Efficient Dynamic Inference in YOLOv9 for Small License Plate

Dian Ning, Dong Seog Han*

School of Electronic and Electrical Engineering

Kyungpook National University

ningdian@knu.ac.kr, *dshan@knu.ac.kr

Abstract

This paper presents a dynamical YOLOv9 architecture that establishes a speedier model for real-time tiny vehicle license detection in edge computing environments. YOLO shows outstanding application prospects in image classification and object detection. The update to YOLOv9 integrates an auxiliary network, which enables domain-wide weight updates and improves performance. However, YOLOv9 still raises challenges regarding heavy computational demands and low inference speed on embedded devices. To address these problems, we incorporate an integration: A dynamic latency-aware mechanism that skips useless backbone layers or channels based on real-time latency monitoring. To prove our new module advancement, we also collected a dataset covering license plates taken in different sizes and within various environments. On the dataset, the optimised architecture achieves 14.67ms inference latency and maintains 88.3% detection accuracy compared to baseline YOLOv9-t. Experimental results demonstrate that the proposed YOLOv9 enhance the time test inference of vehicle license detection, making it more suitable for real-world embedded applications.

Ⅰ. Introduction

In recent years, there have been significant advancements in deep learning models for object detection. YOLO model has emerged as a leading approach for object detection applications nowadays, with various versions, from YOLOv1 to YOLOv9. Ultralytics created YOLOv5, which is an improvement from YOLOv3 and defined the main architecture for the backbone, neck, and head. It has gradually been adopted in the industry. Currently, YOLO models widely used in embedded devices are still based on YOLOv5, largely due to computational constraints that limit the use of newer versions. YOLOv9 is the latest version with programmable gradient information (PGI) to update the low-level layer gradients easily. It also uses a module called generalised efficient layer aggregation network (GELAN), which extends ELEN's method for convolutional layers. This allows the addition of any desired computational blocks within the cross-stage partial (CSP) module for improved feature extraction. To guarantee accurate performance, the lightest YOLOv9-t consumes more than 71% GFLOPS compared to YOLOv5-n. This indicates that computational cost and inference speed are still challenging when deploying YOLOv9 on embedded devices.

To address the aforementioned challenges, this paper presents two key enhancements to the YOLOv9 model, which improve computational efficiency without sacrificing detection accuracy: A dynamic latency-aware mechanism in the backbone of YOLOv9. It takes two parts in the repeated normalized cross stage partial (RepNCSP) module, using channel masker to synergistically suppress redundant inference. Relying on the two maskers, the inference model takes a lighter and faster information flow as it passes through the improved RepNCSP modules in the model backbone and neck.

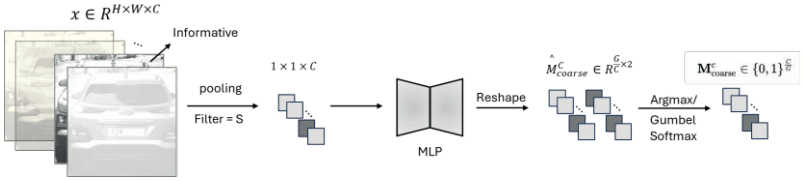

Fig.1. The architecture of RepConvN

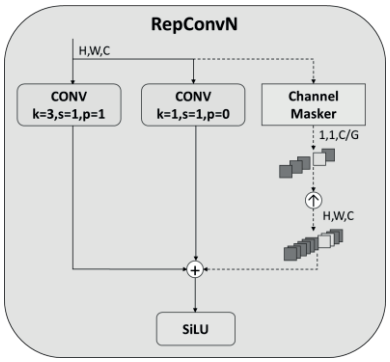

Fig.1. The architecture of RepConvN.

Ⅱ. Method

We built a simulator because the existing CUDA library (e.g., cuDNN) does not support dynamic inference in the current YOLOv9 model. This simulator serves as a foundation for analysing dynamic model contributions and optimising latency across hardware platforms [2].

In YOLOv9, the RepNCSP modules are used in both the backbone and the bottleneck to improve the feature representation during training. Each RepNCSP module is composed of RepNBottleneck blocks, which in turn consist of RepConvN and standard convolutional layers. Although this design improves accuracy, it introduces additional complexity during inference.

RepConvN combines a 1×1 convolution and a 3×3 convolution during training but only retains the equivalent 1×1 convolution after re-parameterisation for inference.

To solve this, we use channel masks before the 1×1 convolution during inference to suppress irrelevant information in the input features and improve efficiency. Channel mask mechanism operates as follows:

1. Global Average Pooling is performed on the input feature map to compress the spatial dimensional information into a channel description vector.
2. This channel vector is passed through a small multilayer perceptron (MLP) or a linear transformation to generate a preliminary channel importance prediction.
3. The preliminary importance vector is binarised by a Gumbel Softmax or Argmax operation to form a channel mask that marks which channels are retained (value 1) and which are skipped (value 0).
4. During the inference process, only the channels with a mask value of 1 will enter the 1×1 convolutional branch and participate in the computation; the rest of the channels will be directly skipped through the Skip Connection, thus reducing the computation overhead.
5. If needed, the final output can be restored to its original resolution by an upsampling operation for subsequent network modules.

Ⅲ. Experiences

Our experience is set in RTX3090, and the model's implementation details follow the YOLOv9 initialisation settings.

We use the dataset prepared by ourselves for the tiny license plate recognition. It includes different scenes such as city road, motorway, mountain road. This dataset has a total of 3000 images, and its train, validation and test parts are 80%, 10% and 10% respectively.

To test the channel mask performance, we used the YOLOv9 model for comparison.

Channel masks	Perception	Recall	Inference time
Y	0.881	0.628	15.32ms
N	0.883	0.659	14.67.ms

Table 1. Comparing the standard YOLOv9-t with our Model

As shown in Table 1, the channel masker reduces the time required for the inference step. Also, the channel masking enhances useful information, our results are better than those of the standard YOLOv9-T model.

IV. Conclusion

In this study, we analyzed the possibility of reducing unnecessary features in YOLO models to achieve faster inference speeds. This study provides valuable experience in dynamic model calculation, and we plan to widely test the results in the future.

ACKNOWLEDGMENT

This work was supported by the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2020-II201808).

REFERENCES

[1] Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. "Yolov9: Learning what you want to learn using programmable gradient information". In European conference on computer vision (pp. 1-21).

[2] Han, Y., Liu, Z., Yuan, Z., Pu, Y., Wang, C., Song, S., & Huang, G. (2024). "Latency-aware unified dynamic networks for efficient image recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence.