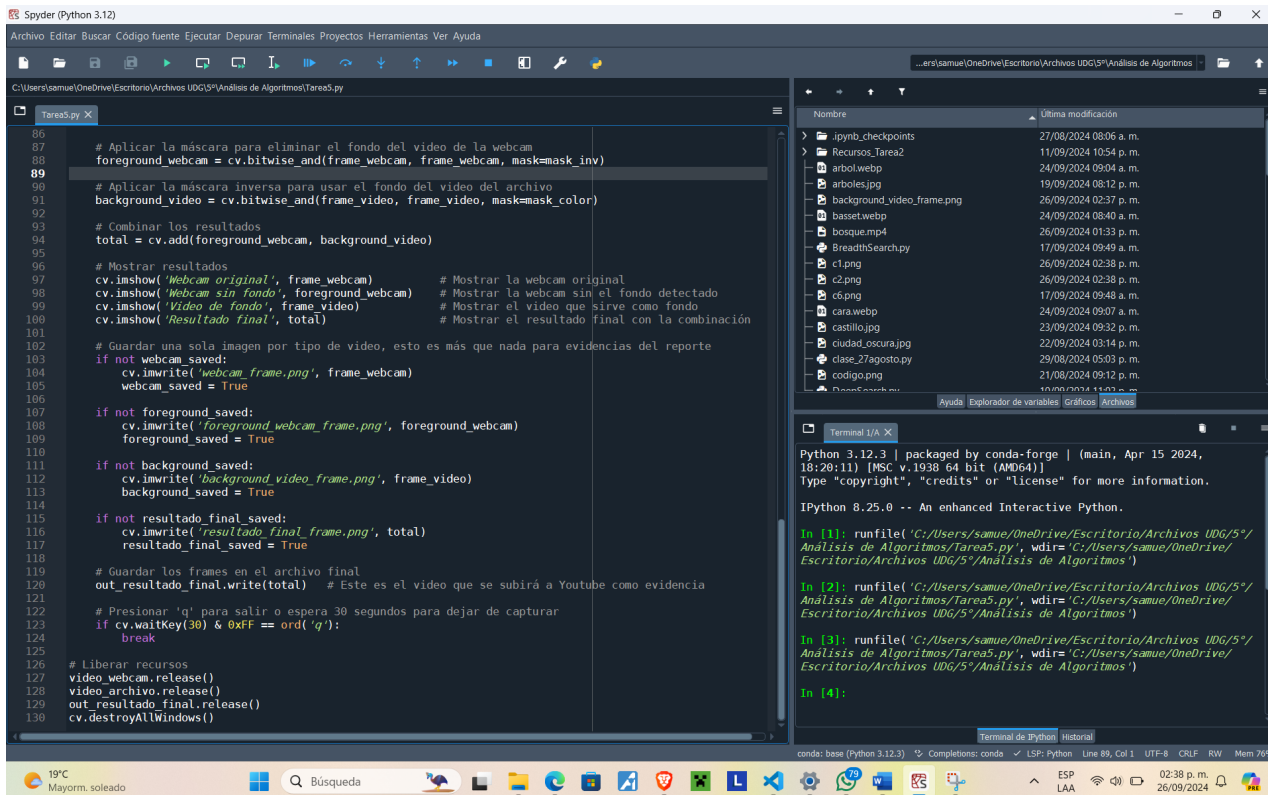
Código:



```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Sep 24 09:47:46 2024
4
5 @author: SamuelBrambila
6 """
7
8 import cv2 as cv
9 import numpy as np
10
11 # Función para detectar el color predominante en una región del video
12 def detectar_color_predominante(frame):
13
14     # Seleccionar una región de la esquina superior derecha
15     frame_width = frame.shape[1]
16     region = frame[0:100, frame_width-100:frame_width]
17
18     # Redimensionar la región para acelerar el procesamiento
19     region = cv.resize(region, (50, 50))
20
21     # Convertir la región a una lista de colores
22     pixels = np.reshape(region, (-1, 3))
23
24     # Encontrar el color promedio de la región (puede ser una aproximación)
25     color_promedio = np.mean(pixels, axis=0)
26
27     # Redondear los valores para tener valores enteros en RGB
28     color_promedio = np.round(color_promedio).astype(int)
29
30     return color_promedio
31
32 # Función para obtener la máscara de color a partir de los límites
33 def obtener_mascara_color(frame, color_predominante, tolerancia):
34     # Generar los límites inferior y superior para el rango de color
35     lower_thresh = np.clip(color_predominante - tolerancia, 0, 255)
36     upper_thresh = np.clip(color_predominante + tolerancia, 0, 255)
37
38     # Crear una máscara basada en ese rango
39     mask_color = cv.inRange(frame, lower_thresh, upper_thresh)
40     mask_inv = cv.bitwise_not(mask_color)
41     return mask_color, mask_inv
42
43 # Abrir webcam (video 0)
44 video_webcam = cv.VideoCapture(0)
```



```
46 # Abrir video desde el archivo
47 video_archivo = cv.VideoCapture('video.mp4')
48
49 # Definir el codec y crear los objetos VideoWriter para guardar los videos
50 fourcc = cv.VideoWriter_fourcc(*'mp4v')
51
52 # Obtener la tasa de FPS original de los videos
53 fps_webcam = video_webcam.get(cv.CAP_PROP_FPS)
54 fps_video = video_archivo.get(cv.CAP_PROP_FPS)
55
56 # Establecer las medidas de los videos para evitar cualquier problema por dimensiones diferentes
57 frame_width, frame_height = 640, 480
58
59 # Crear objeto VideoWriter para guardar el video final en mp4
60 out_resultado_final = cv.VideoWriter('resultado_final.mp4', fourcc, fps_video, (frame_width, frame_height))
61
62 # Banderas para guardar solo una imagen por tipo de video (webcam original, sin fondo, etc.)
63 webcam_saved = False
64 foreground_saved = False
65 background_saved = False
66 resultado_final_saved = False
67
68
69 while video_webcam.isOpened() and video_archivo.isOpened():
70     ret_webcam, frame_webcam = video_webcam.read()
71     ret_video, frame_video = video_archivo.read()
72
73     if not ret_webcam or not ret_video:
74         print("Error capturando video o se ha terminado el video.")
75         break
76
77     # Redimensionar para que ambos videos tengan el mismo tamaño
78     frame_webcam = cv.resize(frame_webcam, (frame_width, frame_height))
79     frame_video = cv.resize(frame_video, (frame_width, frame_height))
80
81     # Detectar el color predominante en la región de la esquina superior izquierda del fondo
82     color_predominante = detectar_color_predominante(frame_webcam)
83
84     # Obtener la máscara para eliminar el fondo con base en el color predominante
85     mask_color, mask_inv = obtener_mascara_color(frame_webcam, color_predominante, tolerancia=52)
86
87     # Aplicar la máscara para eliminar el fondo del video de la webcam
88     foreground_webcam = cv.bitwise_and(frame_webcam, frame_webcam, mask=mask_inv)
89
90     # Aplicar la máscara inversa para usar el fondo del video del archivo
91     background_video = cv.bitwise_and(frame_video, frame_video, mask=mask_color)
```



