Project Report

**Number Plate Detection using YOLOv7 in Python**

Submitted By:

**Parthivi Singh**

3rd Year, Sem V, BTech IT(DS),

Ajeenkya D Y Patil University

**Abstract**

This project aims to develop a real time number plate detection system utilizing the YOLOv7 (You only look once) algorithm. The system will be designed to detect and recognize the license plates from images captured by surveillance cameras and cctv footages.

Leveraging deep learning techniques, the proposed system will contribute to enhancing traffic monitoring and surveillance by efficiently identifying vehicles and extracting their license plate information. The implementation of such a system holds significant potential for improving law enforcement, traffic management, and security measure.

**Contents:**

1. Objectives of the project
2. Need for the project
3. About YOLOv7
4. Source Code

**Objective**

The main objective of this project is to create a real time number plate detection system using algorithm YOLOv7.

This will help in traffic monitoring and will help identify various vehicle’s license numbers just by their images.

These images can be from any source like surveillance cameras or CCTV footages.

This project can identify the license plates.

**Need for this Project:**

There are many ways that this project can be used and it can be pretty useful.

Majority technologies being used in toll plazas for receiving toll from vehicles can use this model for identifying the Number Plate.

The Law officers can also use this project to extract license plates just by looking at the surveillance cameras and CCTV footages.

This project may come in handy when you want to look at license plates deeply and extract information and that too without doing it manually.

**About YOLOv7:**

The YOLO v7 algorithm achieves the highest accuracy among all other real-time object detection models – while achieving 30 FPS or higher using a GPU V100.

Compared to the best-performing Cascade-Mask R-CNN models, YOLOv7 achieves 2% higher accuracy at a dramatically increased inference speed (509% faster).

Source Code for the project

1. Cloning important repositories and important file installation

!git clone https://github.com/WongKinYiu/yolov7.git

%cd yolov7

!pip install -r requirements.txt

1. Adding our Custom Dataset which is uploaded onto a git repository

%cd /content

!curl -L "https://github.com/parthivi-project/Project-Work/raw/main/data\_yolov7.zip" > data\_yolov7.zip; unzip data\_yolov7.zip; rm data\_yolov7.zip

1. Preparing image path files
2. import os
3. train\_img\_path = "/content/images/train"
4. val\_img\_path = "/content/images/val"
5. %cd /content
6. #Training Images
7. with open('train.txt', "a+") as f:
8. img\_list = os.listdir(train\_img\_path)
9. for img in img\_list:
10. f.write(os.path.join(train\_img\_path,img+'\n'))
11. print("Done")
12. #Validation Images
13. with open('val.txt',"a+") as f:
14. img\_list = os.listdir(val\_img\_path)
15. for img in img\_list:
16. f.write(os.path.join(val\_img\_path,img+'\n'))
17. print("Done")

4) Creating a copy of an internal file

%cp /content/yolov7/data/coco.yaml /content/yolov7/data/custom.yaml

1. Downloading Pretrained Weights

%cd /content/yolov7

!wget "https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt"

1. Training our model

!python train.py --batch 16 --cfg /content/yolov7/cfg/training/custom\_yolov7.yaml --epochs 100 --data /content/yolov7/data/custom.yaml --weights 'yolov7.pt' --device 0

1. Inference and testing

#Run

!python detect.py --weights /content/yolov7/runs/train/exp/weights/best.pt --source /content/yolov7/10.png

#Displaying

import glob

from IPython.display import Image, display\_html

i = 0

limit = 10000 #max images to print

#assuming png

for imageName in glob.glob('/content/yolov7/runs/detect/exp4/\*.png'):

  if i < limit:

    display(Image(filename=imageName))

    print("\n")

  i = i + 1

**Note: The output of each and every code is properly displayed in the recordings provided.**

***Thank You!***