Vehicle and License Plate Recognition from Video in Real Time on Embedded Systems

Eirini Denazi

University of Patras, dept. of Electrical and Computer Engineering

ABSTRACT

Subject of this MSc. Thesis is the recognition of vehicles and license plates for a real-time video on an embedded system. In order to achieve vehicle detection three deep learning methods were used: a 5-layer CNN [1] , a Faster R-CNN [2] and a YOLO v3 [3]. The first one was trained by us and the other ones were pre-trained on the COCO dataset [4]. After a vehicle is detected, the frame is cropped and fed into an OCR algorithm. OCR was made possible using 3 methods: PyTesseract [5], a 5-layer CNN and a YOLO v3. For the second one ROI extraction of each character is done first. The YOLO locates and classifies each character on its own. The CNN and the YOLO are pre-trained on Belgian license plates [6]. All the above are tested in real-time on a Raspberry PI [7].

METHODS

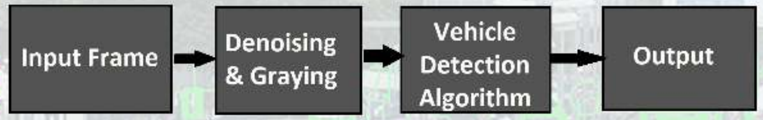
Vehicle Detection:

- Extract ROI using OpenCV [8] , crop frame and feed into simple CNN using KERAS framework. Decide if ROI contains vehicle
- Use pre-trained R-CNN using TorchVision [9] or a pre-trained YOLO v3 using Darkflow framework [10]

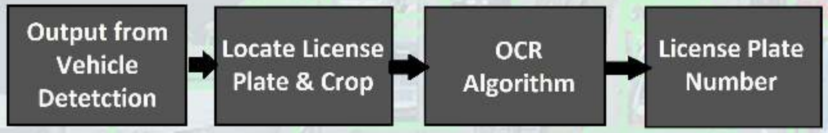
License Plate Recognition: First locate license plate and crop frame and then:

- Use PyTesseract to extract number
- Locate each character and classify it using a simple CNN
- Extract license plate number with a pre-trained YOLO v3 using OpenCV framework

VEHICLE DECTION SYSTEM PIPELINE



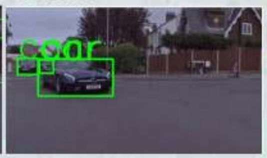
LICENSE PLATE RECOGNITION PIPELINE



VEHICLE DETECTION RESULTS

CNN

R-CNN



YOLO v3

LICENSE PLATE RECOGNITION RESULTS

PyTessaract

```
06-204 |
```

Open CV + CNN

```
RUZ907
```

YOLO + CNN

```
1RUZ907
```

COMPARISSON OF VEHICLE DETECTION METHODS

METHOD	ACCURACY (train set)	TIME (PC)	TIME (Raspberry PI)
Simple CNN	99%	<1 msec	~10 msec
Faster R-CNN	60.6%	7 sec	.
YOLO v3	55%	0.5 sec	6.5 sec

CONCLUSIONS

The choice of algorithm and method for vehicle detection are correlated with the nature of each application. In applications where detection time is crucial but there is no need for accuracy (e.g. traffic control), the simple CNN model could be used. In other cases, where detection time is not that crucial, but there is need in accuracy (like this one) the choice of a YOLO v3 is adequate. In applications where detection is needed over a period of time, but accuracy is crucial (e.g. automatic tolls) the Faster R-CNN is the most suitable algorithm.

License plate recognition is not successful using PyTesseract but is very accurate when using the other two methods on Belgian license plates. Unfortunately, The results are not the same when applied on license plates that aren't Belgian.

FUTURE WORK

- Train a smaller YOLO model to improve car detection time.
- Create a better small CNN model that does not overfit
- Use a better embedded system
- Create a dataset containing more license plates in real-life conditions (noisy, bad resolution and lighting) and perform data augmentation techniques to create a more robust model.
- Try these algorithms on a live demo in real-life conditions.

CONTACT INFO:

Un. Email: up1019256@upnet.gr

Personal Email: irenedenazi@gmail.com

REFERENCES: **[1]** LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. Nature, 521(7553),pp.436-444, **[2]** Ren, S., He, K., Girshick, R. and Sun, J. (2017). Faster R-CNN: Towards RealTime Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), pp.1137-1149., **[3]** Redmon, J. and Farhadi, A. (2017). YOLOv3: An Incremental Improvement. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, Honolulu, Hawaii, United States, **[4]** Cocodataset.org. (2020). COCO - Common Objects in Context. [online] Available at: http://cocodataset.org/#home, **[5]** Medium.(2020). What is Tesseract and how it works? [online] Available at: https://medium.com/@Bytepace/what-is-tesseract-and-how-it-works-dfff720f4a32, **[6]** Drive.google.com. (2020). Licence Plate Recognition – Google Drive. [online] Available at:https://drive.google.com/drive/folders/17gxw7tv7jy3KgJFhQiHX0IiIYObFbIJp, **[7]** Raspberrypi.org. (2020). [online] Available at: https://www.raspberrypi.org/products/raspberry-pi-4-model b/specifications/, **[8]** GitHub. (2020). thtrieu/darkflow. [online] Available at: ttps://github.com/thtrieu/darkflow, **[9]** Pytorch.org. (2020). torchvision — PyTorch master documentation. [online] Available at: https://pytorch.org/docs/stable/torchvision/index.html,**[10]** GitHub. (2020). thtrieu/darkflow. [online] Available at:https://github.com/thtrieu/darkflow