YOLOv5-LC: A Lightweight Classification Deep Learning Detector for Dining Plate

Hanqi Jiang

School of Computer and Information Technology

Beijing Jiaotong University

Beijing, China

20722010@bjtu.edu.cn

Cheng Zeng

School of Computer and Information Technology

Beijing Jiaotong University

Beijing, China

20726043@bjtu.edu.cn

Tiance Wang

School of Computer and Information Technology

Beijing Jiaotong University

Beijing, China

20722024@bjtu.edu.cn

Zhou Guan

School of Computer and Information Technology

Beijing Jiaotong University

Beijing, China

21723045@bjtu.edu.cn

Lan Zeng

School of Computer and Information Technology

Beijing Jiaotong University

Beijing, China

20722002@bjtu.edu.cn

Hao Wang

Dublin International Transportation College

Chang’an University

Xi’an, Shanxi, China

*Abstract*—Plate detection refers to the process of detecting and recognizing the type of plate. This technology has a wide range of applications in fields like catering, smart homes, and healthcare. Current there are many challenges in plate detection methods based on deep learning, such as long training time, high misidentification rate, and inability to deploy on devices. To solve this problem, we propose a lightweight plate classification and pricing algorithm called YOLOv5-LC. Our algorithm (1) has a smaller model size and faster inference speed, (2) achieves accurate plate detection and classification without sacrificing accuracy, and (3) is easier to encapsulate and de-ploy on end devices. First, we improve the anchor box generation process of YOLOv5 and prune and distill the network. Second, we use an improved focal loss as the classification loss function to address the issue of difficulty in distinguishing similar samples. We also introduce Alpha-IoU to improve the convergence speed and accuracy of the algorithm. To evaluate the detection capability of the model, we conducted experiments on an NVIDIA RTX 3060Ti GPU, and the results showed that YOLOv5-LC can achieve a model volume of 1.7M (9.2% of YOLOv5) and better detection accuracy (2.3% mAP increase).

Keywords-component; dining plate classification; YOLOv5; lightweight detector

# Introduction

In recent years, the application and development of plate detection technology has become increasingly widespread, bringing new opportunities and challenges to the restaurant industry. Plate detection refers to the use of computer vision technology to identify and detect plates in a restaurant environment. The development of plate detection technology aims to improve restaurant service efficiency, reduce service costs, and enhance customer dining experiences by automatically identifying the location and number of plates. The main significance of plate detection technology is to improve the efficiency and quality of restaurant services, while reducing the workload and cost of service staff. With the increasing demand for restaurant services, traditional manual service methods are no longer able to meet people's needs. Therefore, the restaurant industry has begun to explore the use of intelligent technologies such as artificial intelligence and machine vision to improve service efficiency and quality. Plate detection technology is an important component of intelligent restaurants, which can provide faster and better service experiences for customers by automatically detecting the location and number of plates. Dining plate detection technology primarily employs computer vision techniques, which process and analyze images and videos using computers to accomplish tasks such as target detection, recognition, and tracking. Currently, several technologies and methods are as follows:

(1) Deep learning, based on neural networks, has achieved significant success in target detection and image recognition, and can extract dish features and detect objects in dish detection.

(2) Single-stage target detection algorithms, such as the SSD (Single Shot Multi-Box Detector) algorithm and the YOLO (You Only Look Once) algorithm, are fast and efficient target detection algorithms that are widely applied in dinner plate detection.

(3) Multi-scale feature fusion technology integrates features of varying scales to enhance target detection accuracy, and can be utilized in dish detection to improve performance.

The advent of deep learning technology has facilitated rapid progress in dinner plate detection technology. In 2015, Redmon et al. introduced the YOLO algorithm, which constituted a significant breakthrough in plate detection. Subsequently, in 2016, Ren et al. proposed the Faster R-CNN algorithm, which further enhanced the accuracy and robustness of plate detection.

Our main contributions are as follows:

1. Conduct a series of ablation experiments on YOLOv5, including (1) using smaller Flops to reduce memory usage and parameter count, and (2) pruning and adding shuffle channels to the YOLOv5 head to increase network inference speed.

2. Remove the Focus layer and avoiding the repeated use of the slice operation make the model easier to deploy and more suitable for processors based on ARM architecture.

3. Avoid using C3 layers multiple times and high-channel C3 layers.

4. Introduce Alpha-IoU and improved focal loss, further improves the recognition accuracy of the model, ensuring its performance in complex scenarios.

The remaining materials are arranged as follows. Section 2 provides an overview of the state-of-the-art research on plate detection technology in recent years. Section 3 describes the principles and specific implementation process of the algorithm.

