Vehicle Detection and License Plate Detection

Omar ElSeadawy

Improving traffic management systems by building a framework that is capable of automatically detecting illegal offences in the streets and being able to identify the cars responsible for such offences would be a great feat. This framework can be designed using different computer vision techniques. In this paper, I design and experiment with the first two phases of the framework that are responsible for vehicle detection, traffic line detection and license plate detection. Those models were built using deep learning approaches using the YOLOv5 pre-trained models and extending those models with extra training using two new data sets. The first data set specifically designed for vehicle detection and traffic line detection while the other data set is customized for the License plate recognition part of the system. I was capable of designing both models and reaching around 0.8 mAP in the vehicle and traffic line detection, and almost 1 mAP in the license plate recognition. Future work for this project, would be to combine both detectors using a common much bigger dataset to improve the speed and accuracy of detections, build a framework for detecting the offences based on the output of the detectors and finally extend the LPR to be capable of extracting characters and identifying the license plate numbers automatically.

Vehicle Detection | YOLO | LPR | Computer Vision | Machine Learning
Correspondence: ***omarelseadawy @aucegypt.edu***

Introduction

For all human beings, mature humans at least, it's reasonably easy for them to describe any image they might see or think about in their imaginations. Humans can describe many items in an image and recognize objects and their types fairly easy. Most people would have similar perspectives in regards to the main objects in an image, and if you ask any human for specific objects, they can easily find it with a simple visual search using their eyes. One example, if one human witnesses a hit and run accident for example, they can easily identify the type of car, if it's a small hatchback, sedan, large truck or any simple type, they can also identify the color in matter of milliseconds; moreover, they can even recognize the license plates although it might not be very easy to memorize its numbers for any person. Gaining huge amounts of information from a single snapshot is a very simple everyday task for humans; however, teaching a machine to do the same thing is not at all the same simple easy task.

Machines have to go through multiple complex phases in order to reach the same outcome as humans, taking much more time in the process as well. Machines need to learn to identify different objects in an image, learn what the type of each object is, learn to translate the visual information from the image such as shape and color of objects and what is even more difficult is to learn to read characters and language written on objects such as the license plates of cars. This is a major point of interest in Computer Vision domain which is mainly concerned with replicating human vision in understanding images and videos. Computer vision applications are now rapidly expanding into almost every industry we might know of. It is used in medical diagnosis and can be seen in the new dermatology by google [1], it is used in self-driving cars to analyze the road and the cars around and it is used in many other different fields.

One primary field of interest for us, is traffic management in which computer vision can be used to detect illegal offences like parking offences and different driving offences. To be able to detect and recognize such offences, the machines first need to learn to recognize distinct types of cars to be able to classify the car under its related offences. Moreover, when a machine is capable of recognizing an offence, it also needs to trace back the vehicle to its owner so they can be punished and this can be done by identifying and recognizing the license plates of those vehicles clearly. Therefore, in this project I am focused on the two main issues that fall under the umbrella of object detection, in which they can be used to pave the way for detecting illegal offences, which are identifying the types of cars and different objects in an image or a video and also be able to recognize the license plates of the cars using a computer vision model.

The rest of the paper will be divided into literature review of existing solutions in section 2, origin and description of the data sets section 3, the proposed solution in section 4, experimental results in section 5 and finally concluding in section 5 followed by the references in section 6.

Literature Review

Years ago, the early research in object detection was based on template matching techniques which started to appear in 1973; however, the computational resources at the time were not very strong to provide high quality detections. Years later, methods started adopting SVMs and Neural networks in order to start the first successful family of object detection techniques. Initially the most adopted use case was human and face detections, later on people started evolving their research to objects that human interacts with, cars, planes, animals and so on. Most of the common object detectors were based on the sliding window scheme in which a window detects objects in an image by sliding over different scales and regions and the extracted windows were ran through a classifier that was capable of indicating or predicting if this as an object it knew or not. Later on, instead of doing exhaustive search of a sliding window, researchers started finding more efficient alternatives that reduce the search

space by looking at key-points. Finally over the years, there were more modern approaches to object detections such as Coarse-to-Fine classifiers, dictionary-based classifiers, deep learning classifiers and others. The advantage of using Deep Learning techniques is that the machine is capable of learning the feature representation of objects instead of following rule-based or user-designed approaches, sadly it introduces the complexity of training a model and finding a decent data set to train such model [8].

