Advances in Computer, Signals and Systems (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2025.090411
ISSN 2371-8838 Vol. 9 Num. 4

Research on Parking Lot Vehicle Counting Based on Simplified YOLOv3

Jiajie Qiu[a], Xiaoying Su[b,*]

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China
[a]3887756843@qq.com, [b]25980669@qq.com
**Corresponding author*

Keywords: Simplified YOLOv3; Vehicle Detection; Parking Lot Management; Vehicle Counting; Fixed View Angle

Abstract: Aiming at the problems of low efficiency and high error rate in traditional parking lot vehicle counting which relies on manual inspection, this paper proposes a vehicle detection and counting method based on the simplified YOLOv3, specially adapted to the parking lot monitoring scenarios with fixed viewing angles and no occlusion. By simplifying the network structure of YOLOv3, the method improves the operation speed while ensuring the detection accuracy, thus meeting the real-time counting requirements. Experimental results show that the vehicle detection accuracy of this method reaches 95.2% on the self-built parking lot dataset, the counting error is controlled within 3%, and the operation efficiency is increased by 40% compared with the original YOLOv3 model. It can effectively realize the automatic and accurate counting of vehicles in parking lots, providing technical support for the intelligent upgrading of parking lot management systems.

1. Research Background and Significance

With the acceleration of urbanization and the continuous growth of car ownership, parking lots, as an important part of the urban transportation system, their management efficiency directly affects the operation order of urban traffic and people's travel experience. Vehicle counting is the core link of parking lot management, and accurate and real-time vehicle count data can provide key basis for parking lot guidance, parking space scheduling, toll management and other work.

At present, there are mainly three methods for vehicle counting in parking lots: manual counting, magnetic frequency induction counting and video surveillance counting. The manual counting method relies on the regular inspection of management personnel, which not only consumes a lot of labor costs, but also is prone to problems such as missing counts and wrong counts when the vehicle flow is frequent, making it difficult to guarantee the counting accuracy and real-time performance. Magnetic frequency induction counting realizes vehicle detection by installing induction devices under parking spaces. Although it can ensure the detection accuracy of a single parking space, it has the defects of high installation cost and difficult maintenance. Moreover, it cannot directly count the total number of vehicles in the area, requiring additional aggregation equipment and algorithm

support.

Video surveillance counting has become the mainstream direction of intelligent parking lot management due to its advantages of low cost, wide coverage and strong traceability. Computer vision-based vehicle detection technology is the core of video surveillance counting. Traditional object detection algorithms such as Haar feature + Adaboost and HOG + SVM [2][5] are easily affected by factors such as illumination changes and vehicle appearance differences in complex scenarios, resulting in unsatisfactory detection effects. The rise of deep learning technology has provided a new solution for vehicle detection. Among them, the YOLO series algorithms [4][6][7] have been widely used in the field of object detection due to their end-to-end detection mode and excellent real-time performance.

Although the original YOLOv3 model has high detection accuracy, its complex network structure and large amount of computation make it difficult to achieve real-time processing in embedded devices or low-configured monitoring systems. However, the characteristics of the parking lot scenario with fixed viewing angle and no occlusion provide the possibility for simplifying the YOLOv3 model. This paper optimizes and simplifies the YOLOv3 model for this specific scenario, improving the operation speed while ensuring the detection accuracy, so as to realize the efficient and accurate counting of vehicles in the parking lot, which has important practical application value.

2. Overview of Related Technologies

2.1. Core Principles of YOLOv3

YOLOv3 is a one-stage object detection algorithm proposed by Redmon et al. Its core idea is to convert the object detection problem into a regression problem, which can complete the localization and classification of objects through a single forward propagation. Compared with two-stage object detection algorithms (such as Faster R-CNN[3]), YOLOv3 omits the step of region proposal generation, which greatly improves the operation speed..

YOLOv3 adopts Darknet-53 as the backbone feature extraction network. Darknet-53 contains 53 convolutional layers, which solves the problem of gradient disappearance in the training process of deep networks through residual connections, and can effectively extract the deep features of images. In terms of feature fusion, YOLOv3 adopts a multi-scale feature fusion strategy, using feature maps of different scales to detect objects of different sizes respectively: the large-scale feature map (13×13) is used to detect large objects, the medium-scale feature map (26×26) is used to detect medium-sized objects, and the small-scale feature map (52×52) is used to detect small objects. This multi-scale detection mechanism enables YOLOv3 to have a good detection effect on vehicles of different sizes.