# Methods

## Data Augmentation

The input of YOLOv5-LC adopts the Mosaic data augmentation method, which is similar to YOLOv5 and YOLOv4. Mosaic data augmentation has several advantages in addressing the issue of imbalanced target sizes in the dataset:

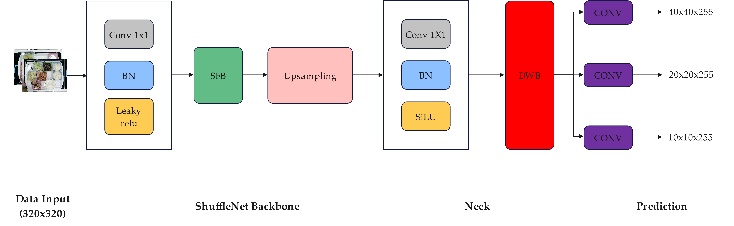
1. By randomly using 4 images and performing scaling and stitching, the dataset has been greatly enriched, especially with the increase in small targets due to random scaling, which further improves the robustness of the network.

2. Compared with regular data augmentation, the Mosaic augmentation can process data from 4 images simultaneously with the same input size, reducing the demand for GPU usage, and achieving good performance even with a single GPU.

3. YOLOv5-LC adopts the adaptive Anchor calculation method, which is the same as YOLOv5. This algorithm requires fixed Anchors to be set for different datasets.

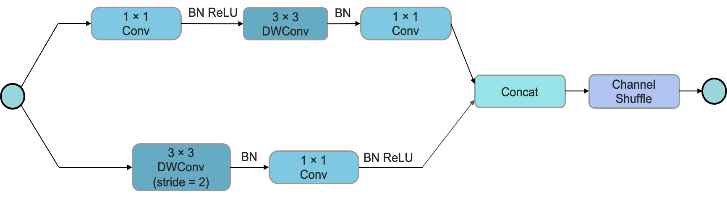
## Network Structure of YOLOv5-LC

To achieve the design philosophy of lightweight, YOLOv5-LC removed the FOCUS layer in the network architecture design and used convolution instead to reduce the number of floating-point operations and parameters, thereby reducing computation and improving speed while ensuring that down-sampling information is not lost. Through experiments, we found that replacing the Focus network layer with convolution can achieve better performance while avoiding the use of slice operations. Particularly, for chips that do not contain GPU or NPU acceleration, frequent slice operations can occupy a large amount of cache space and increase the computational burden.



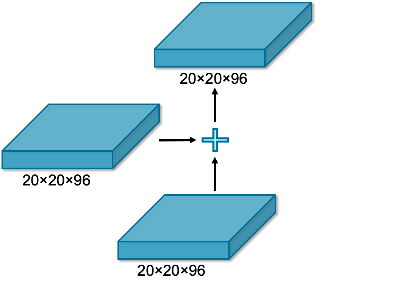
1. Network architecture of YOLOv5-LC

In the backbone section, we opted for ShuffleNet v2 and employed the Channel Split operation to divide the entire feature map into two groups, thereby increasing the number of groups used in convolution to prevent excessive increase in MAC (Multiply-Accumulate) caused by the overuse of group convolution. In addition, YOLOv5-LC also removed the 1024 conv and 5×5 pooling layers from the shufflenetv2 backbone in the Backbone module. The following figure illustrates the network architecture of ShuffleNet v2.



1. Network architecture of ShuffleNet v2

In the Neck module, we adopted the FPN+PAN structure. To optimize memory access and utilization, we chose to use the same number of channels (96 channels) for the Neck module. Furthermore, to further optimize memory usage, we opted to use the original PANet structure and replaced the cat operation in YOLOv4 with an add operation.



1. PAN architecture of YOLOv5-LC

We used the network head of YOLOv5 and performed pruning in the Head module of the network structure.

## Loss Functions

YOLOv5-LC loss function mainly consists of two parts: classified loss and bounding box regression loss. Alpha-IoU was introduced as the bounding box regression loss of the model, which was calculated as follows:

The YOLOv5 model adopts the CIoU as its regression loss function. We have re-placed this portion with Alpha-IoU to improve bounding box regression accuracy and further optimize the model's recognition accuracy.

In order to solve the problem of category imbalance and classification difficulty difference in classification, Kaiming proposed a kind of focal loss in "Focal Loss for Dense Object Detection", which improves the effect of image physical detection. From another perspective, we propose a kind of loss with similar function .

We write the cross-entropy loss function.

Where , and is the predicted value.



Using the sigmoid function:

Then we have:

We modify the loss function of cross entropy. First introduce the unit step function.