Object detection algorithms require a series of image adjustment or video adjustment steps in order to be able to detect objects efficiently at the end according to [5]: Environment modeling or background modeling which has many approaches such as wallflower algorithm by Toyama [6] where he does background subtraction at 3 levels viz pixel level, region level and frame level and the purpose is to whether each pixel is related to the background or not, followed by Motion segmentation which is a technique to detect whether an object in a region is moving or is it stationary which can be done through Temporal Differencing for example in which this technique studies the pixels in 2-3 frames and be able to extract moving regions, and finally the object classification method which takes the extracted regions and distinguish them using different techniques such as shape-based classification based on size or box of an object or motion-based classification which is capable of classifying human actions or even Texture-based classification which classified the object based on the intensity patterns around the pixels [5][7]. All these techniques can be built using deep learning approaches in which we are capable of teaching the machine the shape, movement or texture of those objects instead of defining them in rule-based approaches.

In simple words, object detection is mainly for the machine to be able to identify specific objects it is pre-trained upon and be able to draw bounding boxes to mark those images and even label such objects, the higher quality the images, the easier for the machines to learn and extract information from those pixels. Nevertheless, the constant mobility of cars in traffic introduces the complexity of having low quality snapshots and videos of vehicles in traffic. This introduces a challenge in detecting objects and extracting information from such data; however, it is possible to achieve good detections with satisfying results using modern techniques in computer vision field. In traffic, the perception of vehicles, their types, positions and being able to identify them in the driving environment can be helpful in many environments such as self-driving cars or traffic control systems. With the powerful development of object detection in computer vision technology, extracting vehicle information became much simpler and more efficient than ever. YOLO which is object detection based on deep learning has been a leading pre-trained framework in such field [4].

Some researchers in [2] were capable of building a robust real-time automatic license plate recognition using the YOLO detector which is one of the parts in the framework I'm aiming to build. In their paper, the authors used YOLOv2 and Fast-YOLO approaches which were pre-trained on ImageNet in order to detect a vehicle, find its license plate and extract its license plate numbers. Their results were very impressive, being able to achieve over 95% recall in both vehicle detection and license plate detection. They were also able to extend the detector into a recognizer and extracted the license plate numbers with almost 75% accuracy using the UFPR-ALPR dataset. Other reseachers in [9] were able to reach 95%+ mAP in vehicle detection as well using YOLOv2.

Data Sets

In this project, I used two different data sets to build two different models so I will be discussing both data sets separately. The reason why they are different is because the first data set for detecting vehicles and identify their types was obtained through the competition by Techolution and had around 340 images, while the other UFPR-ALPR data set was obtained from [2] and is only released for academic research purposes and has around 4000 images that were annotated for license plates only; therefore, combining both of them was not very effective as they had different annotation classes. I had to do some minor data preprocessing for both sets as I had to change annotations from xml format and custom format into YOLO format. In addition, for the second data set I only needed some knowledge and did not require the rest of the knowledge so I also had to discard and clean the data to be able to input them into the YOLO training framework.

Techolution Data set. The Techolution data set has around 340 snapshots of street views, either of parked cars or cars moving in the street. There are total of 8 annotated objects that could exist in these images. The first 4 classes classify vehicles into 4 different types :

- Object: Car, Annotation "Car" which is any normal car
- Object: Light Good Vehicle, Annotation "LightGoodsVehicle" which is a vehicle for goods that move limited amount of items or people
- Object: Heavy Vehicle, Annotation "HeavyVehicle" which is large trucks
- Object: Motorcycle, Annotation "Motorcycle"

The other 4 classes identify the traffic lines which can later be used to track offences such as people parking out of lines or vehicles moving against the direction of lines or others.

- Object: Lot Boundary, Annotation "LotBoundary" which are boundary lines drawn in parking lots to mark a box in which vehicles should be parked inside them
- Object: Double Yellow Line, Annotation "DoubleYellowLine" which are lines that vehicles shouldn't park on or next to without a permit

- Object: Directional Arrow, Annotation "DirectionArrow" these lines are drawn in most streets to mark the direction of the traffic and no one should be going against them
- Object: Single Continuous white line, Annotation "SingleContinuousWhiteLine" which mark lanes such that vehicles should abide by and not cross unless allowed to.

These classes are annotated from 0 to 8 in the same order provided above. The data was split into training and testing by YOLO and the results of the vehicle detection will be shown in the results section.

UFPR-ALPR Dataset. This data set obtained from [2] contains 4500 snapshots taken from inside a vehicle, driving through thee streets in an urban environment. The snapshots were taken from a video over 100 videos with duration of 1 second and 30 FPS. The images were available in PNG format and their size was 1920x1080 pixels. There were a variety of license plates: gray, red in both cars and motorcycles. The data was split 40/20 for training and validation and the remaining 40% for testing. The data is annotated such that each snapshot has:

- Bounding Box for vehicle annotating its brand and color
- Bounding Box for License plate
- Bounding Box for each character in the license plate

Proposed Approach

This section will describe the original proposed approach of the paper, but unfortunately due to timing constraints only the first phase of the approach was experimented. Figure 1 shows there were 5 steps in the approach in order to make it a viable deployable system.

