License Plate Detection in Complex Scenes

Kaiyu Zhao
zhaokaiyu124@gmail.com

Yuan Wang
yuan.wang688@gmail.com

Abstract

License Plate detection has many important applications including: traffic & parking management, security surveillance and privacy protection. It is a challenging task because license plates often appear at different scales, orientation, lighting conditions, be partially obscured and be part of complex scenes. In this project, we studied the challenges of applying YOLO V3 model on the license plate detection task: (1) failing to detect small & far-away license plates and (2) mis-identifying similar looking objects as license plates. We explored several strategies aimed at addressing these challenges: (1) increasing input resolution, (2) re-tuning the anchors based on license plates and (3) training on images with & without license plates. We found that all 3 changes improved the model's performance in terms of mAP on both the validation dataset of images with license plate and the validation dataset of images with & without license plate. Combining high resolution & re-tuned anchors produced the best mAP.

1. Introduction

License plate ("LP") detection is a popular application for neural network object detection algorithms. This task has many important applications such as traffic management, digital security surveillance, privacy protection. License plate detection is difficult because license plates often appear at different scales, orientation, lighting conditions, be partially obscured and be part of complex scenes. In this project, we investigated fine-tuning the YOLO-V3 model for the license plate detection task. One potential application of our work is to automatically apply mosaics to license plates in Youtube videos for privacy protection.

2. Related Work

The prior works for license plate detection generally fall into 2 categories: (1) general object detection and (2) specialized license plate detection. In general object detection, there are several state-of-art algorithms such as YOLO-V3[1], Faster R-CNN[2], Single Shot Multibox Detector (SSD)[3]. These algorithms perform very well for detecting large classes of objects within complex scenes (e.g. 80 classes for COCO). Specialized license plate algorithms include RPNet[4] and LPRNet[5]. These algorithms have been trained and evaluated on specialized datasets that only include images of a single license plate (though there is great variation in angle, lighting etc.). The authors have demonstrated that these algorithms perform better than state-of-art general object detection algorithms on these specialized datasets. However, these specialized algorithms have not been evaluated on complex images that include 0 or more license plates.

In this project, we are interested in detecting license plates in complex scenes. Therefore, we used the YOLO-V3 as our baseline model and tried to improve its performance. In particular, we used the Tensorflow implementation of YOLO-V3 by Yun Yang[6]. We also used the k-Means clustering algorithm for re-tuning anchors by Lars Nieradzik[7].

3. Dataset and Features

We used Open-Image dataset in this project. This dataset contains license plates that appears as part of complex scenes. There could be 0, 1 or more license plates in the picture. License plates vary greatly in scale and angle.

We create 3 pairs of train & validation dataset by restricting the Open-Image train & validation dataset to: (1) images with LP, (2) a random sample of images without LP and (3) images containing objects from one of 8 classes specified by us, (namely, Vehicle Registration Plate, Car, Stop Sign, Street Light, Truck, Wheel, Traffic sign and Traffic Light).

Table 1. The statistics of datasets used in the project selected from open-image.

# of Images	Train	Validation
Images w/ LP	5,368	2,065
Images w/ & w/o LP	35,368	24,153
Images for 8-class Dataset	143,686	Not used in this project

Fig. 1 shows a representative sample from the dataset.

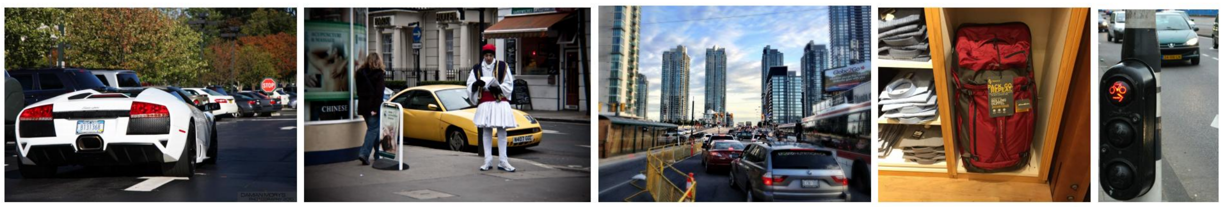


Figure 1: Representative sample of data: (1a), (1b), (1c) Image with LP, (2) Image without LP and (3) Image containing 1 of 8 classes.

