Digital Image Processing - Project
License Plate Detection using YOLOv4

mdesai3@buffalo.edu
ssonawan@buffalo.edu

9th December 2021

Problem Statement

License Plate Detection is a key area of research in the Image Processing field, the technology aims to identify different vehicles based on their license plate numbers. Our goal is to create such a program implementing the YOLOv4 model. License plate detection aims to solve the problem of keeping a track of the ever-increasing number of vehicles for traffic monitoring purposes.

Motivation

As the development of highway infrastructure, motor industry, rise in sales of vehicles, takes place, the bigger the problem of managing rule breaks such as speeding, reckless driving, hit and runs, disobeying traffic laws, stolen cars, parking nuisance, etc. gets.

Managing these infractions become a vital task to increase the safety and well being of people. This project helps in micro-managing the infractions by identifying their license plates which can then be used to identify the people behind them.

Literature Review

In the research paper “YOLOv4: Optimal Speed and Accuracy of Object Detection” conducted by Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao [1], they implemented the YOLOv4 model for object detection on the COCO dataset and conducted a study about its performance with other pre-existing Object Detection Models.

Algorithm

We are implementing the YOLOv4 (You Only Look Once) model for our topic. YOLOv4 is a single-stage object detection model built upon YOLOv3 and its other previous versions. This algorithm uses the concepts of BoF(Bag of Freebies) and BoS(Bag of Specials).

BoF techniques enhance the accuracy without any computation cost e.g. Data Augmentation. BoS techniques are used for pre-processing operations which greatly increase the accuracy but incur small computation costs e.g. Spatial Pyramind Pooling.

In contrast to other object detection models(ODM) which apply classifiers, masks to the image at multiple locations, YOLO uses a single neural network to process the entire image. Some characteristics of YOLOv4 are newer normalization techniques like Cross mini-Batch Normalization (CmBN), Self-Adversarial Training for data augmentation which works in 2 forward-backwards stages, etc,.

YOLOv4 performs faster and is more accurate compared to other ODMs like Learning Rich Features(LRF) single-shot object detector, RFBNet (Receptive Field Block Net), RetinaNet, SSD (Single Shot Multibox Detector). [1]

We first collected our dataset then trained it over the YOLOv4 model creating custom weights for testing. We then applied functions[2] for the detection and cropping of the license plates from video samples.

Dataset

We have used a collection of images from Open Images Dataset and other images from Google Images. The dataset contained 3000 images as the training dataset and 365 images as the testing dataset. The dataset contained images of various license plates under different conditions of light, clarity, size. We will be testing an additional video of a moving car for plate detection.

Evaluation Metrics

Evaluation metrics such as precision, recall which can be defined as the ratio of correct positive predictions over all recorded predictions and the percentage of correct positive prediction over all actual positive cases.[6]

mAP(Mean Average Precision) is the area under the precision and recall curve(PR curve). This mAP value is calculated for the whole area, 50%, 75%. These values are known as IoU thresholds which stands for Intersection over Union i.e. Area of Overlap / Area of Union. [7]

Results & Analysis

Evaluation metrics of our trained model are as follows,

Precision = 0.60%

Recall = 0.77%

F1-score = 0.68%

mAP@0.50 (AP50) = 68.958% ~ 68.96%

An input video sample is passed to the model which outputs a video with a bounding box on license plates detected. These detected License Plates at every 50 frames are cropped and saved as .png images. The cropped images were enhanced and OCR was performed on the cropped images. We will not get the OCR output if the quality of the image extracted from the video is of very low quality.

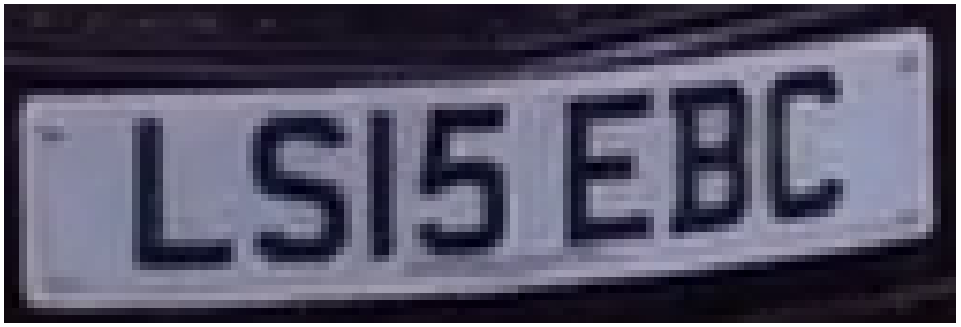

Fig. 1 License Plate 1 at frame 50 (good quality image)

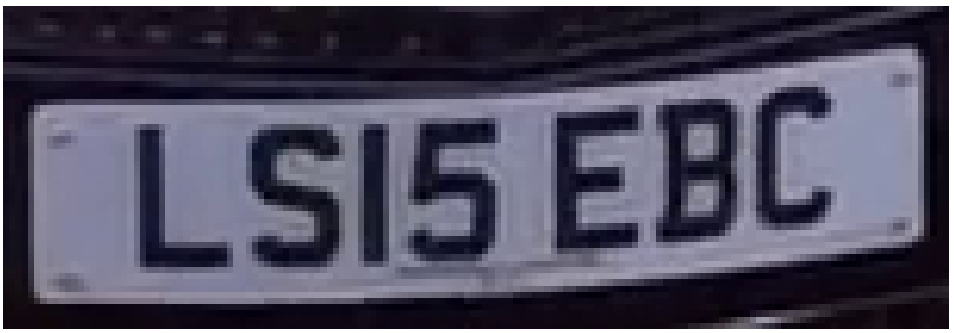

Fig. 2 License Plate 1 at frame 100 (good quality image)

Fig. 3 License Plate 2 at frame 50 (poor quality image)

Fig. 4 License Plate 2 at frame 100 (poor quality image)

```
Image  0  =  LSI5EBC
Image  1  =  rLSI5EBC
Image  2  =
Image  3  =
```

Fig. 5 Output of the OCR Function.

The evaluation metrics of training and testing our model are above average as per the base paper of YOLOv4 model[1]. The output video clearly shows the bounding box around the detected license plates.

Conclusion

We have implemented License Plate Detection via YOLOv4 model with custom dataset and weights with evaluation metrics - mAP@0.50 (AP50) = 68.96%. Thus, showcasing that implementation of the YOLOv4 algorithm provides better accuracy and is faster than other pre-existing object detection models. The future scope for this model will be functioning on live CCTV footage.

References

1. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection" arXiv:2004.10934
2. GitHub Repository: https://github.com/theAIGuysCode/yolov4-custom-functions
3. https://machinelearningknowledge.ai/yolov4-object-detection-tutorial-with-image-and-video/
4. Dataset: https://storage.googleapis.com/openimages/web/index.html
5. Pytesseract installation: https://github.com/UB-Mannheim/tesseract/wiki
6. https://en.wikipedia.org/wiki/Precision_and_recall
7. Evaluation metrics explanations: https://yanfengliux.medium.com/the-confusing-metrics-of-ap-and-map-for-object-detection-3113ba0386ef