In terms of object prediction, each grid cell of YOLOv3 will predict 3 bounding boxes, and each bounding box includes position information (x, y, w, h), confidence and category probability. The redundant bounding boxes are filtered out through the Non-Maximum Suppression (NMS) algorithm, and finally the object detection results are obtained.

2.2. Characteristics of Parking Lot Scenario and Basis for Model Simplification

The parking lot scenario studied in this paper has two prominent characteristics: fixed viewing angle and no occlusion. A fixed viewing angle means that the position and angle of the surveillance camera are fixed, so the position distribution and scale variation range of vehicles in the captured images are relatively stable, and there will be no drastic changes in the object position caused by camera movement. No occlusion ensures that the features of the vehicle can be fully presented,

avoiding the feature loss caused by occlusion.

Based on the above scenario characteristics, there are clear bases for simplifying the YOLOv3 model: first, the vehicle scale is relatively stable, so the complex multi-scale detection mechanism of the original model is not needed, and the number of feature map scales can be appropriately reduced; second, the scenario is single, with less background interference, and the difference between vehicle features and background is obvious, so the backbone feature extraction network does not need to be too deep, and the number of convolutional layers can be reduced to reduce the amount of computation; third, the object category is single (only vehicles need to be detected), so the structure of the classification branch can be simplified to further improve the operation efficiency.

3. Design of Simplified YOLOv3 Model

3.1. Simplification of Backbone Network

The Darknet-53 network of the original YOLOv3 contains 53 convolutional layers. Although it has strong feature extraction ability, it has a large amount of computation. For the parking lot scenario, this paper simplifies the backbone network to Darknet-19, reducing the number of convolutional layers to reduce the computation cost. The simplified Darknet-19 includes 19 convolutional layers and 5 maximum pooling layers. Its specific structure is: 3×3 convolution kernels are used for feature extraction. After each maximum pooling layer (with a stride of 2), the size of the feature map is halved and the number of channels is doubled. At the same time, Batch Normalization (BN) layers and Leaky ReLU activation functions are added after the convolutional layers to speed up the model training and prevent overfitting.

Compared with Darknet-53, Darknet-19 reduces the number of residual connections, and only retains a small number of residual blocks in the latter part of the network to ensure the ability of deep feature extraction. Experiments show that the simplified backbone network can effectively extract vehicle features in the parking lot scenario, and the amount of computation is reduced by 35%.

3.2. Optimization of Feature Fusion Layer

The original YOLOv3 adopts a three-scale feature fusion strategy to adapt to the detection needs of objects of different scales. However, in the parking lot scenario with a fixed viewing angle, the scale variation range of vehicles is small, mainly focusing on medium and small scales, and large-scale vehicles (such as large trucks) appear infrequently. Therefore, this paper simplifies the feature fusion scale to two scales, retaining the medium-scale feature map (26×26) and the small-scale feature map (52×52), and removing the detection branch of the large-scale feature map (13×13).

In terms of feature fusion method, the simplified model still adopts the combination of upsampling and concatenation: the medium-scale feature map is dimensionally reduced through a 1×1 convolution kernel, then upsampled by 2 times, and concatenated with the small-scale feature map to obtain the fused feature map. The fused feature map contains both the detailed features of the vehicle (from the small-scale feature map) and the semantic features (from the medium-scale feature map), which can meet the needs of vehicle detection in the parking lot. By simplifying the feature fusion scale, the amount of computation of the model is further reduced by 15%.

3.3. Adjustment of Prediction Branch

The prediction branch of the original YOLOv3 needs to predict the probabilities of multiple

categories (such as 80 categories in the COCO dataset). However, in the parking lot vehicle counting scenario, only the "vehicle" category needs to be detected, so the prediction branch can be simplified. The output dimension of the category probability in the prediction branch is adjusted from 80 dimensions to 1 dimension, only outputting the probability of the vehicle category, which reduces the number of model parameters and the amount of computation.

At the same time, considering that the shape of the vehicle bounding box in the parking lot scenario is relatively fixed, the number of bounding boxes predicted by each grid cell is reduced from 3 to 2. The K-means clustering algorithm is used to re-cluster the size of the vehicle bounding boxes in the parking lot, so as to obtain the anchor box parameters more suitable for this scenario. The adjusted anchor boxes can more accurately match the shape of the vehicles in the parking lot, improving the detection accuracy while reducing redundant calculations.

