DOI: 10.54254/2755-2721/2026.GU33716

Comparative Study of Traffic Sign Detection Models Based on YOLOv8n

Shouming Li

College of Computer Science, Beijing University of Technology, Beijing, China
23070109@emails.bjut.edu.cn

Abstract. Nowadays, detection of traffic signs is deeply integrated into autonomous driving. This kind of task does not only require accuracy, it also ask for immediate reaction, so that safety can be kept well. For this paper, three representative object detection models are selected—Faster R-CNN, SSDlite, and YOLOv8n—to conduct a comprehensive competitive study. The experiment is carried on a dataset composed of 15 categories of traffic sign images. From the obtained result, Faster R-CNN manages to get the highest mAP@0.5, which is 92.71%, but its inference speed is the lowest one, only 9.51 FPS. Such a slow speed bring a severe limitation when it comes to realtime usage in practice. SSDlite, however, work at a faster speed, still its architecture is somewhat outdated, and this make the mAP@0.5 drop to 66.78%. Different from the above two, YOLOv8n succeed in reaching a balance between precision and velocity. It deliver a robust 89.46% mAP@0.5 and 76.64% mAP@0.5:0.95, and meanwhile the speed achieves 72.52 FPS. Not only that, analysis of the loss curve reveal that YOLOv8n owns more stability during the converging process. Take all these aspects into view, YOLOv8n seems to be the most appropriate one for edge deployment in the field of autonomous driving.

Keywords: Traffic sign detection, Object detection, YOLOv8n, Faster R-CNN, Lightweight network, real-time

1. Introduction

With the rapid development of smart driving systems, road environment perception has become critical to driving safety [1]. Traffic sign detection, the task that requires real-time identification of speed limits and prohibitions, plays an essential role in vehicle decision-making. Recently, Convolutional Neural Networks (CNNs) have dominated computer vision [2], becoming a better baseline for this task than traditional manual feature extraction methods (e.g., HOG+SVM [3]).

Current detection algorithms can generally be divided into two categories: two-stage and single-stage. Two-stage models like Faster R-CNN are known for Region Proposal Networks (RPN) [4], which can provide high accuracy in complex backgrounds, but serial computation slows their inference speed and makes it hard to meet on-board real-time requirements. To improve efficiency, some single-stage detectors reformulate detection as a regression problem, like YOLO and SSD [5, 6]. As Zou et al. noted [7], looking for the optimal trade-off between accuracy and speed remains a core research trend.

DOI: 10.54254/2755-2721/2026.GU33716

In the traffic sign detection task, models must handle variable sign scales and complex backgrounds. Early methods like fully convolutional networks can predict at the pixel level [8], but their parameter costs are massive. To ease the computation stress, lightweight strategies are adopted, such as adopting depthwise separable convolutions or neural architecture search [9, 10]. However, the practical performance of these methods in complex driving scenarios still needs a comprehensive quantitative comparison.

This paper compares three representative models—Faster R-CNN, SSDlite-MV3, and YOLOv8n using a dataset comprising 15 categories of traffic signs. The models are evaluated across multiple metrics, including detection accuracy, recall, and inference speed. Their convergence curves during training are also analyzed.

2. Method

2.1. Dataset

The traffic sign detection dataset published on Kaggle is used in this paper (https://www.kaggle.com/datasets/pkdarabi/cardetection). The dataset contains a total of 5000 traffic scene images, covering common traffic signs such as traffic lights, speed limit signs, and prohibition signs. There are 15 categories, including green light, red light, stop, and 12 speed limit signs from 10 to 120.

In the data preprocessing stage, all images are adjusted to a 416 × 416 pixel resolution. This resolution is enough to recognize small target traffic signs, but it controls the amount of calculation.

2.2. Model

This paper selects YOLOv8n, Faster R-CNN, and the SSDlite model with a MobileNetV3-Large backbone and an input resolution of 320×320 (SSDlite-MV3) for comparative experiments, covering the typical architectures of single-stage and two-stage detectors representing different technical routes in this task.

YOLOv8n, an anchor-free single-stage detector, employs a CSPDarknet backbone and a decoupled head. It was selected as a great lightweight detection method with both velocity and accuracy. It also serves as the baseline since its qualities are highly suitable for resource-constrained edge devices on vehicle.

Faster R-CNN, a classic two-stage detector, adopts a Region Proposal Network (RPN) which filters before detecting when processing. It was chosen as a high-precision benchmark to highlight the computational bottleneck that severely sacrifices speed for precision.