4. Methods

The input to our algorithm is a set of images with/without LPs from Open Image dataset, and the target output is bounding boxes around predicted LPs with confidence level. We use the YOLO-V3 model fine-tuned on the Open Image Dataset's training data (restricted to images with LPs only) as a baseline. We did error analysis to identify weaknesses of this model, and experimented with several changes to improve its performance. This is elaborated further in the Experiments/Results/Discussions section.

5. Experiments/Results/Discussion

5.1 Baseline model

For our baseline model, we chose the YOLO-V3 model with anchors and weights that were trained for the COCO object detection task.

We fine-tuned the model by training it on a one-class ("LP") object detection task using images containing LPs in the Open Image training dataset. We fine-tuned the weights by first training 20 epochs where the weights in the Darknet-53 Backbone layers were frozen (i.e. not trained) and another 30 epochs where all of the weights in the model were trained.

5.1.1Error analysis

We evaluated the baseline model on the following 3 validation sets:

Table 2. Baseline Model Performance

Validation Dataset	**# Images**	**mAP**
Images with LP	2,065	84.5%
Images with & without LP	24,153	51.4%

For image with LP, we see the baseline model's performance is already very good. To see if we can improve it further, we manually examined and categorized the top 20 loss cases.

Table 3. Baseline Model Top 20 Loss Cases Categorized

	# Images	**%**
Missing Bounding Box(es) - Small & Hard to See LPs	8	40%
False Positive Bounding Box(es)	3	15%
Correct Bounding Box(es) with Low IoU	3	15%
Overlapping Bounding Box(es)	3	15%
Correct Bounding Box(es) with Low Confidence	2	10%

We see that the most common loss cases is images with small, far-away, partially obscured LPs where the model either didn't detect a LP or produced a low confidence prediction. To see if we can improve the model's performance on these loss cases, we tried:

- Increase the input image resolution of the model, and
- Re-tune the anchors using k-means clustering on the LP bounding boxes.

We observed the baseline model had significantly lower mAP on the validation set containing images with & without LP which indicates that the baseline model sometimes mispredicted false positive ("FP") bounding boxes. This is further confirmed by the baseline model's precision-recall curve on the validation of images with LP vs. validation set of images with & without LP (see Fig. 4). To see if we can reduce the number and confidence of LP predictions, we tried:

- training a YOLO V3 model with images containing at one of 8 specified classes.
- training a YOLO V3 model with images with & without LP.

These experiments are elaborated on in the following subsections.

5.2 High Resolution Model

Motivated by the fact that ~40% of the top 20 loss cases had small & hard to see LPs that were missed by the baseline model, we hypothesized that increasing the input resolution could increase the model's recall for such small & hard to see LPs. The default YOLO V3 model detects objects at 3 different scales (1/8, 1/16, 1/32). With the default input size of 544 pixels by 544 pixels, objects that occupy only 1-2% of the image (5-10 pixels) would be very difficult to detect even by the finest scale.

Therefore, we tried increasing the input image resolution to 1024 pixels x 1024 pixels. During training, YOLO V3 randomly chooses from a set of input resolutions for each batch in order to generalize better. We also updated the set of input resolutions used in training to cover larger image sizes (i.e. [416, 448, 480, 512, 544, 576, 608, 800, 896, 1024]). This model achieved better mAP on both validation datasets. See "Model Evaluation" subsection.

5.3 High Resolution Model with Re-tuned Anchors

The baseline model used anchors derived from the COCO object detection dataset, which is optimized for objects with diverse height-to-width ratios. We observed that almost all LP bounding boxes are either wide or square. Therefore, we expect only a subset of the default COCO-based anchor boxes would be useful for detecting LPs.