4. Experimental Results and Analysis

4.1. Experimental Environment and Dataset Construction

The hardware environment of this experiment is: Intel Core i7-10700K CPU, NVIDIA GeForce RTX 3060 GPU (12GB memory), 16GB RAM; the software environment is: Ubuntu 20.04 operating system, Python 3.8 programming language, PyTorch 1.10 deep learning framework, OpenCV 4.5 computer vision library.

Since there is a lack of dedicated datasets for the parking lot scenario with fixed viewing angle and no occlusion in public datasets, this paper constructs a custom dataset through on-site shooting. Three parking lots with different fixed viewing angles are selected as the shooting scenes, and high-definition surveillance cameras (resolution 1920×1080) are used to shoot continuously for 72 hours to obtain video data including different lighting conditions (sunny days, cloudy days, nights) and different vehicle types (sedans, SUVs, vans). The video data is intercepted into images at a rate of 1 frame per second, and a total of 20,000 images are obtained.

The LabelImg annotation tool is used for manual annotation of vehicles in the images. The annotation format is the txt format supported by YOLO series algorithms. Each annotation file contains the category information and bounding box coordinates of the vehicle. The dataset is divided into a training set (16,000 images), a validation set (2,000 images) and a test set (2,000 images) in a ratio of 8:1:1, which are used for model training, parameter adjustment and performance evaluation.

4.2. Setting of Model Training Parameters

In the model training process, the Stochastic Gradient Descent (SGD) optimizer is used. The initial learning rate is set to 0.001, the momentum is set to 0.9, and the weight decay coefficient is set to 0.0005 to prevent model overfitting. The batch size is set to 16, and the number of training epochs is set to 100. To improve the generalization ability of the model, data augmentation is performed on the training set images, including random cropping, horizontal flipping, brightness adjustment, contrast adjustment and other operations.

In the training process, the Early Stopping strategy is adopted. When the loss function value of the validation set does not decrease for 10 consecutive epochs, the training is stopped to avoid model overfitting and save training time. Finally, the model with the smallest loss on the validation set is selected as the final test model.

4.3. Evaluation Indicators

To comprehensively evaluate the performance of the simplified YOLOv3 model, detection accuracy, counting error and operation speed are selected as evaluation indicators, and their specific definitions are as follows:

Detection Accuracy: Measured by Mean Average Precision (mAP). Since only the vehicle category is detected, mAP is the precision of the vehicle category. Precision is calculated as the ratio of the number of correctly detected vehicles to the number of all detection results. Recall is calculated as the ratio of the number of correctly detected vehicles to the actual number of vehicles in the image. mAP is the area under the precision-recall curve.

Counting Error: Measured by relative error, which is calculated as the absolute value of the difference between the number of vehicles counted by the model and the actual number of vehicles, divided by the actual number of vehicles, and then multiplied by 100%. A smaller counting error indicates a stronger vehicle counting ability of the model.

Operation Speed: Measured by Frames Per Second (FPS), that is, the number of images that the model can process per second. A higher FPS indicates a faster operation speed and better real-time performance of the model.

4.4. Experimental Results and Analysis

To verify the superiority of the simplified YOLOv3 model, a comparative experiment is conducted between it and the original YOLOv3 model and YOLOv2 model in the same experimental environment and dataset. The experimental results are shown in the following table1.

Table 1 Comparison of Performance Indicators of Different Models

Model	mAP (%)	Counting Error (%)	FPS (Frames/Second)	Model Parameter Quantity (M)
YOLOv2	88.6	6.8	32	19
Original YOLOv3	96.1	2.1	20	61.9
Simplified YOLOv3	95.2	2.8	28	25.3

It can be seen from the experimental results that the mAP of the simplified YOLOv3 model reaches 95.2%, which is only 0.9 percentage points lower than that of the original YOLOv3 model, indicating that the simplified model still maintains high detection accuracy and can effectively identify vehicles in the parking lot. In terms of counting error, the counting error of the simplified YOLOv3 model is 2.8%, which is slightly higher than the 2.1% of the original YOLOv3 model, but much lower than the 6.8% of the YOLOv2 model, which can meet the accuracy requirements of vehicle counting in the parking lot.

In terms of operation speed, the FPS of the simplified YOLOv3 model reaches 28 frames per second, which is 40% higher than the 20 frames per second of the original YOLOv3 model, and close to the 32 frames per second of the YOLOv2 model, which can meet the needs of real-time processing. In terms of model parameter quantity, the parameter quantity of the simplified YOLOv3 model is 25.3M, which is only 40.9% of that of the original YOLOv3 model, greatly reducing the storage cost and computation overhead of the model, and making it more suitable for deployment on resource-constrained devices.

