



[Proyecto - Detector de objetos (teoría Haar Cascade)]

[Sistemas de Visión Artificial]

PROBLEMA

1.- Hacer un detector de objetos (teoría haarcascade) pero utilizando el método que prefieran.

CODIGO

```
Proyecto.py - E:\Personal\CETI\7mo Semestre\Vision Artificial\GitHub\Proyecto\Proyecto.py (3.7.0rc1)
File Edit Format Run Options Window Help
from itertools import count
from tkinter import Frame
import cv2
import numpy
import imutils
import os

Datos = 'n'
if not os.path.exists(Datos):
    print("Carpeta creada: ", Datos)
    os.mkdir(Datos)

cap = cv2.VideoCapture(0)

x1, y1 = 190, 80
x2, y2 = 450, 398

count = 0

while True:

    ret, frame = cap.read()
    if ret == False: break
    imAux = frame.copy()
    cv2.rectangle(frame, (x1,y1),(x2,y2), (255,0,0), 2)

    objeto = imAux[y1:y2,x1:x2]
    objeto = imutils.resize(objeto, width=38)

    k = cv2.waitKey(1)
    if k == 27:
        break

    if k == ord('s'):
        cv2.imwrite(Datos+'objeto_{}.jpg'.format(count),objeto)
        print('Imagen almacenada: ', 'objeto_{}.jpg'.format(count))
        count = count + 1

    cv2.imshow('frame', frame )
    cv2.imshow('Objeto', objeto)

    k = cv2.waitKey(1)
    if k == 27:
        break

cap.release()
```

```
DetectorHaarCascade.py - E:\Personal\CETI\7mo Semestre\Vision Artificial\GitHub\Proyecto\DetectorHaarCascade.py (3.7.0rc1)
File Edit Format Run Options Window Help
import cv2

cap = cv2.VideoCapture(0)

cara = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

if cv2.__version__.startswith('2.4'):
    dmf_flag = cv2.cv.CV_HAAR_SCALE_IMAGE
else:
    dmf_flag = cv2.CASCADE_SCALE_IMAGE

while True:

    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = cara.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),|
        flags=dmf_flag
    )

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x,y),(x+w,y+h), (0,255,0),2)
        cv2.putText(frame, 'Cara', (x,y-10), 2, 0.7, (0,255,0),2,cv2.LINE_AA)

    cv2.imshow('frame', frame)

    if cv2.waitKey(1) == 27:
        break

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

```

from itertools import count
from tkinter import Frame
import cv2
import numpy
import imutils
import os

Datos = 'n'
if not os.path.exists(Datos):
    print("Carpeta creada: ", Datos)
    os.mkdir(Datos)

cap = cv2.VideoCapture(0)

x1, y1 = 190, 80
x2, y2 = 450, 398

count = 0

while True:

    ret, frame = cap.read()
    if ret == False: break
    imAux = frame.copy()
    cv2.rectangle(frame, (x1,y1),(x2,y2), (255,0,0), 2)

    objeto = imAux[y1:y2,x1:x2]
    objeto = imutils.resize(objeto, width=38)

    k = cv2.waitKey(1)
    if k == 27:
        break

    if k == ord('s'):
        cv2.imwrite(Datos+'/objeto_{}.jpg'.format(count),objeto)
        print('Imagen almacenada: ', 'objeto_{}.jpg'.format(count))
        count = count + 1

    cv2.imshow('frame', frame )
    cv2.imshow('Objeto', objeto)

    k = cv2.waitKey(1)
    if k == 27:
        break

cap.release()
cv2.destroyAllWindows()

```

```

import cv2

cap = cv2.VideoCapture(0)

cara = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

if cv2.__version__.startswith('2.4'):
    dmflag = cv2.cv.CV_HAAR_SCALE_IMAGE
else:
    dmflag = cv2.CASCADE_SCALE_IMAGE

while True:

    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = cara.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags=dmflag
    )

    for (x, y, w , h) in faces:
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        cv2.putText(frame, 'Cara', (x,y-10), 2 ,0.7,(0,255,0),2,cv2.LINE_AA)

    cv2.imshow('frame', frame)