Therefore, we tried to re-tune the anchors by running k-Means clustering on the LP bounding boxes in the training set. Here is # of anchors that fall into reach range of aspect (height-to-width) ratio.

Table 4. Anchor Grouped by Aspect Ratio

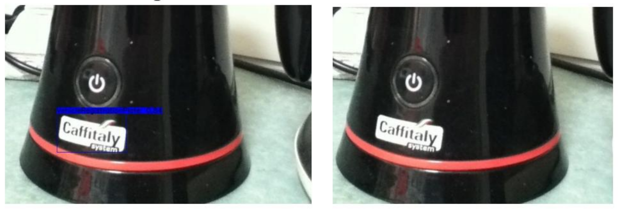
Height-to-Width Ratio	COCO	LP
tall (>1.5)	3	0
square (0.67 to 1.5)	5	1
wide (0.5 to 0.67)	1	4
very wide (<0.5)	0	4

We trained a high resolution model using the new anchors. This model achieved better mAP on both validation datasets. See “Model Evaluation” subsection for results.

5.4 Train on multiple classes (8 Classes)

A YOLO detector trained on only images with LP can mis-predict similar looking objects. For example, Fig. 2. a shows the baseline model mispredicted the brand label on a kettle to be a LP with 0.84 confidence.

We selected seven additional classes that commonly appear together with LPs, including car, stop sign, street light, truck, wheel, traffic signs and traffic lights, etc. This significantly increased the diversity of images in training set. While the new model did correct some LPs (see Fig. 2), we did not achieve satisfactory results due to the significantly increased computation time required to train an 8-class model.



(a) (b)

Fig. 2. (a) The baseline model mispredicted the brand label as a LP with 0.84 confidence. (b) The multiple classes model corrected the misprediction case on the same image.

5.5 Train on images with & without LPs

We also tried the simpler approach of training the YOLO V3 model with a dataset that includes images with & without LPs. We expect the increased diversity of training images would help the model learn to not mis-predict FPs. This model achieved better mAP on both validation sets. See “Model Evaluation” subsection for results.

5.6 Model Evaluation

We evaluate the models by comparing the mAP on the following validation sets: (1) images with LPs, (2) images with & without LPs.

Table 5. Model Performance Comparison

mAP	Image w/ LP	Image w & w/o LP
YOLO V3 (Baseline)	85.85%	51.40%
YOLO V3 High Res	87.74%	54.46%
YOLO V3 High Res & Re-tuned Anchors	**88.39%**	**61.47%**
YOLO V3 Trained on Image with & without LP	87.17%	57.57%

The "High Resolution Model", the "High Resolution & Re-tuned Anchors Model" and the "Model Trained on Image with & without LP" significantly improved the mAP on validation images with LPs. This indicates that these models are better at detecting LPs in images that contain 1 or more LPs. Furthermore, these models improved the mAP on validation images with & without LPs. This suggests these models suffer less mis-prediction of LPs. The "High Resolution & Re-tuned Anchors Model" had the best mAP on both datasets.

Improvements on small & far-away LPs

We see that our changes (in particular the "High Resolution & Retuned Anchors" model) is able to detect small & far-away LPs significantly better than baseline.

