R2Joint: Robust Real-Time Joint Detection Model for Traffic Scenes

**Jinren Ding 23020211153927, Wei Lin 23020211153949, Chen Chen 23020211153916,
Jianlong Hu 31520211154049, Xiang Fei 23020211153891
1549274402@qq.com**

Abstract

Object detection is gaining more and more attention and applications as a common area of deep learning. With the rapid development of Intelligent Transportation Systems (ITS), a greater demand for the detection and classification of traffic scenes has arisen. In addition to the analysis of static objects such as traffic signs, the detection of two types of dynamic objects, namely vehicles and pedestrians, has become an even more pressing problem. Furthermore, ITS also needs to learn more detailed information from regular object detection tasks. Therefore, the detection of license plates of motor vehicles and faces of pedestrians is also a vital goal to be achieved in our project. In this work, we proposed an optimized robust real-time joint object detection model (R2Joint), which is capable of detecting common types of vehicles simultaneously, and recognizes pedestrians' faces and license plates in real time. R2Joint is able to avoid the degradation of detection due to Non-Maximum Suppression (NMS), thus performs better under joint detection task. To achieve more robustly, R2Joint also uses adversarial attack samples for training and modified the network structure to better handle perturbations and noise in data. Our model achieves 62.2% mAP with 71.4FPS, with sufficient performance for the application.

Introduction

Deep learning (LeCun et al. 1998) has achieved remarkable developments in many fields, such as object detection (Bochkovskiy, Wang, and Liao 2020), semantic segmentation (Nirkin, Wolf, and Hassner 2021), and machine translation (Vaswani et al. 2017). As the most common and intuitive application area, computer vision has also gained widespread attention, becoming an important field of artificial intelligence. Computer vision, which is the science of computers and software systems, aiming at recognizing and understanding images and scenes. It is consists of many aspects, including image recognition, object detection, image generation, image super-resolution, etc.

Object detection, considering its large number of applications, can be regarded as the most profound branch in the field of computer vision. Object detection, often used as a collective term for both detection and recognition processes, refers to the process of recognizing and classifying objects in a given scene or image. It has been widely used in scenes such as multi-object recognition, face detection, vehicle detection, pedestrian tracking and so on. Intersecting with other disciplines, object detection can also be applied to more fields.

Moreover, considering the rapid increase of motor vehicles, tremendous pressure has been brought to the public traffic, and the frequent road congestion and traffic accidents have caused great insecurity and increasing complexity in the traffic environment. Therefore, the combination of artificial intelligence technology to improve the traffic environment and alleviate traffic pressure has become a more urgent need nowadays. Intelligent Transportation System (ITS) has emerged to meet this demand by using and combining various information technologies to help improve traffic by means of artificial intelligence.

To combine with, object detection for pedestrian and vehicle becomes a pressing issue. Recently, the object detection algorithm has made great breakthroughs. Based on the network architecture, the most popular deep learning-based algorithms can be divided into two categories, namely ”two-stage” and ”one-stage” methods. The two-stage method, which contain two separate detection and classification steps, is more accurate but relatively slower. The one-stage method, whose networks directly detect and classify objects, is capable of achieving higher speeds, but with reduced performance. In this work, we choose a basic one-stage method and improve its performance based on it.

After the object detection step, we also consider adding structured analysis of the face, such as analyzing whether the person is a male or a female, whether it is a smiling face, and so on. In addition, we also consider adding the processing of license plate (LP) recognition. Automatic LP Recognition is a challenging and important task which is used in traffic management, digital security surveillance and vehicle recognition, which closely related to our task. The robust automatic LP recognition system needs to cope with a variety of environments while maintaining a high level of accuracy. In other words, this system should work well in natural conditions.

To improve the robustness of the algorithm, considering the complexity of the traffic environment, we design adversarial samples for training. We add noise to the commonly used data images, which helps the network to gain better stability and make it less vulnerable to attacks. Our proposed R2Joint model tries to maximize the predict confidence while minimize the detection loss. Tested under carefully designed attack samples, the improved algorithm was able to improve its mAP by 62.6% over the original algorithm.

In summary, the main work of this paper are three fold:

- Training a basic one-stage object detection model to detect pedestrians and vehicles on traffic roads.
- Performing structural analysis on vehicle and pedestrian information, identify the license plate data and facial data within it.
- Using adversarial attack training methods to improve the robustness of the proposed model.

