Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.38059

Energy-Efficient Optimization of YOLO Architectures for Sustainable Autonomous Driving: A Green AI Perspective

Yicheng Duan

School of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an, China
yicheng.duan@stu.xupt.edu.cn

Abstract. With the development of autonomous driving technology, the YOLO series of object detection models have been applied in many vehicle perception systems due to their fast speed and stable performance. However, in practical applications, whether a model is good cannot only be judged by its detection accuracy, but also needs to consider its power consumption, heat generation and deployment cost when running for a long time. As Green AI gains increasing attention, energy-efficient inference with maintained accuracy is a key challenge in autonomous driving. From the perspective of Green AI, this paper examines the evolution of YOLO models and their use in autonomous driving, investigates architectural, inference, and deployment variations across versions, and reviews key optimization methods, such as lightweighting, quantization, dynamic inference, and hardware-software optimization Moreover, it points out that many current evaluation methods focus more on accuracy and speed, and do not consider energy consumption and carbon emissions comprehensively and deeply enough. Therefore, establishing a green evaluation framework that considers accuracy, latency, energy consumption and carbon footprint simultaneously is necessary at present.

Keywords: Green AI, YOLO, Autonomous Driving, Energy Efficiency Optimization, Green Evaluation

1. Introduction

The implementation of the autonomous driving system relies on the stable perception ability of the road environment. Among them, target detection, as the basic task of the perception module, is mainly employed for detecting key targets such as vehicles, pedestrians, cyclists, traffic signs and signals [1]. However, in real road scenarios, factors like light changes, weather disturbances, and target occlusion are common, making the detection task highly uncertain and demanding higher requirements for the model's robustness and real-time performance. The YOLO series, due to its simple structure, high inference efficiency, and good engineering deployment adaptability, has been widely used in vehicle perception tasks. With version iterations, it has continuously optimized in aspects such as feature expression, multi-scale detection, and lightweight design [2]. Current research still mainly focuses on accuracy and speed, and pays insufficient attention to issues such as energy consumption, thermal management, and carbon emissions during the long-term operation of the model. There is still room for further research on green and efficient deployment [3]. According

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.38059

to this, the paper mainly deals with the applications of YOLO in autonomous driving and thoroughly investigates its structural changes, deployment characteristics and energy saving strategies. By means of a literature survey and comparative study, the paper mainly summarizes the general methods of model optimization such as lightweighting, quantization compression and dynamic inference, and also discusses the shortage or deficiency of energy efficiency indicators in the current evaluation system. Thus, this paper examines the performance of YOLO in autonomous driving target detection, assesses its energy consumption and carbon emissions from an analytical perspective, and evaluates the model from the perspective of green computing. The aim is to provide energy-saving suggestions for the design and evaluation of perception models, and promote their long-term application in the automotive field.

2. Green AI and sustainability requirements for autonomous driving

2.1. Definition and key evaluation metrics of Green AI

As the number of AI models continues to increase, the computing resources and energy required during training and recognition processes are also constantly growing. Green AI refers to improving the efficiency of algorithms and systems while maintaining performance, to reduce computing and energy consumption [3]. The evaluation criteria are divided into three categories: the computational complexity indicators (the number of model parameters and the number of floating-point operations) represent the computational load of the model itself; the deployment energy efficiency indicators (the energy consumed per recognition, recognition delay, energy efficiency ratio (TOPS/W), etc.) reflect the energy consumption level that the model can achieve on specific hardware; the environmental impact indicators (carbon dioxide equivalent (CO_2e) generated during training and recognition, etc.) reflect the degree of impact that the model has on the environment [4]. These indicators can provide detailed evaluations of the energy consumption and operating costs of models on vehicle platforms, which is helpful for choosing perception models and optimizing the system in autonomous driving.

2.2. Energy efficiency constraints and deployment demands for autonomous driving

The autonomous driving perception model for vehicles is deployed on the vehicles themselves, rather than in a general computing environment. Therefore, their energy efficiency is different from that of usual computer vision tasks. Because of the high real-time requirement, the perception models should finish the visual tasks such as object detection and localization in a very short time to guarantee the subsequent planning and control, otherwise it may cause safety problems [1]. Bthe onboard computing devices have strict power consumption budget constraints. They must operate for a long time under low power conditions, while also taking into account issues such as battery life, thermal management, and hardware lifespan. Thus, the model size, computational complexity and memory access overhead restrict the feasibility of the deployment [5]. Under conditions of high temperature, vibration and long-term continuous operation, excessive energy consumption may cause the chip frequency to drop, affecting the perception performance and system stability [5]. In addition, the autonomous driving systems often have multi-sensor cooperation characteristics and the vehicle should process information from different sensors like cameras, lidar and millimeter-wave radar simultaneously. This will increase the computational and scheduling demands. Among them, the camera-based target detection (like the YOLO series) only represents a part of the visual perception and its energy consumption should be evaluated together with the lidar point cloud and

DOI: 10.54254/2755-2721/2026.38059

millimeter-wave radar processing modules to maintain the stable operation of the whole perception system [1, 6].