To further analyze the performance of the simplified YOLOv3 model in different scenarios, a

special experiment is conducted for three lighting conditions: sunny days, cloudy days and nights. The results show that the model has the highest mAP on sunny days, reaching 97.5%, with a counting error of only 1.5%; the mAP on cloudy days and nights is 94.8% and 92.3% respectively, and the counting errors are 3.2% and 4.1% respectively. This indicates that the model has better performance in scenarios with good lighting conditions, but still maintains high detection accuracy in scenarios with poor lighting conditions, showing a certain degree of robustness.

5. System Implementation and Application Prospects

5.1. Implementation of Vehicle Counting System

Based on the simplified YOLOv3 model, this paper implements a complete parking lot vehicle counting system, which is mainly composed of four modules: image acquisition module, vehicle detection module, counting module and result display module.

Image Acquisition Module: Real-time parking lot images are collected through high-definition surveillance cameras with fixed viewing angles, and the collected images are transmitted to the back-end processing system with a resolution of 1920×1080. To ensure the stability of image transmission, the RTSP protocol is used for image transmission.

Vehicle Detection Module: The collected images are input into the simplified YOLOv3 model, and the model outputs the bounding box coordinates and confidence of the vehicle. The bounding boxes with confidence lower than 0.5 are filtered out through the non-maximum suppression algorithm to obtain the final vehicle detection results.

Counting Module: The number of bounding boxes in the vehicle detection results is counted to obtain the total number of vehicles in the current parking lot. At the same time, to avoid counting errors caused by short vehicle stay time, a time filtering mechanism is introduced: when a vehicle is detected, it is included in the total count only if it can be detected in 3 consecutive frames; when a vehicle disappears, it is deducted from the total count only if it cannot be detected in 3 consecutive frames.

Result Display Module: The total number of vehicles in the parking lot, the vehicle distribution heat map and the historical statistical data curve are displayed in real time through the Web interface. Managers can intuitively understand the operation status of the parking lot through this interface. At the same time, the system supports exporting statistical data into Excel format, which is convenient for subsequent data analysis and management.

5.2. Application Prospects

The parking lot vehicle counting method based on the simplified YOLOv3 proposed in this paper has high accuracy and efficiency in the scenario with fixed viewing angle and no occlusion. In the future, it can be expanded and optimized from the following aspects:

Multi-scenario Adaptation: Further optimize the model structure, introduce attention mechanism and adaptive illumination adjustment algorithm, improve the performance of the model in complex scenarios such as occlusion and severe weather (such as rainy days and foggy days), and expand the application scope of the system.

Parking Space-level Counting: Combine the parking space layout information of the parking lot to accurately locate the vehicle detection results, and realize the vehicle counting at the parking space level, that is, it can not only count the total number of vehicles, but also clarify the occupancy status of each parking space, providing accurate parking space guidance services for car owners.

Edge Computing Deployment: Deploy the simplified model on edge computing devices (such as embedded development boards and smart cameras) to realize local data processing and real-time

response, reduce the delay and bandwidth consumption in the data transmission process, and improve the real-time performance and reliability of the system.

Intelligent Linkage: Link the vehicle counting system with the parking lot's entrance and exit control, toll management and other systems to realize the full-process intelligent management of the parking lot. For example, when the total number of vehicles in the parking lot is close to saturation, the entrance is automatically closed; the parking fee is automatically calculated according to the vehicle's stay time to realize unmanned toll collection.

6. Conclusion

Aiming at the parking lot scenario with fixed viewing angle and no occlusion, this paper proposes a vehicle counting method based on the simplified YOLOv3 [1][6][7]. By simplifying the backbone network, optimizing the feature fusion layer and adjusting the prediction branch, the operation speed of the model is greatly improved and the number of parameters is reduced while ensuring the detection accuracy. Experimental results show that the simplified YOLOv3 model has an mAP of 95.2%, a counting error of 2.8%, and an FPS of 28 frames per second on the custom parking lot dataset. Compared with the original YOLOv3 model, its operation efficiency is increased by 40%, which can meet the needs of real-time and accurate vehicle counting in the parking lot.

The parking lot vehicle counting system implemented based on this model can provide efficient and reliable data support for parking lot management, which helps to improve the management efficiency and service quality of the parking lot. In the future, through further technical optimization and function expansion, this method is expected to be applied in a wider range of intelligent transportation scenarios.

References

[1] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[2] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[3] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[4] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[5] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Cham: Springer International Publishing, 2016: 21-37.

[6] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[7] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.