Since in principle the threshold only needs to be greater than 0.5, we can get a new loss function using the function.



That is:

Since the is non-differentiable, we use some differentiable functions to approximate this non-differentiable function.



Also, we have

We combinate the equation (10) with (11), the final loss function is as follows:

Regarding the selection of the classification loss function, we have opted for an improved version of the Focal loss, which addresses the issue of imbalanced samples.

Through the coordinated interaction of the focusing factor and weighting factor, EFL significantly improves the performance of the baseline model.

# Experiment

## Datasets and Training Setup

The insufficiency of publicly accessible datasets for dish detection necessitated the gathering of a corpus of approximately 100,000 sample images from the dining hall of Baidu Online Network Technology (Beijing) Co., Ltd. This compendium comprised 17 distinct varieties of dishes, including but not limited to white round bowls and purple square bowls. After meticulous curation and categorization of the dataset, we intend to make it available to the public. The dataset's contents are illustrated in Figure 4.



1. Example images of dataset.

## Results

The detection results of YOLOv5-LC on the food tray dataset are presented in the following figure. It can be observed that the model is able to accurately recognize different types of food trays in various complex scenes. The following Figure is shown.



1. Verify Set Actual Label



1. Validation Set Forecast Label

The figures illustrates that the recognition and prediction of plates containing di-verse food items are characterized by a high level of accuracy. It can correctly identify empty plates, partially occluded plates, plates with food color similar to that of other plates, plates with food color similar to that of other plates, plates with similar back-ground and difficult to separate, plates with incomplete contour, smaller plates in photos, plates with less food quantity, and plates with more food quantity.

## Effects of different equipment models

In addition, we executed the encapsulated models on several devices, including NVIDIA, Intel, and Raspberry Pi, and documented the inference speeds of YOLOv5-LC and YOLOv5s on these respective platforms. Table X below presents the results obtained from the experiments. The findings reveal that our lightweight model outperforms YOLOv5s with significantly improved inference speeds. Therefore, our model is more suitable for industrial scenarios.

1. Comparison of effects of different equipment models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Equipment* | *Computing backend* | *Input* |  |  |
| Inter | @i5-10210U | 640×640 | 131ms | 46ms |
| Nvidia | @RTX 2080Ti | 640×640 | 14ms | 15ms |
| Redmi K30 | @Snapdragon 730G | 320×320 | 163ms | 27ms |
| Xiaomi 10 | @Snapdragon 865 | 320×320 | 163ms | 10ms |
| Raspberrypi 4B | @ARM Cortex-A72 | 320×320 | 371ms | 84ms |

Based on the quantitative analysis of the experimental data, it can be inferred that our model has improved the speed by approximately 64.89% on the Inter device when compared to YOLOv5s. Conversely, on Nvidia devices, the speed is approximately 7.1% lower than YOLOv5s. On the Redmi K30 device, the speed is approximately 83.44% higher than YOLOv5s. For Xiaomi 10 devices, the speed is approximately 93.87% faster than YOLOv5s. Finally, on the Raspberrypi 4B device, the speed is approximately 77.36% faster than YOLOv5s. The average interference speed of YOLOv5s carried by various devices is 168.4ms, whereas the average interference speed of our model carried by different devices is 36.4ms, indicating an average speed increase of 78.38%. Consequently, our model demonstrates more efficient performance when compared to the conventional YOLOv5s model.

## Model Evaluation

PR curve is an important indicator to reflect the prediction. Its significance is that we can intuitively see the change rate of precision with recall. We evaluated the performance of the model using the PR curve, with the following specific evaluation criteria:

Precision (P) is calculated by

Recall (R) is calculated by

The precision-recall curve can be derived by plotting precision as a function of re-call, where recall is the horizontal axis and precision is the vertical axis.

## Ablation Experiments

In order to investigate the effectiveness and optimal performance of our model, we conducted a series of pruning experiments as shown in the following table. After pruning the model, we compared its performance on two image sizes (320×320 and 640×640) and found that our proposed YOLOv5-LC achieved a significantly better mAP.

1. Comparison of Inference Speeds Among Different Neural Network Models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Model* | *Input size* | *Flops* | *Params* | *mAP@0.5* | *mAP@.5:0.95* |
| yolo-fastest | 320×320 | 0.25G | 0.35M | 25.4 | - |
| YOLOv5-LC (Ours) | 320×320 | 0.73G | 0.78M | 34.6 | - |
| NanoDet-m | 320×320 | 0.72G | 0.95M | - | 20.8 |
| yolo-fastest-xl | 320×320 | 0.72G | 0.92M | 34.3 | - |
| YOLOv5s (6.0) | 640×640 | 16.8G | 7.26M | 53 | 39.2 |
| YOLOv5-LC (Ours) | 640×640 | 15.4G | 5.43M | 58.6 | 43.1 |

Furthermore, we also investigated the performance differences of the model when using different loss functions. Considering the trade-off between accuracy and speed, we ultimately selected Alpha-IoU as our baseline.