2.3. Green evaluation dimensions and unified measurement methods

The evaluation of autonomous driving perception models not only includes performance indicators such as accuracy and frame rate but also needs to incorporate resource consumption during training and deployment. In the training phase, the total energy consumption and carbon emissions are mainly considered as the important indices, for example, the total carbon footprint during the training period; in the execution phase, the emphasis is on operational energy efficiency, including energy consumption per frame (J/frame) and, at the vehicle-system level, energy consumption per kilometer (J/km) [4]. These indices measure the model's energy consumption on vehicle platforms and reflect the operational efficiency and deployment costs. To ensure the comparability and repeatability of the evaluation results, the hardware configurations (like Jetson platforms or onboard NPUs) and environmental conditions (such as temperature) should be noted during the test. This is due to the fact that energy consumption and latency are dependent on the hardware and need a specific deployment platform for meaningful comparison. Furthermore, model compression techniques like quantization and pruning can be combined to evaluate their effects on accuracy and avoid discrepancies when comparing models under different optimization approaches. Therefore, one can refer to the green AI evaluation indices like Energy-Accuracy Trade-off, FPS-per-Watt (or Frames-per-Joule) which indicate the real energy efficiency of the inference, to establish an energy efficiency comparison framework with a unified hardware and testing procedure, thus promoting fair comparisons among different algorithms and deployment schemes.

3. Evolution of YOLO architectures and energy efficiency optimization

3.1. Backbone networks and feature efficiency

As the key component influencing YOLO's energy efficiency, the backbone has evolved from merely deepening the network to balancing accuracy and energy consumption. For example, CSPDarknet splits the feature stream into two branches through the Cross Stage Partial (CSP) structure, reducing redundant gradient propagation and computational redundancy. It also decreases feature map memory access overhead, thereby improving overall energy efficiency [7]. In terms of lightweight design, depthwise separable convolution is widely adopted in lightweight networks such as MobileNet and ShuffleNet. Some lightweight YOLO variants also incorporate this structure, which decomposes standard convolution into depthwise convolution and pointwise convolution to reduce parameters, FLOPs, and memory access overhead [8, 9]. In addition, re-parameterization techniques adopt a multi-branch structure during training to enhance feature representation capability, and the branches are equivalently merged into a single structure during inference, thereby reducing computational graph complexity and inference energy consumption while maintaining accuracy [10]. Generally speaking, the evolution path of YOLO backbone network can be summarized as follows: from deepening the network to improving the expressive ability, gradually turning to reducing the computational and memory overhead through structural reconstruction, convolution lightweight and inference stage integration, so as to achieve better end-side deployment energy efficiency [11].

DOI: 10.54254/2755-2721/2026.38059

3.2. Detection architectures and inference optimization

The detection Head not only affects the detection accuracy, but also directly determines the inference delay, memory access overhead and overall energy consumption. The decoupled head proposed in YOLOX separates classification and regression tasks, reducing optimization conflicts and improving training convergence and detection accuracy [12]. In the aspect of structural optimization, YOLOv7 introduces a re-parameterization strategy, which preserves the multi-branch structure in the training stage to enhance the expressive ability, and fuses the multi-branches into a single path in the inference stage, effectively reducing the number of operators and inference delay [10]; YOLOv8 optimizes the feature gradient flow via the C2f module, improving the computational efficiency and potentially reducing the inference cost [13]. Moreover, YOLOv10 presents an NMS-free structure which avoids the candidate box screening step in the usual post-processing, decreasing extra calculations and waiting time. This makes the inference process simpler and greatly boots the real-time performance [14]. Typically, the evolution of the YOLO detector involves enhancing its single-stage detection capability, gradually reducing the reliance on post-processing, shortening the inference steps and reducing memory usage, thereby improving the energy efficiency of the end system [13].

