TOPIC OF CURRENT INTEREST – II

Deep Learning

“Object Detection Using YOLO algorithm”

Group Members :

Muhammad Ahmer Khan B18158029

Muhammad Khubaib Irfan B18158034

Topic

Object Detection by YOLO v3, YOLO v5 Algorithm with the help of image dataset for Predicting the vehicles on the road.

Abstract

YOLO is an algorithm that uses neural networks to provide real-time object detection. YOLO ("you only look once") is a popular algorithm because it achieves high accuracy while also being able to run in real-time, almost clocking 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the map of other real-time detectors. This algorithm "only looks once" at the image in the sense that it requires only one forward propagation pass through the network to make predictions.

Content

The dataset contains media of cars in all views, and we apply the yolo v3, v5 algorithm to detect them.

Dataset

train_solution_bounding_boxes.csv

sample_submission.csv

Folder Path

```
'/content/drive/MyDrive/data/train_solution_bounding_boxes (1).csv'
```

EVALUATION & EXPLAINATION

Importing all necessary libraries:

```
import os, time, random
import numpy as np
import pandas as pd
import cv2, torch
from tqdm.auto import tqdm
import shutil as sh
```

Read the dataset and mounted in Google Drive with Colab:

```
df = pd.read_csv('/content/drive/MyDrive/data/train_solution_bounding_boxes (1).csv')
```

Display first five rows to check the dataset:

```
print(df.head())
```

```
      image_id        xmin        ymin        xmax        ymax
0   vid_4_1000  281.259045  187.035071  327.727931  223.225547
1  vid_4_10000   15.163531  187.035071  120.329957  236.430180
2  vid_4_10040  239.192475  176.764801  361.968162  236.430180
3  vid_4_10020  496.483358  172.363256  630.020260  231.539575
4  vid_4_10060   16.630970  186.546010  132.558611  238.386422
```

Because we are using a public repo, so we clone it from the links:

```
!git clone https://github.com/ultralytics/yolov5  # clone repo
!pip install -U pycocotools
!pip install -qr yolov5/requirements.txt  # install dependencies
!cp yolov5/requirements.txt ./
```

For labelling rows and columns in training_images folder we apply logic:

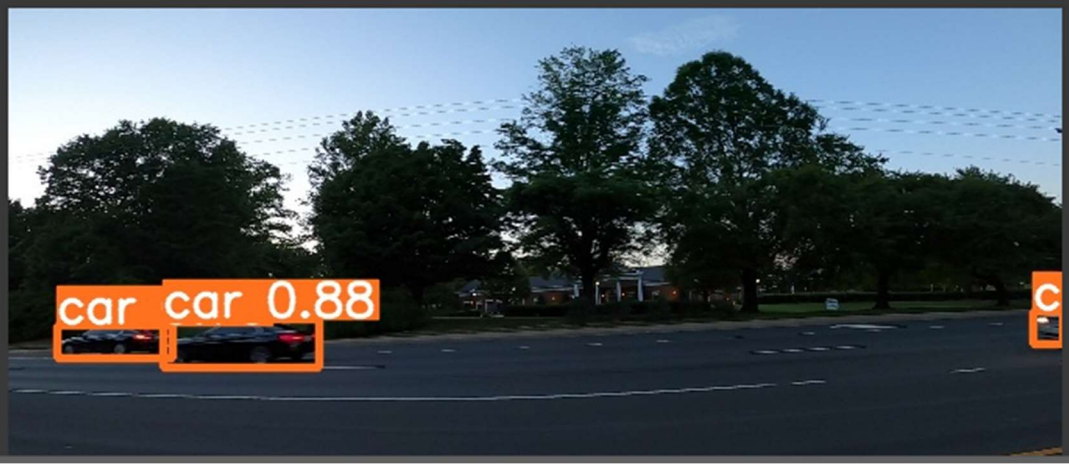
```
Add text cell
source = 'training_images'
if True:
    for fold in [0]:
        val_index = index[len(index)*fold//5:len(index)*(fold+1)//5]
        for name,mini in tqdm(df.groupby('image_id')):
            if name in val_index:
                path2save = 'val2017/'
            else:
                path2save = 'train2017/'
            if not os.path.exists('/tmp/convertor/fold{}/labels/'.format(fold)+path2save):
                os.makedirs('/tmp/convertor/fold{}/labels/'.format(fold)+path2save)
            with open('/tmp/convertor/fold{}/labels/'.format(fold)+path2save+name+".txt", 'w+
                row = mini[['classes','x_center','y_center','w','h']].astype(float).values
                row = row.astype(str)
                for j in range(len(row)):
                    text = ' '.join(row[j])
                    f.write(text)
                    f.write("\n")
            if not os.path.exists('/tmp/convertor/fold{}/images/{}'.format(fold,path2save)):
                os.makedirs('/tmp/convertor/fold{}/images/{}'.format(fold,path2save))
            sh.copy("/content/drive/MyDrive/data/{}/{}.jpg".format(source,name),'/tmp/convert

100% 355/355 [00:01<00:00, 191.77it/s]
```