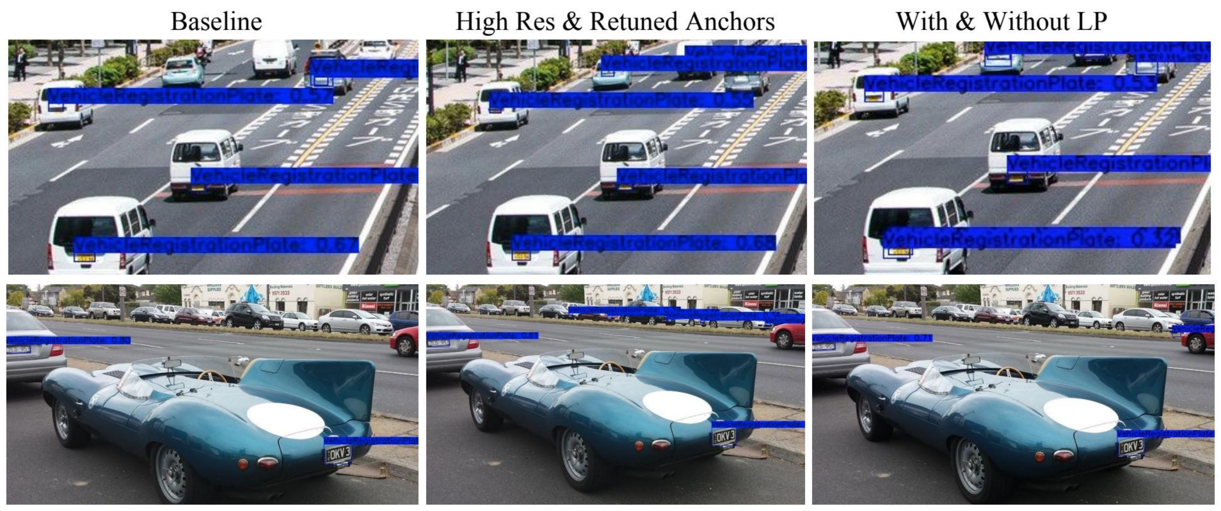

Baseline
High Res & Retuned Anchors
With & Without LP
VehicleRegistrationPlate: 0.57
VehicleRegistrationPlate
VehicleRegistrationPlate: 0.63
VehicleRegistrationPlate: 0.55
VehicleRegistrationPlate: 0.68
VehicleRegistrationPlate: 0.53
VehicleRegistrationPlate: 0.32

Fig. 3: The above images are cropped from the larger output image to focus on differences between model outputs. We see that the "High Resolution & Retuned Anchors" model is able to detect many small & far-away LPs missed by baseline model while the "Trained with & without LP" model is able to detect a few more.

Improvements on FPs

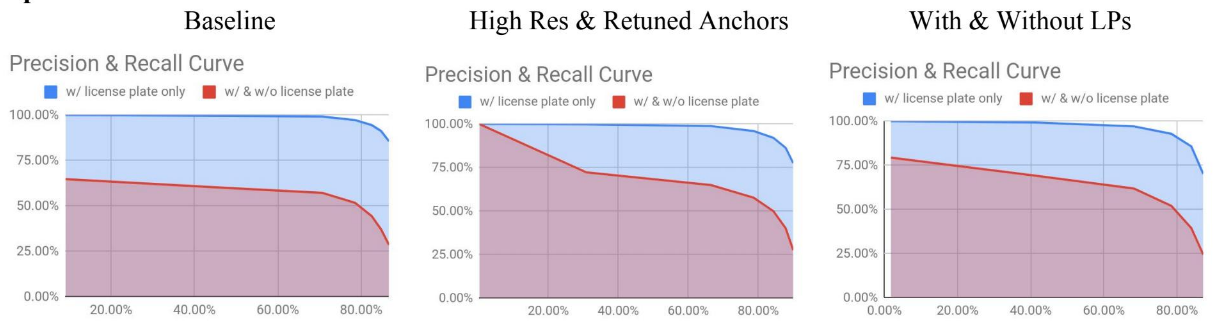


Figure 4: we see that the "High Resolution & Retuned Anchors" model shows very significant improvement in precision over the baseline model on the validation dataset that includes images with & without LP.

Difficult cases:
Our models continue to fail to detect LPs (Fig. 5) or mis-predict LPs (Fig. 6) in some challenging cases.