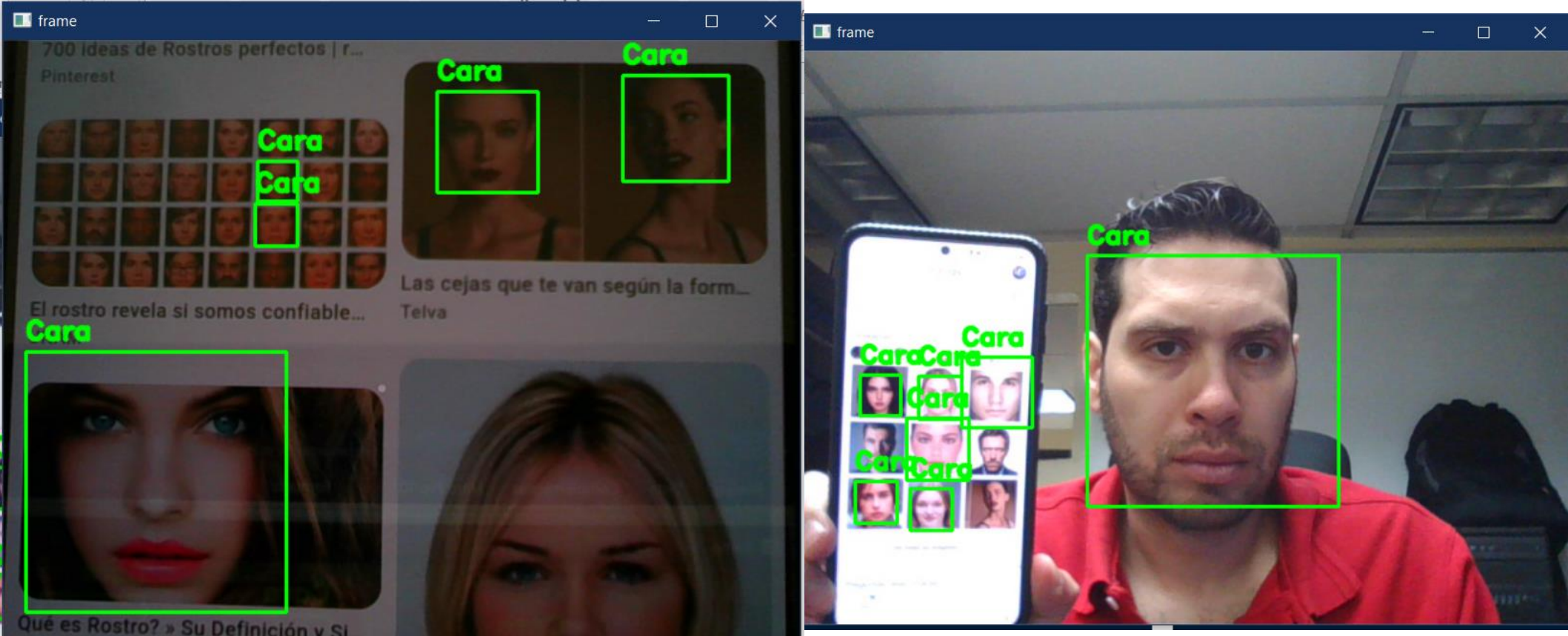
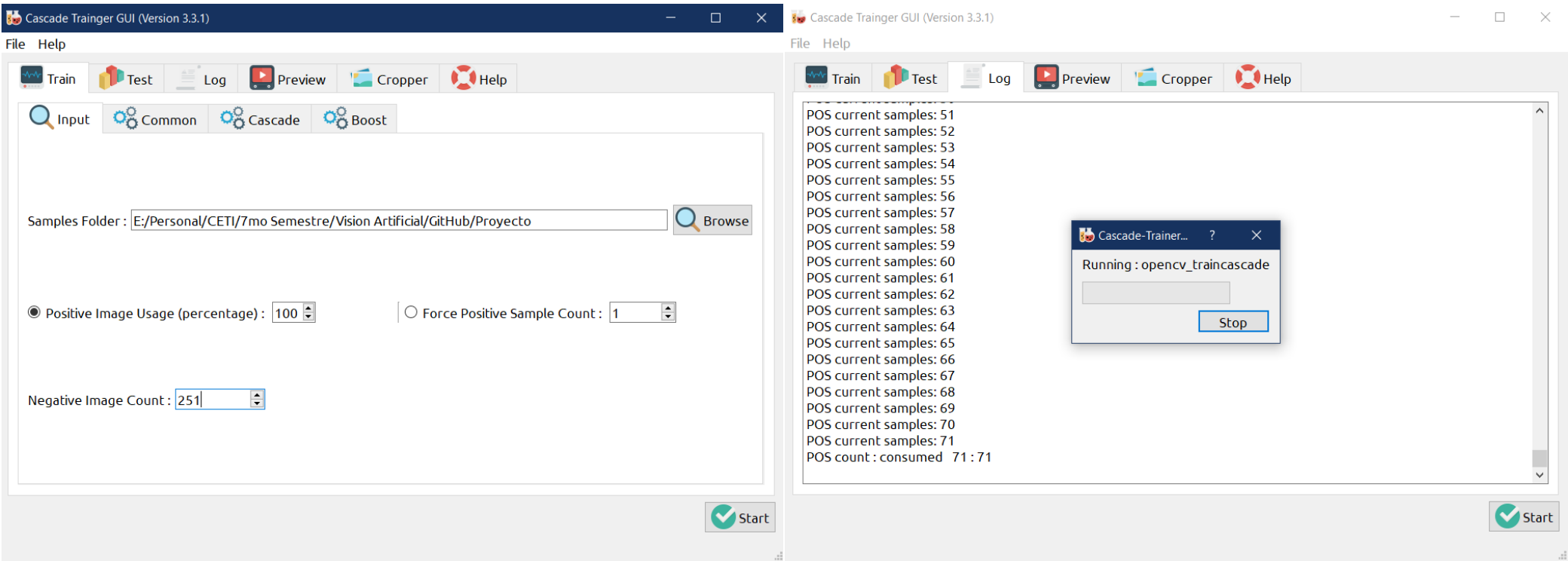
    if cv2.waitKey(1) == 27:
        break

```

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

EVIDENCIAS

Para este proyecto intente hacer mi propio detector de objetos usando Haar Cascade el cual consistía en detectar un funko pop de Dustin de la serie Stranger Things, tome capturas (fotos) tanto positivas como negativas e hice mi entrenamiento usando Cascade-Trainer-GUI no logre que lo reconociera al 100% por lo que creo que me falto realizar más capturas p y n, después realice detección de rostro usando el entrenamiento pre entrenamiento que se encuentra disponible en <https://github.com/opencv/opencv/tree/master/data/haarcascades> logrando la detección del rostro en un 95%.



GITHUB:
<https://github.com/Gylc87/Proyecto.git>
YOUTUBE
<https://youtu.be/QAu2j7Ok-g>