- Step 1: Vehicle detection either from images or videos.
- Step 2: Road Line Detection
- Step 3: License Plate Detection
- Step 4: Check if any illegal traffic offences exist using data from steps 1 and 2
- Step 5: If there is any offence, analyze the license plate detected in part 3 and recognize the characters

Unfortunately, due to time and resource constraints, in this paper I was only able to work on building one model for detecting both vehicles and road lines using the first data set and another model capable of detecting license plates using UFPR-ALPR data set; however, I did not reach steps 4 and 5 in these experiments. I decided to go with YOLO pretrained models, they are implemented in PyTorch and they are super fast compared to other models such as EFficientDet Open sourced by google. I tested out different pre-trained models starting from YOLOv3 to YOLOv5 but some of them were either too small that they took lots of time to reach satisfactory results and others were too big and very computationally expensive and so I decided to stick with YOLOv5l which suited my laptop well and was able to achieve decent results.

Vehicle and Traffic Line Detection. Using the first data set, I built a model using YOLOv5l pre-trained checkpoint. This model was trained for 300 epochs with default settings and no augmentations on 5000 COCO images of size 640 pixels from val2017. The network has 47 Million parameters and is fairly fast compared to larger models. I tested out 50 epochs and 100 epochs but there was not much of a difference because the data set is not huge as it only has around 300 images, so it usually saturated fast and the results will be show in the following section.

License Plate Detection. Using the UFPR-ALPR data set, this time I had a much larger data set, I only extracted the bounding boxes for the license plates and not for each character as I was only interested to find the license plate entirely for now. Analyzing the plate it self is a more complex step in the pipeline that I was not able to reach. I used the exact same YOLOv5l model which produced decent results as well, I ran this for 25 epochs only as it was much slower compared to the first phase thanks to the increase in the data set size.

Experiment and Results

In this section, I am going to show the results of the conducted experiments. The specs of the laptop to train both models: Intel i7-8750H CPU @ 2.2 GHZ, 16 GB RAM and NVIDIA GTX 1060. The image size used was 620, the batch number of images were 6. I am going to show for each experiment, the graphs of precision and recall which measures the accuracy of our predictions and I'm also going to show the mAP@0.5 and mAP@0.5:0.95 which is one of the important evaluation metrics for object detection. I am also going to show some examples of the labeled testing images against the predicted images and show the effectiveness of the model.

A. Vehicle Type and Traffic Line Detection. First experiment for this detection phase, I ran on the 340 Techolution imaeges using YOLOv3l for 50 epochs and here are the results I found. As we can see in figure 2, the precision and recall started increasing and dropped around 20 apochs and then went up and stabilized at around 40+ epochs at around 0.82 in both precision and recall which means that the model was accurate. The second part of figure 2 shows the mAP graphs for the 50 epochs as well which also show satisfactory results reaching around 0.8 @ 0.5 and 0.6 @ 0.5:095.

The following images in figure 3 show the labeled bounding boxes and the predicted detected objects respectively from one of the test batches. The first twelve

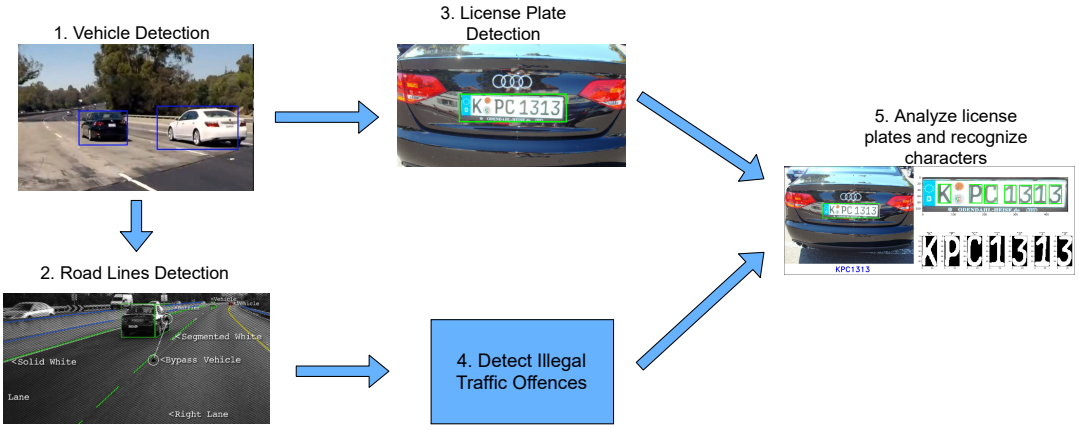

Fig. 1. Full proposed approach.