Applying The YoloV5 algorithm to detect vehicles:

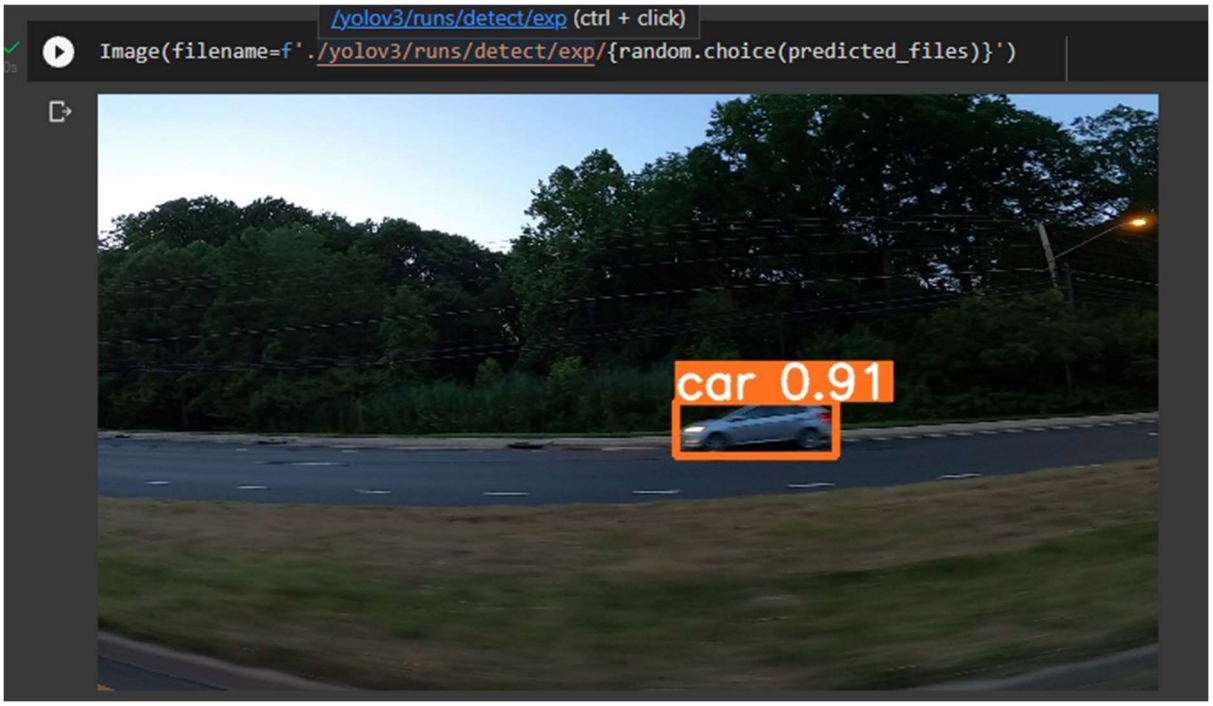
```
predicted_files = []
for (dirpath, dirnames, filenames) in os.walk("./yolov5/runs/detect/exp"):
    predicted_files.extend(filenames)
```

```
Image(filename=f'./yolov5/runs/detect/exp/{random.choice(predicted_files)}')
```



RESULT

Using random image selector for random images:



By using YOLO V5 Algorithm we can predict that a particular vehicle is car and its accuracy is 0.91 percent.