Related Work

Object Detection Models

Deep learning (LeCun et al. 1998) helps object detection models reach higher performance than basic methods, with complex network architecture and sufficient training data. A modern detector is usually composed of two parts, namely the backbone and the detection head. The backbone, which consists of layers, is aiming at extracting features from the raw input. And the task of the detection head is to dict classes and bounding boxes of objects. Different designed backbones and detection heads can be adapted to different hardware environments. For those detectors running on GPU platform, their backbone could be VGG (Simonyan and Zisserman 2014), ResNet (He et al. 2016) or DenseNet (Huang et al. 2017). On the other hand, when using in mobile or CPU platform, family of MobileNet (Howard et al. 2017; Sandler et al. 2018) and ShuffleNet (Zhang et al. 2018) are more suitable.

Based on different detection head and structure, the object detection algorithms are usually categorized into two kinds, namely the one-stage methods and the two-stage methods. The two-stage algorithms, with the first step of generation and extraction of candidate regions and regional features, and the second step of classification based on regional features. These two steps achieve detection and recognition respectively, so it is called the two-stage method. The accuracy of this series of methods is relatively high, but the generation and extraction steps of candidate regions require a large number of repeated operations, which consume more resources and are slower, so it is difficult to achieve the effect of real-time detection. The most commonly used two-stage methods are the family of R-CNN (Girshick et al. 2014), with R-CNN itself and improved methods like Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2015), R-FCN (Dai et al. 2016), Libra R-CNN (Pang et al. 2019), etc. It is also possible to make a two-stage object detector an anchor-free object detector, such as RepPoints (Yang et al. 2019).

The other series of algorithms, one-stage methods, however, directly use single-stage regression to complete the task of object detection and recognition, avoiding the most time-consuming steps of candidate region generation and extraction in two-stage based algorithms. They simplify the network and thus significantly improve the inference speed. Since one-stage methods reduce the size of models and hardware resources consumption, they also potentially reduce the recognition accuracy. Some commonly used methods include the series of YOLO (Redmon et al. 2016; Redmon and Farhadi 2018) and SSD (Liu et al. 2016), whose networks directly predict the categories and positions of different objects.

License Plate Recognition

In the earlier works on general LP recognition such as (Anagnostopoulos et al. 2008), the pipeline consist of character segmentation and character classification stages:

- Character segmentation typically uses different hand crafted algorithms, combining projections, connectivity and contour based image components. It takes a binary image or intermediate representation as input, thus character segmentation quality is highly affected by the input image noise, low resolution, blur or deformations.
- Character classification typically utilizes one of the Optical Character Recognition (OCR) methods which is adopted for LP character set.

Since classification follows the character segmentation, end-to-end recognition quality depends heavily on the applied segmentation method. In order to solve the problem of character segmentation, deep learning-based solutions are proposed. These kind of methods typically take the whole LP image as input and produce the output character sequence. Recent work (Goodfellow et al. 2014) tries to exploit synthetic data generation approach based on Generative Adversarial Networks (Goodfellow et al. 2014) for data generation procedure to obtain large representative license plates datasets.

Adversarial Attack

Neural Networks are vulnerable to adversarial examples like intentionally perturbed images (Szegedy et al. 2013). To improve the robustness of network, various methods have been proposed to generate adversarial samples (Goodfellow, Shlens, and Szegedy 2014; Carlini and Wagner 2017). The attack methods can be divided to two types. In the white-box setting, both the network architecture and parameters are available to the attacker. As for the black-box attack, the attacker only has access to the model's input and the predicted output, and the network itself is invisible. On the field of object detection, a typical kind of attack is adversarial patch (Brown et al. 2017), producing localized and universal perturbations to an image by masking pixels. Another more commonly used method is to add imperceptible noises (Xie et al. 2017), as to generate perturbed images with carefully designed noise, which is invisible to human.

The Proposed Method

We proposed the R2Joint, with robust real-time performance on joint object detection and structural analysis. In the following subsections, we first introduce our basic model. Following, we propose improvements on the R2Joint model,

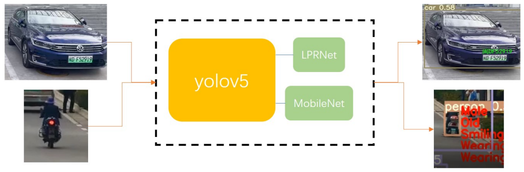

Figure 1: Our proposed model

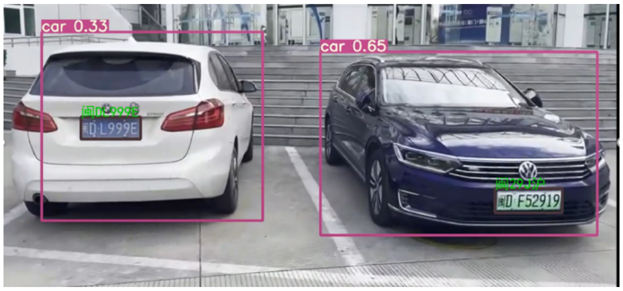

Figure 2: Example of different types of license plates

which consists of two crux techniques: structural analysis and adversarial attack.