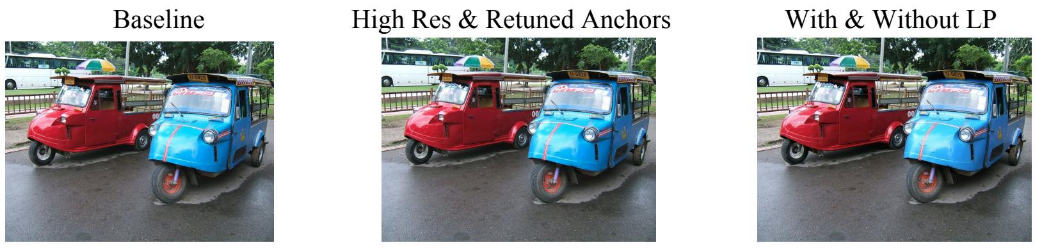


Fig. 5: all models still fail to detect the LP in this image. This suggests the models rely on the context image around the LP (usually a car) to detect the LP. When the context is an auto-rickshaw, these models struggle to find LP.

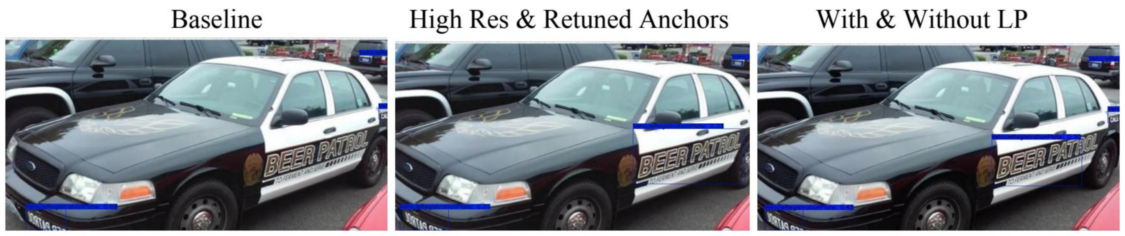


Fig. 6: All models mis-predicted LPs that look similar to LPs. The "High Resolution & Re-tuned Anchors" model and the "Train on images with & without LP" model produced lower confidence for these FPs.

6. Conclusion / Future Work

In this project, we studied fine-tuning the YOLO-V3 object detection model to the LP detection task. We identified two challenges: (1) failing to detect small & far-away LPs and (2) mis-identifying similar looking objects as LPs. We tried 3 different strategies for improving the model (1) increasing input resolution, (2) re-tuning the anchors for LPs and (3) training on images with & without LP. We found these strategies significantly improved mAP and performances on challenging cases. We found that combining high resolution & re-tuning the anchors produced best mAP on validation set.

Potential future work includes:

- Train the "High Resolution & Re-tuned Anchors Model" on images with & without LP,
- Different network architecture such as Faster R-CNN, Mask R-CNN,
- Modify specialized LP recognition models for complex images, and
- Modify the loss function or the training batch to put additional weight on FP loss cases.

7. Contributions

Both team members contributed equally to this project.

8. References

[1] Redmon, Joseph, and Ali Farhadi. 2018. "YOLOv3: An Incremental Improvement," April.
[2] Ren S, He K, Girshick R, Sun J, editors. Faster R-CNN: towards real-time object detection with region proposal networks; Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS-2015); 2015 December 7-12; Montreal, Canada. Cambridge: MIT Press; 2015. pp. 91-99.
[3]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In ECCV, pages 21–37, 2016. 2.

[4] H. Li, P. Wang, and C. Shen, “Towards end-to-end car license plates detection and recognition with deep neural networks,”CoRR, vol.abs/1709.08828, 2017.
[5] S. Zherzdev and A. Gruzdev, “LPRNet: License Plate Recognition via Deep Neural Networks,” arXiv:1806.10447 [cs], Jun. 2018, arXiv: 1806.10447. [Online]. Available: http://arxiv.org/abs/1806.10447.
[6] https://github.com/YunYang1994/tensorflow-yolov3
[7] https://lars76.github.io/object-detection/k-means-anchor-boxes/

9. Code

This is collection of code used in this Project:

- Scripts used to train & evaluate the models (by Kaiyu Zhao and Yuan Wang): Github Link
- Tensorflow Implementation of YOLO V3 (by Yun Yang): Github Link (as at May 20, 2019)