Resultados

Enlace a video: <https://youtu.be/JXUOuDc1tQ>

Imagen webcam



Imagen webcam sin fondo

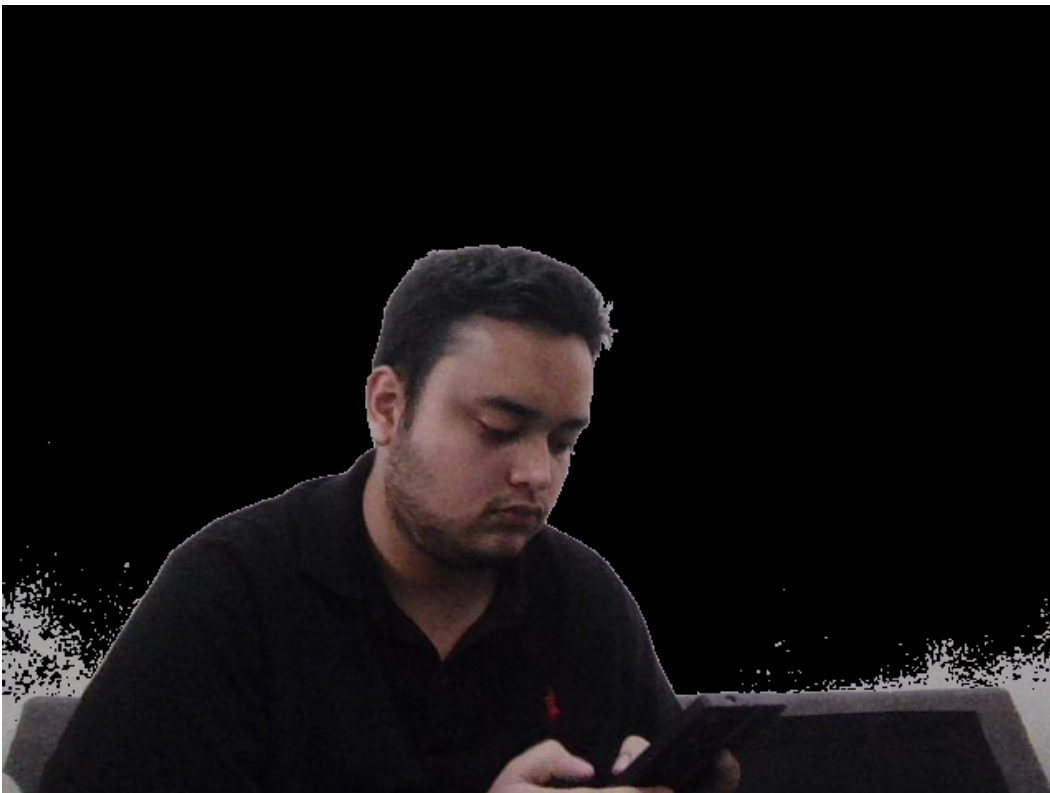


Imagen de video de fondo



Imagen resultado final

