Zense Recruitment 2021

OBJECT DETECTION

-Ankrutee Arora
IMT2020034

Origin of this project:

I have always been interested in exploring Machine Learning and its real-life applications. OpenCV provides real-time optimized computer vision tools and also supports Machine Learning models. So, I decided to start with the basics of ML and OpenCV.

One of the most exciting and intriguing projects that I found in OpenCV is Object detection. The amalgamation of ML and computer vision is a boon in today's world and is in function in multiple domains.

Proposed Idea:

This is a simple implementation as it uses DNN module of open cv and using only two libraries: numpy,cv2. I used Mobile Net SSD (Single Shot Detector), which has been trained on the MS COCO dataset using the TensorFlow deep learning framework.

This project can identify up to 80 different classes of objects ranging from a toothbrush to a person to a car.

This project has uncountable applications , for example, google lens, surveillance and security, traffic monitoring, video communication, robot vision and animation.

Tech Stack:

cv2 : This is a OpenCV library which is used to process the video input and classify the objects.

Numpy: OpenCV makes use of Numpy library which is a highly optimized library for numerical operations and all the OpenCV array structures are converted to and from Numpy arrays.

coco.names : This is a data set which contains names of 80 different objects.

configpath: This uses 'sd_mobilenet_v3_large_coco_2020_01_14.pbtxt' and SSD MobilenetV3 is trained over coco dataset using Tensorflow API. It can detect any single class from the classes provided by coco dataset.

weightsPath : 'frozen_inference_graph.pb' , is a frozen graph that cannot be trained anymore, it defines the graphdef and is actually a serialized graph .

DNN: Using the OpenCV DNN module, we can implement Object Detection in deep learning and computer vision. Moreover, SSD models are generally faster when compared to other object detection models and the MobileNet backbone makes them less compute-intensive.

Future Aspects:

1. I wanted to make it a user-friendly software which can be used with the help of one click. Moreover, this code only works for identifying certain objects so, I can increase the dataset size.
2. This project, as of now gives about 50-95% accuracy, it still doesn't classify the objects accurately.
   In order to make it more accurate, I want to train this model using refined parameters along with advanced concepts of ML.
3. This project can also be enhanced by adding Text detection, which will detect the text on the object along with the Object class.
4. By making few changes, this model can also be used to identify human facial expressions.

References:

Google, YouTube ,StackOverflow and mainly OpenCV documentation have been of great help.