Basic Model of R2Joint

Since our purpose is detecting common types of vehicles simultaneously, and recognizing pedestrians' faces and license plates in real time, the one-stage method is adapted. As it avoids the most time consuming steps of candidate region generation and extraction, the inference speed is significantly improved. As for one-stage object detectors, the most representative models are the family of YOLO series.

Specifically, we used YOLO-v5 as our basic model, which has been widely used recently, and its great power has been proved. Compared to the former YOLO, it has smaller network capacity, faster inference time and higher recognition rate. Our dataset contains eight class: person, bicycle, car, motorcycle, bus, truck, license plate, face. The first six types of data are selected from COCO, as license plate data is only selected from part of CCPD dataset (Xu et al. 2018), and face data is selected from LFW dataset (Huang et al. 2007). The proportion of training set and verification set of the first six categories is the proportion provided by COCO. After the object detection step, we also add structural analysis of the face information, and the processing of license plate recognition. Furthermore, to improve the robustness of the algorithm, considering the complexity of the traffic environment, we design adversarial samples for training. The whole process can be seen in Fig. 1.

Structural Analysis

After the object detection step, we use the structural analysis to get more information about what we have detected, especially the license plate and the facial data.

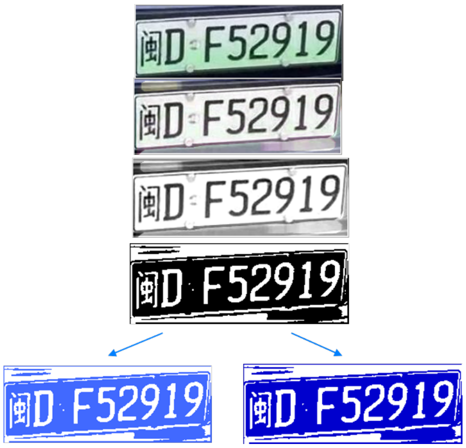

Figure 3: The process of converting colors of license plates

License Plate Recognition There is a very efficient network to help us to do this job, the LPRNet (Wang et al. 2020). It is a real-time framework for high-quality license plate recognition supporting template and character independent variable length license plates. However, we cannot use the pre-training model directly, because it can only recognize the blue plates, not working in the green plates (Fig. 2).

For recognize the green plates correctly, a simple way we can imagine is using a new dataset which contains the green plates to train the proposed model. However, the drawbacks of this approaches are also obvious, as once we want to recognize a new color of license plate, the model has to be retrained. We use another two steps approach to solve this problem. First, we choose the license plates that color is not blue. Second, we convert their colors to blue. In detail, firstly, the license plate image is converted to the HSV color space image, and the blue mask of the image is calculated. Then, the number of non-zero elements (i.e. the number of blue elements) of the mask is calculated. If the number is greater than half of the total pixel points of the image, it is considered as a blue license plate; otherwise, it is not a blue license plate and needs to be converted. For license plate images that are not blue, shadow removal, noise reduction and binarization are carried out, and then the non-white parts are filled with appropriate RGB values as blue, as shown in Fig. 3.

Facial Information Extraction We use the MobileNet (Sandler et al. 2018) as the main framework to extract the information of human face. In the end of the MobileNet, we add a modified classifier, which is implemented with a simple full-connected layer. We classify facial information according to the following attributes: gender, age, is smile, is wearing hat, is wearing glasses. We use the CelebA dataset to train our model, and get more than 90% accuracy during test, as shown in Fig. 4.

Adversarial Attack

Deep learning now has extremely superior performance, as demonstrated in many work. However, because neural net-

Table 1: Performances in AP of each category and mAP (IoU thread=0.5) for our method. The baseline refers to the model trained directly on this dataset using YOLO-v5.