3.3. Deployment adaptation and energy efficiency performance

YOLO deployment optimization mainly focuses on software layer adaptation and hardware layer adaptation. At the software level, the Ultralytics toolchain supports exporting models to intermediate formats such as ONNX and integrating with TensorRT for inference optimization [13, 15]. Through operator fusion, constant folding, and operator rewriting, computational and memory overhead are reduced, enhancing inference efficiency and energy efficiency. At the hardware level, autonomous edge computing platforms are constrained by factors like bandwidth, DDR memory access capability, computational resources, and power budgets. Therefore, network scale, input resolution, numerical precision, and operator implementations need to be adapted according to specific hardware platforms [5]. In addition, studies on efficient YOLO detectors demonstrate that optimizing feature information interaction mechanisms and network structures can maintain detection performance while reducing inference overhead, thus providing insights on the real-time deployment of the YOLO model on edge devices [16]. In practice, the aforementioned optimizations are usually combined with strategies such as quantization and mixed-precision inference to reduce memory usage and enhance throughput.

4. Energy-efficient optimization strategies of YOLO for green autonomous driving

4.1. Model lightweighting and compression optimization

In autonomous driving scenarios, model lightweighting reduces parameter size while addressing real-time performance, computational resource, and energy constraints for edge deployment. Therefore, optimization methods for model lightweighting can generally be divided into three types: structural compression, computational optimization, and knowledge distillation. First, in terms of structural compression, convolutional filters are pruned to remove redundant channels, thereby reducing the number of parameters and computational cost [17]. Second, in terms of computational compression, low-bit quantization reduces the bandwidth requirement of memory access and reduces the energy consumption of inference by reducing the bit width of parameter and feature

DOI: 10.54254/2755-2721/2026.38059

representation [18]. Finally, in terms of knowledge transfer, knowledge distillation transfers the feature representation and discrimination ability of the large-scale YOLO model to the lightweight student model, thus reducing the complexity of the model and keeping the detection accuracy as much as possible [19]. Therefore, model compression needs to achieve a comprehensive balance between accuracy, speed and energy consumption for autonomous driving.

4.2. Dynamic inference and input adaptation

The perceptual load in the autonomous driving environment changes dynamically with the road scene, so the inference process needs to be adaptive. At present, the relationship between scene complexity and computational cost is not yet well established, making complexity-aware dynamic inference an open research issue. The key challenge is to estimate scene complexity with low overhead and adapt the inference path accordingly. In scene complexity estimation, two kinds of light signals, sequential continuity and spatial sparsity, can be used. The former infers the current target density by using the detection results of adjacent frames, and the latter estimates the local target density by downsampling features from shallow layers. The computational complexity of these two methods is lower than that of complete inference, and they can be used for path switching decision. Based on complexity estimation, dynamic inference adjusts computational load via early-exit mechanisms and adaptive resolution strategies. The early-exit mechanism can skip unnecessary computations in consecutive low-complexity frames. Yet, for YOLO's multi-scale detection architecture, feature layer selection or lightweight prediction branches should be considered to avoid the degradation of small-object detection caused by premature termination of deep feature extraction. Adaptive resolution adjusts input resolution or network depth according to scene complexity. For regional resolution adaptation, techniques like regional cropping, feature alignment, and result fusion are required to maintain YOLO pipeline compatibility. This mechanism not only guarantees the accuracy of frame-level detection, but also decreases the average inference overhead by distributing computing resources according to needs, and maintains low latency, thereby achieving adaptive inference for scene changes [20].

4.3. Algorithm-hardware co-design and deployment

The energy efficiency of the model is influenced by the algorithm design as well as the hardware it uses. At the algorithm stage, by means of operator fusion, parallel computation improvement and post-processing procedure simplification, the redundant computations and data processing burden can be decreased. In terms of hardware, the acceleration capabilities of special operators provided by TensorRT or automotive NPU can be utilized to further reduce the inference latency and energy consumption for each task [15]. Therefore, in conjunction with the hardware-aware optimization techniques, the model structure can be adjusted to the computational graph, thus enhancing the utilization of computational resources on various platforms and minimizing the waste of resources. In the actual deployment process, the automated Benchmarking process is introduced to continuously evaluate the inference speed, power consumption and energy efficiency ratio, so as to quantify the effects of different optimization strategies and support the iterative optimization of the model [13].