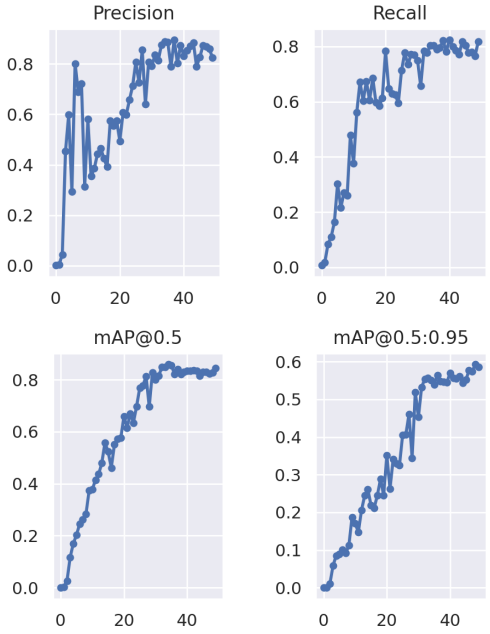

Fig. 2. Precision, Recall and mAP

Fig. 3. Labeled Data vs Predicted Data

photos are compared to the bottom twelve photos and as we can see, the predictions are almost equivalent to the labels and even some of them have extra bit of information detected.

I felt that there might be a possibility that if I trained the network for more time it would be able to achieve better results so I decided to experiment that theory out and re-trained the network for 100 epochs. Moreover, I decided tp upgrade from YOLOv3l to YOLOv5l and also experiment the difference between the two pre-trained models.

As we can see from figure 4, both precision and recall almost stabilized at around the same value as YOLOv3, might be a bit higher but only by one or two decimal points and the same applies for the mean average precision. We can also see from the labeled vs predicted images that the results are very satisfactory such that this model is capable of almost extracting all the annotated objects from the photos and even highlight extra objects.

This also meant that the reason not much change was seen from two version of YOLO pre-trained models and doubling the number of epochs is because the data set it self is not huge, it only has around 300 images and it means the network is only capable of doing so much training until it stops seeing any improvements.

This marks the end of the first phase of the system which is responsible for detecting vehicles and identifying them based on types mentioned in data set section of the Techolution images. It is also capable of detecting the traffic lines well and the output of this model should be the input for the illegal offence checking framework that should be added in the future work. Let us now move on to the next detection phase which is License plate detection using the second provided data set.