1. Effect of different loss functions

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *loss function* | [*mAP@0.5*](mailto:mAP@0.5) | *FPS* |
| YOLOv5-LC | focal loss/CIoU | 53.1 | 95.4 |
| focal loss/DIoU | 53.4 | 102.5 |
| focal loss/GIoU | 53.2 | 98.7 |
| focal loss/RIoU | 55.3 | 111.9 |
| focal loss/Alpha-IoU (Ours) | 58.6 | 109.4 |

# Conclusions

This paper presents a lightweight object detection model, YOLOv5-LC, based on YOLOv5 for efficient classification tasks. The proposed method effectively addresses the issues of slow inference speed and difficulty in deployment faced by the YOLO model, and achieves promising results in plate detection tasks. YOLOv5-LC achieves a high mAP@0.5 of 82.4% on our dataset, with a model size of only 1.7M and a maximum inference speed of 109.4 FPS. The model exhibits high robustness in scenarios with occlusion or a large number of plates. Moreover, it requires relatively low hardware specifications and achieves a fast detection speed with a maximum inference speed of 14 FPS when deployed on a Raspberry Pi 4B, laying a solid foundation for practical applications of the model. In future work, we will continue to improve the model to enable its deployment in a wider range of tasks and achieve faster and more accurate detection performance.

##### Acknowledgment

Thanks to the Computer Laboratory at Beijing Jiaotong University for providing the experimental equipment, including GPUs, and to the staff at the experimental center for their cooperation during the experiments.

##### References

1. Accenture. (January 1, 2021). Share of retailers who adopted Radio Frequency Identification (RFID), by number of omnichannel experiences offered in 2020 [Graph]. In Statista.
2. J. Lu, "The application of smart dinner plates based on RFID technology in college canteens", Inf. Syst. Eng., vol. 27, no. 9, pp. 85-99, 2014.
3. Nie, X. (2013) Security Threats and Countermeasures in the Internet of Things Based on RFID. Applied Mechanics and Materials, 380–384, 2817–2821. Trans Tech Publications, Ltd.
4. Xiao, D. (2020, January). Research on the Application of Internet of Things Technology in the Construction of University Intelligent Health Canteen. In 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS) (pp. 171-175). IEEE.
5. G. Su, "Image-based Chinese dish segmentation and recognition", 2019.
6. Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0\_2
7. Cao, X., Chen, Y., & Cao, W. (2022, June). Proximal PanNet: A Model-Based Deep Network for Pansharpening. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 1, pp. 176-184).
8. He, J., Erfani, S., Ma, X., Bailey, J., Chi, Y., & Hua, X. S. (2021). $\alpha $-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. Advances in Neural Information Processing Systems, 34, 20230-20242.
9. Xu, B., Chen, M., Guan, W., & Hu, L. (2023). Efficient Teacher: Semi-Supervised Object Detection for YOLOv5. arXiv preprint arXiv:2302.07577.
10. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).
11. Meng, L., Li, H., Chen, B. C., Lan, S., Wu, Z., Jiang, Y. G., & Lim, S. N. (2022). Adavit: Adaptive vision transformers for efficient image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12309-12318).
12. Yang, J., Li, C., Zhang, P., Xiao, B., Liu, C., Yuan, L., & Gao, J. (2022). Unified contrastive learning in image-text-label space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19163-19173).
13. Toker, A., Kondmann, L., Weber, M., Eisenberger, M., Camero, A., Hu, J., ... & Leal-Taixé, L. (2022). Dynamicearthnet: Daily multi-spectral satellite dataset for semantic change segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 21158-21167).
14. Yang, J., Liu, S., Li, Z., Li, X., & Sun, J. (2022). Real-time object detection for streaming perception. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5385-5395).
15. Kim, J., Hwang, I., & Kim, Y. M. (2022). Ev-tta: Test-time adaptation for event-based object recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17745-17754).
16. Zhang, H., Meng, Y., Zhao, Y., Qiao, Y., Yang, X., Coupland, S. E., & Zheng, Y. (2022). Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18802-18812).
17. Liu, J., Ji, P., Bansal, N., Cai, C., Yan, Q., Huang, X., & Xu, Y. (2022). Planemvs: 3d plane reconstruction from multi-view stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8665-8675).