

ITFUSION SAS

Informe de Construcción de Extracción del Texto la Foto de una Placa de un Vehículo

Fecha de realización: 25/07/2023

1. Instalación librerías

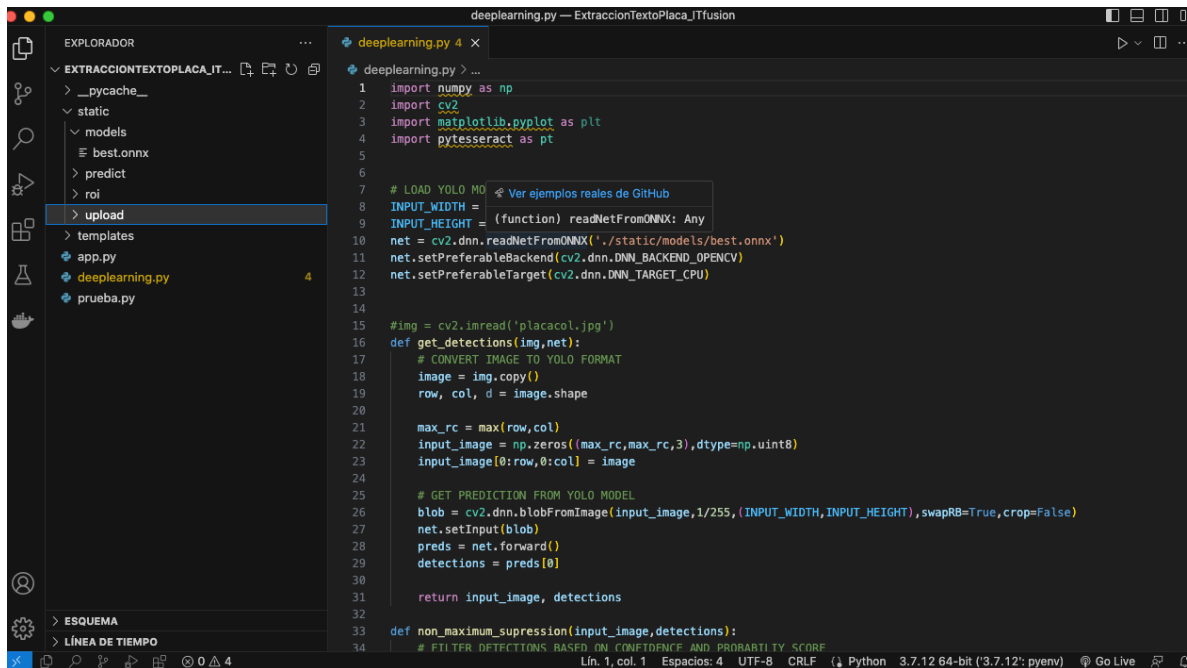
- En la siguiente figura se observa la instalación de las diferentes librerías requeridas para correr el programa que lee la una foto de una placa de un vehículo para extraer el texto y almacenarla en la base de datos:

```
...x0 LaunchInstanceID=7132DDE7-6036-47A9-A60E-DACB80AA799C ...endadoprofesor/Module_5_WebApp.yolo -- python - python app.py ...ubuntu@ec2-3-17-246-145.us-east-2.compute.amazonaws.com:~$  
ubuntu@www:~/backendsindocker$ python3 --version  
Python 3.8.10  
ubuntu@www:~/backendsindocker$ python3 -c "import cv2; print(cv2.__version__)"  
4.8.0  
ubuntu@www:~/backendsindocker$ python3 -c "import numpy; print(numpy.__version__)"  
1.24.3  
ubuntu@www:~/backendsindocker$ tesseract --version  
tesseract 4.1.1  
leptonica-1.79.0  
libgif 5.1.4 : libjpeg 8d (libjpeg-turbo 2.0.3) : libpng 1.6.37 : libtiff 4.1.0 : zlib 1.2.11 : libwebp 0.6.1 : libopenjp2 2.3.1  
Found AVX2  
Found AVX  
Found FMA  
Found SSE  
Found libarchive 3.4.0 zlib/1.2.11 liblzma/5.2.4 bz2lib/1.0.8 liblz4/1.9.2 libzstd/1.4.4  
ubuntu@www:~/backendsindocker$ python3 -c "import matplotlib; print(matplotlib.__version__)"  
3.7.2  
ubuntu@www:~/backendsindocker$
```

Estas librerías fueron instaladas en los servidores Linux en Ubuntu 20.02 en Amazon requeridos para el proyecto.