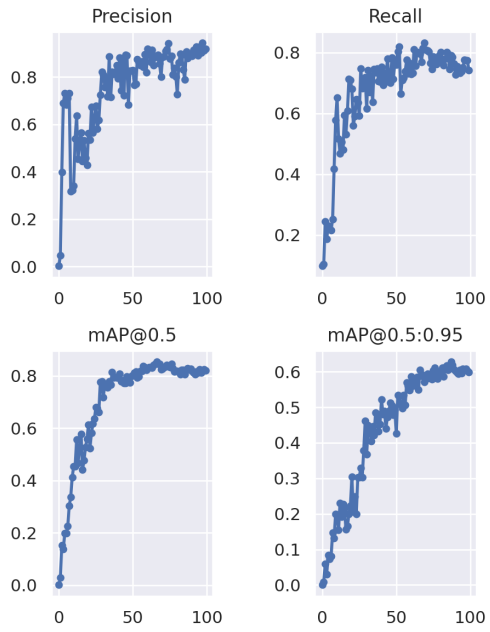

Fig. 4. Precision, Recall and mAP

Fig. 5. Labeled Data vs Predicted Data

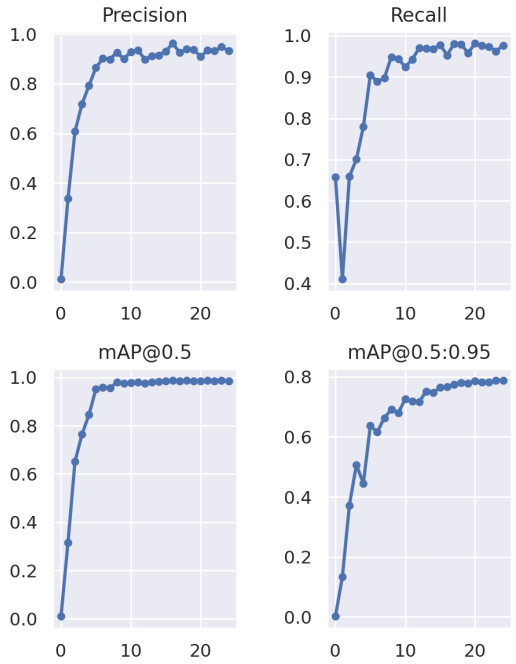

Fig. 6. Precision, Recall and mAP

B. License Plate Detection. The data set for this part of the system was much bigger and thus was much slower when it came to training the model. I was only capable of running the model for 25 epochs over the provided 4000+ images and came up with very satisfying results in very short time using the YOLOv5l model as well. We can see from Figure 6 that LPR using this data was able to achieve higher precision and recall almost reaching up to 1. Same applies to the mAP which gives much better results, even though the number of epochs is half the other and it is using the YOLOv5l pre-trained checkpoints. The reason is most likely the size of the data set which helps train the LPR much better and it is even divided into bigger chunks for testing and validation which also helps improve the overall performance of this model. Figure 7 shows the labeled data vs predicted data and we can actually see the predicted data capable of highlighting all labeled license plates in addition to extra license plates that were not annotated in the data.

The results are very satisfactory for both detection models. These two are the first two parts of the overall system shown in figure 1. For future work, those two models need to attach to a framework that capable of detecting the traffic offences and based on the verdict, it is capable of extending the license plate detection into analysis and recognition of characters and this way it is much easier to automate the entire process. One other approach that should've produced also good results is to combine both detectors at once, sadly for that we need a data set that annotates all 8 classes from the first data set in addition to the license plate class from the second data set to be able to train the model in a more effective manner.

Comparing my results to other results such as [2][9], I

Fig. 7. Labeled Data vs Predicted Data

was able to achieve very high accuracy (almost 100%) using the YOLOv5l in the license plate detection model which is similar to some of the proposed solution in literature; however, I was not able to achieve comparable results in the vehicle detection with the traffic lines detection as I was only able to reach only around 0.8 mAP which is fairly lower than those researchers who were capable of reaching 100% accuracy in vehicle detection. The main difference between us is the data set being used for training as they had data sets with 9000 images and even up to 40000 images in some cases and this massive difference varies the results greatly because the deep learning approaches are heavily dependant on the data set being fed to train the model.

Conclusions

To conclude, after introducing the problem of object detection and recognition, I talked about some of the current technologies and techniques used in this field and some of the early methods. For the project, I decided to try out those techniques in one specific use case which is in the field of Traffic Control. The reason is that building such system capable of detecting offences easily and automatically will help enforcing the laws and improving the overall order of traffic. I built phases 1 and 2 from the system responsible for detecting vehicle types, traffic lines and license plates and I was able to show that the two models I built were capable of achieving very good results in the detection of all the aforementioned objects. For future work, the most important step in my opinion is to find a larger combined data set and combine both detection models in one, then build upon them the framework for detecting offences and extending the License plate detector into a license plate recognizer in which its capable of reading the characters of the license plate from images and videos.

References

1. Peggy Bui, M. D. (2021, May 18). Using AI to help find answers to common skin conditions. Google. https://blog.google/technology/health/ai-dermatology-preview-io-2021/.

2. R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector" in 2018 International Joint Conference on Neural Networks (IJCNN), July 2018.

3. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Ingham, F.; Poznanski, J.; Fang, J.; Yu, L.; et al. YOLOv5. Available online: http://doi.org/10.5281/zenodo.4154370 (accessed on 16 November 2020).

4. Lian, J.; Yin, Y.; Li, L.; Wang, Z.; Zhou, Y. Small Object Detection in Traffic Scenes Based on Attention Feature Fusion. Sensors 2021, 21, 3031. https://doi.org/10.3390/s21093031

5. S., Manjula Tamilselvan, Lakshmi Ravichandran, Manjula. (2016). A Study On Object Detection.

6. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. Proceedings of Int. Conf. Computer Vision, 1999, 255–261.

7. A.J. Lipton, H. Fuijiyoshi, R.S Patil. Moving target classification and tracking from real-time video. Proceedings of the IEEE Workshop on Application of Computer Vision, 1998, 8-14.

8. Verschae Rodrigo, Ruiz-del-Solar Javier, Object Detection: Current and Future Directions, Frontiers in Robotics and AI, https://www.frontiersin.org/article/10.3389/frobt.2015.00029

9. Sang J, Wu Z, Guo P, et al. An Improved YOLOv2 for Vehicle Detection. Sensors (Basel). 2018;18(12):4272. Published 2018 Dec 4. doi:10.3390/s18124272