SSDlite-MV3 is a detector where standard backbones are replaced by MobileNetV3, using depthwise separable convolutions. Its backbone incorporate an SE attention module for refined feature extraction, and a multi-scale fusion strategy is also employed. This model can be seen as an earlier method in object detection; it offers an insight into the architectural advances that modern detectors, like YOLOv8n, have achieved.

2.3. Metrics

For a thorough evaluation on the models, three standard metrics were adopted: mAP@0.5, mAP@0.5:0.95, and Recall. The mAP@0.5 metric serve to assess the overall detection accuracy under common overlap thresholds. As for mAP@0.5:0.95, a more strict version, it evaluates how

DOI: 10.54254/2755-2721/2026.GU33716

precise the bounding box regression is. Recall is particularly critical to traffic sign detection, because missing some signs—like speed limit—could pose threats to driving safety. In addition, Frames Per Second (FPS) was used for measuring inference speed, which is required in real-world autonomous driving scenarios.

3. Result

3.1. Metric comparisons

In this paper, three mainstream target detection models, YOLOv8n, Faster R-CNN, and SSDlite, are compared under the same dataset and experimental environment. The evaluation results are shown in Table 1. The following is an analysis covering detection accuracy, positioning quality, recall, and real-time.

Table 1. Performance comparison of different models

	mAP@0.5	mAP@0.5:0.95	Recall	FPS
YOLOv8n	89.46	76.64	79.29	72.52
FasterR-CNN	92.71	74.19	78.50	9.51
SSDlite-MV3	66.78	53.94	62.61	71.42

Under the mAP@0.5 metric, Faster R-CNN achieved the highest score (92.71%), demonstrating the advantage of two-stage detectors in feature extraction. YOLOv8n followed closely (89.46%), while SSDlite-MV3 performed poorly (66.78%), suffering from complex backgrounds and small-target interference.

The leader changed under the mAP@0.5:0.95 metric. YOLOv8n (76.64%) outperformed Faster R-CNN (74.19%), showing superior bounding box regression accuracy. SSDlite-MV3 fell behind at 53.94%, showing a lack of competence in precise localization.

Regarding recall, YOLOv8n (79.29%) slightly surpassed Faster R-CNN (78.50%), and SSDlite-MV3 got 62.61%. Missed detections are critical risks, so the superior performance of YOLOv8n is of great practical value.

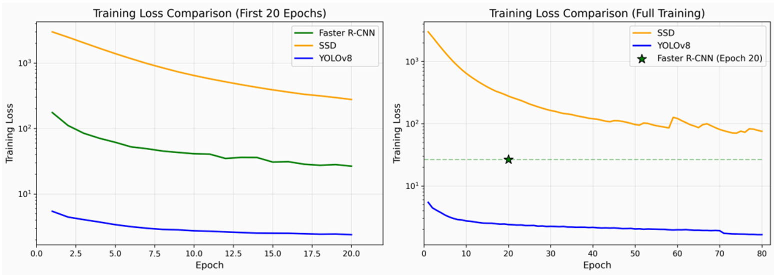
In terms of inference speed, YOLOv8n showed a dominant advantage (72.52 FPS) over Faster R-CNN (9.51 FPS), which fails to meet real-time requirements (>25 FPS). SSDlite-MV3 achieved comparable speed (71.42 FPS), but it's not enough as the sole competitiveness.

In conclusion, while Faster R-CNN is slow in speed and SSDlite-MV3 lacks accuracy, YOLOv8n achieves the optimal trade-off. Its real-time performance alongside high precision makes it the most effective model for this task.

3.2. Training dynamics

This paper uses a two-stage view to compare and analyze the training process of the model: Figure 1 (a) shows the initial phase of 20 epochs, Figure 1 (b) shows the complete training process of 80 epochs, and logarithmic coordinates are used for different orders of magnitude.

DOI: 10.54254/2755-2721/2026.GU33716



(a) First 20 epochs (b) Full training of 80 epochs

Figure 1. Comparison of the training curve

In the first 20 epochs, a rapid loss decline was noticed across all three models, and an initial fit to the features could be established despite different starting points.