2. Programa en python para extracción del texto de la foto de la placa

- En la siguiente figura se puede muestra el ambiente de desarrollo y programa en Python versión inicial que extrae el texto de la foto de una placa de un vehículo:

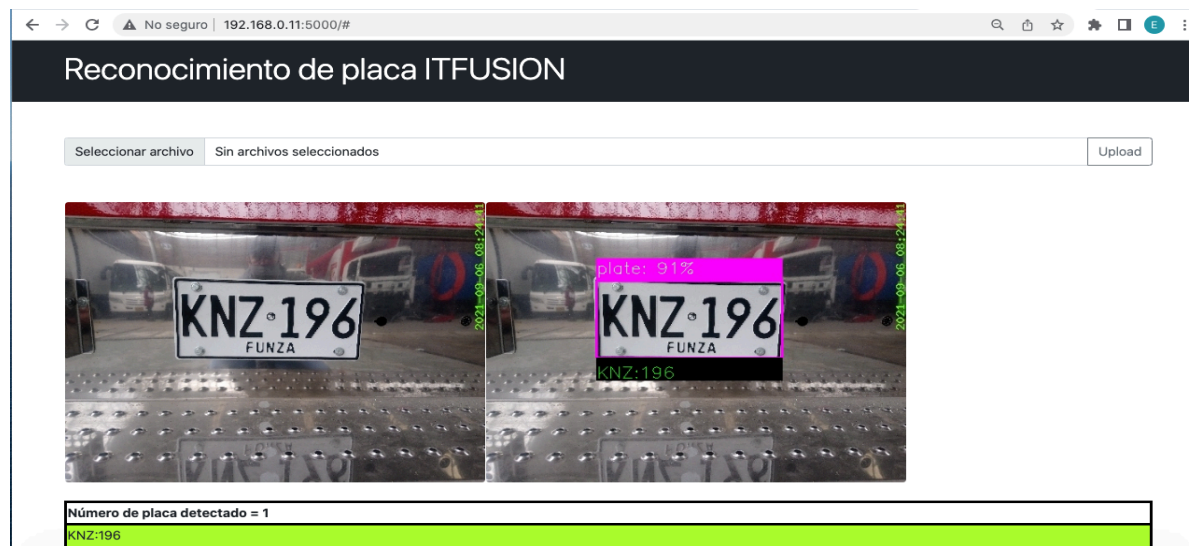


```
1 import numpy as np
2 import cv2
3 import matplotlib.pyplot as plt
4 import pytesseract as pt
5
6
7 # LOAD YOLO MO
8 INPUT_WIDTH =
9 INPUT_HEIGHT =
10 net = cv2.dnn.readNetFromONNX('./static/models/best.onnx')
11 net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
12 net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)
13
14
15 #img = cv2.imread('placacol.jpg')
16 def get_detections(img,net):
17     # CONVERT IMAGE TO YOLO FORMAT
18     image = img.copy()
19     row, col, d = image.shape
20
21     max_rc = max(row,col)
22     input_image = np.zeros((max_rc,max_rc,3),dtype=np.uint8)
23     input_image[0:row,0:col] = image
24
25     # GET PREDICTION FROM YOLO MODEL
26     blob = cv2.dnn.blobFromImage(input_image,1/255,(INPUT_WIDTH,INPUT_HEIGHT),swapRB=True,crop=False)
27     net.setInput(blob)
28     preds = net.forward()
29     detections = preds[0]
30
31     return input_image, detections
32
33 def non_maximum_supression(input_image,detections):
34     # FILTER DETECTIONS BASED ON CONFIDENCE AND PROBABILITY SCORE
```

NOTA:En una versión posterior de este proyecto se complementará la integración con la foto enviada por cada conductor al iniciar y finalizar cada viaje, y guardar el texto de la placa en la base de datos.

3. Ejecución del programa

- En la siguiente figura se puede observar una interfaz web de pruebas donde se selecciona de manera manual una foto de una placa de un vehículo y se muestra en pantalla el texto de la placa extraída, si la encontró:



Proyecto SENA/Minciencias - Itfusion. Documento de construcción de Extracción del Texto de una Placa de un Vehículo

- En la siguiente figura se muestra la ejecución del programa en Python de manera manual, retornando el texto de la placa por línea de comandos desde el servidor Amazon del proyecto:

```
...D=7132DDE7-6036-47A9-A60E-DACB80AA799C | ...dule_5_WebApp_yolo — python • python app.py | ...246-145.us-east-2.compute.amazonaws.com
[ubuntu@www:~/softwareML/extracciontextoplacaITF$ python3 deeplearning.py
Número de placa adecuada encontrada: GZH-386
Numero de placa original leida por algoritmo:GZH386
ubuntu@www:~/softwareML/extracciontextoplacaITF$ █
```

NOTA: Posteriormente haremos la integración de este programa con el envío de la foto desde la app móvil y guardar el texto encontrado en la base de datos.