Session	Model	AP								mAP
		Person	Bicycle	Car	Motorcycle	Bus	Truck	Face	Plate	
YOLO-v5	Baseline	0.625	0.335	0.406	0.549	0.653	0.38	0.986	0.983	**0.606**
YOLO-v5	R2Joint	0.633	0.371	0.727	0.429	0.465	0.369	0.974	0.995	**0.622**

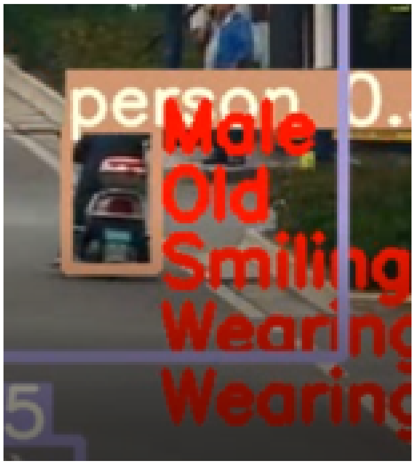

Figure 4: Facial attributes

works are black-box methods, people has little understanding of what the networks do. For some specially generated adversarial samples, models usually show a certain degree of vulnerability. Since the input form of deep learning algorithm is a numerical vector, the attacker may design a specific numerical vector to make the deep learning model make misjudgment, which is called adversarial attack. Unlike other attacks, adversarial attacks mainly occur when adversarial data is constructed, and then the adversarial data is fed into the deep learning model just like normal data to obtain the deceptive recognition results.

To evaluate our model's robustness, we generated some adversarial samples and conducted experiments to observe the performance of our YOLO-v5 model.

$$f = \frac{1}{\|\Lambda\|} \sum_{\lambda \in \Lambda} i : argmax(p_i) = \lambda \mathbb{E} \Bigg\{ \left[b_i^0 \cdot \max\left(p_i\right) - 1 \right]^2 + \left(\frac{b_i^w \cdot b_i^h}{W \times H} \right)^2 \Bigg\} \tag{1}$$

We use the YOLO-v3 (Redmon and Farhadi 2018) model which is trained on the COCO2017 dataset to generate adversarial samples. Our approach is to attack the NMS (Non-Maximum Suppression) mechanism of object detection, more specifically, maximizing the bounding box confidence and minimize the size of bounding box (Equation 1.), reducing their IoU (Intersection over Union), so they are harder to be filtered out by the NMS. This is a white-box setting method. The results show that YOLO-v5 model with adversarial attack mechanism has better detection performance and robustness.

Experiments

We conducted experiments on a user-defined dataset, which includes eight categories, of which the first six categories are from the challenging MS COCO dataset (Lin et al. 2014), the data of license plate category is from CCPD dataset (Xu et al. 2018), and the data of face category is from LFW dataset (Huang et al. 2007). We divide each dataset into the training set and the validation set. The model will be trained on the training set and tested on the latter. The ratio of training set data and verification set data of the first six categories is following the setting under COCO dataset, and the ratio of training set data and verification set data of face and license plate categories is set to 5:1. We evaluate the performance for R2Joint with three metrics: average precision (AP), inference time and model size.

Implementation Details

We train R2Joint with SGD, setting the initial learning rate of $1e-4$ with constant warm-up of 1k iterations and using weight decay of 10^{-4} and momentum of 0.9. For our comparative experiment, we train for 300 epochs with a learning rate drop by a factor of 10 at 100 and 200, respectively. Our experiments were performed on a single Nvidia Tesla V100 GPU.

Results

We take the model trained by YOLO-v5 on the user-defined data set as the baseline, and compare our method R2Joint with it. As shown in Table. 1, the map of R2Joint is nearly one percentage point higher than baseline, which shows that our method is effective. Among them, the AP in car category is 40% higher than baseline, which may be due to the impact of adding countermeasure samples.

Because the model of our project is designed to be deployed to the actual scenes, we also compare the parameter size of the model and the time required for prediction, as shown in Table. 2. Considering the balance of accuracy and speed, our method is nearly 0.2ms slower than the fastest YOLOv5s model in reasoning speed, due to the structural analysis module. However, the inference speed of R2Joint still reaches over 70fps, which meets the requirement of real-time detection.