DOI: 10.54254/2755-2721/2026.38059

5. Green evaluation framework and future development directions

Existing autonomous driving perception models are mainly evaluated based on mAP and inference speed, while lacking a unified evaluation of energy consumption and on-board deployment capability. The variations in the computing structures and power consumption of different hardware systems make it difficult to compare the energy efficiency of the models [3, 4]. Hence, it is essential to develop a multidimensional assessment scheme, in which the perception performance is measured by mAP and latency, the operational overhead is calculated by the energy consumption per frame, and the hardware resource consumption is indicated by parameters like memory bandwidth utilization [4, 5]. Two main problems remain to be solved for practical application. Firstly, the energy consumption measurements are different on different platforms due to variations in power sampling, accuracy, memory access and thermal conditions, which should be unified to reduce the differences in assessment. Secondly, it is still a difficult task to estimate the complexity of the scene in dynamic inference, because there is no definite connection between complexity and computation, leading to inappropriate adaptive inference. After overcoming these difficulties, a cross-platform benchmark evaluation can be carried out to facilitate fair model comparisons with the same data sets and fixed hardware conditions, thus improving the repeatability and comparability of the results [2, 4]. At the same time, the promotion of this direction relies on the cooperation of hardware manufacturers, data set builders and algorithm researchers, so the more feasible path in the short term is to form a local evaluation consensus on mainstream vehicle computing platforms, and then gradually expand to cross-platform unified standards.

6. Conclusion

From the viewpoint of Green AI, this paper investigates the evolution of the YOLO structure, the limitations in its deployment, and the optimization methods in the field of autonomous driving. The research shows that the perception model should ensure high detection accuracy, meet the vehicle power supply limitations, thermal design requirements, and long-term operational stability. Therefore, energy efficiency has become an important evaluation metric. By analyzing the lightweight backbone network, detector structure optimization, and the deployment optimization of YOLO on different hardware platforms, it has been found that reparameterization and NMS-free design can improve the inference efficiency by reducing the computational redundancy in the inference process. Moreover, the decoupled detection head improves the training convergence and detection accuracy by adjusting the relationship between the classification task and the regression task, enhancing the practicality of edge deployment. In order to reduce the energy consumption of each inference and enhance flexibility in various scenarios with limited computing resources, further optimization can focus on model compression, flexible allocation of computing resources, and integration of algorithms with hardware.

Additionally, the standardization and repeatability of the green evaluation systems will provide consistent standards for the optimization of energy efficiency.

References

[1] Janai, J., Güney, F., Behl, A., et al. (2020). Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends in Computer Graphics and Vision, 12*(1-3), 1-308.

[2] Mao, Z. H., Zhu, J. L., Wu, X., et al. (2022). Research review of YOLO-based object detection for autonomous driving. *Computer Engineering and Applications, 58*(15), 68-77.

[3] Schwartz, R., Dodge, J., Smith, N. A., et al. (2020). Green AI. *Communications of the ACM, 63*(12), 54-63.

DOI: 10.54254/2755-2721/2026.38059

[4] Henderson, P., Hu, J., Romoff, J., et al. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research, 21*(248), 1-43.
[5] Sze, V., Chen, Y. H., Yang, T. J., et al. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE, 105*(12), 2295-2329.
[6] Lyu, P., Xu, J., Li, T., Xu, W., et al. (2021). Survey on edge computing technology for autonomous driving. *Journal on Communications, 42*(3), 190–208. DOI: 10.11959/j.issn.1000-436x.2021045.
[7] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[8] Sandler, M., Howard, A., Zhu, M., et al. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510–4520).
[9] Ma, N., Zhang, X., Zheng, H. T., et al. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *Proceedings of the European Conference on Computer Vision* (pp. 116–131).
[10] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7465–7475).
[11] Feng, S., Qian, H., Wang, H., & Wang, W. (2024). η-repyolo: Real-time object detection method based on η-RepConv and YOLOv8. *Journal of Real-Time Image Processing, 21*(3), 81.
[12] Ge, Z., Liu, S., Wang, F., et al. (2021). YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*.
[13] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8. *Ultralytics*. https://docs.ultralytics.com/models/yolov8
[14] Wang, A., Chen, H., Liu, L., et al. (2024). YOLOv10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*.
[15] NVIDIA. (n.d.). TensorRT documentation. *NVIDIA*. https://docs.nvidia.com/deeplearning/tensorrt/ (accessed 2024-06-02).
[16] Wang, C., He, W., Nie, Y., et al. (2023). Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *Advances in Neural Information Processing Systems, 36*, 51094–51112.
[17] Li, H., Kadav, A., Durdanovic, I., et al. (2017). Pruning filters for efficient ConvNets. *International Conference on Learning Representations*.
[18] Jacob, B., Kligys, S., Chen, B., et al. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713.
[19] Yang, Z., Li, Z., Jiang, X., et al. (2022). Focal and global knowledge distillation for detectors. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4633–4642. IEEE.
[20] Han, Y., Huang, G., Song, S., et al. (2022). Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(11), 7436–7456.