Over the full training period (Epochs 1–80), the convergence quality and stability offered by YOLOv8n turned out to be superior. Its loss curve moved downward smoothly from 5.436 to 1.646, with no significant fluctuations appearing. By contrast, although SSDlite-MV3 realized the largest loss reduction (from 2987 to 72), some noticeable fluctuations still existed in the later stage of its curve, and this points to gradient instability. For Faster R-CNN, only 20 epochs were evaluated, as the time cost was too heavy to bear. However, a smooth downward trend was observed there, which suggests that more epochs might bring further optimization for this model.

Beyond convergence, training efficiency is also a matter of importance to feasible engineering. Both YOLOv8n and SSDlite-MV3 managed to finish 80 epochs in less than 2 hours, while Faster R-CNN took around 6 hours for just 20 epochs. If the full 80 epochs were to be completed, the training time could become 12 times longer. Even though a longer training might allow Faster R-CNN to reach higher accuracy, its computational cost makes the model far from practical in real-world deployment. In the end, YOLOv8n distinguishes itself with decent performance in all the metrics, proving its excellent training efficiency and engineering value.

4. Discussion

Based on the experimental findings, this section discusses the advantages, limitations, and future directions of YOLOv8n in traffic sign detection.

4.1. Advantages analysis

The superiority of YOLOv8n benefits from its architecture and algorithm innovations. It resolves the misalignment between classification and localization tasks with a task alignment strategy, which better filters positive samples. Beyond that, the integration of Distribution Focal Loss (DFL) makes bounding box regression more accurate than traditional losses (e.g., Smooth L1 in SSDlite), which explains YOLOv8n's smooth convergence process.

4.2. Limitations

Despite its robust performance, this study has three main limitations. First, the model has never been validated under disruptive environmental conditions (e.g., heavy rain, fog, low illumination), which could severely impact feature extraction. Second, the generalization ability is evaluated on constrained regional datasets, lacking further experiments on diverse signs from different countries.

DOI: 10.54254/2755-2721/2026.GU33716

Finally, while desktop FPS indicates real-time potential, it is unclear how it works when actually deployed on resource-constrained edge devices (e.g., Jetson Nano).

4.3. Future work

To deal with these limitations, the future work will be arranged around three main directions. The first direction is about strengthening the environmental robustness, which can be done through data augmentation. For instance, Mosaic and GAN-based synthetic weather generation can simulate complicated conditions. The second thing is making edge deployment more feasible. Pruning along with quantization would be applied to bring down the computational cost on vehicle chips that have less power. Lastly, a more comprehensive benchmark is to be constructed. Such a benchmark is supposed to cover diverse weather, varied lighting, and international sign styles, in order that real-world driving performance can be properly assessed.

5. Conclusion

This study conducted a comparison of Faster R-CNN, SSDlite-MV3, and YOLOv8n for traffic sign detection. The main conclusions are as follows:

Faster R-CNN achieved the highest mAP@0.5 due to its well-designed region proposal mechanism, but it was outperformed by YOLOv8n under the stricter mAP@0.5:0.95 metric, indicating its limitation in bounding box regression. Although both single-stage models had the velocity for real-time inference, SSDlite-MV3 failed to keep up with accuracy due to its outdated fusion strategies. YOLOv8n took the lead in mAP@0.5:0.95, recall, and inference speed, while loss curve analysis showed that it excelled at optimization and stable convergence.

In conclusion, YOLOv8n provides the optimal balance between precision and speed for this task. Future work will focus on data augmentation and model compression techniques to push deployment on edge devices.

References

[1] Shalev-Shwartz, S., Shammah, S. and Shaknarovich, S. (2017). On a Formal Model of Safe and Scalable Self-Driving Cars. arXiv preprint arXiv: 1708.06374.
[2] LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.
[3] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. Proceedings of the IEEE CVPR, San Diego, CA, USA, 886-893.
[4] 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 (NeurIPS), Montreal, Canada, 91-99.
[5] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE CVPR, Las Vegas, NV, USA, 779-788.
[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, C.Y., Fu, C.Y. and Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. Proceedings of the ECCV, Amsterdam, Netherlands, 21-37.
[7] Zou, Z., Shi, Z., Guo, Y. and Ye, J. (2023). Object Detection in 20 Years: A Survey. Proceedings of the IEEE, 111(3), 257-276.
[8] Long, J., Shelhamer, E. and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE CVPR, Boston, MA, USA, 3431-3440.
[9] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, T., Weyand, T., Andreetto, M. and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv: 1704.04861.
[10] Tan, M. and Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the ICML, Long Beach, CA, USA, 6105-6114.