Ablation experiments

In order to verify the effectiveness of our structural analysis and adversarial attack methods, we conducted ablation

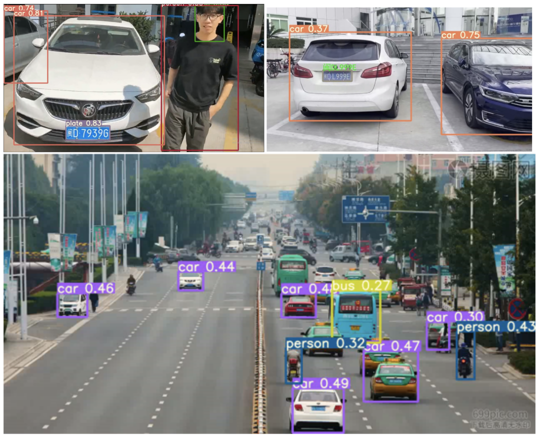

Figure 5: Examples for the results of traffic scenes detection from our method.

Table 2: Comparison between R2Joint and the original YOLO-v5 models.

Model	Inference Time(ms)	Model Size(MB)
YOLOv5s	0.9	7.2
YOLOv5m	1.7	21.1
YOLOv5l	2.7	46.5
R2Joint	1.4	14.0

Table 3: Effects of each component in our work.

YOLO-v5	Structural Analysis	Adversarial Attack	mAP	Inference Time(ms)	Model Size(MB)
✓			0.606	**1.2**	14.0
✓	✓		0.608	1.4	14.0
✓	✓	✓	**0.622**	1.4	14.0

experiments, as shown in Table. 3. The performance of the model using both structural analysis and adversarial attack is better than that of the model using only structural analysis and baseline, which proves the improvement of R2Joint.

Influence on Structured Analysis We added a new module to the original network to perform structured analysis of license plate and face information. The results show a 0.2% improvement in our training effect, but considering that adding the new module does not have an impact on the recognition effect itself, the small improvement in the training effect probably comes from errors. However, by adding the structured analysis module, our algorithm only slightly reduces the recognition speed and is still able to achieve real-time detection.

Influence on Adversarial Attack We compare the performance effects of the original dataset with the expanded dataset. The results show that the adversarial samples can improve the recognition accuracy of our algorithm by about 1.6%. Considering that our adversarial samples are obtained by mainly adding noise to the original images using a black-box attack, it is not possible to design attack samples with higher targeting. However, the results also show that our algorithm is able to produce slightly improved performance when trained in combination with adversarial samples. This may be due to drift in the real traffic environment faced by the model when it is actually used, which deviates from our dataset. Therefore, by introducing adversarial samples, it can help the model to obtain better robustness and thus better performance in the real environment.

Visualization

To better demonstrate the superiority of R2Joint, we provide several qualitative results in Fig. 5. We find that our method can precisely detect plate and car at the same time, and faces and people can also be detected at the same time.

Conclusion

In this paper, we proposed R2Joint, a robust real-time joint detection model for traffic scenes. Our proposed model is able to detect the vehicles and pedestrians in the traffic environment, and then identify license plates and face information with structural analysis. To achieve this goal, we first utilize YOLO-v5 in object detection step. After that, we use structural analysis to obtain more in-depth information, which consists of two important modules: license plate recognition and facial information extraction. We implement these modules with two sub-networks in R2Joint, which achieves the effect of joint model detection. In order to verify the robustness of our proposed model, we then train R2Joint through adversarial attack. Experimental results and comparisons with YOLO-v5 demonstrate better performance of the proposed R2Joint model, with real-time detection performance and sufficient performance for the application.

References

Anagnostopoulos, C.-N. E.; Anagnostopoulos, I. E.; Psoroulas, I. D.; Loumos, V.; and Kayafas, E. 2008. License plate recognition from still images and video sequences: A survey. *IEEE Transactions on intelligent transportation systems*, 9(3): 377–391.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665*.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. IEEE.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, 379–387.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, G. B.; Ramesh, M.; Berg, T.; and Learned-Miller, E. 2007. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, 740–755. Cham: Springer International Publishing. ISBN 978-3-319-10602-1.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C. Y.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision*.

Nirkin, Y.; Wolf, L.; and Hassner, T. 2021. Hyperseg: Patchwise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4061–4070.

Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; and Lin, D. 2019. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 821–830.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28: 91–99.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, D.; Tian, Y.; Geng, W.; Zhao, L.; and Gong, C. 2020. LPR-Net: Recognizing Chinese license plate in complex environments. *Pattern Recognition Letters*, 130: 148–156.

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1369–1378.

Xu, Z.; Yang, W.; Meng, A.; Lu, N.; Huang, H.; Ying, C.; and Huang, L. 2018. Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yang, Z.; Liu, S.; Hu, H.; Wang, L.; and Lin, S. 2019. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9657–9666.

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6848–6856.